

https://dspace.org/

The Wits intelligent teaching system (WITS): a smart lecture theatre to assess audience engagement

Klein, Richard

2017

http://hdl.handle.net/10539/23528

Klein, Richard (2017) The Wits intelligent teaching system (WITS): a smart lecture theatre to assess audience engagement, University of the Witwatersrand, Johannesburg, http://hdl.handle.net/10539/23528

Downloaded from a DSpace repository. https://dspace.org/

THE WITS INTELLIGENT TEACHING SYSTEM (WITS)
A SMART LECTURE THEATRE TO ASSESS AUDIENCE ENGAGEMENT

SCHOOL OF COMPUTER SCIENCE AND APPLIED MATHEMATICS
UNIVERSITY OF THE WITWATERSRAND

RICHARD KLEIN
0707074G

SUPERVISED BY
PROF TURGAY CELIK

MAY 10, 2017



A Thesis submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Doctor of Philosophy

Ethics Clearance Number: H14/03/06

Abstract

The utility of lectures is directly related to the engagement of the students therein. To ensure the value of lectures, one needs to be certain that they are engaging to students. In small classes experienced lecturers develop an intuition of how engaged the class is as a whole and can then react appropriately to remedy the situation through various strategies such as breaks or changes in style, pace and content. As both the number of students and size of the venue grow, this type of contingent teaching becomes increasingly difficult and less precise. Furthermore, relying on intuition alone gives no way to recall and analyse previous classes or to objectively investigate trends over time. To address these problems this thesis presents the WITS INTELLIGENT TEACHING SYSTEM (WITS) to highlight disengaged students during class.

A web-based, mobile application called Engage was developed to try elicit anonymous engagement information directly from students. The majority of students were unwilling or unable to self-report their engagement levels during class. This stems from a number of cultural and practical issues related to social display rules, unreliable internet connections, data costs, and distractions. This result highlights the need for a non-intrusive system that does not require the active participation of students. A non-intrusive, computer vision and machine learning based approach is therefore proposed.

To support the development thereof, a labelled video dataset of students was built by recording a number of first year lectures. Students were labelled across a number of affects – including boredom, frustration, confusion, and fatigue – but poor inter-rater reliability meant that these labels could not be used as ground truth. Based on manual coding methods identified in the literature, a number of actions, gestures, and postures were identified as proxies of behavioural engagement. These proxies are then used in an observational checklist to mark students as engaged or not.

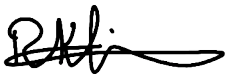
A Support Vector Machine (SVM) was trained on Histograms of Oriented Gradients (HOG) to classify the students based on the identified behaviours. The results suggest a high temporal correlation of a single subject's video frames. This leads to extremely high accuracies on seen subjects. However, this approach generalised poorly to unseen subjects and more careful feature engineering is required. The use of Convolutional Neural Networks (CNNs) improved the classification accuracy substantially, both over a single subject and when generalising to unseen subjects. While more computationally expensive than the SVM, the CNN approach lends itself to parallelism using Graphics Processing Units (GPUs). With GPU hardware acceleration, the system is able to run in near real-time and with further optimisations a real-time classifier is feasible.

The classifier provides engagement values, which can be displayed to the lecturer live during class. This information is displayed as an Interest Map which highlights spatial areas of disengagement. The lecturer can then make informed decisions about how to progress with the class, what teaching styles to employ, and on which students to focus. An Interest Map was presented to lecturers and professors at the University of the Witwatersrand yielding 131 responses. The vast majority of respondents indicated that they would like to receive live engagement feedback during class, that they found the Interest Map an intuitive visualisation tool, and that they would be interested in using such technology.

Contributions of this thesis include the development of a labelled video dataset; the development of a web based system to allow students to self-report engagement; the development of cross-platform, open-source software for spatial, action and affect labelling; the application of Histogram of Oriented Gradient based Support Vector Machines, and Deep Convolutional Neural Networks to classify this data; the development of an Interest Map to intuitively display engagement information to presenters; and finally an analysis of acceptance of such a system by educators.

Declaration

I declare that this thesis is my own, unaided work. It is being submitted for the Degree of Doctor of Philosophy at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



Richard Klein
May 10, 2017

In memory of my father,
David Klein (1955 – 2013),

and dedicated to my mother,
Margie Klein.

Acknowledgements

I would like to sincerely thank my supervisor, Prof. Turgay Celik, for his guidance, motivation, help and mentorship. He has always been available when I needed assistance – including many late nights. His help has been invaluable and I consider him to be the model on which to base an academic career.

Thank-you to my family. To my mother and brother, Margie and Peter, for their unwavering support – encouragement when things were broken, celebrations when things were working, and frozen meals for when I ran out of food.

Thank-you to my colleagues in the School of Computer Science and Applied Mathematics. Specific thanks go to Prof. Michael Sears for all our chats and to Mr Pravesh Ranchod for all the free cokes. I would also like to extend my appreciation to Dr. Ann Cameron, from the Wits Science Teaching and Learning Centre for her support and enthusiasm.

My gratitude goes out to all my friends who have supported me during this time. Special mentions go out to Benji Sim, Nick Parker, Jade More-Sim, Chloe Shaw, Kirsty Parker, Donna MacLeod, Nikki Seiler, Sarah Everson, Roxy Klein, Vaughan Phillips, Jarryd Bekker, Ernst Shea-Kruger, and Andrew Estill. Thank-you to Donna MacLeod for her assistance editing the final document.

Thank-you to the National Research Foundation (NRF) of South Africa which partially funded this research through their Thuthuka programme (Unique Grant No. 94006).

Finally, I would like to express my gratefulness to all the students that took part in the various experiments and to all the research assistants that tirelessly labelled data.

Publications

Aspects of this work have been submitted as follows:

Klein and Celik (2017a)
Richard Klein and Turgay Celik. Engage: Live Self Reported Engagement for Large Classes. IEEE Transactions on Learning Technologies, 2017. Under review.

Klein and Celik (2017d)
Richard Klein and Turgay Celik. Wits Intelligent Teaching System: A video dataset and computer vision system for student action recognition in the classroom. IEEE Transactions on Learning Technologies, 2017. Under review.

Klein and Celik (2017c)
Richard Klein and Turgay Celik. Visualization of Audience Interest: An Interest Map for Reporting Live Audience Engagement. IEEE Transactions on Learning Technologies, 2017. Under review.

Klein and Celik (2017b)
Richard Klein and Turgay Celik. The Wits Intelligent Teaching System: Detecting Student Engagement During Lectures Using Convolutional Neural Networks. In 2017 International Conference on Image Processing. ICIP'17. IEEE, 2017. Accepted.

Contents

Preface

Abstract . . . i
Declaration . . . ii
Acknowledgements . . . iv
Publications . . . v
Table of Contents . . . vi
List of Figures . . . x
List of Tables . . . xiii
Nomenclature . . . xv

1 Introduction **1**

1.1 Introduction . . . 1
1.2 Contributions . . . 2
1.3 Overview . . . 2

2 Background **4**

2.1 Introduction . . . 4
2.1.1 Computer Vision . . . 4
2.1.2 Machine Learning . . . 5
2.1.3 Overview . . . 6
2.2 Computer Vision Techniques . . . 7
2.2.1 Convolution and Linear Filtering . . . 7
2.2.2 Optical Flow . . . 7
2.2.3 Histogram of Oriented Gradients (HOG) . . . 10
2.2.4 Local Binary Patterns . . . 13
2.2.5 Haar-like Features . . . 14
2.3 Supervised Machine Learning Techniques . . . 16
2.3.1 Feature Vectors . . . 17
2.3.2 Linear Separability and Kernel Functions . . . 18
2.3.3 A Linear Classifier: The Neuron . . . 21
2.3.4 Support Vector Machines . . . 23
2.3.5 Boosting and Viola-Jones . . . 27
2.3.6 Deep Learning and Convolutional Neural Networks (CNN) . . . 29
2.4 Incorporating Multi-Sensor Data . . . 38
2.4.1 Data Fusion . . . 38
2.4.2 Feature Fusion . . . 40

2.4.3 Decision Fusion . . . 40
2.5 Dimensionality Reduction . . . 40
2.5.1 Principal Component Analysis (PCA) . . . 41
2.6 Accuracy Measures and Validation . . . 44
2.6.1 Classification Validation . . . 44
2.6.2 Cross-Subject Validation . . . 45
2.6.3 Imbalanced Datasets . . . 46
2.6.4 Approaches Used . . . 47

3 Related Work **48**
3.1 Contingent Teaching . . . 48
3.2 Affective Computing . . . 49
3.3 Emotions . . . 51
3.3.1 What to Measure? . . . 51
3.3.2 Perspectives on Emotion . . . 52
3.4 Affect Detection . . . 57
3.4.1 Facial Expression . . . 58
3.4.2 Paralinguistic Speech Features . . . 58
3.4.3 Body Language and Posture . . . 59
3.4.4 Physiology . . . 60
3.4.5 Multimodality . . . 61
3.4.6 Real versus Posed Data . . . 62
3.4.7 Ground Truth . . . 62
3.4.8 General Remarks on Affective Computing and Assumptions . . . 63
3.5 Affective Computing in Education . . . 64
3.5.1 Affect in Education . . . 64
3.5.2 Detection of Affect for Educational Purposes . . . 65
3.6 Prediction and Detection of Affect . . . 68
3.6.1 Boredom and Engagement . . . 68
3.6.2 Confusion . . . 71
3.6.3 Frustration . . . 72
3.6.4 Fatigue . . . 72
3.7 Conclusion . . . 73

4 The Intrusive Approach: Live Self-Reported Engagement **75**
4.1 Introduction . . . 75
4.2 Related Work . . . 77
4.2.1 Introduction . . . 77
4.2.2 SMS in and out of the Classroom . . . 77
4.2.3 Clickers in the Classroom . . . 78
4.2.4 Systems for Real-Time Formative Lecturer Feedback . . . 79
4.3 Design . . . 79
4.4 Architecture . . . 83
4.5 Results . . . 84
4.5.1 Uptake . . . 84
4.5.2 Student Comments . . . 91

4.6 Conclusion . . . 94

5 The Non-Intrusive Approach: The Wits Intelligent Teaching System 96
5.1 Introduction . . . 96
5.2 Architecture . . . 98
5.3 Data Collection . . . 100
5.3.1 WITS Annotation Tool (WITSAT) . . . 100
5.3.2 Data Export . . . 103
5.3.3 Classification . . . 103
5.3.4 Label Comparisons . . . 104
5.3.5 WITS Database (WITSDB) . . . 104
5.4 Computational Pipeline for Training and Testing . . . 112
5.5 Conclusion . . . 115

6 Non-Intrusive Approach using Engineered Visual Features 117
6.1 Introduction . . . 117
6.2 Validity . . . 117
6.3 Classroom Actions . . . 119
6.3.1 Writing . . . 119
6.3.2 Cellphone Use . . . 126
6.3.3 Laptop Use . . . 128
6.3.4 Talking . . . 128
6.3.5 Raised Hand . . . 128
6.3.6 Yawning . . . 130
6.3.7 Head on Desk . . . 130
6.4 Posture . . . 131
6.5 Conclusion . . . 131

7 Non-Intrusive Approach using Automated Visual Features 132
7.1 Introduction . . . 132
7.2 Hardware Acceleration . . . 133
7.3 Validity . . . 133
7.3.1 Cross-Frame Validation . . . 133
7.3.2 Cross-Subject Validation . . . 134
7.3.3 Cross-Lecture Validation . . . 136
7.4 Classroom Actions . . . 136
7.4.1 Writing . . . 137
7.4.2 Cellphone . . . 139
7.5 Posture . . . 142
7.6 Horizontal Head Pose . . . 144
7.7 Vertical Head Pose . . . 146
7.8 Conclusion . . . 147

8 Interest and Visualisation: The Interest Map 150
8.1 Introduction . . . 150
8.2 The Interest Map Computational Pipeline . . . 151
8.3 Defining Interest: The Observational Checklist . . . 153

8.4 Recognising Interest . . . 155
8.4.1 Direct and Indirect Interest Classification . . . 155
8.4.2 Data Export . . . 156
8.4.3 Cross-Frame Validation . . . 157
8.4.4 Cross-Subject Validation . . . 158
8.4.5 Sequence Length . . . 158
8.4.6 Image Size . . . 158
8.4.7 Recognising New Subjects . . . 159
8.4.8 Cross-Lecture Validation . . . 160
8.5 Acceptance . . . 161
8.5.1 Respondents . . . 161
8.5.2 Response to the Interest Map . . . 164
8.6 Respondent Comments . . . 164
8.6.1 Comments from a Teaching & Learning Expert . . . 166
8.6.2 Performance Monitoring . . . 167
8.6.3 Summative Feedback . . . 168
8.6.4 Definition of Engagement and Accuracy . . . 168
8.6.5 Teachers Do This Already . . . 169
8.6.6 Lecturer Distraction . . . 171
8.6.7 Student Privacy, Discomfort and the Hawthorne Effect . . . 171
8.6.8 Resolution . . . 172
8.6.9 Display Colours . . . 172
8.6.10 Temporal Considerations . . . 173
8.7 Conclusion . . . 173

9 Conclusion **175**
9.1 Summary . . . 175
9.2 Accuracy and Generalisation . . . 177
9.3 Computational Performance . . . 178
9.4 Future Work . . . 179
9.4.1 Engage . . . 179
9.4.2 WITS Database . . . 179
9.4.3 Classifiers . . . 180
9.4.4 Interest Map . . . 181
9.4.5 Automatic Teaching Assistant (AutoTA) . . . 181
9.4.6 Educational Research . . . 181
9.5 Conclusion . . . 181

A Questionnaire Results **182**

References **219**

List of Figures

2.1 Linear Convolution . . . 8
2.2 Convolution Filters . . . 8
2.3 Gradient Detection . . . 11
2.4 Gradient Vectors . . . 12
2.5 HOG Feature Descriptor . . . 12
2.6 HOG Feature Descriptor Over a 256×256 Image . . . 13
2.7 Local Binary Patterns . . . 14
2.8 Haar-Like Features (Viola and Jones 2001) . . . 15
2.9 Integral Image Regions . . . 15
2.10 Using the Integral Image . . . 16
2.11 Classification Boundaries (Marsland 2015) . . . 17
2.12 South African Rands (South African Reserve Bank 2017) . . . 17
2.13 Linearly Separable Data (Kiyoshi Kawaguchi 2000) . . . 19
2.14 Adding Features for Linear Separability (Friedman *et al.* 2001) . . . 19
2.15 McCulloch and Pitts (1943) Neuron . . . 22
2.16 Neuron with Bias . . . 22
2.17 Separating Hyperplanes (Weinberg 2012) . . . 24
2.18 Optimal Separating Hyperplane . . . 25
2.19 Cascade Classifier . . . 29
2.20 Single Layer Perceptron . . . 30
2.21 Neural Network/Multi Layer Perceptron . . . 31
2.22 Activation Function Candidates . . . 32
2.23 Feature Learning (Goodfellow *et al.* 2016) . . . 35
2.24 AlexNet Architecture (Krizhevsky *et al.* 2012) . . . 38
2.25 HOG Features with Data Fusion . . . 39
2.26 Highly Correlated 2D Data . . . 42
2.27 Eigenvalues of the Principal Components . . . 42

3.1 Specimen Questions for the Turing Test (Turing 1950) . . . 49
3.2 Proposed Emotional Framework for Learning (Kort *et al.* 2001) . . . 64

4.1 Student Views of the Engage System . . . 80
4.2 Instructor Dashboard . . . 81
4.3 System Architecture . . . 85
4.4 Experiment 2: First Years, feedback to lecturer only. . . . 87
4.5 Experiment 3: First Years, feedback projected for all students. . . . 87
4.6 Experiment 4: First Years, feedback projected for all students. . . . 88

4.7 Experiment 5: First Years, feedback projected for all students with comments enabled. . . . 89
4.8 Experiment 6: First Years, feedback projected for all students with comments enabled.* . . . 89
4.9 Experiment 7: First Years, lecture streamed on YouTube feedback and comments available on students' devices at home.† . . . 90
4.10 Experiment 8: Fourth Years, feedback accessible through lab computers with comments enabled. . . . 91
4.11 Experiment 9: Fourth Years, feedback to lecturer only.* . . . 91

5.1 Standard Lecture Venue . . . 99
5.2 System Level View of WITS . . . 99
5.3 Annotation Tool . . . 101
5.4 Automated Face Labelling . . . 102
5.5 Face Detectors . . . 102
5.6 Label Export Format as an SQLite Table. . . . 104
5.7 Compare Labels . . . 105
5.8 Invalid Frames . . . 107
5.9 Action Labels . . . 109
5.10 Action Labels – Special Cases . . . 110
5.11 Subject Postures . . . 110
5.12 Subject Head Poses . . . 111
5.13 Training and Testing Methodology . . . 114

6.1 Cross Validation Results Over Subjects – Writing . . . 124
6.2 Good (a) and Bad (b) Subject Views. . . . 124
6.3 High Performing Subjects for Cellphone Recognition. . . . 126
6.4 Cross Validation Results Over Subjects – Cellphone Use . . . 126
6.5 Talking Recognition . . . 128

7.1 CNN Accuracy and Loss (Validity) . . . 135
7.2 CNN Accuracy and Loss (Writing) . . . 138
7.3 CNN Accuracy and Loss (Cellphone) . . . 140
7.4 CNN Accuracy and Loss (Posture) . . . 143

8.1 Interest Map . . . 150
8.2 Temporal Interest Map . . . 152
8.3 Computational Pipeline for Interest Map Creation . . . 153
8.4 Extract of SQLite Database of Labels . . . 156
8.5 Interest Map Questionnaire . . . 162
8.6 Respondent Information . . . 163
8.7 Responses to the Interest Map . . . 165
8.8 Comment from a Teaching & Learning Expert . . . 166
8.9 Comments Regarding Performance Monitoring . . . 168
8.10 Comments Regarding Summative Feedback . . . 169
8.11 Comments Regarding Accuracy . . . 170
8.12 Comments Regarding Teachers' Abilities . . . 171

8.13 Comment Regarding Lecturer Distraction 171
8.14 Comments Regarding Student Discomfort 171
8.15 Comment Regarding Display 172
8.16 Comments Regarding Interest Over Time 173

List of Tables

2.1 Interpretation of Cohen's κ Scores . . . 46

3.1 Classification Accuracies in Bosch *et al.* (2015b) . . . 65

4.1 Student Responses Using Engage in Live Lectures . . . 85
4.2 Did you use Engage to report distraction during the lecture? . . . 91
4.3 Do you think the lecturer was responsive to the information received? . . . 92
4.4 Would you like to use this app in all your classes? . . . 92
4.5 When asking short questions do you prefer... . . . 92
4.6 Did you find the program distracting? . . . 92
4.7 What buttons should the program display? . . . 92
4.8 Please provide any further comments you might have. . . . 92
4.9 Did you use Engage to report distraction during the lecture? . . . 93
4.10 Do you think the lecturer was responsive to the information received? . . . 93
4.11 Would you like to use this app in all your classes? . . . 93
4.12 When asking short questions do you prefer... . . . 93
4.13 Did you find the program distracting? . . . 93
4.14 What buttons should the program display? . . . 93
4.15 Please provide any further comments you might have. . . . 94

5.1 Number of Valid Frames in WITSDB . . . 112
5.2 Number of Valid Frames by Action Labels in WITSDB . . . 113
5.3 Number of Valid Frames by Posture in WITSDB . . . 113
5.4 Number of Valid Frames by Head Pose in WITSDB . . . 114

6.1 HOG SVM Validity Results: 5-Fold Cross Validation on 10,000 Frames . . . 118
6.2 Validity Results: 5-Fold Cross Validation on 50,000 Frames . . . 119
6.3 Writing Results: 5-fold Validation on Stratified Datasets, PCA Retains 90% Variance. . . . 121
6.4 Writing Results: 5-fold Validation on Stratified Datasets, PCA Retains 99% Variance. . . . 121
6.5 Cross Validation Results Over Frames – Writing . . . 122
6.6 Cross Validation Results Over Subjects – Writing . . . 125
6.7 Cross Validation Results Over Subjects – Cellphone Use . . . 127
6.8 Cross Validation Accuracy Over Subjects – Talking . . . 129
6.9 Cross Validation Accuracy Over Subjects – Raised Hand . . . 130

7.1 CNN Accuracy Detecting Validity . . . 134

7.2 CNN Confusion Matrix for the Validation Set (Validity) . . . 134
7.3 CNN Confusion Matrix for Cross-Subject Validation (Validity) . . . 136
7.4 CNN Accuracy Detecting Writing . . . 137
7.5 CNN Confusion Matrix for the Validation Set (Writing) . . . 137
7.6 CNN Confusion Matrix for the Cross-Subject Validation Sets (Writing) . . . 139
7.7 CNN Confusion Matrix for the Cross-Lecture Validation Sets (Writing) . . . 139
7.8 CNN Accuracy Detecting Cellphone . . . 141
7.9 CNN Confusion Matrix for the Validation Set (Cellphone) . . . 141
7.10 CNN Confusion Matrix for the Cross-Subject Validation Sets (Cellphone) . . . 141
7.11 CNN Confusion Matrix for the Cross-Lecture Validation Sets (Cellphone) . . . 142
7.12 CNN Confusion Matrix for the Validation Set (Posture) . . . 143
7.13 CNN Confusion Matrix for the Cross-Subject Validation Sets (Posture) . . . 144
7.14 CNN Confusion Matrix for the Cross-Lecture Validation Sets (Posture) . . . 144
7.15 CNN Confusion Matrix for the Validation Set (Horizontal Head Pose) . . . 145
7.16 CNN Confusion Matrix for the Cross-Subject Validation Sets (H. Head Pose) . 145
7.17 CNN Confusion Matrix for the Cross-Lecture Validation Sets (H. Head Pose) . 146
7.18 CNN Confusion Matrix for the Validation Set (V. Head Pose) . . . 146
7.19 CNN Confusion Matrix for the Cross-Subject Validation Sets (V. Head Pose) . 147
7.20 CNN Confusion Matrix for the Cross-Lecture Validation Sets (V. Head Pose) . 147

8.1 Interest Frames per Lecture . . . 156
8.2 CNN Confusion Matrix for the Validation Set . . . 157
8.3 CNN Confusion Matrix for cross-subject validation on all lecture 1 frames . . . 158
8.4 CNN Test Set Accuracy by Sequence Length . . . 159
8.5 CNN Test Set Accuracy by Image Size . . . 159
8.6 CNN Confusion Matrix for Cross-Subject Validation with 1,000 Frame Bootstrap 160
8.7 CNN Confusion Matrix for Cross-Lecture Validation (Interest) . . . 161
8.8 Respondent Disciplines . . . 164

A.1 Raw Demographic Responses . . . 182
A.2 Raw Interest Map Responses . . . 185

Nomenclature

Mathematical Notation

$\mathbf{A}, \mathbf{B}, \mathbf{W}$	An upper case, boldface letter is a matrix.
X, Y, Z	An upper case, light (non-boldface) letter is a set.
$\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{x_i}$	A lower case, boldface letter is a vector.
f, g	Arbitrary functions.
x, y, z, x_i	A lower case, light (non-boldface) letter is a real number.
$(\cdot)^T$	Matrix transpose operator.
$\dot{x}, \dot{y}$	Derivatives of variables x and y with respect to time.
$\exp(x)$	The exponential function, e^x
$\nabla_{\mathbf{w}}$	Gradient operator with respect to a specific set of weights, $\mathbf{w}$.
κ	Cohen's Kappa (Cohen 1960).

Images

$\mathbb{D}$	A dataset.
$\mathbf{I}$	An image or image sequence in dataset $\mathbb{D}$.
$\mathbf{I}(x, y)$	The pixel value at position (x, y) of some image $\mathbf{I}$.
$\mathbf{I}(x, y, t)$	The pixel value at position (x, y) at time t of some image sequence.
$\mathbf{I}_x, \mathbf{I}_y, \mathbf{I}_t$	Partial derivatives of image intensity with respect to x, y and t respectively.
$\mathbf{II}$	An integral image.
w, h	Width and height of some image.
HOG	Histogram of Oriented Gradients.

LBP	Local Binary Patterns.

Machine Learning

X	Set of all feature vectors over some dataset $\mathbb{D}$.
$\mathbf{x}, \mathbf{x_i} \in X$	A feature vector.
Y	Set of all label vectors corresponding to the feature vectors in X.
$\mathbf{y}, \mathbf{y_i} \in Y$	A label vector.
$\mathbf{p}$	Parameters of a machine learning model.
SVM	Support Vector Machine.
η	Learning Rate.
HOG SVM	Support Vector Machine using HOG descriptors.
RBF	Radial Basis Function.
CNN	Convolutional Neural Network.

Chapter 1

Introduction

1.1 Introduction

STUDENT engagement is a pivotal concept when considering the utility of lectures. In an age where even tertiary level content is available online, it becomes important for instructors to differentiate themselves from online content sources. This is particularly true when teaching large classes where interaction with the presenter is already limited.

Disengagement and boredom are fundamental and chronic problems with traditional modes of instruction that effect up to 60% of students (Astin 1984; Larson and Richards 1991; MacHardy *et al.* 2012; Shernoff *et al.* 2003). While research into student engagement originally focussed on concerns relating to large drop-out rates, it should now be considered in relation to new teaching modes that have emerged as a result of modern technology. As class and venue sizes grow, the ability of the presenter to engage with the entire audience diminishes. In extreme cases, the value of an expert lecturer can be lost completely if students are unable to interact with him or her. In such cases, the efficacy of *chalk-and-talk* lectures over video and online systems becomes questionable.

Online tutoring systems can provide video lectures, animations, spot tests with immediate feedback, and custom lesson paths to each student. This is in stark contrast with lectures that often degrade into frantic transcription or sleeping sessions.

The committed lecturer need not despair, however, as there are many examples of highly effective lectures. Expert teachers and lecturers are able to gauge the emotional state of the class and take steps to positively impact learning (Kapoor *et al.* 2007). A passionate lecturer's enthusiasm in class is contagious and can help instil a curiosity and excitement in learners that may not be present through newer teaching modes (Picard 1995). Furthermore, lecturers practising *contingent teaching* scaffold their classes in such a way that they go 'off-script' to cater to the current state of the class (Draper and Brown 2004). This is considered by many to be the definition of good teaching, albeit scarce and difficult to perform (van de Pol *et al.* 2011). This type of teaching is particularly difficult in large classes where just keeping track of all the students can become a challenge.

The purpose of this thesis and through it the WITS INTELLIGENT TEACHING SYSTEM

(WITS) is to support both lecturers and institutions in understanding and improving the quality of their lectures. WITS uses computer vision and machine learning techniques to monitor students and report back about their engagement during class. Such a system could be used live in class to assist lecturers with contingent teaching or as a self-evaluation tool with which they can retrospectively assess how well they managed to engage their students. It could be used when training new lecturers, or to understand the intuition developed by already established ones. WITS may be able to identify students that are 'at risk' or even venues that show consistent spatial patterns of disengagement that may be a result of some physical problem such as lighting, acoustics or airflow. Finally, as a tool to support research, WITS can be used to assess teaching strategies and to understand when students disengage and re-engage.

1.2 Contributions

This work contributes a number of novel aspects to the literature.

- A system to perform real-time analysis of students in large classes is proposed. Based on a survey of the literature, it is believed that this is the first work to deal with the topic of affect detection in a wild learning environment where subjects are not seated in front of a computer.
- Engage, a web-based, open-source system to allow real-time, anonymous feedback about interest during class, was developed and released (Klein and Celik 2017a).
- The WITS ANNOTATION TOOL (WITSAT) was developed and released. It is new cross-platform and open-source software that allows the user to label videos along spatial, action, and affective axes (Klein and Celik 2017d).
- The WITS DATABASE (WITSDB) was constructed and is a database of lecture videos where students were labelled using WITSAT (Klein and Celik 2017d).
- Histograms of Oriented Gradients with Support Vector Machines were applied to WITSDB to recognise an approximation of Interest (Klein and Celik 2017c).
- Convolutional Neural Networks with GPU hardware were used to improve accuracy and achieve near real-time performance (Klein and Celik 2017b).
- An Interest Map to visualise the feedback was proposed, implemented and its acceptance by educators at the University of the Witwatersrand was assessed (Klein and Celik 2017c).

1.3 Overview

Chapter 2 presents background in Computer Science and Mathematics relating to the methods used in this work. This primarily encompasses a discussion of Computer Vision and Machine Learning, as well as their respective techniques. Chapter 3 presents related work in Affective Computing and Education.

Chapter 4 presents Engage, a web-based tool for real-time feedback from students and discusses the problems associated with intrusive approaches like this. In Chapter 5 the need for an automatic, non-intrusive approach to monitoring interest is argued, which also discusses the development and creation of WITSAT and WITSDB. Chapter 6 applies Histogram of Oriented Gradient features and Support Vector Machines to the labelled image sequences in WITSDB and presents results thereof. Chapter 7 then documents the application of Convolutional Neural Networks to the data which vastly improves accuracy, generalisation, and performance.

Chapter 8 focuses on the definition of interest using an observational checklist, as well as the development of an Interest Map to visualise and report this information to the lecturer. Responses from educators at the institution regarding the use and acceptance of the system are also discussed.

Finally, Chapter 9 concludes with a discussion of the system in its entirety. This includes ethical and practical aspects that should be considered when deploying the system, interpreting the results of previous sections, and finally future work is proposed.

Chapter 2

Background

2.1 Introduction

2.1.1 Computer Vision

Arguably one of the most impressive abilities of humans is our ability to almost effortlessly perceive the world around us. While a young child can describe a scene or count the objects in it, it is over thirty years since the inception of computer vision and the same is not (yet) a solved problem for computers. The goal of computer vision is to imitate the human ability to interpret and extract information from images and image sequences (Mallat 1990).

The difficulty of computer vision is that it is an inverse problem. Given a detailed model of the world, computers can generate hyper-realistic images. Driven by the gaming and movie industries, the field of computer graphics is mature and generates spectacular results that can be unsettling in their realism. We understand how light interacts with different surfaces and how lenses work to focus it on a retina or hardware sensor. Computer graphics can be considered the forward problem, and computer vision the inverse.

Computer graphics aims to take a model and render it as an image. Conversely, computer vision aims to take an image and construct a model of the world. The difficulty here is that it must take incomplete data and build a solution. Given this incomplete data, often the need arises for physics-based and probabilistic models (Szeliski 2010).

Computer vision already finds real-world applications in optical character recognition, machine inspection, retail checkout lanes, photogrammetry (building 3D models from aerial photographs), medical imaging, automotive safety (a fundamental aspect for self-driving cars), motion capture, surveillance, biometrics, remote sensing, robot guidance, and many more (Rosenfeld 1988; Szeliski 2010). Disciplines of computer vision include image processing, feature detection and mapping, segmentation, alignment, structure from motion, sparse and dense motion estimation, image stitching, stereo correspondence, 3D reconstruction, rendering, and recognition techniques (Szeliski 2010).

In contexts like facial recognition, companies like Facebook are starting to approach human-level performance with unprecedented accuracies of up to 97.35% over a dataset with four mil-

lion images and over 4000 identities (Taigman *et al.* 2014). For facial recognition problems, this means that over the last 20 years error rates will have decreased by well beyond three orders of magnitude (Phillips *et al.* 2011).

Traditional recognition approaches such as Eigenfaces (Turk and Pentland 1991) used mathematical and statistical techniques to find patterns in pixel values over a database of images. These approaches did not necessarily make use of the strong spatial correlations identified in images. New features, such as wavelets, local binary patterns, histograms of oriented gradients, and custom convolutional filters were developed to make use of the local and global spatial correlations and to extract features that better describe the aspects that make up an image – such as edges, gradients, textures, highlights and others.

These features are then usually the input into some machine learning process that uses techniques such as support vector machines, linear or logistic regression, decision trees, and neural networks. In recent years, new breakthroughs in deep learning have propelled computer vision forward particularly with regard to object recognition and classification (Krizhevsky *et al.* 2012). In deep learning, many-layer neural networks are used, often in conjunction with convolutional and pooling layers, where the learning agent itself is responsible for learning features that encapsulate the spatial information found in images.

Modern methods of recognition are primarily built upon machine learning approaches. Machine learning is considered in the following section.

2.1.2 Machine Learning

The fundamental goal of machine learning is to use experience to improve performance in some way (Langley 1996; Shavlik and Dietterich 1990).

Learning involves...

> “...changes in a system that... enable it to do the same task or tasks drawn from the same population more efficiently and more effectively the next time.
> Simon (1983)

The idea is that by using prior experience, the machine itself updates its approach so that similar problems can be solved in the future. It is inherently related to data analysis and statistics. Primarily a science driven by data, the machine must use various computer science, statistics, and optimisation techniques to identify patterns in a large data repository (Mohri *et al.* 2012). The important aspects of learning include remembering, adapting, and generalising (Marsland 2015).

Generally computers can only do what they are told to do: step-by-step recipes or algorithms are required to solve a problem. In contrast, the learning machine is not told how to represent data, concepts, or even how to solve the problem at hand. It is presented with data and must develop its own internal representation of knowledge based on what it sees and the underlying model of the method. This approach is particularly useful when we do not know

how to solve a problem, so the computer is programmed to explore some larger search space to find a solution (Minsky 1961).

There are three main groups of machine learning algorithms: supervised, unsupervised, and semi-supervised. In supervised approaches, the method is presented with data features, $\mathbf{x_i} \in X$, and labels, $\mathbf{y_i} \in Y$, and must learn some function, $f : X \rightarrow Y$, that satisfies $f(\mathbf{x_i}) = \mathbf{y_i} \forall i$. In essence f must learn to capture the general patterns in the dataset so that it can be applied to predict labels ($\mathbf{y}$ values) for new, unseen data ($\mathbf{x}$ values)(Shavlik and Dietterich 1990). The structure of the model, f, is usually chosen beforehand according to the method used and learning involves finding or optimising parameters for the function that best represents the data by minimising some cost or *loss* function.

The second approach to machine learning is unsupervised learning or density estimation. In this paradigm the process searches for similarities in the dataset and, based on its findings, clusters data points into similar groups. No labels are presented to the system in this approach and the algorithms rely purely on the identification of similar traits in each datum. This type of learning is useful for clustering data where there are no labels or when doing feature selection for other approaches (Marsland 2015).

The final paradigm is a semi-supervised approach where the system is given incomplete feedback about its performance. The primary candidate for semi-supervised learning is called Reinforcement Learning in which the system has probabilistic knowledge about the state of the world and performs actions which probabilistically alter that state and produce some penalty or reward. An agent's goal is to take actions or make decisions that maximise the expected reward. The agent is never told which actions to take, so this approach is not supervised, but is given incomplete information about how its actions affected the world and is provided some penalty or reward. Reinforcement learning was originally coined by Minsky (1961), but remained largely dormant until the early 1980's (Sutton and Barto 1998).

This work makes use of supervised learning approaches to vision. In all cases the approach follows the same pipeline:

1. Labelled images undergo feature extraction to develop a data representation that emphasises the high spatial correlation thereof.
2. This feature representation undergoes dimensionality reduction to more compactly express the information in the feature vectors.
3. The reduced dimensionality feature vectors along with the corresponding labels are presented to the system and some numerical optimisation technique is used to find the set of parameters, $\mathbf{p}$, so that the model, f, best describes the data.

2.1.3 Overview

The rest of this chapter presents techniques from computer vision and machine learning that are used later in this work. Section 2.2 shows some common vision and feature extraction methods. These include convolutional filtering, optical flow, Haar Wavelets, Histogram of Oriented Gradients (HOG), and Local Binary Patterns (LBP). Section 2.3 then presents some

machine learning techniques including Support Vector Machines (SVM), the Viola-Jones Cascaded Classifier, and Deep Convolutional Neural Networks. Section 2.4 considers how to use multi-sensor and/or temporal data with methods that do not explicitly support it. Section 2.6 presents the validation techniques that are used to report accuracy and examine generalisability.

2.2 Computer Vision Techniques

2.2.1 Convolution and Linear Filtering

A pervasive technique in computer vision and signal processing is linear filtering or convolution. A convolution *kernel* linearly combines the local neighbourhood of a pixel to produce a response. By placing the kernel over each pixel in the original image, all the responses together form the filtered image. Formally, the convolution (g) of an image, signal or function (f), and a kernel (h), is

$$g(i,j) = \sum_{k,l} f(i-k, j-l) \cdot h(k,l), \tag{2.1}$$

where g is the filtered output. This can be written using the convolution operator ($*$) as

$$g = f * h. \tag{2.2}$$

This linear convolution is illustrated in Figure 2.1 on the following page. The centre of the kernel (0.62) is placed over the 87 in the second row of the image. The values in the image are multiplied with the corresponding values in the kernel and the response is calculated as

$$\begin{aligned} g(1,1) &= 70 \cdot 0.01 + 245 \cdot 0.08 + 129 \cdot 0.01 \\ &\quad +173 \cdot 0.08 + 87 \cdot 0.64 + 178 \cdot 0.08 \\ &\quad +167 \cdot 0.01 + 149 \cdot 0.08 + 227 \cdot 0.01 \\ &= 121. \end{aligned} \tag{2.3}$$

The strategy to handle borders varies between implementations. Often the filtered image is smaller than the original image as pixels where the kernel goes outside the image are excluded. Other approaches calculate the response based on the numbers that are available (as illustrated in Figure 2.1), otherwise the neighbourhood around the border can be repeated or mirrored beyond the image boundary as well.

The filter in Figure 2.1 is a smoothing filter that performs a Gaussian blur over the image. Filters can perform a number of different functions based on the weights of the kernel. The results of a Gaussian blur and Laplacian edge detector are shown in Figures 2.2b and 2.2c.

2.2.2 Optical Flow

Optical flow tracks the perceived motion of objects relative to the image coordinates. This motion may be apparent due to movement of the camera or of the object itself. By allowing the

70	245	129	38	237	90
173	**87**	178	66	89	212
167	149	227	214	50	149
41	57	245	65	64	140
30	192	140	208	157	234
127	65	35	62	121	73

$*$

0.01	0.08	0.01
0.08	**0.64**	0.08
0.01	0.08	0.01

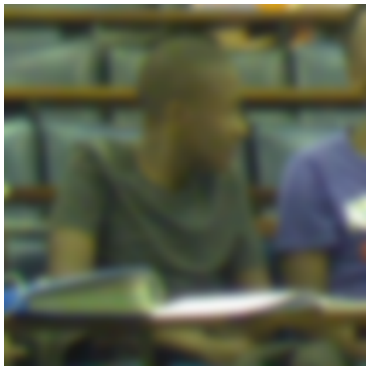
$=$

79	183	121	62	172	94
141	**121**	161	90	107	165
137	145	211	175	78	129
50	92	204	106	82	127
49	151	146	172	154	181
91	72	48	72	105	77

Figure 2.1: Linear Convolution



(a) Original Image



(b) Gaussian Blur Filter



(c) Laplacian Filter

Figure 2.2: Convolution Filters

tracking of pixels' trajectories through consecutive frames, optical flow is used in a number of applications. Structure from motion uses monocular vision to build a 3D reconstruction based on camera movement – this is similar to the way that humans detect depth when one eye is closed (Koenderink 1986). Video compression codecs, such as h264, track the movement of objects to minimise the number of pixel values to be stored for frame reconstruction (Richardson 2004). Video stabilisation can be performed using optical flow to estimate shaking of the camera (Chang *et al.* 2002; Liu *et al.* 2014). Optical flow has even been used to perform action recognition by tracking dense trajectories of pixels (Wang *et al.* 2011).

Optical flow algorithms generally make two assumptions (OpenCV 2015):

1. The pixel intensities of objects do not change between consecutive frames.
2. Neighbouring pixels have similar motion paths.

Points are tracked using the *flow velocity* equations. The derivation of the flow velocity equation, shown in Equation (2.6), is based on Horn and Schunck (1981). A similar derivation based on the Taylor series expansion can be found in OpenCV (2015).

Let the intensity of a point (x, y) in the image at time t be $\mathbf{I}(x, y, t)$. Based on the first assumption, the pixel intensity of the point does not change between consecutive frames, therefore:

$$\left.\frac{d\mathbf{I}}{dt}\right|_{(x,y,t)} = 0. \tag{2.4}$$

x and y are functions of t as the point is moving in the image over time, so applying the chain rule yields:

$$\frac{\partial \mathbf{I}}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial \mathbf{I}}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial \mathbf{I}}{\partial t} = 0 \tag{2.5}$$

This can be rewritten as

$$I_x u + I_y v = -I_t, \tag{2.6}$$

where $u = \dot{x} = \frac{dx}{dt}$, $v = \dot{y} = \frac{dy}{dt}$, and I_x, I_y, I_t are the partial derivatives of intensity with respect to x, y, and t respectively at position (x, y). This is a linear equation in two unknowns and to solve this equation a second constraint is required. Various approaches are considered in literature, but this work makes use of the Lucas-Kanade (LK) algorithm (Lucas *et al.* 1981).

The LK algorithm uses pixels in the neighbourhood of the current one to provide the missing information. If the second assumption holds, then the neighbouring 8 pixels will move in the same way. The image brightness gradients (I_x, I_y, and I_t) can be calculated at each of the nine points. This creates Equation (2.7), a set of nine linear equations with two unknowns.

$$\begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_8} & I_{y_8} \\ I_{x_9} & I_{y_9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_{t_1} \\ I_{t_2} \\ \vdots \\ I_{t_8} \\ I_{t_9} \end{bmatrix} \tag{2.7}$$

To solve this overdetermined linear system a linear least squares fit is used. For some overdetermined linear system,

$$\mathbf{A}\mathbf{b} = \mathbf{y}, \tag{2.8}$$

the linear least squares fit is given by (Strang *et al.* 1993)

$$\mathbf{b} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}. \tag{2.9}$$

Substituting Equation (2.7) into Equation (2.9) gives:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_{x_i}^2 & \sum_i I_{x_i} I_{y_i} \\ \sum_i I_{x_i} I_{y_i} & \sum_i I_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i I_{x_i} I_{t_i} \\ \sum_i I_{y_i} I_{t_i} \end{bmatrix} \tag{2.10}$$

where $i \in \{1, 2, \ldots, 9\}$. This system yields values for u and v which in turn describe the movement of the point being tracked across frames.

This method can be used with image pyramids to efficiently handle movement over multiple scales (Bouguet 2001). The base of the pyramid is the original image and the width and height of the image is halved on subsequent levels. This is the method used for optical flow throughout this work.

2.2.3 Histogram of Oriented Gradients (HOG)

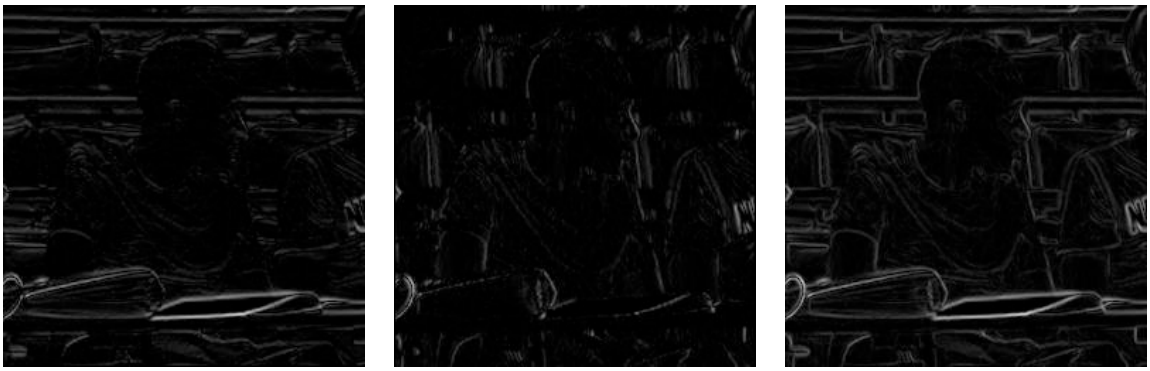
When performing machine learning over images or image sequences, features that take into account the high spatial correlations of visual data should be used. In particular, gradients in local neighbourhoods of an image often prove useful. Dalal and Triggs (2005) developed the Histograms of Oriented Gradients (HOG) feature descriptor for use identifying pedestrians through computer vision.

At a high level, to calculate the HOG feature the image is divided into *cells*. Gradients over the different pixels within each cell are calculated and a histogram of the directions is built. The histograms are then normalised over larger regions called *blocks*. These normalised histograms are known as the HOG descriptors. These features are then presented to machine learning methods (such as support vector machines) which then learn to classify the image according to given labels. A more detailed description of process is given below.

First, the image colour is normalised. Performing convolution using the filters in Equation (2.11) yields the image gradients in the x and y directions respectively. This method resembles the Laplacian operator from Figure 2.2c. Other gradient detecting operators such as the Sobel operator could also be used. Based on the original image in Figure 2.2a, Figure 2.3 shows the horizontal and vertical gradient responses.

$$[-1, 0, 1]^T \text{ and } [-1, 0, 1] \tag{2.11}$$

The gradient can be converted from Cartesian coordinates to polar coordinates by combining the x and y derivatives (g_x, g_y) as shown in Equations (2.12) and (2.13). The gradient magnitude image is shown in Figure 2.3c.



(a) Horizontal Edges (g_y) (b) Vertical Edges (g_x) (c) Gradient Magnitude (g)

Figure 2.3: Gradient Detection

$$g = \sqrt{g_x^2 + g_y^2} \tag{2.12}$$

$$\theta = \arctan \frac{g_y}{g_x} \tag{2.13}$$

Once in polar coordinates, each pixel corresponds to both a magnitude (Equation (2.12)) and direction (Equation (2.13)). The gradient directions can range from 0 to 2π radians or can be converted to range between 0 and π. These correspond to *signed* and *unsigned* approaches. Figure 2.4 shows a small region of the same image with the gradient vector field superimposed. Experimental analysis by Dalal and Triggs (2005) showed that the unsigned approach provides better results when detecting humans.

These gradient directions are then quantised into 9 bins. The image is divided into *cells* that are each 8×8. Each pixel in the cell has a vote to that cell's overall histogram binned by its quantised direction and weighted by its magnitude. Four neighbouring cells are then used to form 2×2 overlapping *blocks*. Using the L2 norm, the 4 histograms that make up block are normalised.

Finally, the histograms over the entire image are concatenated into a single feature vector which can be used for training and testing. The histograms of each cell are illustrated in Figure 2.5. An example of the HOG descriptor over a larger 256×256 image is provided in Figure 2.6.

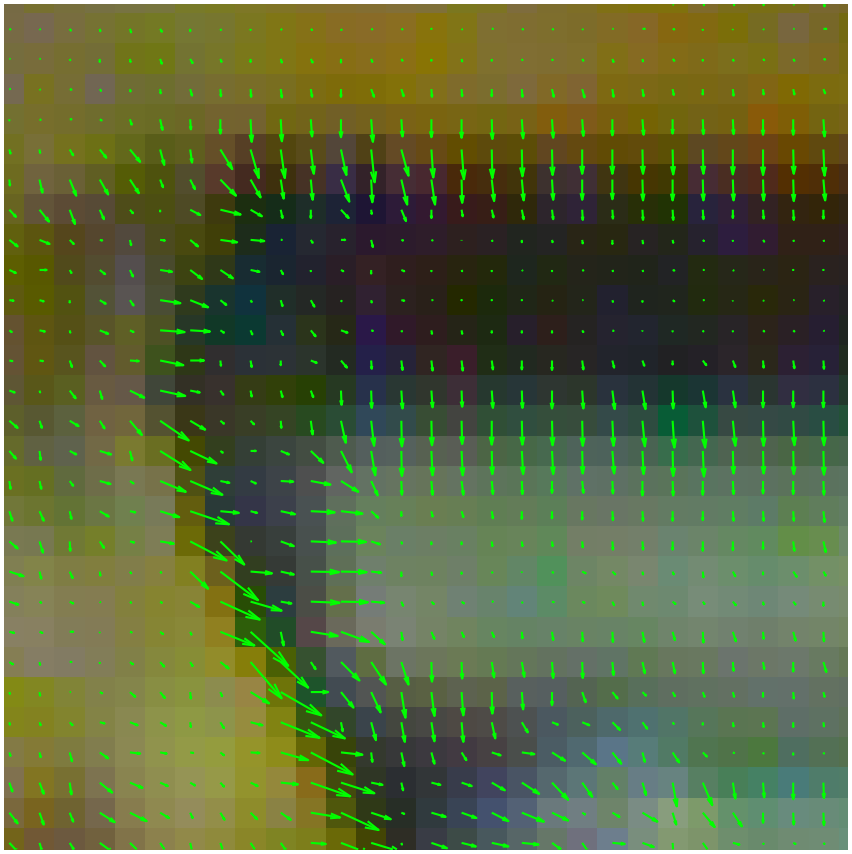


Figure 2.4: Gradient Vectors

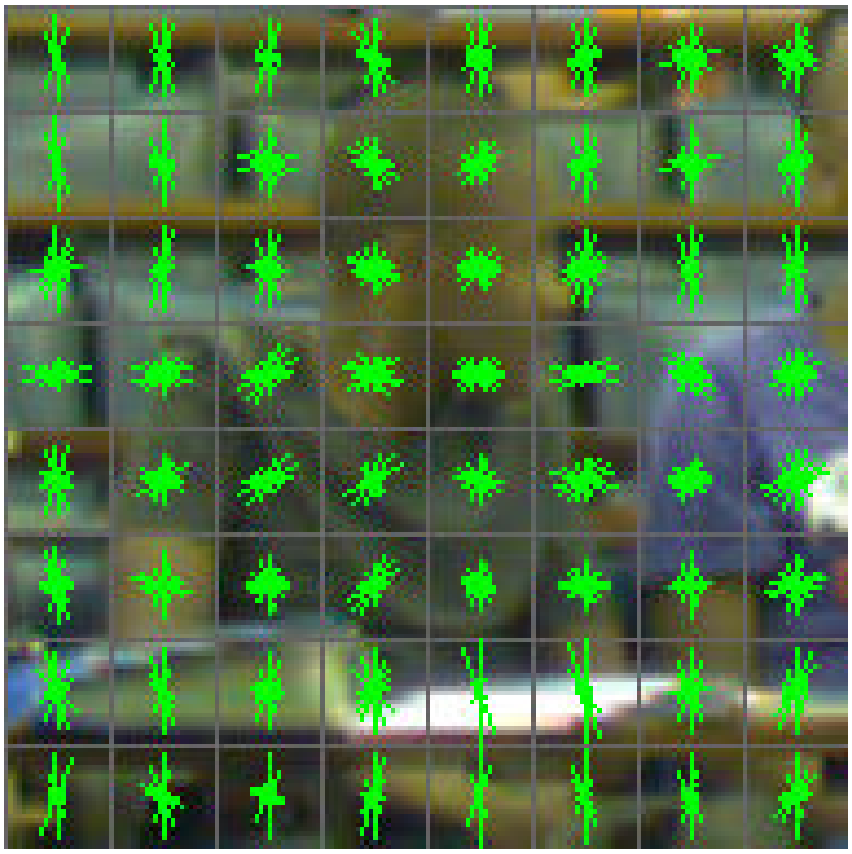


Figure 2.5: HOG Feature Descriptor

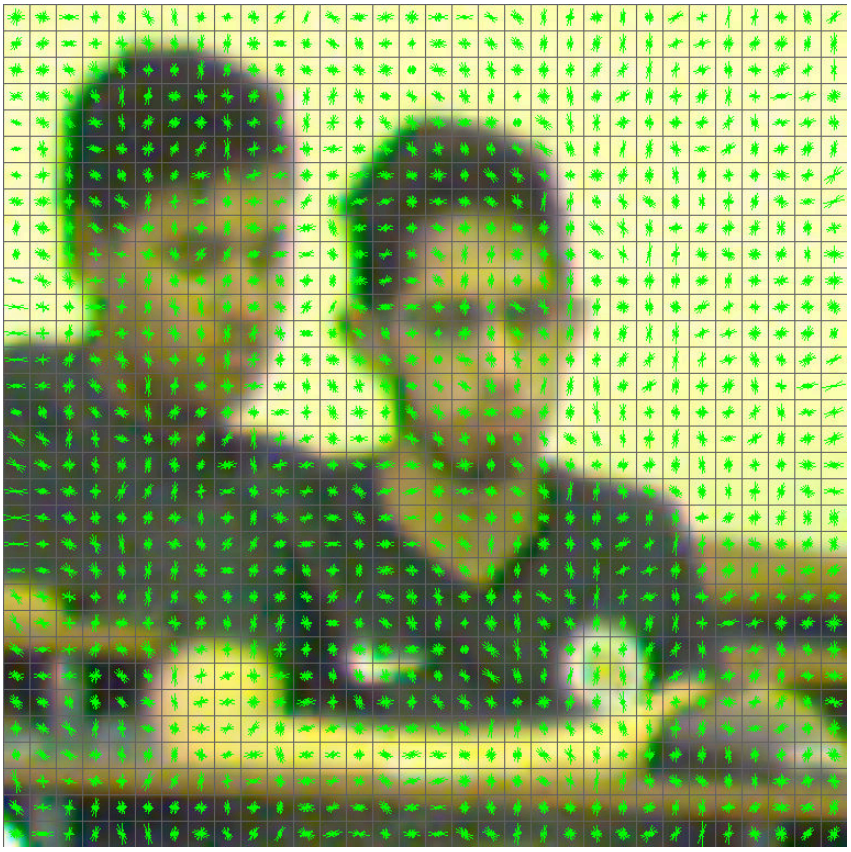


Figure 2.6: HOG Feature Descriptor Over a 256×256 Image

2.2.4 Local Binary Patterns

Originally proposed by Ojala *et al.* (1996), Local Binary Patterns (LBP) are used as texture descriptors. They work well for unsupervised segmentation tasks (Ojala and Pietikäinen 1999) and as features for face detection (Ahonen *et al.* 2006).

The original formulation uses 3×3 neighbourhoods. To calculate the response of the current pixel, its eight neighbours are thresholded using the current pixel's intensity. For the pixel at (x, y) consider its eight neighbours, (x_i, y_i), by moving around the centre pixel in a circle.

$$s(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & otherwise \end{cases} \tag{2.14}$$

$$LBP = \sum_{i=0}^{7} s(\mathbf{I}(x, y) - \mathbf{I}(x_i, y_i)) \cdot 2^i \tag{2.15}$$

This is illustrated in Figure 2.7. The current pixel has an intensity of 87. Each neighbour with a value greater than 87 is considered a 1, otherwise a 0. Moving circularly around the centre pixel starting at the top left, the eight neighbours are written to build an 8-bit binary number. This number is the LBP descriptor for the centre pixel.

LBPs were later extended to support multiple size neighbourhoods (Ojala *et al.* 2002). A circle is still traced around the centre pixel, but at some radius, r, and with p sampling points

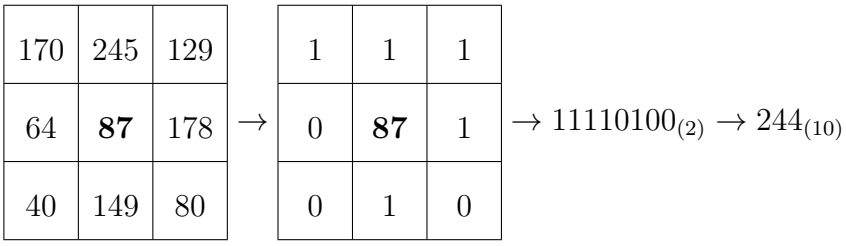


Figure 2.7: Local Binary Patterns

rather than eight. Bilinear interpolation is used to calculate the intensity where the sample point along the circle does not land directly on a pixel.

Once a local binary pattern is calculated for each pixel, they are grouped into cells and blocks just as the HOG features were. A normalised histogram over the different blocks can be calculated, which in turn can be used as the feature descriptor for subsequent machine learning methods.

Note that the HOG and LBP feature descriptors are neither scale nor rotationally invariant.

2.2.5 Haar-like Features

Haar-like features are based on the Haar Wavelet which can be used to represent a function as a decomposition of orthonormal functions in a manner analogous to a Fourier decomposition. The Haar wavelet is the discrete step function on a fixed interval shown in Equation (2.16).

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 0.5, \\ -1 & 0.5 \leq t < 1, \\ 0 & otherwise. \end{cases} \tag{2.16}$$

Viola and Jones (2001) used Haar-like features for the purpose of face detection. These features involve a number of rectangles as seen in Figure 2.8. The black and white areas correspond to values of -1 and 1 in a convolutional kernel. In general they make use of three types of features: 2-rectangle, 3-rectangle, and 4-rectangle type features. They used the boosting machine learning technique in conjunction with many cascaded weak classifiers to design a real-time face detector that has become the standard in face detection.

Importantly, Viola and Jones (2001) introduced the concept of an *integral image*. The integral image can be computed once for an image in $O(wh)$ operations where w is the width and h is the height. Once the integral image is available, the rectangular Haar-like features can be calculated at any location, over any scale in constant time – $O(1)$.

Given some image, $\mathbf{I}$, the integral image, $\mathbf{II}$, at some point (x, y) is defined as the sum of all the pixels above and to the left of (x, y) inclusive:

$$\mathbf{II}(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} \mathbf{I}(x', y'). \tag{2.17}$$

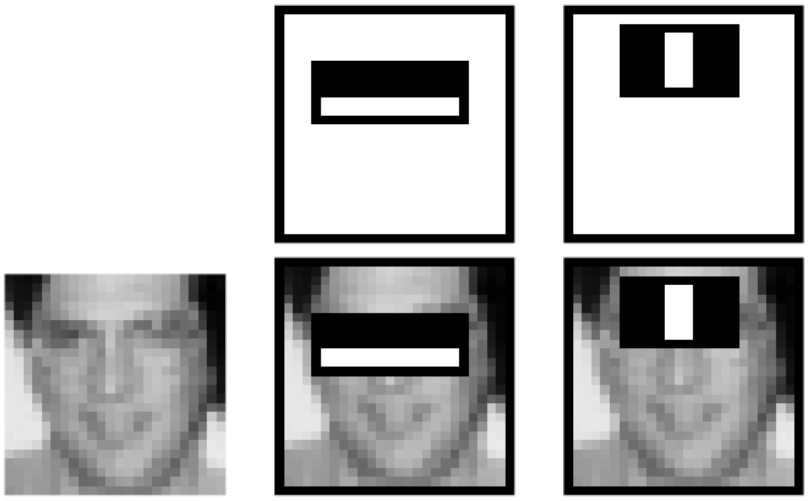


Figure 2.8: Haar-Like Features (Viola and Jones 2001)

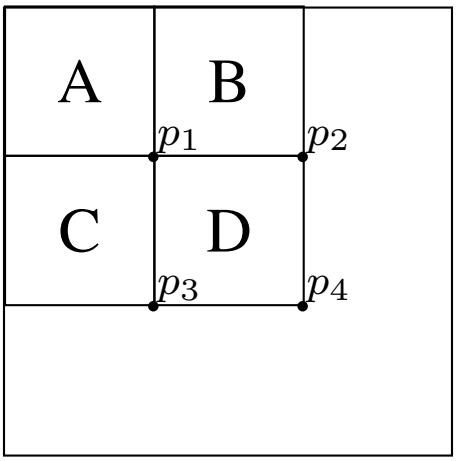


Figure 2.9: Integral Image Regions

It can be implemented using a dynamic programming approach with the following two recurrence relations:

$$\mathbf{S}(x, y) = \mathbf{S}(x, y - 1) + \mathbf{I}(x, y) \tag{2.18}$$

$$\mathbf{II}(x, y) = \mathbf{II}(x - 1, y) + \mathbf{S}(x, y). \tag{2.19}$$

$\mathbf{S}$ is the cumulative row sum and using these two formulae, the full integral image is computed from a single pass over the image.

The rectangular features are all constructed using blocks of ± 1. Contrary to normal convolution, the rectangular filter is convolved with the image at a particular position only and each position of the filter is considered a different feature. When performing convolution with these filters, it is sufficient to sum all the pixels in the positive region, all the pixels in the negative region, and then subtract the two subtotals. Using the integral image, the sum for each region can be computed in constant time by looking at the locations of the four corners.

70	245	129	38	237	90
173	87	178	66	89	212
167	149	227	214	50	149
41	57	245	65	64	140
30	192	140	208	157	234
127	65	35	62	121	73

(a) Original Image

70	315	444	**482**	719	809
243	575	882	986	1312	1614
410	891	1425	1743	2119	2570
451	989	1768	2151	2591	3182
481	1211	2130	**2721**	3318	4143
608	1403	2357	3010	3728	4626

(b) Integral Image

Figure 2.10: Using the Integral Image

Consider Figure 2.9: the value at point p_1 contains the sum of region A, the value at p_2 contains A+B, point p_3 contains A+C, and point p_4 contains A+B+C+D. To get the sum over D only the formula is $p_4 - p_3 - p_2 + p_1$

Figure 2.10 shows an image and its corresponding integral image. The sum over the highlighted region is just $2721 - 481 - 482 + 70 = 1828$. Summing the 12 pixel values from the original image results in the same value but takes linear time in the number of pixels being summed.

The Cascade Classifier detection algorithm from Viola and Jones (2001) is based on these features and is presented in detail, in Section 2.3.5.

2.3 Supervised Machine Learning Techniques

There are primarily two applications of supervised machine learning techniques: regression and classification (Marsland 2015). Regression focuses on function approximation or interpolation: fit a curve through the data so that it passes as close as possible to each point. For example, given a number of points generated by some linear function with noise, linear regression will try to learn coefficients for a straight line that best describes the data.

On the other hand, classification aims to divide the data into discrete categories. Based on the features presented to it, the system tries to learn boundaries between the different classes of data. For example, in Figure 2.11 there are three different classes. When learning, the system tries to find the parameters of a curve, hyperplane, or surface that best separates the different categories. A datum is then classified according to its position above or below these boundaries.

This work aims to classify students into a number of categories, therefore the focus is on classification and not regression. This section presents a number of classification techniques that are used in the rest of the thesis.

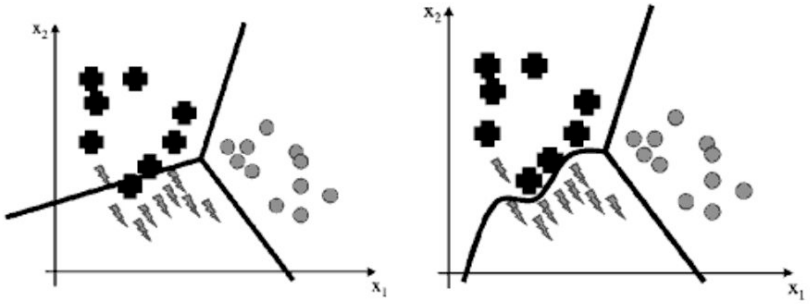


Figure 2.11: Classification Boundaries (Marsland 2015)

2.3.1 Feature Vectors

As discussed in the previous sections, machine learning algorithms perform learning, classification and regression on some *feature vector*. The feature vector is some numerical encoding of the original data which is the input into the classification or regression. The features can map directly onto the input data, or can provide statistical or summary information that encodes more abstract concepts.

For example, suppose a system is being trained to recognise South African bank notes as seen in Figure 2.12. A nïave way to pass the data to the system is to present a vector of the raw pixel intensities. This approach presents a challenge to the system due to the high dimensionality of the input. Because of this, the classifier will take longer to train, will need significantly more samples, and may not work as accurately or as fast as it otherwise could.



Figure 2.12: South African Rands (South African Reserve Bank 2017)

Looking at the notes, it is immediately evident that the notes could be easily classified by looking at the colour and shape only. By extracting the average colour or hue and measuring the length of the note, the classifier could be trained based on these two features only. Such a classifier will need fewer training samples and parameters which means it will both train and classify faster. It is likely that such a classifier will achieve a higher accuracy as a fewer number of parameters make it less prone to over-fitting where the classification boundary more resembles an interpolation of the points rather than finding overall trends in the data.

In general the selected features should be fast to compute, and should separate the data in a manner that the learning algorithm will be able to classify. For example, if the classifier can only differentiate between linearly separable input vectors then the features must represent the data in a way that supports this model. Poor feature selection may mean that the classifier is unable to accurately represent the underlying structure and it will perform poorly regardless of the number of training samples and training time.

Common features used in computer vision applications include the responses of various convolutional filters, edge detectors, Laplacian of Gaussians, Local Binary Patterns, and Haar-like features.

2.3.2 Linear Separability and Kernel Functions

Linear classifiers learn the parameters or weights of a straight line (2D), plane (3D), or hyper-plane (higher dimensionality) that separates the different categories of data. For example, in Figure 2.13 there are two categories of data and the classifier has learnt to separate them using the straight line

$$y(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \tag{2.20}$$

where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the input feature vector, and b is some constant bias. Classification on some input vector, $\mathbf{x_i}$, is performed by

$$L(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.21}$$

Training the classifier focuses on finding the best weight vector $\mathbf{w}$ so that Equation (2.21) classifies optimally.

As alluded to in the previous section, for this kind of classifier to work the feature vectors representing the data must be linearly separable. The data shown in Figure 2.13 on the next page is indeed linearly separable as it is possible to draw a straight line that correctly divides the data into the relevant categories. In many real-world examples the data is not linearly separable as seen on the left-hand-side of Figure 2.14.

In such cases a linear classifier will fail as no hyper-plane exists that correctly partitions the data. Alternative features should be considered and added to the representation. If there is no alternative data available, kernel methods can be used to help embed the features in a higher dimensional space.

As seen in Figure 2.14, by adding features corresponding to a second degree polynomial over the original values, the data now becomes linearly separable and a separating hyper-plane

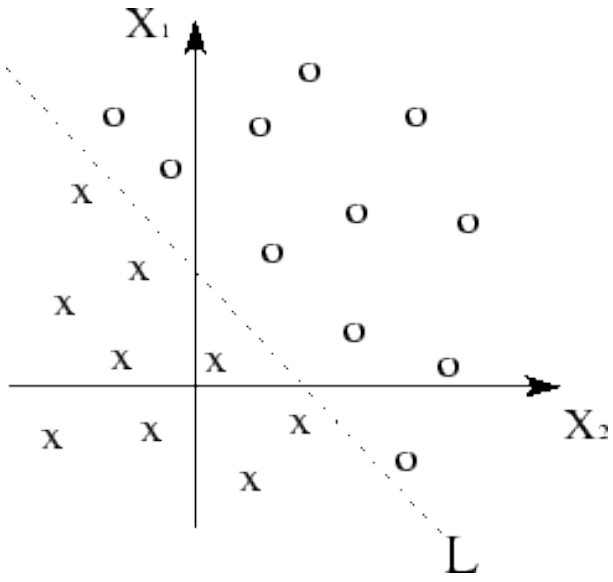


Figure 2.13: Linearly Separable Data (Kiyoshi Kawaguchi 2000)

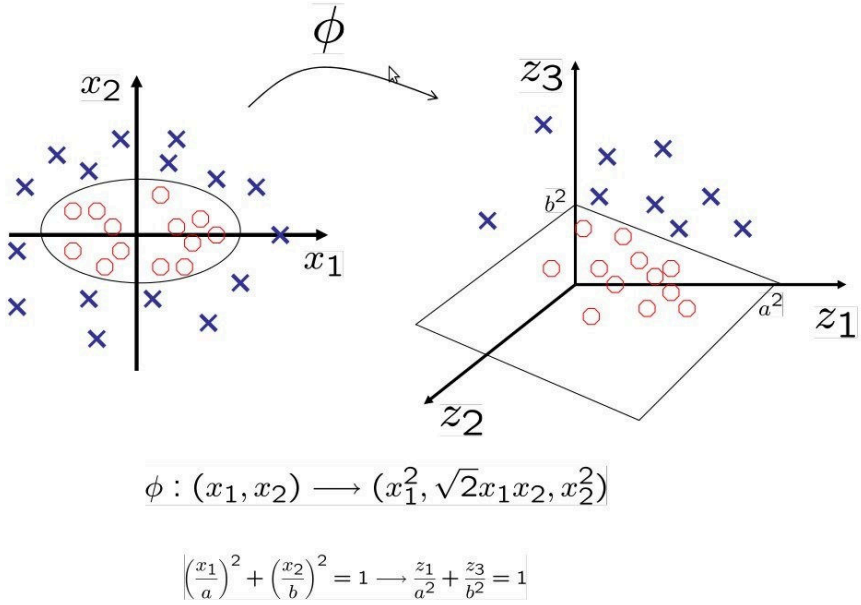


Figure 2.14: Adding Features for Linear Separability (Friedman *et al.* 2001)

can be found. The problem now is that the dimensionality of the feature space explodes in size and the algorithms generally become computationally expensive.

For the original 3D feature, $[x_1, x_2, x_3]$, the second degree polynomial feature $\Phi(\mathbf{x})$ becomes:

$$\Phi(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3]. \tag{2.22}$$

The size of the feature vector has gone from 3 to 10 dimensions. The number of dimensions is $O(d^s)$ where d is the original number of dimensions and s is the degree of the polynomial. The feature vector can be transformed using polynomials of degree s, sigmoid functions with parameters κ and δ, or with radial basis functions with parameter σ. Other transformations can even map the features to an infinite number of dimensions. In general the amount of work done increases exponentially or in the infinite case numerically considering these vectors becomes impossible (Marsland 2015).

In these cases the computational costs are too high and the methods fail. However, in linear classification the fundamental operation is usually the dot product of two vectors. To calculate the dot product of vectors of length n, $O(n)$ multiplications and additions are performed. Using the extended basis shown in Equation (2.22) this results in $O(d^s)$ operations. For example, using the second degree polynomial above, the dot product of two transformed vectors, $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$, is:

$$\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) = 1 + 2\sum_{i=1}^{d} x_i y_i + \sum_{i=1}^{d} x_i^2 y_i^2 + 2\sum_{i=2}^{d}\sum_{j=1}^{i-1} x_i x_j y_i y_j, \tag{2.23}$$

where d is the dimensionality of the original feature vector $\mathbf{x}$ (Marsland 2015). This results in $O(d^2)$ operations.

However, a *kernel function* that yields the same result can be used to replace the original dot product. Using the binomial theorem, or the multinomial theorem for higher degrees, the kernel, $\mathbf{K}$, is

$$\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) = \mathbf{K}(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^s \tag{2.24}$$

In the polynomial kernel in Equation (2.24), not only is the number of operations reduced from $O(d^2)$ to $O(d)$, but the extra values in the extended basis need not be calculated or stored at all. The kernel provides an efficient method to calculate the dot product of an extended feature set, without even having to calculate the extended features.

Equivalent kernels for the sigmoid (Equation (2.25)) and radial basis function (RBF, Equation (2.26)) expansions are provided below.

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta) \tag{2.25}$$

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) = \exp(-(\mathbf{x} - \mathbf{y})^2 / 2\sigma^2). \tag{2.26}$$

In summary, for a linear classifier to function, the data must be linearly separable. If it is not, expanding the feature set using polynomial, sigmoid, and radial basis functions can often provide the required separability. However, these expansions introduce a large number of extra dimensions to the feature space which can have a significant impact on the computational costs.

In linear classifiers the main operation is usually the dot product of vectors. Kernel functions provide an efficient way to calculate dot products over the extended feature vectors without needing to evaluate or even store the extended features.

2.3.3 A Linear Classifier: The Neuron

In the previous section, linear classifiers were briefly introduced alongside the importance of the linear separability of data. Kernel functions provide an efficient mechanism to deal with data that is not linearly separable and are used extensively by Support Vector Machines (SVMs) which are discussed in Section 2.3.4. Before considering the SVM, however, this section introduces the simplest linear classifier, known as the McCulloch and Pitts (1943) neuron. The neuron is illustrated in Figure 2.15 and is a simplified mathematical model of an actual neuron.

There are three parts to the neuron:

1. n input weights, labelled w_i, that correspond to the synapses that connect real neurons,
2. an adder that sums the weighted input, just as a neuron cell's membrane would accumulate an electrochemical charge, and
3. an activation function that decides whether the cell fires based on its input.

The features, x_i, are presented as input to the neuron and the neuron calculates the weighted sum as

$$h(\mathbf{x}) = \sum_{i=1}^{n} w_i \cdot x_i = \mathbf{w} \cdot \mathbf{x} \tag{2.27}$$

This value goes into the activation function, $g_\theta(h)$,

$$g_\theta(h) = \begin{cases} 1 & \text{if } h > \theta \\ 0 & \text{if } h \leq \theta, \end{cases} \tag{2.28}$$

which decides whether the neuron fires or not. This is equivalent to the linear classifier from Equation (2.21) where $b = -\theta$. This threshold or *bias* allows the adjustment of the neuron's response when the input values are all 0. Training the neuron involves the iterative adjustment of both the weights and the bias until the neuron classifies the data optimally.

To simplify the training process, the bias can be folded into the weights with a *bias node* that always has an input value of 1 as illustrated in Figure 2.16. w_0 corresponds to the bias weight, but can be trained exactly the same as the other weights.

The weights can be updated using the following rule:

$$w_i \leftarrow w_i + \eta(t - y) \cdot x_i, \tag{2.29}$$

where t is the target output, y is the actual output, and η is a parameter called the *learning rate* which is equivalent to the step size in numerical optimisation. If $t = y$ then the input is correctly classified and no adjustment is made. If $t = 1$ and $y = 0$ then the neuron did not fire when it should have. This means that the weighted sum of the inputs was not large enough and each weight is increased by some proportion, η, of the corresponding input. If the input for x_i

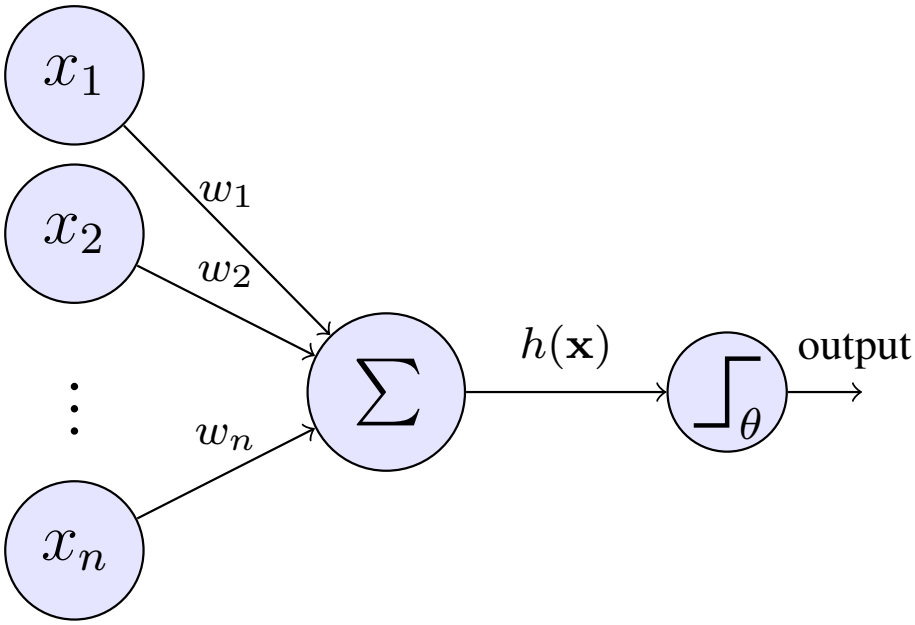


Figure 2.15: McCulloch and Pitts (1943) Neuron

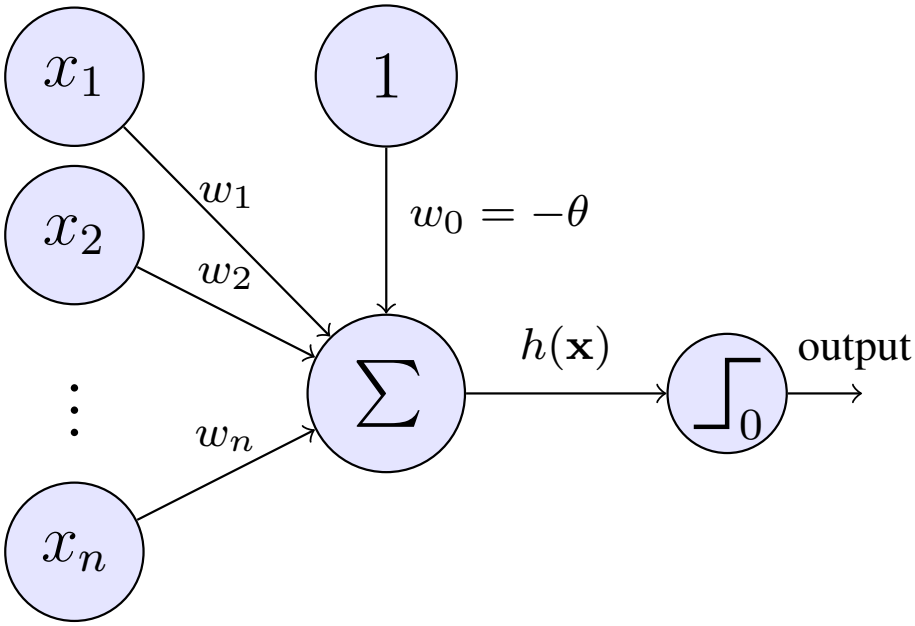


Figure 2.16: Neuron with Bias

was 0, then it did not contribute to the summation and the relevant weight is left unchanged. Contrastingly, if $t = 0$ and $y = 1$ then the neuron did fire when it should not have. In this case, the rule decreases the weight of all contributing synapses.

Performing this procedure over all the training samples corresponds to a single *epoch*. Multiple training epochs should be done, usually with a decreasing learning rate, until some maximum number of iterations or the required accuracy is reached.

The neuron finds a linear hyperplane to classify the data into one of two categories – where the neuron fires, and where it does not. However, there are no constraints on the actual hyperplane that separates the data. The SVM considered in the following section imposes an optimality constraint on the hyperplane that separates the two classes.

2.3.4 Support Vector Machines

The Support Vector Machine (SVM) is one of the most popular algorithms in modern machine learning. It was introduced by Vapnik (1995) and in its simplest form is a linear classifier that attempts to place a hyperplane between two categories of data. The neuron update rule in Equation (2.29) iteratively updates the decision boundary until an acceptable number of data points are correctly classified. The SVM, on the other hand, attempts to find a decision boundary that is *optimal* in some way.

Figure 2.17 illustrates three hyperplanes, H_1, H_2, and H_3. The first hyperplane, H_1, is obviously a bad choice as it incorrectly classifies most of the black circles. Both H_2 and H_3, however, classify all the data points correctly. This invites the question whether one plane is better than the other and if so, by what criteria are they are judged.

> “However, if you had to pick one of the lines to act as the classifier for a set of test data, I’m guessing that most of you would pick the line shown in the middle picture. It’s probably hard to describe exactly why you would do this, but somehow we prefer a line that runs through the middle of the separation between the data points from the two classes, staying approximately equidistant from the data in both classes. Marsland (2015)

This intuition stems from a sense that if the training data truly indicates some underlying process, then when new points are drawn some will be closer the decision boundary. Without more information there is no way of knowing where the actual dividing line sits, but by placing the separator down the middle of the gap it minimises the probability that new points will fall on the wrong side of the line and therefore be incorrectly classified. By building in an equal sized *margin* on each side, it also ensures that small variations in the features do not cause unintended misclassifications.

By requiring the margin to be maximal, it decreases the number of ways the decision boundary can be placed. As a result, the memory capacity of the model is decreased which can lead to a decrease in over-fitting and an increase in generalisability due to the *bias-variance trade-off* (Manning *et al.* 2008). This optimal hyperplane is illustrated in Figure 2.18.

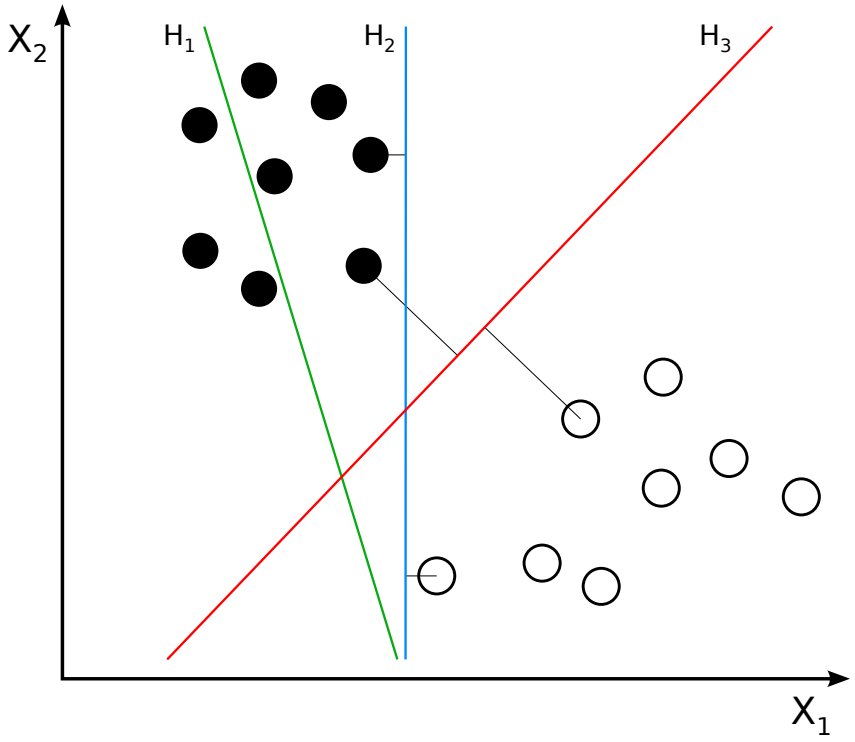


Figure 2.17: Separating Hyperplanes (Weinberg 2012)

The following derivation of the optimal hyperplane is based on those in Friedman *et al.* (2001), Manning *et al.* (2008), and Marsland (2015) and assumes that the data is linearly separable. Notation is similar to that of Manning *et al.* (2008).

Let the decision surface be the hyperplane, $\langle \mathbf{w}, b \rangle$ defined by

$$\mathbf{w} \cdot \mathbf{x} + b = 0. \tag{2.30}$$

Let the linear classification, $g(\mathbf{x})$, for the SVM be

$$g(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b), \tag{2.31}$$

where $\mathbf{w}$ contains the parameters of the decision boundary and the intercept, b, is some bias that is *not* folded into the weight parameters. g yields values from $\{-1, 1\}$ that correspond to each of the two classes. This differs from conventional linear classifiers that use $\{0, 1\}$.

Let the set of data points $\mathbb{D} = \{(\mathbf{x_i}, y_i)\}$, where $\mathbf{x_i} \in \mathbb{R}^l$ is the input vector, $y_i \in \{-1, 1\}$ is the label and l is the length of the input features.

Define the *functional margin* of the i^{th} data point, $\mathbf{x_i}$, with respect to $\langle \mathbf{w}, b \rangle$, to be

$$m_i = y_i \cdot (\mathbf{w} \cdot \mathbf{x} + b). \tag{2.32}$$

The *functional margin* of the entire dataset is then

$$m = \min_i (2m_i), \tag{2.33}$$

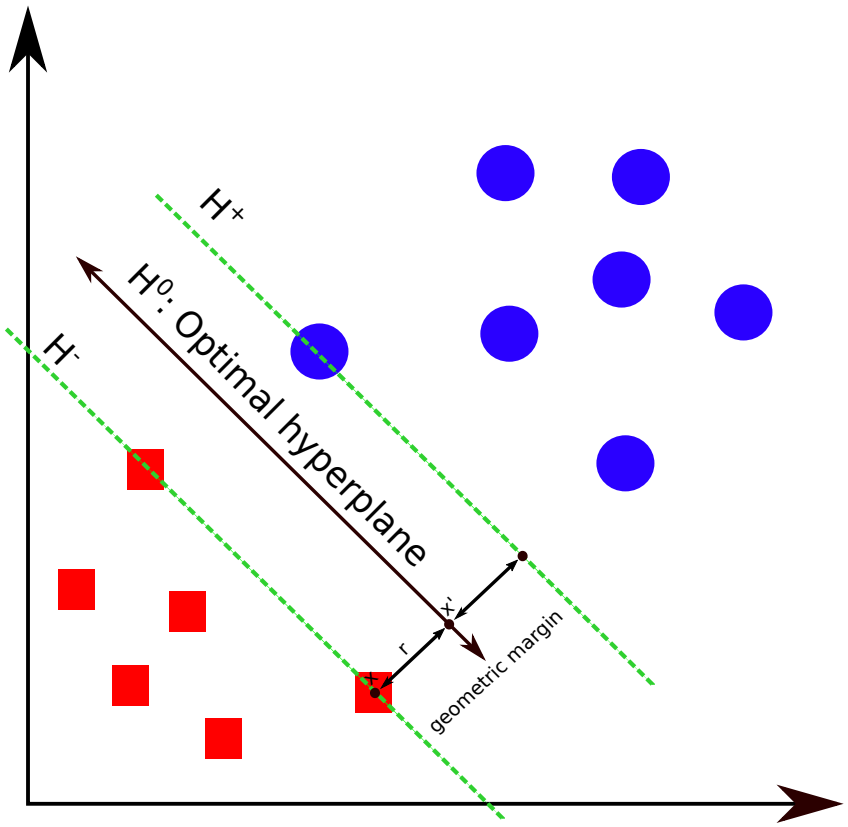


Figure 2.18: Optimal Separating Hyperplane

which is twice the minimum functional margin over all the points. The factor of 2 comes from measuring the width of the margin on each side of the hyperplane. At this stage there is still an unconstrained degree of freedom in the hyperplane – and therefore in the functional margin – as a constant scaling factor can be applied to $\mathbf{w}$ and b without changing the plane, i.e. $\langle \mathbf{w}, b \rangle$ and $\langle \alpha\mathbf{w}, \alpha b \rangle$ are equivalent hyperplanes, but result in different functional margins. Because of this, a constraint on the size of $\mathbf{w}$ is required.

Let r be the Euclidean distance from some point $\mathbf{x}$ to the decision boundary. The shortest path is the line that passes through $\mathbf{x}$ perpendicular to the plane. The unit vector in this direction is $\frac{\mathbf{w}}{||\mathbf{w}||}$. Let $\mathbf{x}'$ be the intersection of the decision boundary and this line. Therefore,

$$\mathbf{x}' = \mathbf{x} - yr\frac{\mathbf{w}}{||\mathbf{w}||} \tag{2.34}$$

where multiplying by y swaps the sign for when the point is above or below the decision boundary. Because $\mathbf{x}'$ is on the decision boundary,

$$\mathbf{w} \cdot \mathbf{x}' + b = 0. \tag{2.35}$$

Substituting Equation (2.34) into Equation (2.35) gives the following formula for r:

$$\mathbf{w}\cdot(\mathbf{x} - yr\frac{\mathbf{w}}{||\mathbf{w}||}) + b = 0 \tag{2.36}$$

$$\mathbf{w}\cdot\mathbf{x} - yr\frac{\mathbf{w}\cdot\mathbf{w}}{||\mathbf{w}||} + b = 0 \tag{2.37}$$

$$\mathbf{w}\cdot\mathbf{x} - yr\frac{||\mathbf{w}||^2}{||\mathbf{w}||} + b = 0 \tag{2.38}$$

$$\mathbf{w}\cdot\mathbf{x} - yr||\mathbf{w}|| + b = 0 \tag{2.39}$$

$$yr||\mathbf{w}|| = \mathbf{w}\cdot\mathbf{x} + b \tag{2.40}$$

$$r = \frac{\mathbf{w}\cdot\mathbf{x} + b}{y||\mathbf{w}||} \tag{2.41}$$

$$r = y\left(\frac{\mathbf{w}\cdot\mathbf{x} + b}{||\mathbf{w}||}\right). \tag{2.42}$$

In the final line, y can be moved from the denominator as its value can be only -1 or 1.

The data points with a minimal distance between them and the decision surface are termed *support vectors* and these vectors are ultimately responsible for the placement of the decision surface. The *geometric margin* is the maximum width of the band that can separate the support vectors in each class and is independent of the scaling mentioned earlier due to the normalising $||\mathbf{w}||$ term in r. Because of the independence of the geometric margin on scale, any scaling constraint can be imposed on $\mathbf{w}$. Therefore, we can choose to require that the functional margin of each point in the dataset is at least 1, and for at least one datum in each category to equal 1. In other words,

$$y_i(\mathbf{w}\cdot\mathbf{x_i} + b) \geq 1 \tag{2.43}$$

$$\exists \text{ support vectors } S \subset \mathbb{D}, \text{ such that } (\mathbf{x_i}, y_i) \in S \Rightarrow y_i(\mathbf{w}\cdot\mathbf{x_i} + b) = 1. \tag{2.44}$$

To calculate r for the support vectors, substitute Equation (2.44) into Equation (2.42):

$$r_i = \frac{y_i(\mathbf{w}\cdot\mathbf{x_i} + b)}{||\mathbf{w}||} \tag{2.45}$$

$$= \frac{1}{||\mathbf{w}||}. \tag{2.46}$$

Hence, the geometric margin, ρ, is

$$\rho = \frac{2}{||\mathbf{w}||} = \frac{2}{\sqrt{\mathbf{w}\cdot\mathbf{w}}}. \tag{2.47}$$

The aim of training the SVM is to find the decision surface that maximises this geometric margin. Therefore, training the SVM involves maximising ρ subject to the condition that

$$\forall(\mathbf{x_i}, y_i) \in \mathbb{D}, y_i(\mathbf{w}\cdot\mathbf{x_i} + b) \geq 1. \tag{2.48}$$

It is easy to see that the maximisation problem is solved by minimising the objective function $L(\mathbf{w})$, subject to the same condition.

$$L(\mathbf{w}) = \mathbf{w}\cdot\mathbf{w} \tag{2.49}$$

The solution can be found through the dual problem using Lagrange multipliers. A Lagrange multiplier, α_i is associated with each constraint in Equation (2.48) and the problem becomes:

$$\max_{a_i} \left(\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \right) \tag{2.50}$$

$$\text{such that} \sum_i \alpha_i y_i = 0, \text{ and } \alpha_i \geq 0 \; \forall \alpha_i \tag{2.51}$$

The SVM parameters are then:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x_i} \tag{2.52}$$

$$b = y_k - \mathbf{w} \cdot \mathbf{x_k}, \text{ for any } k \text{ where } a_k \neq 0, \tag{2.53}$$

and the support vectors, $\mathbf{x_i}$, are those that correspond to $a_i \neq 0$. This shows how the placement of the decision boundary relies on the support vectors only.

To handle cases where the SVM may have misclassified some data points because the data is not linearly separable, the objective function can be updated to:

$$L(\mathbf{w}) = \mathbf{w} \cdot \mathbf{w} + \lambda \sum_{i=1}^{R} \epsilon_i, \tag{2.54}$$

where ϵ_i is the distance to the correct side of the margin known as the *slack variable* and R is the number of incorrect classifications (Marsland 2015; Ramanan 2008). λ is then a parameter that controls the trade-off between a large geometric margin and a misclassified data point. This is called a *soft margin* SVM and α_i from Equation (2.51) is now bounded above by λ:

$$0 \leq \alpha_i \leq \lambda \; \forall \alpha_i. \tag{2.55}$$

Note that the only computation that ever involves $\mathbf{x}$ is the dot product seen in Equation (2.50). This means that while the method is fundamentally designed for data that is linearly separable, it lends itself to the types of kernel functions discussed in Section 2.3.2. By including polynomial, radial-basis, or sigmoid functions, data that is not linearly separable can be embedded in higher dimensional spaces. When using the data directly, this results in an explosion in the number of dimensions and therefore the storage and computational costs. However, the kernels provide a way to calculate the dot product of the extended feature vectors without ever having to construct them, and therefore retain the original computational complexity. Using the *kernel trick* in this way, SVMs can be used successfully in a number of cases where the data is otherwise not suitable to linear classification.

2.3.5 Boosting and Viola-Jones

Ensemble Learning is an approach where a number of classifiers are used to construct a single, strong, highly accurate classifier. The idea behind ensemble learning is that classification or

regression is made by votes cast by a committee of predictors. *Boosting* is an ensemble learning approach where multiple weak classifiers are trained in this way. Each weak classifier need only perform slightly better than chance. When querying the system, the response from each of these weak learners is then combined to form a composite predictor (Friedman *et al.* 2001; Marsland 2015).

Adaboost (Freund and Schapire 1995) is the most popular boosting algorithm. In Adaboost, some number, m, of weak classifiers, G_i, are trained on weighted data points. At the end of the algorithm the final output is the sign of the weighted sum of the classifiers.

At the start of the algorithm the observation weights, $w_i, i = 1, 2, \ldots, n$, are initialised to $\frac{1}{n}$. The first classifier, G_1, is trained on the data and its error rate is calculated. The weight, w_i, of each incorrectly classified data point or observation is then increased while the weights corresponding to incorrectly classified points are decreased. The next classifier, G_2, is then trained on the data so that it more carefully focuses on observations with a heavier weight, i.e. those that were misclassified by the previous stage. The error rate of this stage is calculated and the observations' weights are updated accordingly. The process repeats until all classifiers have been trained.

The final output is then:

$$G(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i G_i(\mathbf{x})\right) \tag{2.56}$$

where the coefficients α_i are related to the error rate of the classifier. By adjusting the weights at each level, the classifiers are forced to focus their attention on data points that were incorrectly classified on previous levels (Friedman *et al.* 2001).

The Viola and Jones (2001) algorithm performs visual object detection in images and uses the Haar-like features and integral image discussed in Section 2.2.5. They note that with a 24×24 base resolution, the exhaustive set of rectangle features has over 180,000 members. An algorithm would need to calculate these features for each position of a sliding window over multiple image scales. This is unsuitable for any real-time system and a more efficient approach was needed in spite of the dramatic performance improvements due to the integral image.

Inspired by the Adaboost method, they hypothesised that even with only a few the available features an effective classifier could be constructed. The challenge was finding the useful features in the set and constructing a strong classifier to use them. A weak classifier, $h_j(\mathbf{x})$, is defined that consists of a single feature, f_i, a threshold, θ_j and a parity, p_j, that adjusts the direction of the inequality:

$$h_j(\mathbf{x}) = \begin{cases} 1 & \text{if } p_j f_j(\mathbf{x}) < p_j \theta_j, \\ 0 & \text{otherwise,} \end{cases} \tag{2.57}$$

where $\mathbf{x}$ is a 24×24 image-patch and a f_i is a feature (2/3/4-rectangle Haar-like feature) in a specific position and scale. In practice none of these weak classifiers can function with significant accuracy, but through boosting significant improvements are found.

Viola and Jones (2001) propose an adaptation to the Adaboost approach where they use a degenerate decision tree in a *cascade* as illustrated in Figure 2.19. The cascade works by

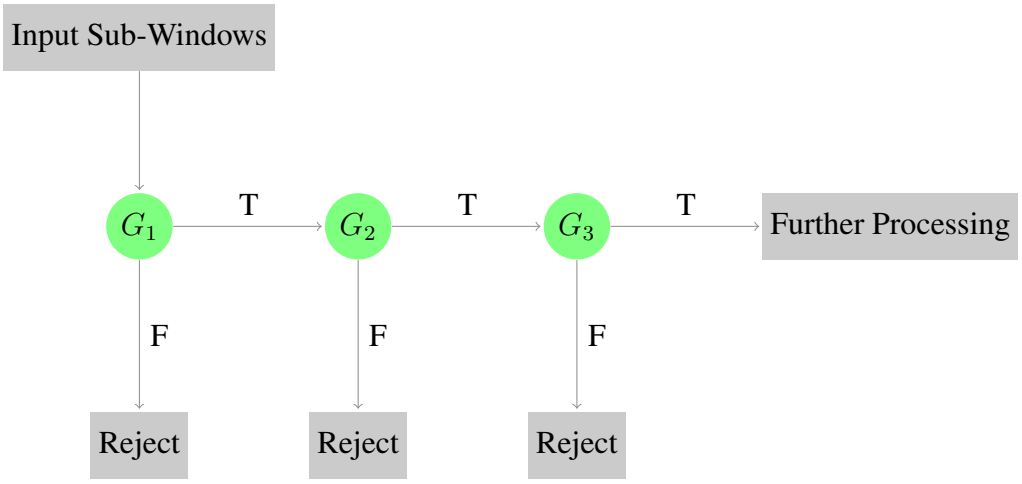


Figure 2.19: Cascade Classifier

identifying classifiers that are able to reject negative samples accurately, while still allowing all the potential positive candidates through. The idea is that a simple classifier can quickly reject image-patches or sub-windows that are definitely negative allowing the algorithm to focus instead on more probable candidates. In such a case, each stage of the process should prioritise a low false negative rate, over a false positive rate. This means that the classifier would rather pass a negative image on to the following stage than to accidentally reject a positive image.

Viola and Jones (2001) show that for the first stage a two-feature strong classifier can be built using Adaboost. The threshold can then be adjusted to minimise the false negatives at the expense of more false positives. They were able to get their classifier to detect 100% of faces, with a false positive rate of 40%. Suppose a hypothetical dataset of 50 faces and 50 non-faces: such a classifier would accept all 50 faces as well as 20 of the non-faces. By performing two extremely simple, constant time feature calculations this classifier is able to reject 30% of that hypothetical data which means that a more complicated classifier can be implemented for the second stage that needs to check fewer images.

The cascade design exploits the fact that the vast majority of image-patches are non-faces and should be rejected early. With each layer of the cascade, a stronger but more computationally expensive classifier is used. Training a cascade classifier is computationally expensive as the best weak classifiers need to be identified and then merged into a number of strong classifiers – this involves a lot of time spent training classifiers that are ultimately discarded. The trade off is that when correctly configured, such a classifier is able to work with very high reliability and in real-time.

2.3.6 Deep Learning and Convolutional Neural Networks (CNN)

The final approaches to machine learning presented in this work are related Artificial Neural Networks. Section 2.3.3 presented an artificial model of the neuron. The following methods all

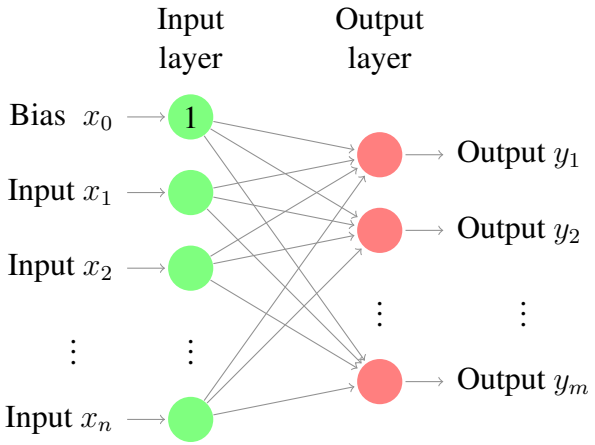


Figure 2.20: Single Layer Perceptron

use collections of these neurons together in some way to try improve the learning abilities of the overall algorithm.

The Perceptron

The Perceptron is simply a collection of multiple independent neurons. As shown in Figure 2.20 the feature vector presented to the network is augmented with an input that is always 1 – this bias node allows the threshold of the neurons in the output layer to be trained as a weight. The inputs are connected to a number of separate output neurons so that input x_i is linked to output y_j by weight w_{ij}. Together the output neurons form a vector $\mathbf{y}$, that is used to predict some target vector $\mathbf{t}$.

As each neuron is independent of the others, training is performed using the same rule as for a single neuron:

$$w_{ij} \leftarrow w_{ij} + \eta(t_i - y_i) \cdot x_i, \tag{2.58}$$

where the initial weights are randomly initialised to small values and η is the learning rate. Training continues until some acceptable accuracy or maximum number of iterations, T, is achieved. The complexity of the training algorithm is $O(Tmn)$.

Once trained, the weights can be represented in a matrix, $\mathbf{W} \in \mathbb{R}^{m \times (n+1)}$, and the output vector $\mathbf{y}$, is the activation function applied to the multiplication of $\mathbf{W}$ and augmented input feature vector:

$$\mathbf{y} = \text{sign}(\mathbf{W}\mathbf{x}). \tag{2.59}$$

Just as the neuron by itself is a linear classifier, the perceptron is also a linear classifier. It tries to find a hyperplane that separates the data for each output neuron. As such, a major limitation of the perceptron is that the data needs to be linearly separable.

A fundamental issue with the perceptron is the requirement of linearity in the data. However, by stacking neurons in multiple *layers* the model is able to represent more complex patterns and, given enough neurons and layers one can build a universal approximator that is able to represent any function. When stacking these layers, the model is called a Multilayer Perceptron (MLP) or Artificial Neural Network (ANN). As seen in Figure 2.21 the network is separated into the input layer, the hidden layer(s) and the output layer.

The input layer represents the feature vector, augmented with a 1 which acts as a bias for the activation function. Each input is connected to every neuron in the first layer. Each neuron in that layer is connected to every neuron in the following layer, and this pattern continues until the output layer is reached. Note that each hidden layer is also augmented with a bias node that always fires.

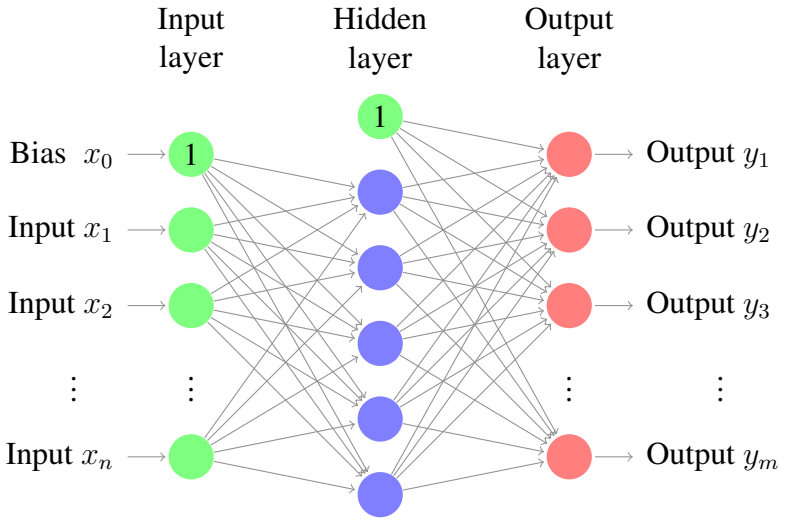
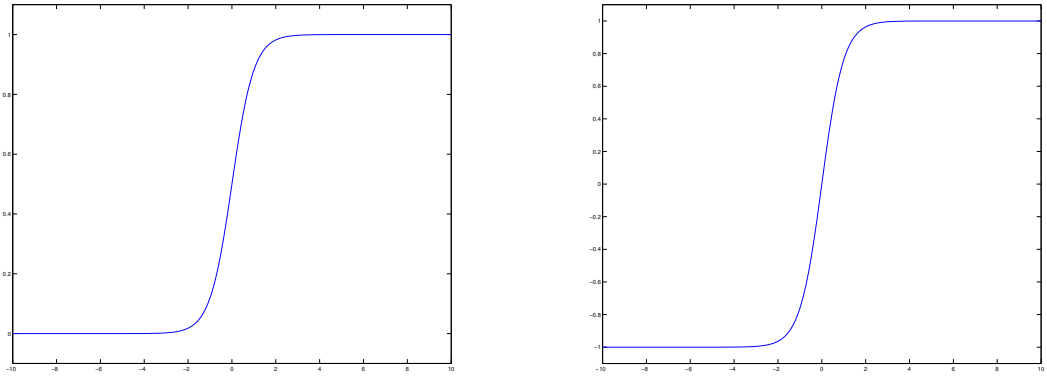


Figure 2.21: Neural Network/Multi Layer Perceptron

There are two aspects to the training algorithm, the forward propagation of activations and the backward propagation of error. In the forward part of the algorithm, the feature vector is presented at the input layer and the activation of each neuron in the hidden layer is calculated. These activations are propagated forward through each of the following layers until it reaches the output layer. The error, $E \in \mathbb{R}$, between the actual output, $\mathbf{y}$, and the target output, $\mathbf{t}$, is calculated using the sum of the square error at each neuron:

$$E(\mathbf{t}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^{m} (t_i - y_i)^2, \tag{2.60}$$

where m is the number of neurons in the output layer. The goal of training is to minimise this function using some numerical optimisation technique – usually (stochastic) gradient descent with momentum. The process of updating weights to account for errors is called back-propagation. The derivation of the back-propagation algorithm given below extends that presented in Marsland (2015).



(a) $\text{sig}(x), \beta = 2$ (b) $\tanh(x)$

Figure 2.22: Activation Function Candidates

In the version of the MLP described above there is a discontinuity at each node due to the threshold activation function. This threshold can be replaced by either the sigmoid or hyperbolic tan functions which are both smooth, continuous, and can be used to approximate the threshold step function. The curves of each function are shown in Figure 2.22 with definitions in Equations (2.61) and (2.62). Note that the range of $\tanh(\cdot)$ is $(-1, 1)$ and should be scaled to $(0, 1)$.

$$g(x) = \text{sig}(x) \quad = \frac{1}{1 + e^{-\beta x}}, \tag{2.61}$$

$$g(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.62}$$

Now that the activation function at each node, $g(\cdot)$, is continuous and differentiable, so is the error function. As the error function is also convex, usual numerical optimisation techniques can be used.

Note that the derivative of the sigmoid activation function, $g(x)$, is:

$$\frac{dg}{dx} = g(x) \cdot (1 - g(x)) \tag{2.63}$$

and that the descent direction of a convex function, f, is $-\nabla f$.

The standard back-propagation algorithm is based primarily on the chain rule from calculus. Suppose that we are training an MLP with N_i inputs labelled x_i, N_h hidden neurons labelled H_i, and N_o output neurons labelled O_i where neurons use the sigmoid activation function.

Let v_{ij} be the weight connecting the i^{th} input and the j^{th} hidden neuron. Let w_{jk} be the weight connecting the j^{th} hidden neuron and the k^{th} output neuron.

In the forward pass, the activation of each hidden neuron, h_j, and output neuron, y_k, is:

$$h_j = g\left(\sum_i v_{ij}x_i\right) = \left(1+\exp(-\beta\sum_i v_{ij}x_i)\right)^{-1} \tag{2.64}$$

$$y_k = g\left(\sum_j w_{jk}h_j\right) = \left(1+\exp(-\beta\sum_j w_{jk}h_j)\right)^{-1} \tag{2.65}$$

Using the chain rule, the derivative of the error function (Equation (2.60)) with respect to w_{jk} is:

$$\frac{\partial E}{\partial w_{jk}} = -(t_k - y_k)\cdot y_k \cdot (1-y_k)\cdot h_j = -\delta_k^{\text{out}}\cdot h_j \tag{2.66}$$

where the error contribution for output k is

$$\delta_k^{\text{out}} = (t_k - y_k)\cdot y_k\cdot(1-y_k). \tag{2.67}$$

The weights between the hidden and output layer, $\mathbf{w}$, are adjusted by taking a small step in the descent direction:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla_{\mathbf{w}}E \tag{2.68}$$

$$w_{jk} \leftarrow w_{jk} + \eta\cdot(t_k - y_k)\cdot y_k\cdot(1-y_k)\cdot h_j, \tag{2.69}$$

$$w_{jk} \leftarrow w_{jk} + \eta\cdot\delta_k^{\text{out}}\cdot h_j, \tag{2.70}$$

where η is the learning rate that adjusts the step size and $\nabla_{\mathbf{w}}$ is the gradient operator with respect to $\mathbf{w}$ only.

Using the chain rule, the change in error with respect to the first layer weights is:

$$\frac{\partial E}{\partial v_{ij}} = \sum_k (t_k - y_k)(-1)(y_k)(1-y_k)(w_{jk})(h_j)(1-h_j)(x_i) \tag{2.71}$$

$$= -\sum_k (\delta_k^{\text{out}})(w_{jk})(h_j)(1-h_j)(x_i) \tag{2.72}$$

$$= -\delta_j^{\text{hidden}} x_i \tag{2.73}$$

where error attributed to each hidden neuron can be calculated as the weighted sum of its connections to the output nodes:

$$\delta_j^{\text{hidden}} = h_j\cdot(1-h_j)\sum_k \delta_k^{\text{out}} w_{jk}. \tag{2.74}$$

Again, the weights between the input and hidden layer, $\mathbf{v}$, are adjusted by taking a small step in the descent direction:

$$\mathbf{v} \leftarrow \mathbf{v} - \eta\nabla_{\mathbf{v}}E \tag{2.75}$$

$$v_{ij} \leftarrow v_{ij} + \eta\sum_k (\delta_k^{\text{out}})(w_{jk})(h_j)(1-h_j)(x_i) \tag{2.76}$$

$$v_{ij} \leftarrow v_{ij} + \eta\delta_j^{\text{hidden}} x_i, \tag{2.77}$$

where $\nabla_{\mathbf{v}}$ is the gradient operator with respect to $\mathbf{v}$ only.

In general, if the neuron layers of the network are numbered from 0 to N where layer 0 corresponds to the input values and N the output values. Let h_j^l be the activation of neuron j in layer l. Let w_{ij}^l be the weight of the connection from h_i^{l-1} to h_j^l. Let the error contribution of h_j^l be δ_j^l. Then:

$$\delta_j^N = (t_k - h_j^N) \cdot h_j^N \cdot (1 - h_j^N) \tag{2.78}$$

$$\delta_j^l = \sum_k \delta_k^{l+1} w_{jk}^{l+1} \tag{2.79}$$

where $l \in \{1, 2, \ldots, N-1\}$. The update rule for w_{ij}^l becomes:

$$w_{ij}^l \leftarrow w_{ij}^l + \eta \delta_j^l h_i^{l-1}, \tag{2.80}$$

for $l \in \{1, 2, \ldots, N\}$.

This allows training of the weights in a network with an arbitrary number of layers and neurons per layer. The order in which training samples are presented to the network should be randomised and the network should train on each data point multiple times throughout the training phase. If the dataset is very large, random subsets or batches of the data can be used for each epoch and the optimisation step becomes stochastic. It is useful to add momentum to the optimisation to help avoid local minima, in this case the update rule becomes:

$$w_{ij}^{l,t+1} \leftarrow w_{ij}^{l,t} + \eta \delta_j^l h_i^{l-1} + \alpha \Delta w_{ij}^{l,t-1}, \tag{2.81}$$

for $l \in \{1, 2, \ldots, N\}$, t represents the current training update, $\Delta w_{ij}^{l,t-1}$ represents the previous update, and $\alpha \in [0, 1]$ is a parameter affecting the influence of momentum.

Given the activation functions above, enough layers and enough neurons, the MLP is able to approximate any function and is known as a *universal approximator*. This means that is has significant learning capacity, even for data that is not linearly separable, but that it potentially suffers from *over-fitting*. This causes poor generalisation on new data because the network has enough parameters to simply interpolate the training data. In this case, it does not have to identify the underlying process that generated the data. When using neural networks with many parameters, one should be cautious to employ proper validation techniques to ensure that the model is not over-fitting the data.

Deep Learning with Neural Networks

For a long time the number of layers in neural networks were limited to only a few. The field of deep learning focuses on the use of networks with many layers. In such networks, each subsequent layer is forced to learn more and more abstract concepts in some hierarchy. In general, machine learning techniques are extremely sensitive to the representation of the data. For example, in some representations the data may not be linearly separable, but by performing some transformation of the data it could be represented in a way that was. This fundamentally changes what methods will be effective when trying to classify such data.

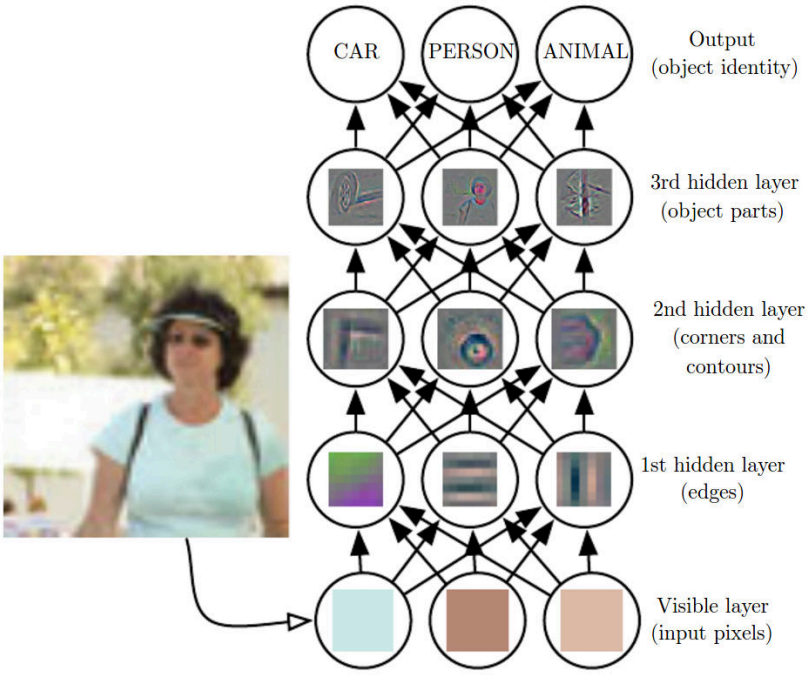


Figure 2.23: Feature Learning (Goodfellow *et al.* 2016)

In deep learning approaches, the idea is that the learning system must learn to extract the features themselves. This is known as *representation learning* (Goodfellow *et al.* 2016). In a deep neural network, for example, the first layers must learn to extract low level features from the raw data, while the latter layers each build a progressively more abstract representation based on these features.

Figure 2.23 shows how a deep neural network may learn to classify images into different types of objects. The raw image pixels are used as input into the network. In this case, the first layer has learned to recognise various types of edges – discontinuities in colour and brightness. These edges are then passed to the second layer which has learnt to identify corners and contours from them. While the first layer may only be able to recognise horizontal and vertical edges, the second layer is able to merge them together into some more abstract concept, namely contours. The next layer then takes the contours and starts to look for parts that correlate with the objects it is trying to recognise. In this case, it will have learnt that contours in a specific pattern usually indicate that the image is a person rather than a car.

A number of different approaches to deep learning exist. This work makes use of Deep Convolutional Neural Networks (CNNs) which are popular in computer vision.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were first introduced by LeCun *et al.* (1989). They noticed that feature maps based on convolutional filtering were popular and effective choices when engineering features for use with MLPs on computer vision problems. They proposed that by forcing hidden neurons to use only local sources of information at lower levels, local

features could be extracted from the underlying image. At the same time, these features could often be found in multiple places in the image, so by using *weight sharing* between the nodes in the layer, the layer would be able to recognise these local features at any location in the input.

This approach effectively means that all the neurons in a hidden layer are forced to use the same weights and that the connections of these weights to the input data are local only. In practice, a *convolutional layer* was designed. Each node in the hidden layer uses the same linear convolution kernel, which is applied over all the pixels in the input data. Training the layer means adjusting the weights inside the kernel so that it produces the best possible feature map across the input image. This architectural change means that the layer is forced to try learn features of the image that the other layers can use to build more abstract representations for classification. Over time different ways of applying the filter have been identified. Some parameters of a convolutional kernel include its size (width and height) and stride (apply the filter every n pixels). A number of different types of layers have since been developed and are used to build different CNN architectures. Some of these layer types are discussed below.

Layer 2.1. Convolution
This is the primary vision oriented layer. A convolution layer contains a number of learnable filters that each produces a feature map. Parameters include: the number of filters, the kernel width and height, and the stride.

Layer 2.2. Pooling
Pooling layers down-sample neighbourhoods of neurons in a feature map produced by convolution. The layer applies a pooling unit that returns a single number as a summary of that neighbourhood. Different types of pooling layers include max-pooling, min-pooling, and average-pooling. These layers return the maximum, minimum and average of the values in the neighbourhood. Parameters include: the width and height of the pooling units, a stride, and a pooling type. Traditionally the pooling unit size and the stride are equal. Krizhevsky *et al.* (2012) found that by having a smaller stride, and therefore *overlapping pooling*, networks were less prone to over-fitting, meaning accuracy and generalisation improved.

Layer 2.3. Rectified Linear Unit (ReLU)
The ReLU is an activation layer that transforms some input through the following activation function:

$$a(x) = \begin{cases} \alpha x & \text{if } x \leq 0, \\ x & \text{otherwise,} \end{cases} \tag{2.82}$$

where α is the negative slope parameter. When α is 0 the unit is an ReLU, otherwise it is a Leaky-ReLU. The unit passes positive input to the next layer, but generally forces the network to ignore input values that are negative. This increases non-linearity and sparsity in the network which also assist with generalisation. This type of layer uses only the one parameter.

Layer 2.4. Local Response Normalisation (LRN)
Local response normalisation performs *lateral inhibition* which is exhibited by real neurons in the human brain. Adjacent neurons must compete with each other where highly active layer

inputs inhibit others. The normalisation output is

$$b^i_{x,y} = a^i_{x,y} \cdot \left(k + \alpha \sum_j (a^j_{x,y})^2 \right)^{-\beta}, \tag{2.83}$$

where the sum runs over the adjacent kernel maps at position (x, y), k, α, and β are hyper-parameters, $a^i_{x,y}$ is the response of kernel i at (x, y) and b is the output of the LRN. Over the ImageNet database, Krizhevsky *et al.* (2012) found $k = 2$, $\alpha = 10^{-4}$, and $\beta = 0.75$ worked well after applying an ReLU to the input first. They suggest that neighbourhoods of 5 consecutive feature maps should be used for the sum, where the order of the maps is arbitrarily chosen and fixed before training begins.

Layer 2.5. Dropout

Dropout can substantially help to avoid over-fitting. A dropout layer works by randomly setting the value of each input neuron to zero, with probability of 0.5. This means that roughly half of the neurons in the layer are ignored during a training pass (both forwards and backwards). This results in a number of architectures being trained simultaneously as neurons are forced to learn to represent features without relying on the credibility of other neurons. By forcing the system to learn features with random subsets of neurons, co-adaptations of neurons can be avoided which helps stop over-fitting (Hinton *et al.* 2012). When actually using the network, the neurons' activations are all multiplied by 0.5 to approximate the geometric mean of all the random subsets of neurons used throughout training, but in this case neurons are not zeroed.

Layer 2.6. Softmax

Softmax normalises the values of the inputs so that the sum is one. It is usually used as the final layer in a network where each of the output neurons correspond to a specific category. In this way, the output of each neuron can be considered to be the probability of the input belonging to each trained class. Where a_i is the i^{th} input, Softmax applies the following normalisation:

$$b_i = \frac{a_i}{\sum_i a_i}. \tag{2.84}$$

Layer 2.7. Fully Connected

The fully connected layer is the standard neuron layer illustrated in Figure 2.21. If there are n inputs and m neurons, then $m(n + 1)$ weights, $\mathbf{W} \in \mathbb{R}^{m\times(n+1)}$, are learned and the output, $\mathbf{y}$, is

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \tag{2.85}$$

where $\mathbf{x}$ is the input vector augmented with a 1 at the front to handle the bias.

The layers introduced above were used in the AlexNet CNN and achieved ground breaking accuracy on the ImageNet database which consists of 1.3 million high-resolution images (Krizhevsky *et al.* 2012). When predicting the top five labels for an image from a pool of 1,000 options, they achieved an error rate of 18.9% – which considerably outperformed the state of the art. AlexNet itself consists of 60 million parameters (weights) and 500,000 neurons. There are five convolutional layers, each followed by pooling and normalisation layers, which feed into two layers of fully connected neurons. Finally a Softmax layer normalises the output over

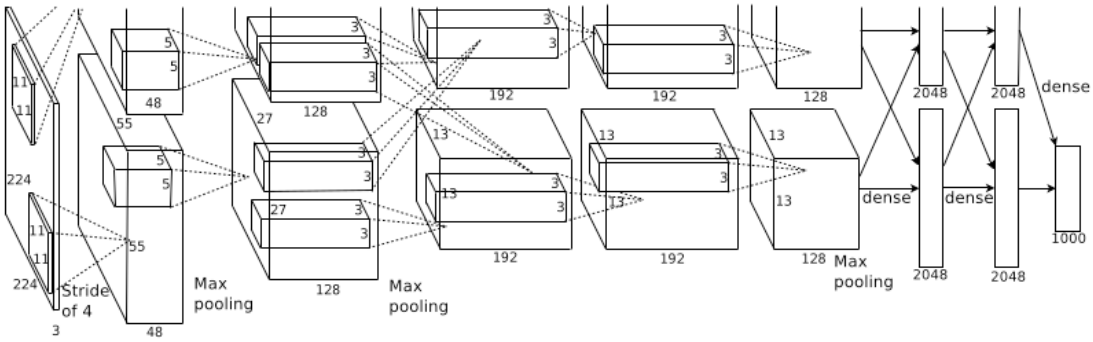


Figure 2.24: AlexNet Architecture (Krizhevsky *et al.* 2012)

1,000 output neurons that each correspond to a different category of image. This is the foundation of the Deep Convolutional Neural Network in Chapter 7. The AlexNet architecture is illustrated in Figure 2.24.

2.4 Incorporating Multi-Sensor Data

There are three primary approaches that can be used to merge data from multiple sensors. Specifically these are Data Fusion, Feature Fusion and Decision Fusion (Sharma *et al.* 1998). If the selected learning algorithm and feature representation does not support temporal data directly, these approaches can be used to merge temporal data as well. This section explains these different approaches to multi-sensor and temporal learning.

In the sections that follow, the methods will be illustrated using an image sequence, $\mathbf{I}_s$, where s is a student or subject. $\mathbf{I}_s(n)$ is then a specific frame in $\mathbf{I}_s$, where n is the frame number in the video. $\mathbf{x} = [x_1, x_2, \ldots, x_l]^T \in \mathbb{R}^l$ is then a feature vector of length l. The methods each describe how to fuse m video frames to produce an output.

2.4.1 Data Fusion

If the sensors have common temporal resolutions, the raw data of the one sensor can simply be appended to the raw data of the other. For example, the information about pixels in a video frame could be appended to information from a physiological sensor that provides data at the same time frequency. This is termed Data Fusion. The same approach can also be used to handle video information with features that do not directly support temporal information.

For example, temporal information can be included by concatenating consecutive frames and then calculating the HOG features over the resulting image. Let $I_s^m(n)$ be the image from which to extract features:

$$I_s^m(n) = [I_s(n); I_s(n-1); \ldots; I_s(n-m-1)]. \tag{2.86}$$

A feature vector, $\mathbf{x}$, for training and classification can then be constructed by calculating the HOG features over the image $I_s^m(n)$. Images and feature vectors resulting from this equation are illustrated in Figure 2.25 for $m = 1$ (Figure 2.25a) and $m = 4$ (Figure 2.25b). A careful examination of the features at the top of Figure 2.25a and the corresponding cells in Figure 2.25b, show immediate issues with data fusion in this context.

Consider the cell in the top right of Figure 2.25a. This cell indicates a strong horizontal gradient in that area, shown by the dominant vertical green lines. Once the previous frames are concatenated the HOG response in that area changes substantially. The HOG features are now strongly influenced by the discontinuity in the gradients over the image seam. The classification method must now learn to ignore the noise introduced by these gradients which in turn may influence the training time and classification accuracy.



(a) 1 Frame (b) 4 Frames

Figure 2.25: HOG Features with Data Fusion

2.4.2 Feature Fusion

In the feature level approach, the features are calculated over each sensor input or video frame separately and then concatenated.

Let the feature vector of each frame be $\tilde{\mathbf{x}}(n) \in \mathbb{R}^{\tilde{l}}$, the final feature vector $\mathbf{x} \in \mathbb{R}^{l}$ is then

$$\mathbf{x} = [\tilde{\mathbf{x}}(n); \tilde{\mathbf{x}}(n-1); \ldots; \tilde{\mathbf{x}}(n-m-1)] \tag{2.87}$$

where $l = \tilde{l} \cdot m$ where ; indicates concatenation of the column vectors.

Feature fusion helps to avoid problems that are caused by data concatenation when employing features that make use of spatial information. By concatenating the features, artificial discontinuities in the data are not introduced.

2.4.3 Decision Fusion

When using decision fusion the data from separate sensors or temporal measurements are each classified separately first and then joined in some logical way. This is a common approach in Human Computer Interaction (HCI) research.

Using this approach, each frame of a video may be processed independently both for feature extraction and classification. Once each frame is classified, the results may be merged in some logical or statistical way. For example, to detect engagement, the system may classify each frame individually and then use a rolling average over the last m frames to construct a sensible output.

Other more sophisticated techniques such as Hidden Markov Models (HMMs) may be used to merge the independent classifications. For example an HMM could track the state of a subject over time and then view the next frame as a new observation to update the current state.

2.5 Dimensionality Reduction

An image consisting of 4 frames, such as Figure 2.25b, generates a HOG feature vector of length 44,100. For 80,000 images this results in 52GB of data, which is not feasible for off-line machine learning methods like Support Vector Machines (SVM), that require all training data be loaded into memory at once.

Most interesting problems in data mining and machine learning involve the use of data with very high dimensionality (a large number of measurements per record). While more measurements often mean more information, the state/search/parameter space of many methods increases exponentially in the number of dimensions. This means that to adequately cover the space more records are required for both training and testing, which has implications for training and testing times, as well as the storage and memory requirements. This is known as the *curse-of-dimensionality* (Vlachos *et al.* 2002).

In many cases the measurements in the various dimensions are highly correlated meaning that there is a lower dimensional subspace that effectively captures the true variance in the data. In this event, dimensionality reduction techniques can project the data onto this subspace in a way that keeps only the important information. This yields a more compact representation of the underlying structure of the data (Witten *et al.* 2016).

Vlachos *et al.* (2002) identify both non-linear and linear methods which can be classified into two categories:

1. Local / Shape preserving
2. Global / Topology preserving

The former category attempts to find a simpler representation of each record in the dataset by retaining the most important information for each record. For example, a Fourier (Agrawal *et al.* 1993; Faloutsos *et al.* 1994) or wavelet (Chan and Fu 1999) decomposition of a signal can discard the high frequency coefficients while preserving most of the original energy. The latter category focuses on finding a minimal subspace that preserves the overall variance in the dataset. The most common approach in this category is Principal Component Analysis (PCA).

This work makes use of PCA for dimensionality reduction. An explanation of the method is presented in the next section.

2.5.1 Principal Component Analysis (PCA)

Figure 2.26 shows a number of highly correlated 2D data points. While this data initially looks like 2D data, the majority of the variance is accounted for by the green line. By rotating and translating the axes to the green and red lines, the data can be perfectly represented. However, by only looking at the value along the green-axis, one can approximate the position of each data point with high accuracy. By doing so, the 2D data has been reduced to 1D data. In fact, by making the proposed projection of each point onto the green-axis, 98.94% of the variance in the dataset is retained, while the size of each data point has been halved.

The goal of PCA is to final a new orthonormal basis set for the given data. This is done by finding the *principal components* that best describe the points. In the example above, the first principal component corresponds to the green line, while the second principal component corresponds to the red one. The components are the eigenvectors of the covariance matrix and the variance accounted for by each one is proportional to the size of the corresponding eigenvalue. Figure 2.27 shows the eigenvalues corresponding to the first (green) and second (red) principal components. The rest of this section provides a mathematical definition of the principal components.

PCA aims to find the orthonormal transformation $\mathbf{A} \in \mathbb{R}^{l \times k}$ that maps between a data point or feature vector $\mathbf{x} \in \mathbb{R}^l$ and the reduced vector $\mathbf{y} \in \mathbb{R}^k$, where k is the dimensionality of the reduced feature space. The forward mapping is performed using

$$\mathbf{y} = \mathbf{A}^T \mathbf{x}. \tag{2.88}$$

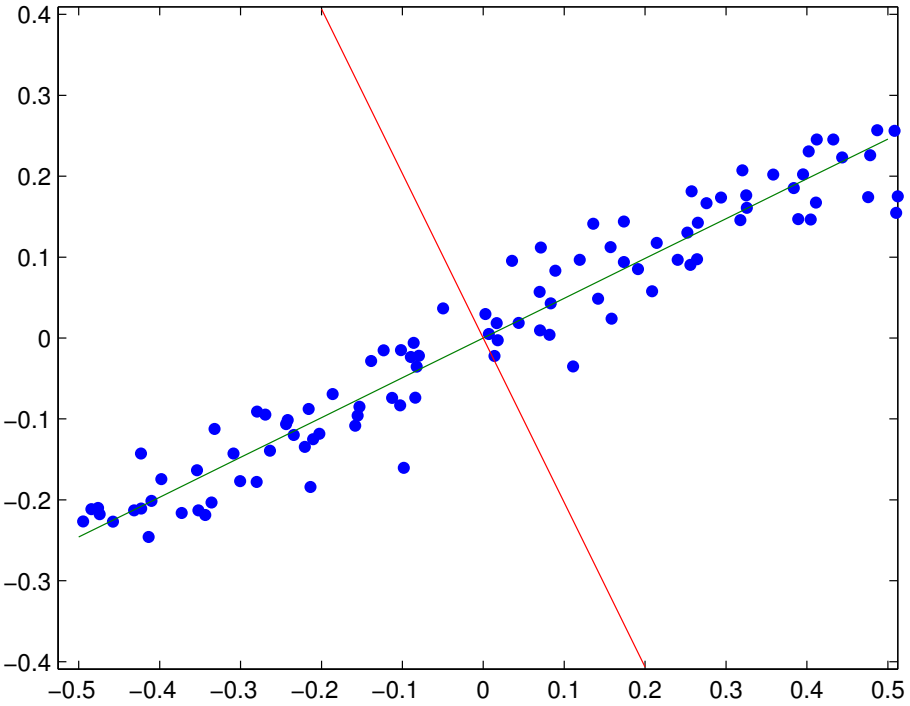


Figure 2.26: Highly Correlated 2D Data

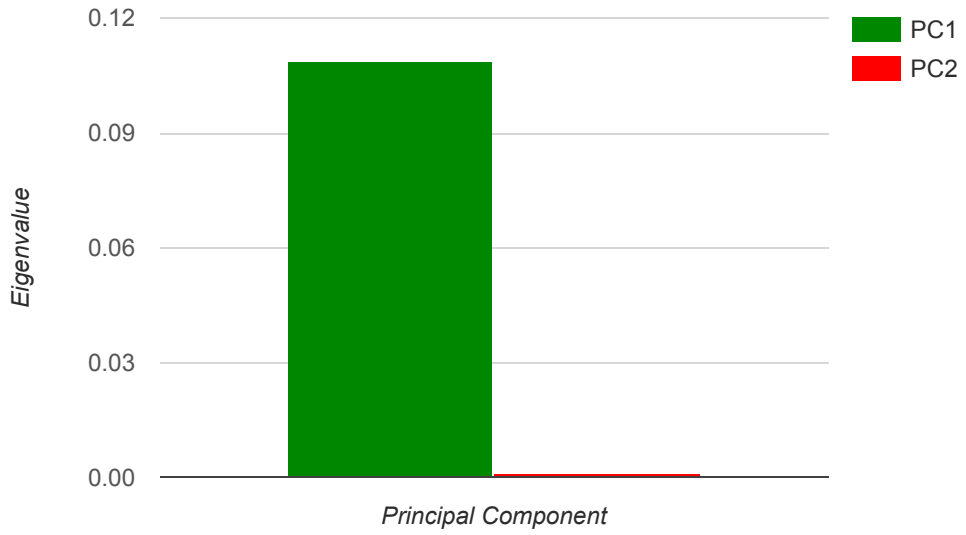


Figure 2.27: Eigenvalues of the Principal Components

As $\mathbf{A}$ is orthonormal ($\mathbf{A^T A} = \mathbf{I}$) it is its own inverse, so the inverse mapping becomes

$$\tilde{\mathbf{x}} = \mathbf{A}\mathbf{y} \tag{2.89}$$

where $\tilde{\mathbf{x}}$ is the closest reconstruction of $\mathbf{x}$ given the basis. $\mathbf{A}$ is calculated according to the following steps.

Suppose you have r feature vectors, $\mathbf{x}_i \in \mathbb{R}^l$. Let these vectors form the columns of the data matrix $\mathbf{D} \in \mathbb{R}^{l \times r}$. The mean feature vector, $\bar{\mathbf{x}}$, is

$$\bar{\mathbf{x}} = \frac{1}{r} \sum_{i=1}^{r} \mathbf{x}_i. \tag{2.90}$$

Let $\mathbf{X} \in \mathbb{R}^{l \times r}$ be the mean centred dataset so that each column, $\mathbf{x}'_i$, satisfies

$$\mathbf{x}'_i = \mathbf{x}_i - \bar{\mathbf{x}}. \tag{2.91}$$

The covariance matrix, $\mathbf{C} \in \mathbb{R}^{l \times l}$, of $\mathbf{D}$ is given by

$$\mathbf{C} = \frac{1}{r-1} \mathbf{X}\mathbf{X^T}. \tag{2.92}$$

The eigenvectors, $\mathbf{v}_i \in \mathbb{R}^l$, and eigenvalues, $\lambda_i \in \mathbb{R}$, of $\mathbf{C}$ are then calculated. Let $\mathbf{V} \in \mathbb{R}^{l \times l}$ be the matrix where column i is the unit length eigenvector that corresponds to the i^{th} largest eigenvalue. As the columns of $\mathbf{V}$ are both orthogonal ($\mathbf{v}_i \cdot \mathbf{v}_j = 0, \forall i \neq j$) and unit length ($\mathbf{v}_i \cdot \mathbf{v}_i = 1$), $\mathbf{V}$ is orthonormal.

Because there are only r images, there will be at most r meaningful eigenvectors. As l can be quite large in comparison to r, calculating the eigenvectors of the covariance matrix can be time consuming. Turk and Pentland (1991) show how to speed up the process, by rather calculating the eigenvalues and eigenvectors of $X^T X \in \mathbb{R}^{r \times r}$ and then using them in a linear combination with X to get to the original r eigenvectors.

To choose a subspace that retains θ proportion of the variance, $k \in \mathbb{N}$ should be chosen as the first integer such that,

$$\frac{\sum_{i=0}^{k} \lambda_i}{\sum_{i=0}^{l} \lambda_i} \not< \theta, \tag{2.93}$$

where the eigenvalues are summed in decreasing order of magnitude.

The first k columns form the columns of $\mathbf{A}$. If all columns of $\mathbf{V}$ were used then $\mathbf{A}$ would be exactly orthonormal and the transformation would retain all of the data's variance. As only the first k columns are used, the columns $\mathbf{A}$ are orthonormal ($\mathbf{A}^T\mathbf{A} = \mathbf{I}$) but the rows are not ($\mathbf{A}\mathbf{A}^T \neq \mathbf{I}$). For this reason the inverse reconstruction becomes less precise as k is reduced.

2.6 Accuracy Measures and Validation

2.6.1 Classification Validation

When learning to perform some classification task, it is important to be able to validate the accuracy of the model given its learned parameters. Upon each iteration of the learning algorithm the model parameter values are updated so that it better reflects the underlying structure in the training data. It is expected that a model with sufficient degrees of freedom will be able to classify the training data with very high accuracy. This is primarily because the training process itself updates the model parameters so that they correctly classify the training samples. Measuring the classification accuracy over the training dataset gives no indication as to the generalisation capabilities of the of model.

Therefore, it is important to consider how the model performs on new data that was not seen during training. To measure this generalisation, the full dataset should be partitioned into two mutually exclusive groups of data: one group for training, another group for testing. During the training phase, the model parameters are optimised so that the classifier represents the training data as best as possible. Following this phase, the testing data is presented to the model to assess how well the classifier works on unseen data. In many algorithms, this process is repeated and the accuracy over the unseen data is reported at the end of each of these *epochs*.

Furthermore, if the model has hyper-parameters, then a third partition is required and is called the *validation set*. In this case, the model is trained on the training data multiple times with different hyper-parameters. The accuracy over the test set is used to establish the best set of hyper-parameters. This approach however, allows the system to learn aspects of the test set through the hyper-parameters, so to be sure that the model generalises, the accuracy over the completely unseen validation set is calculated at the end.

Partitioning the data into three separate sets, however, can decrease number of samples that can be used for actual learning. To overcome this problem, *n-fold cross validation* divides the data into n different partitions or *folds*. Training is then performed on $n-1$ partitions and testing on the remaining one. As the algorithm proceeds, it rotates through the different fold combinations so that each fold is used for testing once. The final accuracy is then the average accuracy over all iterations. While this approach is computationally expensive, the process ensures that the training data is not wasted.

Formally, suppose that there is a dataset that contains k images.

$$X = \{x_1, x_2, \ldots, x_k\} \tag{2.94}$$

The data can be randomly partitioned into n mutually exclusive sets of images, $X_1, X_2, \ldots, X_n$ so that

$$X_i \cap X_j = \emptyset \text{ where } i \neq j, \tag{2.95}$$

$$\bigcup_{j=1}^{n} X_j = X. \tag{2.96}$$

In each iteration, i, of 5-fold cross validation the training (T_i) and testing/validation (V_i) sets are

$$T_i = \bigcup_{\substack{j=1 \\ j \neq i}}^{5} X_j, \tag{2.97}$$

$$V_i = X_i \tag{2.98}$$

2.6.2 Cross-Subject Validation

As this work focuses primarily on the use of video, it is important to consider the content of each frame within an image sequence. The students are seated in a lecture venue, therefore the background does not change and students' range of movement is relatively limited so the difference between consecutive frames can be fairly small. For example, consider the image sequence in Figure 2.25b.

While the student moves between frames, his head and shoulders, neighbour, and the background barely move at all. For a system learning to recognise some action or affect on a single subject, it can easily learn what parts of the image are important and what parts are not. In an image of another student, however, the background may be significantly different, as well as the position of the head, shoulders etc. A different student will have different skin tones, clothes, and mannerisms. The student may sit completely differently, may be viewed from a different angle and may write in a completely different way.

This indicates that there are two important factors to consider when validating generalisation accuracy, namely *intra-subject* and *inter-subject* variation. By performing n-fold cross validation and partitioning over all the video frames in the dataset, the reported accuracy will primarily reference intra-subject generalisation. This speaks to the system's ability to classify the subjects that it has already seen as all subjects are likely to be part of the training set.

To measure the system's ability to classify subjects that it has not seen before requires a different approach. In this case, rather than randomly partitioning over the video frames the partition lines should be drawn between the students themselves. This means that to measure inter-subject generalisation, hold-one-out validation or *cross-subject validation* is appropriate.

This approach takes the entire set of students/subjects, S, and selects one. That subject's data is used for testing/validation purposes, while all other frames are used for training. Formally, the training and validation sets at iteration i become

$$T_i = \bigcup_{s_j \in S \setminus \{s_i\}} S_j, \tag{2.99}$$

$$V_i = S_i, \tag{2.100}$$

where $s_i \in S$ is the subject being considered and S_i is the set of images relating to that subject specifically.

Due to the increased variation seen between subjects it is expected that when holding out a subject, the accuracy values will be lower than when validating across frames.

2.6.3 Imbalanced Datasets

The number of training samples in a category is an important factor when considering any machine learning technique. For example, the action of yawning is infrequent relative to not yawning. Therefore, if one records a video of a subject in class and labels each frame accordingly the latter category will be significantly under-represented. In a five minute recording (300 seconds), the subject may only spend a few frames actually yawning. Being generous, assume a yawn takes 10 seconds. Suppose the student yawns 5 times in those 5 minutes (50 seconds of yawn) that means that only 16.7% of the time recorded will be of the student yawning.

A classifier that decides to classify every image as *not yawning* immediately has an accuracy of 83.3% accuracy. This is an extremely misleading number that speaks to the distribution of the data more than the abilities of the classifier. For this reason, the balance of samples in the dataset should be carefully considered alongside the reported validation accuracy. The performance of a random classifier called *chance* is usually presented to benchmark what happens when there is 'no learning' taking place.

One way to prevent this bias from effecting the system is to construct a balanced dataset by re-sampling the original one so that there are an equal number of positive and negative exemplars from which the system can learn. With such a 50:50 split of data, one expects a classifier that is randomly guessing to get an accuracy of 50%. Any accuracy that is significantly above this should be due to some learning effect in the algorithm.

This approach, however, discards information about the representation of categories in the data. In the case where 90% of the data belongs to a specific category, it means that it is more important to predict one category correctly than the other. *Stratification* ensures that the different categories of data are represented according to their presence in the original data (Witten *et al.* 2016). When performing n-fold cross validation, stratification can be used to ensure a balance of data over the partitions. In this case the methods can be referred to as *stratified n-fold cross validation* or *stratified holdout validation*. In this case, it is important to examine the accuracy of chance, the confusion matrix which shows how labels were categorised, and even a reliability metric such as Cohen's Kappa (Cohen 1960) which relates the expected number of agreements due to chance and the observed number of agreements. Landis and Koch (1977) provide general guidelines to interpret various κ values as shown in Table 2.1.

Table 2.1: Interpretation of Cohen's κ Scores

κ	Agreement
< 0.00	worse than chance
0.00 – 0.20	slight
0.21 – 0.40	fair
0.41 – 0.60	moderate
0.61 – 0.80	substantial
0.81 – 1.00	near perfect

Another approach that can be used is to penalise the learning algorithm for getting the classification wrong in the minority set more than in the other. In such a case, the predictor will

learn that it is more important to classify the one group correctly even though it has seen more samples from the larger one. This is called *cost sensitive learning* but requires that the learning method's objective function is adjusted to consider this penalty.

Finally, the last approach considered here is to present samples from the smaller category to the system multiple times. The danger of this approach, however, is that the model may have issues with over-fitting. An approach that is used in a number of systems to prevent this is to pre-process the data differently before it is presented multiple times. This is context and method dependant. For example, when training the OpenCV (Bradski 2000) cascade classifiers for face detection (Lienhart *et al.* 2003; Viola and Jones 2001 2004) the training utility takes the face samples and performs a number of affine transformations (linear transformations such as rotation, scaling and 2D perspective warp) to the positive training set to increase the number of samples without causing over-fitting.

2.6.4 Approaches Used

This work uses stratified n-fold cross validation over the full set of labelled video frames as well as stratified and balanced holdout over the subjects (cross-subject validation). In all cases the stratification performed ensures that the proportion of samples representing each category is the same in both the testing and training sets. A random classifier (chance) is also provided in each case to provide a benchmark from which to measure accuracy. This is considered the standard approach when reporting error rates (Witten *et al.* 2016).

In many experiments, a *balanced* dataset is used. In this case, each category in the full dataset is randomly sub-sampled to build training and testing sets that have equal proportions of each category. This means that the probability of a sample belonging to any category is uniform and the expected accuracy of the random classifier is $\frac{1}{c}$ where c is the number of categories.

Chapter 3

Related Work

3.1 Contingent Teaching

Contingent teaching focuses on the ability of the teacher to scaffold the class in such a way that the presentation of material depends on the actions of the students. When applied, contingent teaching allows the teacher to go 'off script' and cater to the current state of the class more effectively (Draper and Brown 2004). There are arguments that this form of teaching reflects what good teaching actually is:

> "Although scaffolding is considered an effective teaching method, it appears to be rather scarce in classrooms, mainly because it is so difficult to perform... Scaffolding presumably reflects what good teaching is: being responsive to a learner and supporting a student within his/her ZPD [Zone of Proximal Development]. van de Pol *et al.* (2011)

In education generally, large class lecturing is often identified as a weak point. Common diagnoses include the lack of interactivity which is often manifested as an inability to get discussion going. The lecturer therefore loses the sense of how well the material sits with the audience (Draper and Brown 2004).

In large classes contingent teaching is difficult for two primary reasons. As there are more students in the class, the current state with regard to both content and affect becomes less homogeneous, which means that it becomes more challenging for interventions by the lecturer to focus on the issues of the individual student. On the other hand, as the numbers increase, it can also become more difficult to elicit responses from the students that the lecturer can use to effectively gauge understanding and sentiment.

The effect on student morale can be disproportionately positive when the student sees their response having a direct impact and the teacher responds on the spot to their specific learning needs (Draper and Brown 2004). This raises the question of how to effectively extract this information from a class, particularly in cases where the students may be reluctant to overtly provide it, or where the number of students makes direct questioning infeasible.

Literature identifies the use of clickers and SMS as ways of interacting with students in large classes or in cases where the students may be unwilling to directly engage. This work terms these approaches *intrusive* as they intrude into the class and require active responses from the students. This is discussed more completely in Chapter 4.

On the other hand we propose a *non-intrusive* approach based on computer vision that can be used in the general classroom to automatically assess students. Current approaches to this all make use of a webcam where the student is interacting with a computer and these are discussed in Sections 3.4 to 3.6.

3.2 Affective Computing

> “ Can machines think? — Turing (1950)

Whether machines will ever be able to really think or have some form of consciousness is a controversial philosophical debate that continues even today. Answering the question itself requires a practical definition of ‘thinking’ that is decidable – a definition that for which some test can yield a yes or no answer. Turing (1950) proposes a test based on imitation: if an interrogator cannot tell the difference between a computer and a human, then the computer can be considered intelligent.

Turing (1950) proposed the original *Imitation Game* and what has come to be known as the *Turing Test*. The Turing Test considers the text of a conversation between a human and an unknown *agent*. The test is ultimately about whether the human can tell if he is talking to a computer or another human. If he is unable to distinguish the two, then the machine can be considered intelligent.

Q : Please write me a sonnet on the subject of the Forth Bridge.
A : Count me out on this one. I never could write poetry.
Q : Add 34957 to 70764
A : (Pause about 30 seconds and then give as answer) 105621.
Q : Do you play chess ?
A : Yes.
Q : I have K at my K1, and no other pieces. You have only K at K6 and R at R1. It is your move. What do you play ?
A : (After a pause of 15 seconds) R-R8 mate.

Figure 3.1: Specimen Questions for the Turing Test (Turing 1950)

Illustrations provided by Turing are shown in Figure 3.1 and immediately highlight the type of problems that an intelligent machine must be able to solve. There are straightforward questions such as `add 34957 to 70764`, where the machine can simply perform the calculation and return it after some time delay to obfuscate how long the computation took. Problems

such as chess illustrate how the machine must be able to solve logical problems in varying problem domains. Current artificial intelligence systems are able to produce world champion level play in games like Checkers (Schaeffer *et al.* 2007), Chess (Hsu 1999), and Go (Silver *et al.* 2016) and significant work has been done on developing AI methods to apply to many varying real-world problem domains. In the first question posed however, we see that the agent opts out of a meaningful response.

While this may be acceptable in the conversation illustrated by Turing, it is easy to imagine situations where an intelligent machine must be able to converse with creativity and emotions in mind. For example, to pass the test while in a discussion of the latest music or responding to good or bad news, the system must at the very least be able to recognise, express and respond to some level of emotion. The Turing Test attempts to limit the effect of sensory expression by working through a textual conversation, but in spite of this, emotions can still be recognised and expressed in text (Cooper 1960). To pass the test, the computer's responses should be indistinguishable from a human's, and therefore to truly pass the test for intelligence, the machine must be able to perceive and express emotions (Picard 1995).

Beyond the Standard Turing Test, a modern intelligent system interacting with humans may be required to understand the state of the user, or even moderate its expression to effectively convey information. Consider a video game that can detect the player's emotional state. When frustration is detected it may moderate the difficulty to ensure that the game is not too hard, on the other hand if it detected confidence, then it could increase the difficulty to make sure the user is challenged. In the same vein, one might consider an *Intelligent Tutoring System* (ITS) that monitors your affect throughout an online learning episode. It should present work that is hard enough to be engaging, but not so hard that the student becomes demoralised and defeated (D'Mello *et al.* 2007). A piano teaching system should go even further: in addition to detecting interest, pleasure, and distress in its users, it should be able to recognise expression and phrasing within the music being played (Picard 1995).

Affective Computing is the field of computing that aims to solve these problems and reduce the gap between technology and emotionally driven human beings. The intention is to allow computers to recognise and respond to users' affective states during interactions in a way that increases usability, efficacy, and user enjoyment (Calvo and D'Mello 2010). The book, Affective Computing (Picard 1997) was instrumental in the inception of the area.

Picard (1997) outlines three main spheres of affective computing: affect detection, expression, and engenderment. Affect detection allows the computer to recognise the various affective states observed in humans. Expression allows a computer to convey emotion in a way that a human would intuitively understand, for example through the use of an avatar with facial expressions, gestures on physical robots, or even through natural language processing and speech synthesis. Finally, the last area considers whether computers can *'have'* feelings. It is argued that emotion plays a number of important roles in intelligent human functioning, and if computers are to achieve comparable abilities they will need to be endowed with them in some way. Picard (1995 1997 2003b) strongly argues this case, and considers the effect of the limbic brain on perception.

> “All sensory inputs, external and visceral, must pass through the emotional limbic brain before being redistributed to the cortex for analysis, after which they return to the limbic system for a determination of whether the highly-transformed, multi-sensory input is salient or not. Cytowic (1996)

The limbic parts of the brain responsible for emotion, are also largely responsible for perceptual processing: signals are processed by it before and after being processed by the more logical cortex. It is hypothesised that when faced with an infinite number of logical conclusions, emotion assists to bias the governing analytical process. This is supported by research on patients with frontal-lobe disorders and their impaired ability to make decisions (Damasio 1994). Picard (1995) expresses ‘serious doubt’ about its feasibility of computers that feel, but also draws parallels with debates surrounding machines and consciousness.

Minsky (2007) considers emotion to be a substantial part of what we call intelligence but rather than considering it an influencing factor, he considers emotion “not especially different from the processes that we call *thinking*.” Rather emotions are “ways of thinking” that increase a person’s resourcefulness. The implication is that by understanding a concept in multiple ways – through the lens of multiple emotions – then when one approach fails, frustration grows and you switch between the different points of view. This results in a diversity of thinking that should be incorporated into Artificial Intelligence:

> “Accordingly, when we design machines to mimic our minds – that is, to create Artificial Intelligences – we’ll need to make sure that those machines, too, are equipped with sufficient diversity... Minsky (2007)

Whether computers will ever be *able to feel* remains a mystery for now. However, machines with affective abilities – including recognizing, expressing, modelling, communicating and responding to emotion – have been built. There has been progress in all three key areas of affective computing (Calvo and D’Mello 2010; Zeng *et al.* 2009).

Affect detection is considered the primary goal of this research and the other areas of Affective Computing will not be considered further.

3.3 Emotions

3.3.1 What to Measure?

Emotions are constructs with fuzzy boundaries and are considered conceptual quantities that cannot be directly measured. Although this implies that there is often no absolute ground truth even among human experts, there is consensus that affect detection does not need to be perfect but rather approximately as good as humans (Calvo and D’Mello 2010).

In general, it is easy to identify big outbursts of emotion, such as swearing at your computer

and hitting the mouse on the desk out of frustration. However, if you were asked to write down the emotional state of the person sitting next to you right now, you would probably find this extremely challenging (Picard 2003a). In many cases people would even struggle to label their own feelings in various states. Kagan (1984) expresses this eloquently:

> “The term emotion refers to relations among external incentives, thoughts, and changes in internal feelings, as weather is a superordinate term for the changing relations among wind velocity, humidity, temperature, barometric pressure, and forms of precipitation. Occasionally, a unique combination of these meteorological qualities creates a storm, a tornado, a blizzard, or a hurricane – events that are analogous to the temporary but intense emotions of fear, joy, excitement, disgust, or anger. But wind, temperature, and humidity vary continually without producing such extreme combinations. Thus meteorologists do not ask what weather means, but determine the relations among the measurable qualities and later name whatever coherences they discover... Kagan (1984)

With this analogy in mind, the reason one often struggles to articulate emotions, is because of a failure to recognise emotion as the combination of many different factors. Just as there are many different weather states for which we do not have labels that we simply refer to as ‘fine, OK or bad,’ there are many different emotional states that are difficult to articulate because the labels have only been defined for discrete points in some large continuous space. For this reason, not all aspects of one’s affective state will be relevant or useful to observers (Picard 2003a). Therefore, it is reasonable to simply identify with equivalents to ‘fine, OK or bad’ with regards to different aspects of a subject’s affective state.

Furthermore, it is possible that there are combinations of affect with labels for which we have no use and conversely, there are combinations that are useful but not labelled. In this case, rather than specifically looking for states like pleasure, anger, or joy one should, for example, attempt to identify “the state you are in when all is going well with the computer” versus “the state you are in when encountering annoying usability problems” (Picard 2003a). Within this framework, the definition of ‘engagement’ that is used throughout in this work primarily considers whether the student is in “a state where the student is registering, listening to and considering information provided by the lecturer” versus “a state where the student is distracted, sleeping, or otherwise no longer focused on the lecturer.” These ideas are further discussed in Section 8.3.

3.3.2 Perspectives on Emotion

Primarily in the realm of philosophers and psychologists, founding theories of emotion originate from Charles Darwin (Darwin 1872) and William James (James 1884). In the 1980’s and 1990’s the idea of using computation to study emotion came to the fore, and the first expert systems were developed. Amongst discussions of machine intelligence the idea of including affective capabilities in systems developed and in the mid 90’s Affective Computing was born.

Much of the literature in affective computing is driven primarily by computer scientists and

artificial intelligence researchers. Because of this, much of the work in the field is agnostic to debates and controversies in the underlying psychological theory (Calvo and D'Mello 2010). There is danger in this, as all systems – usually implicitly – make assumptions about the various underlying models of emotion.

Within a specific emotional theory or framework, there are important concepts that should be considered. For example, in a system detecting frustration it is important to consider whether all users exhibit the same set of behaviours when frustrated. The universality of specific emotions and resulting behaviours should be established not only for multiple individuals within a population, but especially when considering different populations with different cultural standards. These considerations are even more important when considering machines that could express or even feel various emotions.

The remainder of this section briefly summarises six perspectives on emotion identified in the literature by Calvo and D'Mello (2010). The purpose of this section is to convince oneself, that regardless of the underlying philosophy of emotion, there are physiological and behavioural signals that indicate different aspects of emotion.

Emotions as Expressions

Among the first to scientifically explore emotions, Darwin (1859 1872) noticed similarities in the facial and body expressions of humans and animals. His conclusion was that these *behavioural correlates of emotional experience* were the result of evolutionary processes. He presented "serviceable associated habits," which evolved not for the primary purpose of expressing emotion, but were rather associated with other important actions. For example, a disgusted face would have been associated with the rejection of an offensive object and indeed communicates this message effectively.

This theory of emotion does not explain a large number of other emotional expressions, although there is some evidence that there are some universal facial expressions. This view has been challenged (Ortony and Turner 1990; Russell *et al.* 2003a; Russell 1994), but there is indication that at the very least there are six 'basic' emotions in the way they that they are innate and cross-cultural (Ekman and Friesen 2003; Ekman 1971; Izard 1971 1994; Tomkins 1962). These are anger, disgust, fear, joy, sadness, and surprise.

> "... the young and the old of widely different races, both with man and animals, express the same state of mind by the same movements. Darwin (1859)

Frijda (1987) noted the concept of "action tendencies." These are states of readiness to act in a particular way when confronted with an emotional stimulus. It is primarily linked to our need to solve problems in the environment, for example, "desire" makes us want to approach something while "fear" makes us want to avoid it.

There is considerable research that identifies facial correlates of emotion and this line of thinking has given rise to many affective computing systems based on facial expressions as a way to detect emotion. Most of this type of work is performed using camera. Other systems use

alternate channels such as voice, body language and posture and gestures. (Calvo and D'Mello 2010)

Emotions as Embodiments

James (1884) proposed that both expression and physiology play a vital role in our experience of emotion. His model put forward that the physiological changes experienced by the body actually are the emotion, rather than just its expression.

> "If we fancy some strong emotion, and then try to abstract from our consciousness of it all the feelings of its characteristic bodily symptoms, we find that we have nothing left behind...
>
> ...We feel sad because we cry, angry because we strike, afraid because we tremble, and neither we cry, strike, nor tremble because we are sorry, angry, or fearful, as the case may be. James (1884)

An example of this theory in action, would be that one experiences some emotion-evoking stimulus such as seeing a predator. Immediately the body triggers physiological responses (such as releasing adrenalin, increased heart rate, faster breathing) that the brain then interprets or labels as fear. It is generally accepted that the body influences emotions, but here the primary issue is that of causation. The question is whether the brain experiences fear that then causes these changes in the body, or whether these changes in the body are what the brain interprets as fear.

Independently, Lange (1885) put forward a similar model. Their work is together considered the 'James-Lange Theory' and focuses on the changes in the Sympathetic Nervous System as part of the Autonomic Nervous System, which controls bodily functions such as breathing and heart beats.

Dalgleish (2004) notes that, while in its original form it has been abandoned, the theory has remained influential and "most contemporary neuroscientists would endorse a modified James-Lange view in which bodily feedback modulates the experience of emotion." This theory implies that emotional experience is embodied in peripheral physiology and that by analysing these physiological changes one may be able to find a typical physiological response for each emotion. (Calvo and D'Mello 2010)

Historically, sensors to measure these physiological changes have been intrusive making the collection of natural data difficult. With the improvements in wearable sensors, however, in the future there may be large banks of data measured in real-world contexts. With the recent improvements in statistical data mining and machine learning techniques, these sensors may offer exciting research avenues in the future. For this work's purpose and in their current form, these sensors are too intrusive and expensive. The possibility of incorporating thermal imaging to try detect some of these physiological signals offers a potentially significant increase to what may be possible in the future.

Cognitive Approaches to Emotions

Probably the leading view of cognitive psychologists, this model states that for an emotion to be experienced, the object of some event should be appraised as directly affecting the person in some way based on their experience, goals and opportunity for action (Roseman *et al.* 1990; Schachter and Singer 1962; Scherer *et al.* 2001; Smith and Ellsworth 1985). This appraisal is considered to be an unconscious process evaluating the events based on "novelty, urgency, ability to cope, consistency with goals etc." Cognitive-motivational-relational theory states that to predict a person's reaction in a situation, that one must know their expectations and goals in relation to that situation. (Calvo and D'Mello 2010)

Roseman *et al.* (1990); Roseman (1984) describes 14 emotions and associates them with a cognitive appraisal process that models events based on five dimensions:

1. *Consistency Motives*, beneficial versus harmful
2. *Probability*, how likely is it that the event will actually occur?
3. *Agency*, who causes the event?
4. *Motivational State*, appetitive (rewarding) versus aversive (punishment)
5. *Power*, is the subject in control of the situation?

Ortony (1990) presents a model with four sources of evidence to test emotional theories: language, self-report, behaviour, and physiology. These two models both agree that there is some appraisal process and provide a framework in which to predict the emotional state of a person as a point in the respective *appraisal space*. The computational models that follow from these works have proven useful in developing affective computing systems. (Calvo and D'Mello 2010)

Emotions as Social Constructs

Averill (1980) argues that emotions are social constructs and cannot be explained only on the basis of physiological or cognitive terminology. The language of emotion is considered a vital part of its experience. Glenberg *et al.* (2005) stress the importance of language to emotion. There are even researchers who question the universality of Ekman's studies (Ekman and Friesen 2003; Ekman 1971) arguing that the labels used to code emotions were not universal labels, but rather labels used in the US. There are cultures where labels cannot be literally translated, and it is argued that in these cultures different emotions are associated with different social connotations, thus creating a different experience.

Reinforcing these arguments, Stets and Turner (2008) reviewed how each society has *display rules* which dictate when and how we express our emotions.

Gendron *et al.* (2012) and Lindquist *et al.* (2006) present experiments where they found that when interpreting emotion the "perceiver unknowingly contributes to emotion perception with emotion word knowledge." This output reinforces the notion that our vocabulary to describe emotions certainly affects the way we perceive it in others, and potentially even ourselves. With

this in mind, it is important to consider how the language used in a study or survey affects the subjects as they read/hear the various labels.

As mentioned in Section 3.2 humans tend to label a large continuous space of emotions using a small number of discrete names. By providing subjects with a list of these discrete labels, the researcher is encouraging them to 'round off' to some nearest label, rather than consider where they sit on some multidimensional spectrum.

Calvo and D'Mello (2010) note that little work in affective computing has focused on this social constructionist approach to emotions. Most researchers seem to prefer the approaches outlined in Sections 3.3.2 to 3.3.2 on pages 53–55, as they lead to a more practical list of features for computer scientists.

Neuroscience

Only in the past two decades has neuroscience become part of the study of emotional phenomena. In this time, new techniques have been developed to study emotional processes and their neural correlates. This is the birth of affective neuroscience, and is the study of the neural circuitry that underlies emotional experience. (Dalgleish *et al.* 2009; Damasio 2000; Davidson *et al.* 2003; Panksepp 1998)

The advances in neuroscience now provide alternate data sources for affective computing, rather than relying solely on facial expressions and self-reporting. The methods developed include brain imaging (fMRI), lesion studies, genetics and electro-physiology (EEG). These provide data about the physiological processes never before available to researchers.

One important development is the large body of evidence which suggests that the neural layers of cognition and emotion overlap substantially (Dalgleish *et al.* 2009). Cognitive processes such as memory encoding and retrieval, causal reasoning, deliberation, goal appraisal, and planning operate continually throughout the experience of emotion. (Calvo and D'Mello 2010)

Another important discovery is that emotional learning can occur without awareness and elements of emotional behaviour do not require explicit processing (Calvo and Nummenmaa 2007; Öhman and Soares 1998). It suggests that some emotional phenomena are not consciously experienced, but still influence memory and behaviours. This indicates that studies based solely on self-reports may not give thought to more subtle phenomena.

Recent evidence from the field aligns with a modified version of the *Emotion as Embodiments* theory outlined in Section 3.3.2, although with a lower level approach. There is a growing interest in *emergent variable models* where emotions can be considered the labels given to certain neural states, rather than emotion causing these states.

Core Affect and Psychological Construction of Emotion

Russell (2003) argues that there are so many different emotional theories because there are multiple facets to emotion. This is in contrast to the argument that these theories describe the same thing from different points of view. He puts forward the idea that if emotion is a

> “. . . heterogeneous cluster of loosely related events, patterns and dispositions, then these diverse theories might each concern a somewhat different subset of events of different aspects of those events. Russell (2003)

One of the main features of his theory centres around the idea of a *core affect*, which is a consciously accessible neuro-physiological state which sits as a two dimensional point between pleasure and displeasure (*valence*), and sleepy and activated (*arousal*). He realises that emotions are not exact and produced by programs, but are rather the result of loosely coupled components. These components need not be coherent or in agreement, but a person is generally able to make sense of the different sources of information. The *prototypical case* of an emotion occurs when all these sources are coherent.

3.4 Affect Detection

Affect Detection and its associated methods are of primary interest to this work. In the previous section, underlying theories of emotion were briefly considered and hinted to the way that features may emerge from the different subsets thereof. In this section we examine different behavioural and physiological modalities through which emotions may be indicated. There are four important considerations when examining the value of a modality (Calvo and D'Mello 2010):

1. Validity
2. Reliability
3. Time Resolution
4. Cost and Intrusiveness

Validity focuses on the natural ability of the measured signal to identify or discriminate an affective state. Reliability looks at the signal in real-world environments; it considers whether the signal generalises outside of the lab and in what situations it is actually valid. Time resolution of the signal as it relates to the relevant application should be considered. Potentially valid and reliable methods might have resolutions too coarse to be useful. Finally, the cost of the equipment and the intrusiveness of the sensors is the final important factor. For example, an fMRI machine is too expensive, intrusive, and distracting to use in a classroom, a camera is not.

The rest of this section outlines a number of different detection modalities as well as some of the systems that use them. This list is based primarily on Calvo and D'Mello (2010) and mentions facial expressions, paralinguistic features of speech, body language and posture, physiology and multi-modal approaches. Other methods where the equipment is either too intrusive or not applicable to a live lecture venue are excluded. Finally, a brief consideration of real versus posed data for ground truth is presented, and some last remarks on affective computing and necessary assumptions are made.

3.4.1 Facial Expression

This modality encompasses the majority of affect detection research. Its is based on the view from Section 3.3.2 that basic emotions are associated with expressions. A system that detects these expressions is then able to infer affect.

Ekman and Friesen (1978) developed the *Facial Action Coding System* (FACS). This system codes the independent motions of the face as *Action Units* (AUs) which then make up basic expressions. There are 6 basic expressions, namely anger, disgust, fear, joy, sadness, and surprise. (Ekman 1992). This system has become the de facto standard for classifying facial expressions in the behavioural sciences. Donato *et al.* (1999) are mindful that manually encoding a video is an extremely expensive task, where a human encoder can spend up to an hour encoding a single minute of video. There has been much progress in getting computers to automatically encode AUs, but their reliability is not yet comparable to human encoders (Asthana *et al.* 2009; Brick *et al.* 2009; Calvo and D'Mello 2010; Hoque *et al.* 2009). El Kaliouby and Robinson (2005ab) have used the FACS method for some basic affect detection and achieve accuracies in between 60% and 90%.

There is a large amount of work in the development of *Intelligent Tutoring Systems* (ITS) that use webcams to recognise facial expressions – not necessarily with FACS – to adjust the content and questioning the student receives based on the student's affective state. Some of these works include Arroyo *et al.* (2009) and McDaniel *et al.* (2007). For more work on ITSs see D'Mello *et al.* (2006).

Zeng *et al.* (2009) reviewed 29 vision-based affect detection systems. Most of the research focuses on detecting the six basic emotions listed above and not necessarily those emotions that are required for general affective systems. Most of the systems are trained based on posed facial expressions as opposed to natural or spontaneous data. This has implications for the reliability of these methods in real-world settings. Only 6 of the 29 systems were able to operate in real-time on single faces. This indicates the difficulty of achieving a real-time system for affect detection – heavy use of parallelisation and high performance computing techniques are necessary to achieve real-time performance.

3.4.2 Paralinguistic Speech Features

When speaking, one transmits affect information both explicitly and implicitly. Explicit information is transmitted by what is said, while implicit information is transmitted through volume, rhythm, intonation/pitch, and stress. These are also known as *prosody*. While the all mechanisms are not fully understood, it appears that listeners can decode many emotions from prosody and other non-linguistic vocalizations, such as laughing and crying. This establishes a feasible level of validity.

Surveys and research conducted by Johnstone and Scherer (2000); Juslin and Scherer (2005); Russell *et al.* (2003b); Zeng *et al.* (2009) outline a number of important results. The most reliable finding, according to this analysis, is the link between pitch and arousal – sleepiness/fatigue versus activation/excitement (Calvo and D'Mello 2010). Detection rates of sadness, anger and fear are best, while disgust is the worst. Across the board, however, accuracy

rates are lower than those of facial expression based systems.

Although less reliable, the cost is low both in monetary terms and the amount of training required. Calvo and D'Mello (2010) note that 16 of the 19 systems surveyed were trained on spontaneous speech. Couple this with its non-intrusive nature and fine resolution, paralinguistic features are considered a useful feature which may provide useful information in the future.

The difficulty of speech based features in a lecture venue is that most of the time, it is only the lecturer talking. If the students are talking they are often whispering to one another, which masks the useful features. If the students are all talking then unmixing the voices is a non-trivial task, but more importantly, this is something the lecturer will definitely notice without a computer. Finally, if a single student is talking then that student is likely interacting with the lecturer already. For these reasons, paralinguistic features of speech specifically do not appear useful as the primary data source for large scale monitoring of classes and this feature is not currently used in the system.

There may be useful information in this medium and audio should be considered as a supplementary modality in the future. Work should be done to analyse audio recordings of lecture venues, where volume and dominant frequencies may provide useful information. For example, increased high frequency noise in an area might indicate that students are whispering to each other.

3.4.3 Body Language and Posture

The signals examined by methods in the previous two sections are by far those most commonly used in affective computing in spite of the fact that many theories regarding expressions, including Darwin's, focused heavily on body language. Body language and posture have been overlooked in state-of-the-art detection systems (Calvo and D'Mello 2010).

Body language appears to be a valid and useful approach for affect detection primarily due to the relatively large size of the human body and the high number of degrees of freedom. This allows the body to exhibit a large number of different positions and movements. These factors are considered to make body posture a potentially ideal affective communication channel. (Bull 1987; Coulson 2004; De Meijer 1989; Mehrabian 1977; Montepare *et al.* 1999)

Particularly, the size of the body makes posture an ideal channel to detect affect over longer distances as it requires much less detail than facial expressions (Walk and Walters 1988). D'Mello and Graesser (2009) note that the "greatest advantage to posture based affect detection is that gross body motions are ordinarily unconscious, unintentional, and thereby not susceptible to social editing, at least compared to facial expressions, speech intonation and some gestures."

This is an extremely important issue when attempting to detect what society considers to be negative feelings. The theory of *Social Display Rules* put forward by Ekman and Friesen (1969) is consistent with the findings of D'Mello and Graesser (2009), that students "do not readily display frustration on the face, perhaps due to the negative connotations associated with this emotion." Ekman and Friesen (1969) refer to *non-verbal leakage*, which alludes to the difficulty liars face when disguising deceit through channels usually controlled subconsciously.

A large body of research uses the Tekscan *Body Pressure Measurement System* (BPMS) which is a thin film with a matrix of pressure sensors that can be placed on the horizontal and vertical surfaces of a chair. Mota and Picard (2003) were the first to use such a system to detect the affective states of a student working in front of an ITS. By analysing the spacial and temporal pressure maps produced by the BPMS, they were able to classify nine postures with an accuracy of 87%. Following this, the system recognised three levels of interest: high, low or taking a break. The system analysed postures over three second intervals and was accurate to 82%.

D'Mello and Graesser (2009) extended the work above to predict engagement/flow, boredom, confusion, frustration and delight of a student during a session with their ITS, *AutoTutor*. Their experiments yielded positive results with accuracies of about 75% when predicting these outcomes.

Posture based detection is quickly gaining traction (Bianchi-Berthouze and Lisetti 2002; Castellano *et al.* 2008b) and is a big, open field that has largely remained unexplored.

Most of the work in posture detection in affective computing seems to focus on the use of pressure sensors in chairs of some sort. In this work, it is proposed that the gross posture detection is done through vision, as pressure sensors on each seat in a large classroom would be prohibitively expensive.

This method of posture tracking will be closely related to gesture detection and the system will track the gross body position and movements, rather than the slight movements detected in the pressure plates.

3.4.4 Physiology

While the three modalities outlined above primarily relate to the *expression* perspective of emotion, physiology and brain imaging relate primarily to the *embodiment* perspective of emotion. The aim using this approach, is to find physiological patterns that correlate with the various relevant emotions. There are two underlying psychological fields from which this body of work draws: *physiological psychology* and *psychophysiology*. These relate to investigations of how behaviour changes based on forced physiological changes, and how physiological factors change when different stimuli are presented. These fields attempt to understand both the physiological effects of behaviour and the behavioural effects of physiology. Behaviour, in this case, refers to both emotional and cognitive processes. (Calvo and D'Mello 2010)

Physiological measurement techniques relating to affective computing include:

1. Electromyogram (EMG) - Muscle Activity
2. Electrodermal Activity / Galvanic Skin Response (EDA/GSR) - Skin Conductivity due to Sweat
3. Electrocardiogram (EKG / ECG) - Heart Activity
4. Electroencephalography (EEG) - Brain Activity
5. Electrooculogram (EOG) - Eye Movement

6. Magnetic Resonance Imaging (MRI) - Brain Imaging

7. Functional Magnetic Resonance Imaging (fMRI) - Blood Flow in the Brain

While *non-invasive* or non-surgical in nature, these sensors are often intrusive, in that the individual needs to wear or sit inside the sensors, this makes them less useful to a classroom setting. There seems to be a growing body of work that is focusing on the use of infra-red or thermal imaging techniques. For example, Sun *et al.* (2005) had success when detecting a subject's pulse using a sensitive thermal imaging system. Shastri *et al.* (2009) found that they were able to detect peripheral sympathetic responses using only imaging techniques with an accuracy comparable to those found using GSR sensors. Puri *et al.* (2005) found a correlation between times of increased stress and blood flow in the frontal vessel of the forehead. This increased flow dissipates convective heat and can be monitored through the use of thermal imaging. Their system was able to successfully detect stress in the 12 subjects they evaluated.

Of most importance to this work, is that of Merla and Romani (2007). Through the use of thermal imaging and the change of thermal patterns exhibited over a subject's face, they were able to detect stress, fear and emotional arousal. They note how the topographic distribution of cutaneous (skin) temperature on the face provides an alternative to the standard sensors and is a touchless method for assessing a subject's emotional arousal.

At the current time, high resolution thermal imaging technology is prohibitively expensive. A typical thermal camera can be used to track one or two students at a resolution high enough for facial analysis; however, when considering a small classroom of even 50 students, the expense is too restrictive. As technology advances the feasibility of thermal methods may increase, but currently it appears that while there may be significant amounts of useful data, it is infeasible to roll out the technology on a large scale. For these reasons, thermal imaging is not considered further in this work.

3.4.5 Multimodality

The psychological theories for expressions and embodiments both suggest that for some emotional episode, there will be changes to both expressions and physiology. Calvo and D'Mello (2010) use anger to illustrate this concept. When angry one expects standard manifestations through facial, vocal and body language. At the same time, a rise in heart rate is likely to occur. By monitoring all of these channels of communication, a system can detect this affect more reliably. Some may consider it surprising that, while multimodal affect detection systems have been widely advocated, they are rarely implemented. This is largely because challenges in the single modal setting become exponentially more difficult to deal with in multi-sensory environments. In general, however, researchers seem to agree that multimodal sensor environments will ultimately be part of a comprehensive affective computing solution (Calvo and D'Mello 2010; Jaimes and Sebe 2007; Sharma *et al.* 1998).

Calvo and D'Mello (2010) analyse only two systems that use three or more modalities. These are Scherer and Ellgring (2007b) and Castellano *et al.* (2008a). The former uses a mixture of facial features, vocal features and body movements to discriminate among 14 emotions. With classification based on only facial or vocal features, the system performed at 52% each

time. When using the combined 37 channel model, this accuracy increased to 79%. The latter used a mixture of facial features, speech contours and gestures. For single modal cases the accuracy of the system was 48%, 67% and 57% for each feature respectively. When used together, the system's accuracy increased to 78%.

Using facial features, posture features and context from an ITS, Kapoor and Picard (2005) were able to infer the level of interest of a child. Using the upper face, lower face, context and posture yielded accuracies of 64%, 53%, 57% and 82% respectively. When combining all the modalities, their system performed at 86% accuracy. This suggests that their other features were mostly redundant when taken alongside posture.

Calvo and D'Mello (2010) present a number of other cases where multimodal sensory data outperforms the corresponding single channel system. In spite of these results, they warn against making the assumption that there is coherence among multiple components of emotion. A common assumption made by researchers working in affective computing is that the emotional response to some stimulus is synchronized over the different communication channels. There is evidence that – other than the prototypical cases – there is actually low correlation between the signals and it is important to identify when these signals are strongly and loosely coupled. (Barrett 2006; Calvo and D'Mello 2010; Castellano *et al.* 2008a; Russell *et al.* 2003a; Scherer and Ellgring 2007a)

3.4.6 Real versus Posed Data

Literature shows that there are important differences between real and posed data. Afzal and Robinson (2009); Cohn and Schmidt (2004); Ekman *et al.* (1990); Pantic and Patras (2006) show how systems trained on posed expressions do not generalise to real-world contexts. One primary difference is seen in the synchronicity of multimodal responses mentioned in the previous section. Castellano *et al.* (2008a); Scherer and Ellgring (2007a) find that across different channels, the responses of posed emotions align more than those exhibited in real life. It may be interesting future work for deception experts, but based on the literature, real data should be used to train the system where ever possible.

3.4.7 Ground Truth

Establishing ground truth with regards to the fuzzy borders of affect is a challenge. Whitehill *et al.* (2014) note that the current popular methods are *self-reporting*, *observational checklists* and *automated measurements*. Each method has its own limitations.

One major drawback of self-reporting is that students may be influenced by the same pressures associated with social display rules. These rules tell students that it is not acceptable to be bored in class and that you certainly would not want the lecturer to know that you are bored. This is considered in more detail in Chapter 4.

A second major problem is the definition of 'being engaged.' As students assess themselves based on different factors, they are likely to come up with arbitrary levels of engagement that

do not necessarily generalise between them and other students. Considering the temporal nature of engagement and other affects, the best way to asses through self-reporting may be to measure levels of engagement over time and assess the relative changes reported by a subject. As long as a subject is consistent in their reporting of their own state, it may be possible to generalise results more easily. In this case, direct claims such as "slumped posture correlated with high boredom levels" should be replaced with claims focusing on the changes over time: "as boredom levels increased from some initial level, more slumped postures were observed." The emphasis is on the *relative change* in current levels, rather than the absolute levels themselves.

Observational checklists, on the other hand, require that some observer checks for certain behaviours. Whitehill *et al.* (2014) notes questions such as whether the students sit quietly, do their homework, ask questions and are punctual. These checks, however, are only tangentially related to the students emotional engagement as a compliant student may check all the boxes, but remain emotionally disengaged.

The third method examined by Whitehill *et al.* (2014) relates to *Automated measurement*, which is the purpose of this research, but requires that we already have a ground truth database on which to train. Automated tools can rely on features such as context in ITSs, physiological signals, facial expression, and body posture.

This work presents work in both self-reporting (Chapter 4)and an observational checklist (Section 8.3) and discusses benefits and problems with each approach in the relevant sections.

3.4.8 General Remarks on Affective Computing and Assumptions

There are many open questions in the affective sciences. For example there are many questions surrounding the use of labels like anger, shame, joy or sleepiness versus points in some continuous space, like valence and arousal. A pragmatic approach by researchers has lead to a fast expansion in Affective Computing, but practitioners should always bear in mind what their fundamental assumptions are when developing affect sensitive systems.

One assumption of interest to these systems is the internal representation of emotion within a subject. If the research tries to impose a representation – such as labels and scales – the outcome of self-reporting is influenced. These representations are not necessarily global and the language with which subjects are provided influences their ability to report. For example, when not provided with labels, Lindquist *et al.* (2006) found subjects did not necessarily identify the same emotions that they did when labels were provided.

Another fundamental assumption in many systems is that when provided with a stimulus, subjects express their experienced emotion. There are two aspects to consider that are important to this work. On the one hand, social display rules mean that subjects filter what expressions they show, particularly when experiencing negative emotions (Ekman and Friesen 1969). This was explicitly validated in Schneider and Josephs (1991) with school children, where they find that learners try to appear fine even when they are not. The other important aspect to be considered is that of emotional *regulation*, where the subject might find a way to offset their experience of a negative emotion, such as boredom, by showing interest in something else – such as the environment or through doodling.

Doodling, for instance, may help offset boredom and can actually lead to an increase in cognitive performance over time. Andrade (2010) found this in an experiment with recollection of information from telephone calls. Maclay *et al.* (1938) found that doodles were produced during states of "idleness, boredom, leisure, meditation, and affective tension – indecision, concentration, expectation, and impatience." Note the appearance of both boredom and concentration in that list. Scott (2011) notes that doodling happens when the individuals focused attention is partly or completely elsewhere.

Regulation, such as this, may make a subject appear bored, disinterested or disengaged, when the opposite is actually true. Care must be taken when building an 'observational checklist' to ensure that the items being checked do indeed correspond to the affect being measured.

3.5 Affective Computing in Education

3.5.1 Affect in Education

The importance of affect during learning episodes has been established in literature. Picard (1997) asserts that the reason humans are so good at many tasks, is because we process data through the lens of emotion, particularly when it comes to perception. Learning is significantly enhanced when students feel empathy from their teacher (Graham and Weiner 1996; Zimmerman 2000). The interpersonal relationships between teachers and students are linked to long term increases in motivation (Royer and Walles 2007; Wentzel and Asher 1995). D'Mello and Graesser (2009); D'Mello *et al.* (2006); Graesser *et al.* (2006); Kort *et al.* (2001) argue that emotions play a significant role in the deep learning of conceptual information.

Axis	-1. 0	-0. 5	0	+0. 5	+1. 0
Anxiety-Confidence	Anxiety	Worry	Discomfort	Comfort	Hopeful	Confident
Boredom-Fascination	Ennui	Boredom	Indifference	Interest	Curiosity	Intrigue
Frustration-Euphoria	Frustration	Puzzlement	Confusion	Insight	Enlightenment	Ephipany
Dispirited-Encouraged	Dispirited	Disappointed	Dissatisfied	Satisfied	Thrilled	Enthusiastic
Terror-Enchantment	Terror	Dread	Apprehension	Calm	Anticipatory	Excited
	-1. 0	**-0. 5**	**0**	**+0. 5**	**+1. 0**

Figure 3.2: Proposed Emotional Framework for Learning (Kort *et al.* 2001)

Figure 3.2 provides a framework for the assessment of student affect during learning episodes. While there are a number of 'negative emotions,' the proposed model of learning has learners cyclically moving between a number of different stages, such as curiosity to confusion, to frustration and ultimately hopefulness. This cycle of positive and negative affect is supported by Shute *et al.* (2015).

General affect detection focuses on anger, fear, sadness, happiness, disgust, and surprise, but these basic emotions are infrequent in the learning context (Bosch *et al.* 2015b). Rather, students' experiences during learning episodes centre around engaged concentration, boredom, confusion, frustration, happiness, and anxiety (DMello 2013). Although they are not exactly

the same, the affects listed by Kort *et al.* (2001) and DMello (2013) are fundamentally in agreement.

The universality of facial expressions and mappings with the basic emotions were largely established by Ekman *et al.* (1980) and Reisenzein *et al.* (2013). The same process is still under way for the learning centred affects (McDaniel *et al.* 2007). As such, there is very little work done detecting the students' affects during learning episodes using vision and most systems primarily rely on contextual and other sensor information. Current systems are discussed in the next section.

3.5.2 Detection of Affect for Educational Purposes

Bosch *et al.* (2015b) present one of the first works detecting learning-centred affective states in the wild using vision. They perform supervised learning on data collected from 137 students in 8^{th} and 9^{th} grade computer laboratories. The students each sat at a computer and played a physics based computer game while being recorded by webcams. The students were coded using the five affects shown in Table 3.1 as well as being *on-task* or *off-task*. Being off-task involved watching other students, using a cellphone, or having unrelated conversation. They use the *area under receiver operating characteristic curve (AUC)* to measure accuracy and achieve impressive results considering the challenges. Their results are reported in Table 3.1.

Table 3.1: Classification Accuracies in Bosch *et al.* (2015b)

State	AUC
Off task behaviour	0.816
Boredom	0.610
Confusion	0.649
Delight	0.867
Engagement	0.679
Frustration	0.631
5-way classification accuracy	0.655

Commercial software was used to extract the likelihood estimates for the presence of 19 action units (AUs), as well as head pose, orientation and position information. Poor lighting, extreme head pose or position, occlusions from hand-to-face gestures, and rapid movements resulted in 25% of data being discarded. Using clustering and Bayesian approaches, the extracted features were used to classify the relevant affects. The work by Bosch *et al.* (2015b) is the most similar to the work presented in this thesis and will be used a benchmark for results.

Problems identified primarily related to issues with noisy data, which was a problem for the work presented in this thesis as well. As data was collected using webcams on computers in front of the students it provided an unoccluded, consistent, frontal view of the subjects. The data presented in this thesis is more challenging as it considers students sitting in a lecture theatre viewed from a single camera at the front of the class. This means that there is significantly more variation due to occlusions, background changes, resolution, angle from the camera to the student, and others. Kai *et al.* (2015) expands upon Bosch *et al.* (2015b) and again notes

problems with video based approaches and noisy classroom environments. Again the students were playing the physics-based game and were recorded from cameras on the machines in front of them. The issues related to poor video data are discussed further in Chapter 5.

Raca and Dillenbourg (2013); Raca *et al.* (2015) focus on the use of head motion to identify students that are paying attention. They show that head pose can be used to approximate eye gaze and, in turn, use that to analyse whether students are behaviourally engaged in the class. They also make use of optical flow to track how much the student is moving. Their work gives accuracies between 61.86% and 65.72% classifying students on a 3-point scale of attention. The data was collected in a manner similar to this work, but on a smaller scale. Attention was labelled with students self-reporting their attention levels periodically using a questionnaire during class.

A further literature survey was unable to find any work seeking to apply vision based affect detection methods to real-life '*wild*' learning environments where subjects were not seated in front of a computer. It is believed that this is the first work to focus on issues of affect detection in the wild, without subjects seated in front of computers. However, there has been a lot of work in the creation of Intelligent Tutoring Systems (ITSs) in order to make an affect-aware online tutor. This was proposed by Kort *et al.* (2001) and the research groups of D'Mello and Picard have contributed much literature towards the development of ITSs to maximise learning.

A system that responds appropriately to negative affective states is more likely to re-engage the student (Kapoor *et al.* 2007). The work already done in this field is very promising (Alyuz *et al.* 2016; Arroyo *et al.* 2007; Conati and Maclare 2004; D'mello and Graesser 2007; D'Mello *et al.* 2007 2010; Graesser *et al.* 2007; Johns and Woolf 2006; Kapoor *et al.* 2007; McQuiggan and Lester 2006). *AutoTutor* has received the most attention recently. Systems like this use a combination of hardware monitors and contextual information from the current activity to predict the student's current affect.

In these settings, the researchers have used various methods and sensors to detect emotions such as boredom, fatigue, joy, surprise, anger, disgust, and interest while the student works on problems with the ITS. It then adjusts both the difficulty of the problems as well as the phrases it uses to respond to the user. The ITS applications developed so far have been based primarily on vision for facial expressions. Other sensors, including a Body Pressure Measurement System (BPMS) (D'Mello and Graesser 2009) and skin conductance sensors (Arroyo *et al.* 2009) have been used. Almost all of the current systems use a pressure sensitive mouse which provides vast information about frustration (Dennerlein *et al.* 2003).

The ITS developed in Arroyo *et al.* (2004 2007 2009) was equipped with mouse pressure, posture, video, and skin conductance sensors. It was then applied in both High School and Undergraduate settings. They used self-reported student data as ground truth. A major component of the system was the use of *MindReader* which performs facial expression recognition to infer aspects of a subject's state of mind (El Kaliouby and Robinson 2005a). They found that the extra sensors improved the predictive power of the system. They also identified correlations between *Sitting Forward* and both frustration and excitement. With contextual information from the online tutor, they are able to differentiate between the two. Confidence and concentration were strongly correlated and were well predicted by the facial expression software.

In D'Mello and Graesser (2009), the *AutoTutor* ITS was equipped with the BPMS pressure

system to measure posture. The BPMS system is placed on the horizontal and vertical parts of a chair, it then feeds pressure information to the system that can be used to infer posture. Based on posture alone, machine learning experiments found detection accuracies of 73%, 72%, 70%, 83% and 74% when differentiating boredom, confusion, delight, flow and frustration from neutral affects. When differentiating between two, three, four and five of the states, accuracies of 71%, 55%, 46% and 40% were achieved respectively. Although facial expressions were once considered the gold standard for emotional expression, they argue that "there is converging evidence that disputes the adequacy of the face in expressing affect... at the very least, it is reasonable to operate on the assumption that some affective states are best conveyed through the face, while others are manifested through other non-verbal channels." Of interest to this research, D'Mello and Graesser (2009) note the face is particularly sensitive to social display rules when dealing with negative emotions, such as frustration and boredom.

Still applicable to education, but with many other uses, MacHardy *et al.* (2012) and Benzaid and Dewan (2010) look to detect the affect of subjects in the audience of a distributed presentation. When live streaming a presentation over the Internet, presenters have no way to assess the interest levels of their audience. Some new online presentation tools allow audience members to indicate their level of interest, but this feature is both rare and can be distracting for the user. Their work looks to use built-in webcams in the computers of each audience member to perform facial expression recognition and report levels of interest back to the main server. Aggregate information could then be supplied to the presenter automatically to assist presenters in understanding their audience.

Their system was trained on self-reported data, where a user would watch a video while being recorded. They would then review the video afterwards and tag sections as bored or engaged. They were unable to detect frustration better than chance. The studies were too small to be of statistical significance, with only five subjects in Benzaid and Dewan (2010) and twenty in MacHardy *et al.* (2012). In spite of this, their results show promise, in that a simple SVM trained only on the pixel values of the mouth, eyes and nose was able to distinguish between the two affects.

More recently work has focused on the development of affect detection for distance learning systems (Gogia *et al.* 2016). Their goal was to detect and report affect of remote learners. A multi-modal approach was developed based on a EEG hardware worn by the participants as well as a facial tracker based on the Kinect camera. While the results are promising and show the power of a single user system, the hardware constraints are unsuitable for large scale roll-out. Poulsen *et al.* (2017) show that EEG sensors can be used to reliably recognise engagement and that subjects in classroom settings exhibit reliable neural responses that were correlated across both subjects and experiments. This shows that EEG sensors are a potentially feasible way to collect ground truth data in the future. The idea of providing EEG sensors to all students in a large class could be problematic and will likely introduce issues practical when scaling up even if students were prepared to wear the equipment for extended periods of time.

Paquette *et al.* (2016) use a mix of the Kinect, software logs and quantitative field observations for subjects in front of a computer based learning game. Their field observations were made using the Baker-Rodrigo Observation Method Protocol (BROMP) (Ocumpaugh 2012) where a certified coder is used to provide labels. To be BROMP certified, the new coders must have a Cohen's κ score of 0.6 when compared with other previously certified coders. Unfortu-

nately such coders were not available for this research. Paquette *et al.* (2016) found issues with subject posture and interaction-based (software interaction logs) detectors, but ultimately found that the interaction-based detectors perform better. Conversely, Bosch *et al.* (2015a) found that face detector based systems functioned better than the interaction-based systems, although the latter suffered less from missing data. They note the difficulty of feature engineering and how it is difficult to know a priori which features will prove effective.

3.6 Prediction and Detection of Affect

The previous sections identified a number of learning-centred affects from the literature, some of which include: engaged concentration, boredom, confusion, frustration, and fatigue. It is useful for a teacher to be able to track all of these affects during class as the intervention required when a student is *disengaged and anxious* will be different from when the student is *disengaged and confident*. This section surveys some work on detecting these affects. Other affects such as happiness and anxiety are not considered in this work.

3.6.1 Boredom and Engagement

Engagement can be organised into three primary categories: behavioural, emotional and cognitive (Whitehill *et al.* 2014). *Behavioural Engagement* refers to a student's willingness to participate: Will they submit work? Will they attend class? Will they follow directions? *Emotional Engagement* concerns the students' emotional attitude towards the task at hand. This could be a student fulfilling all the requirements, but disliking or being bored by the task. *Cognitive Engagement* relates to the way in which the students' cognitive abilities are being used. These focus on attention, memory and creative thinking. In this first research endeavour, WITS is interested primarily in identifying lapses in cognitive engagement.

Whitehill *et al.* (2014) labelled users of an ITS as engaged or not-engaged using several untrained assessors. They were given specific instructions about labelling, using features like blinks and eye gaze. 10 second clips were labelled by the observers. The assessors had very high agreement labelling engagement as high or low, with a Cohen's Kappa (Cohen 1960) value of $\kappa = 0.96$. When discriminating between 4 levels of engagement, the reliability decreased to $\kappa = 0.56$. Frame-by-frame recognition was used to classify the video frames, which over time contribute to an engagement score for the segment. For classification they used GentleBoost with Box Filter features from Viola and Jones (2004). Support Vector Machines (SVM) with Gabor features, as well as Multinomial Logistic Regression (MLR) were used for emotion detection. When performing binary classification on high versus low engagement labels, the classifier accuracy was comparable to that of humans. The work reinforces the idea that automatic recognition of student engagement is possible.

The Augmented Multi-party Interaction (AMI) Consortium is a community looking to improve the effectiveness of meetings by creating software to properly support and regulate them. Kroes (2007) used the AMI video database of meetings to perform boredom analysis. He notes

three primary uses of such a system: assessing general efficacy of meetings, assessing communication skills of meeting participants, and generating meeting annotations that allow for sorting and searching based on participant boredom or interest. These uses naturally translate from the meeting setting to a lecture venue.

Kroes (2007) makes a number of interesting observations applicable to boredom in meetings, that apply more generally as well. He notes the temporal nature of boredom, that one should not be interested in just a single value, but rather a range of values over time. He also notes the existence of some mutually exclusive states – for example, excitement which can serve as a counter-indication of boredom.

Boredom specifically can be defined as follows:

> “Boredom is a state of mental weariness and dissatisfaction produced by lack of interest or activity.
> Blaszczynski *et al.* (1990)

> “Boredom is a state of relatively low arousal and dissatisfaction, which is attributed to an inadequately stimulating environment.
> Mikulas and Vodanovich (1993)

There are two components that are observed to create what is termed boredom: low arousal and dissatisfaction. These two-dimensional axes correspond to the valence-arousal axes mentioned previously and referenced throughout literature. The alternatives within these axes correspond to:

- Low arousal & Satisfaction: relaxation and peacefulness
- High arousal & Dissatisfaction: anger or fear
- High arousal & Satisfaction: excitement

It is important to distinguish between different causes of the state of ‘low arousal and dissatisfaction’. The definitions above both note that this feeling is a result of lack of stimulation. This is an important consideration as an affect detection system will not be able to tell the difference between apathy towards the subject or lecturer induced boredom. It is important that any user ultimately using the system is properly informed of these nuances.

By viewing the video corpus from AMI, Kroes (2007) identified a number of common signs of boredom in meetings. This was performed by untrained assessors. The assessors watched the videos and determined when a subject was bored. They then compared all the video segments of bored subjects against the video segments of engaged subjects. This resulted in a number of observations listed below which relate to either low arousal (relaxation) or dissatisfaction.

The identified signs are grouped into postural, facial and social signs. The postural signs include:

1. Slumped posture, which indicates low arousal as muscles are relaxed. This leads to a decrease in stature as the shoulders, back and head are slouched. The shoulders and head

are usually pointed forwards while the subject leans back into their chair.

2. Attentive subjects generally showed higher levels of movement than bored subjects. Low levels of movement with periodic, sudden changes was typical of both low arousal and dissatisfaction.

3. Some bored subjects let their hands lay idle, but most tended to use their hands to support their head, rub or clutch their face or placed them on their neck or behind their head. These are known as hand-over-face gestures and can be interpreted as signs of low arousal, but not necessarily dissatisfaction. Attentive subjects tended to place their hands on the table or somewhere in front of them, which may be a counter-indication of boredom.

The second group is made up of facial signals.

1. The amount that one's eyes open is an important indicator of arousal. Low arousal is typically evidenced by partially closed eyes and an increase in blinking.

2. Subject's gaze is indicative of arousal. Subjects who are bored tend to stare and generally show more attention to the surroundings of the room than to the speaker. It is also noted that attention is shifted more frequently between note taking and the speaker when the subject was engaged.

3. Someone who is bored tends to keep their mouth closed more except for possibly the occasional yawn. Periods of time without smiling may be an indicator of dissatisfaction.

Finally, Kroes (2007) mentions social signs:

1. Talking, writing, typing and other activity may be an indication of attentiveness, lack thereof may indicate low arousal or even boredom.

2. Acknowledging peers when they speak is an indication of attentiveness. This acknowledgement may take the form of looking at them, nodding/shaking ones head in agreement/disagreement or other facial cues to indicate focus.

In particular, the observations on posture and head movements are strongly supported by Wallbott (1998), who found a raising of the chin, tilting back and sideways of the head, lack of erect body posture and low movement all indicative of boredom.

Kroes (2007) made use of a *Mean Shift Embedded Particle Filter (MSEPF)* to track objects, such as hands, in real time. This performed well even when the object was in a cluttered environment and moving rapidly. He used *AdaBoost*, *Support Vector Machines (SVMs)* and *Linear Discriminant Analysis (LDA)* to classify Gabor wavelet representations of the video frames to detect facial expressions. This classifier worked with 93% accuracy to classify among neutral, anger, disgust, fear, joy, sadness and surprise. The classifier was trained on 626 examples from AMI, it ran in real time and required that subjects were in frontal positions to the cameras.

Fatigue detection was performed using pupil shape, frequency of eye blinking and yawning and a model based on Jin *et al.* (2007). To do this a neural network was trained using the three dimensions as input and the desired output as calculated in the model. Gabor filters were used to find the position of the facial features which were then monitored over time to detect blinks

and yawning. Context in the meeting was also considered but was detected with very high error rates.

Postural signs were detected by using naïve methods. Based on the fact that the cameras were stationary and that the background was unchanging, to detect a slumped posture, background subtraction was performed and the hight of the remaining object in the scene was measured. Similarly, detecting the average movement over a portion of the video allows the system to detect low or high periods of movement. Hand usage was tracked using the object tracking mentioned above. More sophisticated methods based on thermal imaging, stereo vision, Haar-like features and LDA are seen in Iwasawa *et al.* (1997); Jeong *et al.* (2008); Van den Bergh *et al.* (2008). Alternate methods can be found in surveys by Moeslund *et al.* (2006) and Gavrila (1999). Using segmentation, silhouettes and heuristic rules based on body shape characteristics, Juang *et al.* (2009) have created an efficient and effective posture estimator.

The detection of other facial and social signs, and actually detecting boredom were not done in his work. Gaze was not detected, and Kroes (2007) notes that this is a very difficult problem to solve accurately. Significant, successful work has since been done on tracking head poses and detecting the visual focus of attention with Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) (Ba and Odobez 2009 2011; Ba 2007; Smith *et al.* 2008), this is still a complex problem and in a lecture theatre where the camera is far from the subject it is difficult to establish student focus from their eyes. In this case, using the face is an acceptable approximation. Of particular interest is a survey on head pose estimation by Murphy-Chutorian and Trivedi (2009).

Work in boredom detection for the AutoTutor ITS is found in D'Mello and Graesser (2010) and much of the other work by D'Mello's group. These works are broad and primarily rely on multimodal approaches including conversational cues, gross body language and facial features. D'Mello and Graesser (2010) makes use of LDA to classify between a number of emotions, including boredom. They found that facial features and context were sufficient to detect boredom. Craig *et al.* (2008) found a strong correlation with AU 43 (eye closure) as well as general eye and mouth movement which supports the observations made above.

3.6.2 Confusion

Comparatively little work has been done in the detection of confusion. D'Mello and Graesser (2010) accurately used facial features to detect confusion. They found that the inclusion of discourse enabled the detection of confusion independently; when using the two signals together, their accuracy decreased. This is probably due to the *curse of dimensionality*. McDaniel *et al.* (2007) found correlations with AU 4 (lowered brow), 7 (tightening of the eye lids) and a lack of 12 (lip corner puller). This conflicts with D'Mello *et al.* (2006) who found a correlation with the presence of AU 12, rather than the lack thereof. Craig *et al.* (2008) established that AU 1, 2, 4 and 7 strongly correlate with confusion and frustration with near perfect scores.

Murphy-Chutorian and Trivedi (2009) note the use of head pose in conveying emotions such as confusion. While this seems intuitive, they do not specifically cite any literature or experiments to support their claim.

3.6.3 Frustration

As part of Picard's research group at the MIT Media Laboratory, working on an Intelligent Tutoring System, Kapoor *et al.*'s (2007) work specifically looks to detect frustration. Their research methodology is of particular interest. They note that self-reported feelings at the end of a task are "notoriously unreliable," while the use of external coders can quickly become an enormous and expensive task. They admit that less is known about the reliability of labelling by a subject in the heat of the moment, but giving them a way to indicate that they are – in this case – frustrated appears more reliable than self-reporting after the fact.

Their work involved the creation of a system that asks subjects aged 12 – 13 to solve the *Towers of Hanoi* problem on a computer. While solving the problem, a number of sensors monitored the students. They used a camera, a pressure sensing chair, a pressure sensing mouse and a wrist band to monitor skin conductance. While solving the problem, there was also a button on the screen that says "I'm Frustrated." They acknowledge that there may be learners that are frustrated but do not click the button. They argued, however, that prior studies from human computer interaction suggest that collecting self-perceived negative information by a computer in this way is at least more accurate than collecting the same information through trained experts. They cite Card *et al.* (1974); Lucas *et al.* (1977); Robinson and West (1992), drawing comparisons of computers and doctors taking embarrassing medical histories. A button click can then be interpreted as an indication of a subject being both *frustrated and aware of it*.

When a user clicks the button, the period leading up to the click is marked as frustrated. The machine learning system then used a number of features to track the subject. They used the IBM Blue-Eyes camera to track pupils based on the red-eye effect (Haro *et al.* 2000). A Hidden Markov Model (HMM) from Kapoor and Picard (2001) was then used to detect head nods and shakes. The system from Kapoor and Picard (2002) was used to detect shape information about the eyes and eyebrows. Based on the location of the eyes, a Support Vector Machine (SVM) was used to differentiate between 'fidgets' and 'smiles.' This vision system was able to run in real time at 27–29 frames per second on a 1.8GHz Pentium 4 machine.

The pressure on the chair was processed with the use of a mixture of Gaussians and a feed forward neural network which classified postures. Pressure on the mouse and skin conductance were also included. Data were classified using 5 methods: random (control), 1-nearest neighbour, SVM (radial basis function kernel), Gaussian process, and SVM (kernel of Gaussian process). Accuracies when predicting periods of frustration were 58% for control, 67%, 71%, 79% and 79% respectively.

This approach to real-time self-reporting is investigated further in Chapter 4.

3.6.4 Fatigue

All works found relating to fatigue detection were in the context of driving. The most accurate techniques for doing this are through the use of physiological sensors monitoring brain waves, heart rate, pulse rate and respiration (Healey and Picard 2000). These methods are not feasible for driver fatigue detection as it requires the use of distracting equipment.

For these reasons, Devi and Bajaj (2008) advocate the use of non-intrusive methods using video cameras to detect changes in eye blinking, sagging posture, leaning of the driver's head and the open/closed state of the eyes. Horng *et al.* (2004), Dong and Wu (2005) and Devi and Bajaj (2008) rely on eyelid movement to detect fatigue. They note that eye blink frequency increases beyond the normal rate when the subject is in a fatigued state. They also consider the notion of micro sleeps, which are short periods of sleep lasting 3 to 4 seconds. These are good indications of fatigue and can be detected by lengthened periods of closed eyes in subjects. They use skin tone to detect the face, and template matching to detect the eyes on the face. They monitor the number of consecutive frames in which the subject's eyes are closed by looking at the average intensity of the frames. If 5 consecutive frames are found, then a micro sleep is detected and the system alerts the driver of fatigue.

Another visual technique for detecting drowsiness is the detection of yawning. The integration of a number of signals such as eye blinks and yawning can produce a robust fatigue detection system (Wang and Shi 2005). Active Shape Models for this purpose are functional, but computationally intensive and therefore not practical for real-time implementations. Wang and Shi (2005) first use the Viola and Jones (2001) real-time face detection algorithm along with a Kalman filter to locate the face. Using template matching as well as colour, they then detect the mouth. By using the aspect ratio of the bounding box around the mouth as an estimator of openness, they consider any opening above some threshold to be a yawn.

Wang *et al.* (2006) present a survey of driver fatigue detection techniques. The only other visual methods surveyed include the direction of gaze, blinking rate, eye closure, mouth shape and head position. Other approaches rely on physiological sensors, such as EEG, and driver specific performance, both of which are not relevant to this work.

3.7 Conclusion

This chapter presented an overview of affective computing, emotions, and affect detection. Affective computing is the field of computing that aims to give machines affective abilities such as detection and expression. There are debates surrounding whether computers will ever be able to *have* feelings and if so, whether that is even desirable. There are arguments that emotions provide a bias when searching for logical conclusions in infinite search spaces, that they relate to multiple ways of thinking and even play a significant role on our perceptual abilities. The debate contends that for machines to reach human levels of intelligence, there must be some level of emotional intelligence.

Regardless of whether machines will ever be able to feel, literature already indicates that they can be endowed with skills in recognition and expression. Recognition can be used to improve human computer interactions so that computers can moderate work flows based on the user's current emotional state. Among others, this has implications for gaming, education, safety, and therapy. Affect expression may allow a computer to act as a proxy or even express itself to humans in a way that is intuitive and allows for more natural working environments. For example a long distance therapist may talk to a rendered avatar with realistic facial expressions, or a robot may show expressions that indicate its intentions to those around it.

Much work has now been done in understanding the physiological and expressive responses to a number of basic emotions, like anger, disgust, fear, joy, sadness, and surprise. These emotions, however, do not always relate directly to an educational context where one is more interested in engagement/concentration, boredom, confusion, frustration, fatigue, satisfaction/happiness, and anxiety. Understanding these emotions is the focus of ongoing work in the field of affective computing.

This chapter identified intrusive, in-the-moment, self-reporting as well as observational checklists to measure and asses these learning centred affects. Based on the work discussed here, Chapter 4 presents an investigation into a self-reporting and problems are identified. Following that, Chapter 5 presents the process of constructing a labelled dataset using assessors and then considers how this dataset can be used as an observational checklist to construct labels relating to engagement.

Chapter 4

The Intrusive Approach: Live Self-Reported Engagement

4.1 Introduction

The previous chapters outlined the importance of student affect in relation to learning and highlighted the need to be able to monitor and report this information to the lecturer. This chapter presents a web-based system called `Engage`, to allow students to self-report engagement levels, provide comments, and vote on questions using their smart phones. The system then presents a graph to the lecturer in real time that allows the lecturer to better understand the current affective state of the students in class.

There are a number of requirements for such a system, the most important of which is that is does not further distract students that may already be struggling to concentrate. The student interface must be simple and easy to navigate. It should not require any substantive reading, should not require in-depth instructions, and should not have any functionality beyond what is necessary.

In this chapter two main types of disengaging students are identified: those that think the presenter must speed up, and those that think the presenter should slow down. Students that quickly understand what is going on generally disengage from the lecture as the lecturer may be teaching at a pace suited to the average student in the class. While this is not a fundamental problem as those students now understand what is happening, there are dangers that these students become restless and disturb others in the class. These students may disengage and start playing on their mobile device, but then not notice the point where their initial level of understanding runs out.

On the other hand, there are students that may disengage because they do not understand what is going on. These students initially try to focus on the content, but if they have missed a fundamental building block they may find it difficult to catch up with the rest of the class. Over time these students become despondent and disengage with the lecture.

These two types of students are considered in terms of *positive sentiment* and *negative sentiment*. Considering the affects shown in Figure 3.2 the first case corresponds to students who

have gone through a cycle of affect including enlightenment and confidence, after which the lecturer remained on the topic for too long which then resulted in boredom. In the second case, students experience confusion, discomfort, and worry which eventually leaves them dispirited and frustrated. Ultimately both of these students disengage and boredom sets in.

The system caters for these two types of students with two buttons on the main interface. The buttons were labelled `Faster` and `Slower` and the explanation given to the students is paraphrased below:

> “If you feel bored because you understand what the lecturer is presenting and you want the lecturer to move on, then you should click `Faster`.
>
> If you feel bored because you've lost track of what the lecturer is presenting and you want the lecturer to re-explain something, then you should click `Slower`.
>
> Instructions to students.

In both cases, the system targets students who are already losing focus on the lecture. Whether the student falls into the first or second category, the student is starting to opt out of the lecture and at this stage it becomes useful for the lecturer to intervene if possible. Students that are concentrating are not encouraged to pick up their phone and respond as this may itself be a distraction.

Literature surrounding online classrooms finds that shy students are more likely to participate online than in a physical environment (Palloff and Pratt 1999; Vonderwell 2003). To cater for these students in a physical classroom where they would otherwise disengage, the system provides a text field where a student can post messages that appear for the lecturer in the same way as a twitter feed, but anonymous. This allows more reserved students to comment or ask questions where they are shy or believe that other students would consider something a ‘stupid question.’

Due to the real-time nature of the system, however, it is important that the students do not spend too much time typing the messages, and that the lecturer does not have to spend too much time reading them. If the lecturer feels that the messages take too much time to digest, he will simply stop reading them or will read them less frequently by which time the information may be out of date. For this reason the message box is limited to 140 characters.

Allowing students to comment in this way not only allows them to succinctly communicate with the presenter, but if their comments are related to the current topics, it provides text data that can be mined for further information. For example, if suddenly the number of ‘slow clicks’ increase one might be able to look at the comments to see more specifically what the problem was. If a large enough corpus were constructed, there may be scope to mine for sentiment in the text in the same vein as Go *et al.* (2009); Kouloumpis *et al.* (2011); Pak and Paroubek (2010).

A number of the students reported that they would prefer labels such as `I Understand` and `I don't Understand`. For these students the original labels were viewed as a criticism of the lecturer, rather than as a report of their current state. In a set up where the students like or dislike the lecturer, factors like this may heavily influence the responses. This is particularly important when considering the culture and social display rules that may be in play

during such an exercise. Fundamentally, changing the labels does not have an effect on what the system is trying to measure, as long as the meaning of the buttons is such that it separates the aforementioned groups of students.

The remainder of this chapter is organised as follows. Section 4.2 introduces background and work related to live reporting during class. Section 4.3 and Section 4.4 discuss the design and architecture of the implemented system: Engage. Section 4.5 presents the results of using the system during a number of first and fourth year classes. These results are considered in terms of both uptake and a questionnaire that was administered after using the system. Finally, Section 4.6 considers the results and presents a discussion thereof. Proposed future work relating directly to this portion of the research is presented towards the end of the thesis in Section 9.4.1 on page 179.

4.2 Related Work

4.2.1 Introduction

A survey of the literature identifies two primary methods of eliciting these responses using mobile devices. Specifically researches focusing on the use of SMS to receive comments and questions, or through the use of physical 'clickers' where the students can vote for solutions to problems or to vote as formative feedback for the lecturer (Boyle and Nicol 2003; Draper and Brown 2004; Lau *et al.* 2014; Lindquist *et al.* 2007; Markett *et al.* 2006; Patten *et al.* 2006).

This section provides an overview of this literature, starting with SMS communication (Section 4.2.2), and followed by approaches that make use of clicker hardware (Section 4.2.3). Section 4.2.4 concludes the survey with comments regarding the lack of available systems to provide formative lecturer feedback specifically.

4.2.2 SMS in and out of the Classroom

A number of works have focused on the use of Short Message Service (SMS) technology for teaching. Many of these works make use of SMS or even Rich Site Summary/Really Simple Syndication (RSS) as a way for teachers to send messages outside of class times as a means to motivate students, answer questions and handle class administration (Brett 2011; Faure and Orthober 2011; Lan and Sie 2010; Lau *et al.* 2014; Rau *et al.* 2008; Tarhini *et al.* 2015). These works are beyond the scope of this research, where the focus is instead on works where students are able to send messages to the lecturer in real-time during the class.

The first such system appears in Markett *et al.* (2006) where they developed a system for students to SMS their thoughts to the lecturer in real time. These comments would then be accessible to the lecturer both during and after class. They identified a number of different types of comments: Clarifying, Administrative, Content Comments, Greetings/Jokes, Direct Questions, and Spurious Comments. Using the system resulted in a more interactive learning environment where there was more opportunity for greater and ongoing feedback. The students

in the study reported an increase in both interest and motivation. The authors reported that the system gave people who were normally shy, the chance to have their say and that a number of issues were raised that usually would not have been if the student asking had been publicly identified. These results are supported in literature that followed their study (Huang *et al.* 2008; Lau *et al.* 2014; Lindquist *et al.* 2007). In these systems, the sender of the SMS is anonymised, and the papers do not report displaying the messages to other students in the class. In the works cited, a common issue raised by students is the recurring cost of messaging. This is especially relevant to this work as South African SMS text messages are considered to be prohibitively expensive and are rarely used in comparison to data based messaging services. South African students are unwilling to spend prepaid credit to send any messages for the purpose of class. A system that works over the internet allowing the use of the university's Wi-Fi or comparatively cheaper data bundles is vital.

There are works that make use of Twitter to allow students to comment and speak to presenters. In many cases the classes are run with a *hashtag* to which students can tweet. Two issues immediately come to light: anonymity and the separation of social and academic lives. Unless the student signs up for a new Twitter account specifically for the class at hand, there are issues that these comments are both public and no longer anonymous. Even then, a thorough analysis of tweets may give away a student's identity. If a student is hesitant to ask a 'stupid' question in class, one can be sure that the same student will not want to broadcast that same question to the whole world.

There are also issues involving the separation between the students' studying and social life raising ethical issues of displaying or reading their feeds during the class (Lowe and Laffey 2011). For example, if the lecturer projects the feeds onto a screen during the class, or if one student follows another's private account, then there may be personal issues that arise. At the same time, a public platform such as Twitter is open to tweets from outside parties that may disrupt proceedings.

These issues inform the decision to build a web-based system completely separate from the normal social networking of students and lecturers. The comment system of Engage specifically avoids the use of SMS technology, optionally allowing real-time display to all users while also anonymising all the displayed data.

4.2.3 Clickers in the Classroom

Other works make use of specific hardware to allow students to vote on questions during the class. Literature contains multiple examples where this is beneficial to learning (Draper and Brown 2004). It seems that these approaches are especially adept at providing an upward shift for the tail end of mark distributions (Boyle and Nicol 2003).

The idea of voting in class primarily replaces instances where the lecturer might ask for a show of hands to gauge how students respond to a question. Issues with asking for a show of hands primarily involve anonymity and the inability to actually count the number of votes in a reasonable amount of time (Boyle and Nicol 2003; Draper and Brown 2004; Huang *et al.* 2008; Lau *et al.* 2014). The introduction of clicker based voting systems effectively solve these problems.

The clicker systems usually provide the students with a 'Who wants to be a millionaire?' type multiple choice interface. A number of effective uses of these devices are outlined in literature. The most effective of theses approaches appears to be one where the students vote on the answer to a question, the result of the vote is displayed and students then discuss with their neighbour whether they agree with the answer that emerged. Following this discussion, a second vote is held after which the lecturer discusses the solution with the class. This mix of *peer-instruction* and *class-instruction* shows positive results when measuring both class interaction, student engagement and summative marks (Boyle and Nicol 2003; Draper and Brown 2004).

Complications with these systems, however, involve the use of custom hardware that students or universities must purchase and maintain. These systems often involve hardware that requires support staff or lecturer training to operate in classes.

A final issue with these systems is the lack of flexibility as the clicker hardware is designed with a very specific use case in mind. For example, they do not allow for text messaging that has been positively considered in literature above. As modern students are part of a generation that has grown up in a world of pervasive digital technology (Jones *et al.* 2009), there is a strong argument that the use of mobile phones allows for significantly greater flexibility and that future exploration should take advantage of this. A mobile phone based system can easily provide the functionality of both a text comment system and custom clicker hardware.

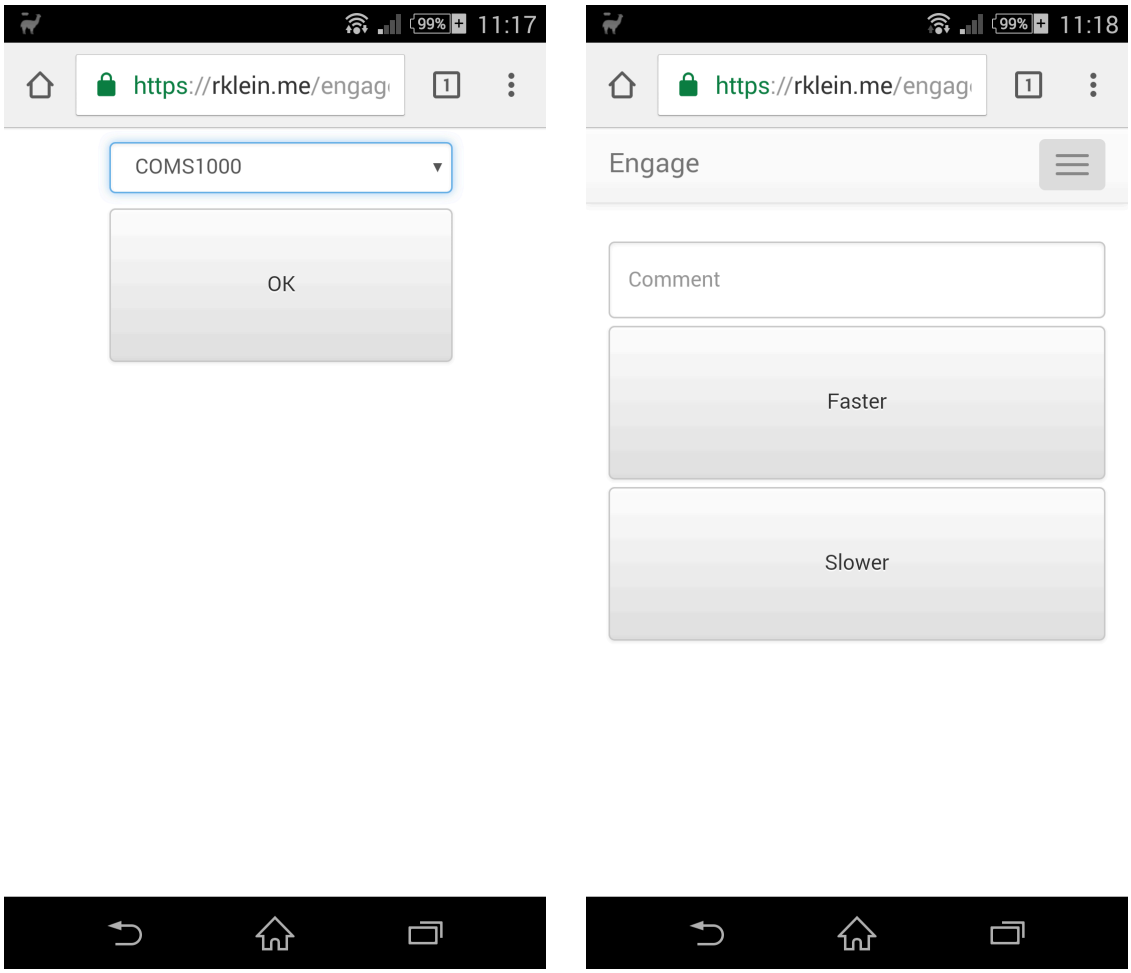
4.2.4 Systems for Real-Time Formative Lecturer Feedback

A review of the literature returned only one case of the research using the real-time feedback mechanism for formative lecturer feedback. The idea of using the clicker system to provide formative feedback on teaching is mentioned as a use case in Draper and Brown (2004), but is suggested as a vote in the same way a lecturer would stop and ask students to vote on a question. No continuous real-time manual systems for this purpose were found nor any further discussions regarding its use in education.

It is presented as the primary purpose and novelty of Engage, with voting and commentary as secondary applications. Throughout class Engage allows the students to provide formative feedback regarding their sentiment, and optionally comments and questions as necessary. The system can easily be changed into a voting mode to allow clicker type functionality. After the voting is complete, then Engage goes back to requesting and providing feedback for the lecturer.

4.3 Design

As discussed above, in the South African context it is imperative to avoid using costly SMS messaging and rather to use a purely internet based approach. This left the options of developing a smart phone app or a mobile site. A responsive mobile website was more popular with students as they did not have to install anything, did not have to allow an app any permissions on their phone, and for those with older feature phones a website was actually the only option.



(a) Landing Screen

(b) Main Student Screen

Figure 4.1: Student Views of the Engage System

When students first visit the site, they are shown the screen in Figure 4.1a which asks them to select their current lecture from a dropdown box. Only lectures taking place at that current time (with a small margin before the lecture starts) will be displayed.

When the student selects an option, the system displays the main student screen seen in Figure 4.1b. This screen shows two buttons and a text field. The main screen shows the `Faster` and `Slower` buttons already discussed, although these captions can be customised by the lecturer prior to use. The text box provides space for the student to enter a comment of up to 140 characters. The character limit can also be customised by the lecturer beforehand.

The presenter dashboard is shown in Figure 4.2. The dashboard displays a graph that tracks the responses submitted by students. These responses are placed into a number of *bins*, which allow the lecturer to adjust how the system displays information on the graph. The time resolution of the bins should be coarse enough that the graph is not just showing individual student clicks. But also fine enough that one can see responses while still discussing a single topic. Bins of 10 to 30 seconds seem appropriate. The lecturer can also select a specific starting time and session length to display. This allows the lecturer to go back and review previous presenta-

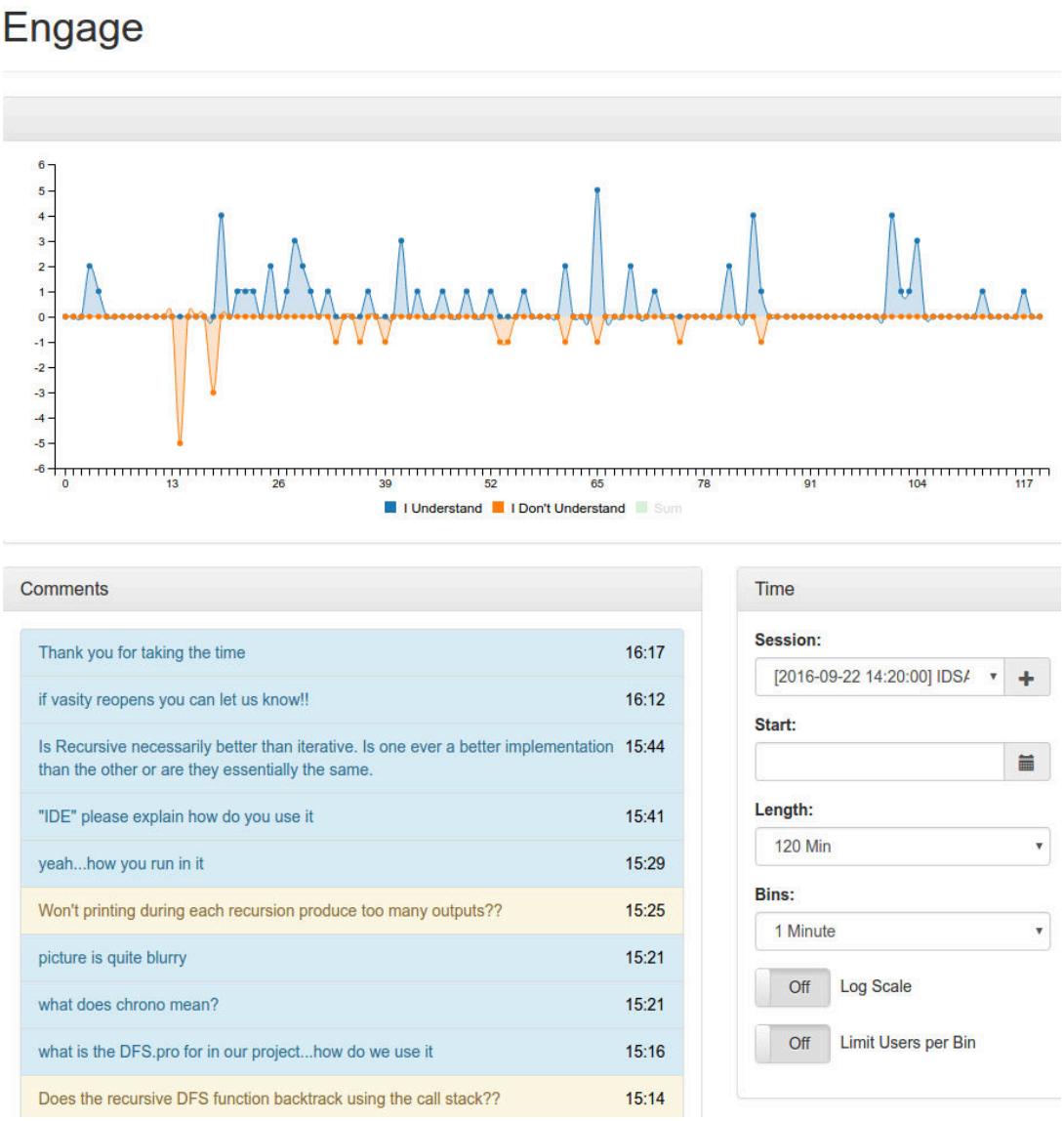


Figure 4.2: Instructor Dashboard

tions. To view live information, the lecturer should set the starting time, and the length should be set to `Now`. The dashboard graph will then update every 5 seconds to display new data as it arrives.

The graph itself displays 3 values. The blue and orange graphs display `Button0` (`Faster/I Understand`) and `Button1` (`Slower/I Don't Understand`) clicks respectively. The blue graph always grows upwards with the number of clicks in the current bin, while the orange graph grows downwards. A green graph displays the difference between the two types of clicks. Specific graphs can be turned on and off by clicking on the legend of the graph. Figure 4.2, for example, has the sum deactivated, and the two buttons shown.

Underneath the graph, the lecturer can choose a number of display settings. The lecturer can choose which lecture to display by setting the session. The start value allows the user to select a specific time to start; this is useful when one may want to display data from halfway

through a session. The length to display can be set to a specific time or one of two special values: `End` which shows data up until the scheduled end of the lecture or `Now` which shows all information up until the current time – this is the best view for live display of the graph. Other values of length can be selected from 10m, 30m, 60m, 90m, 120m. The bin's setting adjusts the length of each time bin and adjusts the resolution of the graph, which is effectively a histogram showing the number of clicks in a bin. This setting can be selected from 1s, 5s, 10s, 30s, 1m and 5m.

The remaining two settings allow scaling of the graph and rate limiting of users. In large classes the presenter may be more interested in the log scale of clicks, which is easier to read in some cases. Let U be the set of all unique users using the system. Let f_i^u and s_i^u be the number of times that user $u \in U$ clicked faster and slower in time period i. Let F_i, S_i and Σ_i be the 3 plots.

With log scale and user limiting turned off, graphs are just the total number of clicks in the relevant time period:

$$F_i = \sum_{u \in U} f_i^u, \tag{4.1}$$

$$S_i = -\sum_{u \in U} s_i^u. \tag{4.2}$$

When log scale is on and user limiting is off the system just scales everything by $\log_2$:

$$F_i = \log_2(1 + \sum_{u \in U} f_i^u), \tag{4.3}$$

$$S_i = -\log_2(1 + \sum_{u \in U} s_i^u). \tag{4.4}$$

These two approaches, however, allow a single student clicking n times to shift the graph as though n different students had each clicked once. These are two very different situations from the perspective of the lecturer, and so user limiting allows one to get a better sense of how the class is feeling. When log scaling is off, and user limiting is on, then each unique user may only contribute 1 click per bin, i.e.,

$$\delta(x) = \begin{cases} 1 & x > 0 \\ 0 & otherwise, \end{cases} \tag{4.5}$$

$$F_i = \sum_{u \in U} \delta(f_i^u), \tag{4.6}$$

$$S_i = -\sum_{u \in U} \delta(s_i^u). \tag{4.7}$$

Finally, if both log scaling and rate limiting are on, then every click that a student makes is logarithmically scaled. So a single student clicking 8 times is counted as though $\log_2(8) = 3$ unique users had each clicked once.

$$F_i = \sum_{u \in U} \log_2(f_i^u), \tag{4.8}$$

$$S_i = -\sum_{u \in U} \log_2(s_i^u). \tag{4.9}$$

In all cases, the third `sum` graph is defined as:

$$\Sigma_i = F_i + S_i. \tag{4.10}$$

Finally, the dashboard also shows the presenter the comments that are arriving from the students in real time. If a student clicks *Faster* when entering a comment, it will appear in blue. If the student clicks *Slower*, it will appear in orange. The comments appear with the most recent comment on the top as well as a time stamp.

4.4 Architecture

The server is implemented with a restful web interface over three PHP pages. The first, `index.php`, handles interaction with the students. The page checks for a cookie containing a sequential user id and lecture id. If this is not found then there is a new user and the landing screen (Figure 4.1a) is displayed for them to select a lecture. Once this is done, the user is directed to the main student page (Figure 4.1b) which shows the two buttons along with the lecturer specified captions, or multiple buttons if the system is in voting mode. When the student clicks a button, the form is submitted using http POST.

The lecturer dashboard, `dash.php`, is built using the C3 JavaScript library[1] and Twitter Bootstrap[2]. JavaScript updates the graph and comment box once every five seconds, by invoking an AJAX call to `data.php`.

`data.php` accepts the following GET variables:

- `lecture` - The lecture id. Only clicks that correspond to the specified lecture are considered. If set to -1 or if it is missing, then clicks across all lectures during the specified times are displayed.
- `start` - The start date and time as "YYYY-MM-DD HH:mm". For example, 12 September 2016 at 15:20 is formatted as 2016-09-12 15:20. Only clicks after this time are considered. This field may be omitted if a `lecture` is provided, as each lecture in the system is loaded with a specified start time.

[1] c3js.org
[2] getbootstrap.com

- `len` - Length of time, in seconds, to fetch. If this is set to -1 (`Now`), then the server will respond with all data from the start until the current server time. If set to 0 (`End`), then the server will calculate the length using the scheduled end time for the lecture.
- `div` - The length of each bin, in seconds.
- `logscale` - If set to 1, then log scale is activated.
- `user` - If set to 1 then users are rate limited.

When visiting `data.php`, the server will respond with a JSON object of the following form:

```
{
  "data": {
    "t" : [0,1,2...],
    "Button0Caption": [1,0,2...],
    "Button1Caption": [0,-1,-1...],
    "Sum":[1,-1,1...]
  },
  "comments": {
    "t" : ["TimeStamp1","TimeStamp2"],
    "button" : [0,1],
    "comment" : ["Comment1","Comment2"]
  }
}
```

The response contains `data` and `comments`. The data field contains the number of times each button was clicked in a given time period, where `Button0Caption` and `Button1Caption` are replaced by the actual captions. Values associated with the second button are always reported as negative so that the `data` object can be plotted directly by C3. The `comments` object contains the all comments sorted by the comment time with most recent comments first as well as which button was pressed when the student submitted the comment.

Figure 4.3 shows a schematic representation of the system along with a login module that integrates with the university course management software. Lecturers can create new lectures, or automatically pull information about lectures from the university database. Lecturers can add private notes about the class, set start and end times, and customise the button captions. The lecturer can also decide whether non-authenticated users can see the graph, the comments, or both.

4.5 Results

4.5.1 Uptake

The implemented system was used in a number of Computer Science lectures at the university. In these lectures the system was explained to the students as a way that they could report their engagement and understanding to the lecturer that was both live and anonymous. Table 4.1

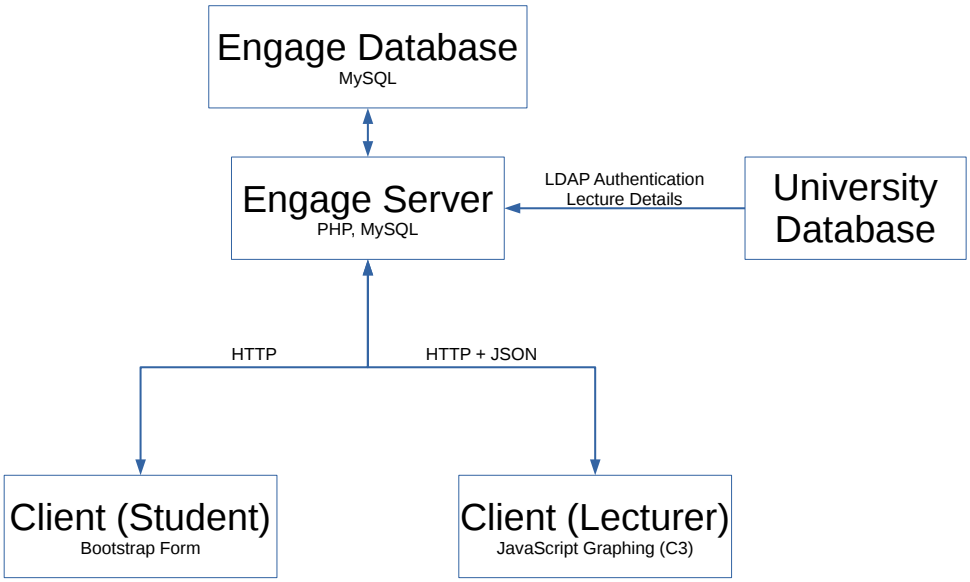


Figure 4.3: System Architecture

Table 4.1: Student Responses Using Engage in Live Lectures

#	Year	Class Size	Total Clicks	Total Users	Live Feedback
1	First	120	40	3	None
2	First	200	46	12	Lecturer Only
3	First	200	364	21	Projected for Students
4	First	200	1068	26	Projected for Students
5	First	200	638	20	Projected for Students (comments on)
6	First*	35	94	10	Projected for Students (comments on)
7	First†	24	74	11	Students on home computers
8	Fourth	18	131	16	Lab computers available
9	Fourth*	12	17	7	Lecturer Only, 45 minute lecture

shows the lectures in which the system was used and reports on the number of students that used the system in each class. All lectures took place as two 45 minute sessions with a 15 minute break in between. This means that for some time $t \in [0; 45]$, a lecture was in progress. For $t \in (45; 60)$ the students were on break and for $t \in [60; 105]$ the second half of the lecture was in progress.

Due to protest action on campus, some live classes had significantly decreased numbers (marked by *). Classes presented through live streaming on YouTube are marked with a dagger (†). There was no graph for Experiment 1. Experiments 2 to 7 (inclusive) were a single first year cohort and Experiments 8 and 9 were a fourth year cohort. The graphs for the experiments are shown in Figures 4.4 to 4.11. The x-axis of the graphs corresponds to the number of minutes since the start of the lecture. The graphs exclude 'testing clicks' when the users were first introduced to the system.

Experiment 1 (120 First Years)

Originally the system was launched in a Computer Science class of approximately 120 first year students. The experiment was run in such a way that neither the lecturer nor the students had live access to the responses during the lecture, but that the lecturer would have access to this data afterwards. There was an extremely low uptake of the system with only 3 students using it for the first 25 minutes, after which even they stopped using it. Given that the lecturer would only be made aware of student problems after the class, the students lacked motivation to report anything during the class. Generally students preferred directly asking the lecturer questions during the class. The students were not interested in using this system unless the results were immediate.

Experiment 2 (200 First Years)

Subsequent experiments were performed with the next year's cohort of first year Computer Science students. There were approximately 200 students attending this class. This time the graph was loaded and displayed to the students once at the start of the lecture as the system was first introduced. Following that the graph was visible to the lecturer but not the students.

In this experiment, there were 68 clicks in total, made by 12 different students (6% of students). When the system was first explained to the students there was an initial rise in the number of clicks in both faster and slower directions as students tested the system. After the initial experimentation died down and the lecture began, there were 46 clicks during the lecture itself. Every user that experimented made at least 1 further click once the lecture started. All but two of the clicks took place in the first half of the double lecture. After the 15 minute break, most students had closed the browser on their phone and did not visit the program again. There were 4 clicks during the break and 2 clicks after it. Students' interest in using the system diminished over time as they had no feedback to see that their specific clicks were visible to the lecturer. The graph for experiment 2 is shown in Figure 4.4 and shows how the number of clicks were heavily weighted towards the start of the lecture.

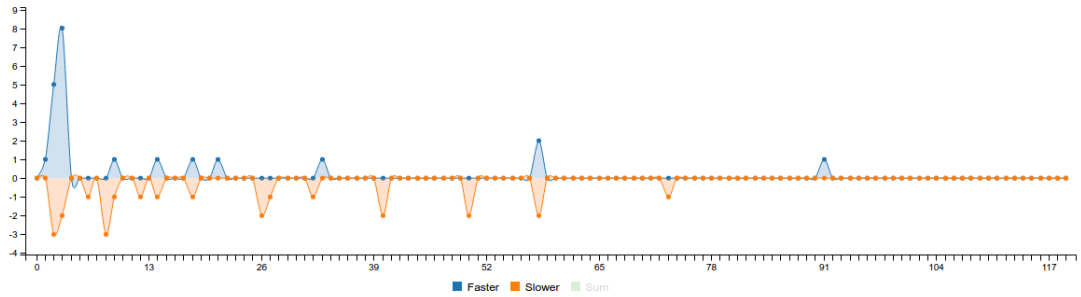


Figure 4.4: Experiment 2: First Years, feedback to lecturer only.

Experiment 3 (200 First Years)

In the third experiment, the graph was projected above the chalk board throughout the entire lecture. The graph updated once every 5 seconds and the students were able to see for themselves that their click affected the graph in real-time. In this experiment there were 364 clicks in total, made by 21 different students (11%). These numbers exclude experimental clicks that preceded the start of the actual lecture. Now that students were able to see their clicks affecting the graph immediately, they were more likely to click multiple times as they could see their influence over the graph increase. This time there were an average of 17 clicks per student, and a median of 7. The 4 students with the most number of clicks had 87, 70, 64, and 53 clicks respectively, while the remaining 17 students clicked on average 5.3 times (ranging between 1 and 10).

There was a large downward spike just before the 105 minute mark (45min lecture + 15min break + 45min lecture = 105 min) which was caused by 3 students obviously ready for the lecture to finish. These results are shown in Figure 4.5.

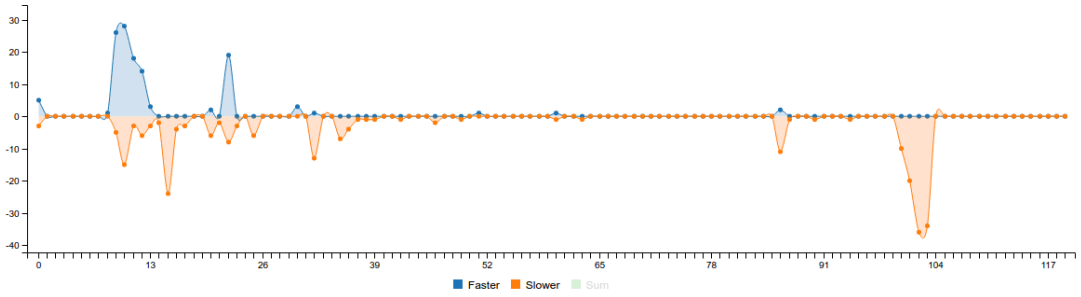


Figure 4.5: Experiment 3: First Years, feedback projected for all students.

Experiment 4 (200 First Years)

The fourth experiment was done in the same way as experiment three. This time there were 1216 clicks from 26 different students (13%). The average number of clicks per student was 46 with a median of 14 clicks. The top 2 students clicked 336 and 236 times. The remaining click counts were fairly evenly distributed between 1 and 70 clicks per student. There are generally more clicks where the students ask the lecturer to go slower. While these clicks are still skewed

towards the first half of the lecture, we see that they are more spread out over those first 45 minutes. It appears that this skew still occurs as students do not go back into the system after the break (between minutes 45 and 60).

Of interest in this lecture, is that for the first 35 minutes, there were between 6 and 10 unique users clicking at least once every 5 minutes. These students were using the system correctly as requested and the feedback was incorporated into the lecturer's decisions during class. Later on, there was a downward spike just before the break as students anticipate it coming. There is no spike at the end of the lecture as all students had stopped using the system by that time.

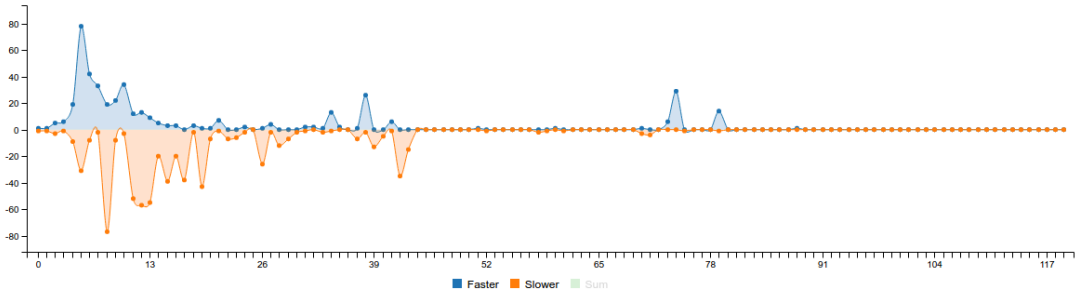


Figure 4.6: Experiment 4: First Years, feedback projected for all students.

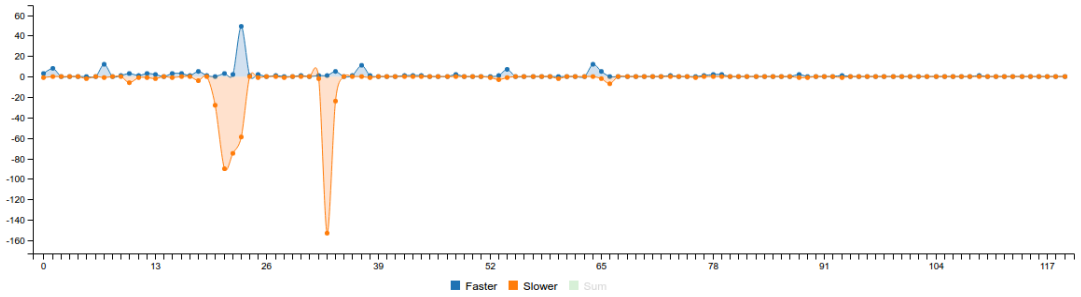
Experiment 5 (200 First Years)

This experiment was run in the same way as the previous one except that the comment system was now enabled. Comments would appear projected above the blackboard in real-time. There were 638 clicks in total made by 21 different students (11%). At 37 minutes into the lecture, there is a large downward spike of 122 slower clicks over 3 minutes. These clicks were all done by a single user that asked a question using the comment box, and continued clicking slower until the question was answered. Figure 4.7a and Figure 4.7b show the original and log scale graphs for this lecture. When there are large spikes in the number of clicks, the log scale helps maintain the usefulness of the graph as the y-axis rescales.

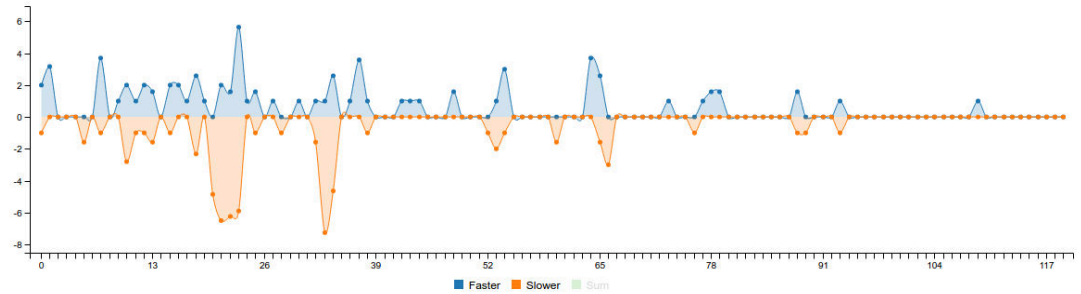
Over the course of the lecture there were 35 comments typed into the box. Thirty-three (33) were jokes and unrelated comments and two serious questions for the lecturer.

Experiment 6 (35 First Years*)

In this week student unrest erupted on campus and only 34 students attended the lecture. In this lecture 10 students used the system. Interestingly, the clicks were spread out fairly evenly across the lecture. There were 42 comments throughout the lecture. 8 comments were directly related to work being covered, while the remaining 34 were unrelated.



(a) Original Graph



(b) Log Scale Graph

Figure 4.7: Experiment 5: First Years, feedback projected for all students with comments enabled.

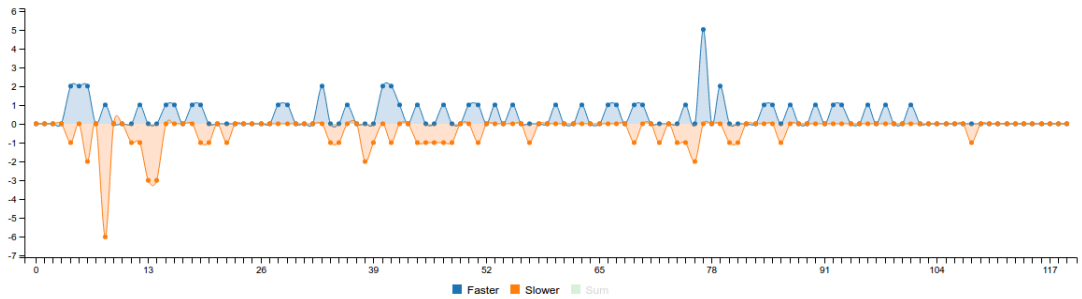


Figure 4.8: Experiment 6: First Years, feedback projected for all students with comments enabled.*

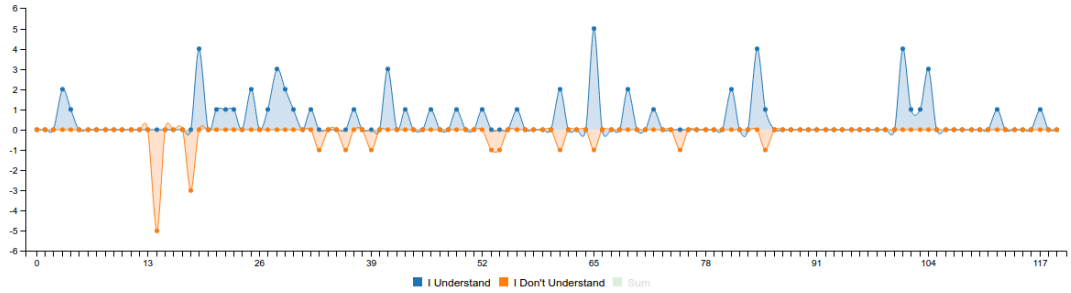


Figure 4.9: Experiment 7: First Years, lecture streamed on YouTube feedback and comments available on students' devices at home.†

Experiment 7 (24 Remote First Years†)

With protests in full swing campus was closed. The opportunity was taken to live stream a lecture over YouTube while the students solely used Engage to communicate with the lecturer. Students were able to stream the live video from home and were told to use Engage to communicate with the lecturer. Before the lecture began and for the first time, the students were explicitly asked to keep the comments focused as it was the only channel with which students could ask questions and provide feedback. There were 74 clicks in total with 11 different students using the system. There were 24 comments throughout the lecture and all of them were directly related to content the lecturer was talking about in real time. The clicks were very evenly spread as students responded when the lecturer asked if students understood the concepts being presented.

Experiment 8 (18 Fourth Years)

The following two experiments were conducted with a class of fourth year Computer Science students. Experiment 8 was a 2 hour lecture that took place in a computer lab. The students were able to have Engage and the graph open on the computer in front of them. The graph was shown on an iPad to the lecturer; this was not the first year lecturer from the previous experiments. In this class the lecturer used the projector for slides. 16 different students clicked a total of 131 times. There were 7 comments in total and all of them were directly related to what was being discussed in class at the time. In this case the students were not specifically prompted to stay on topic, and this may be an indicator of the maturity of the older students.

Experiment 9 (12 Fourth Years*)

This experiment took place in the week when protests had broken out on campus, the class size was smaller than usual and the students were forced to move to a different lecture theatre. The lecture ran for only a single 45 minute session. The graph could not be shown to students live as they were not in front of computers and the lecturer was using the projector for slides. There were 17 clicks in total by 7 unique users out of the 12 that came to the lecture.

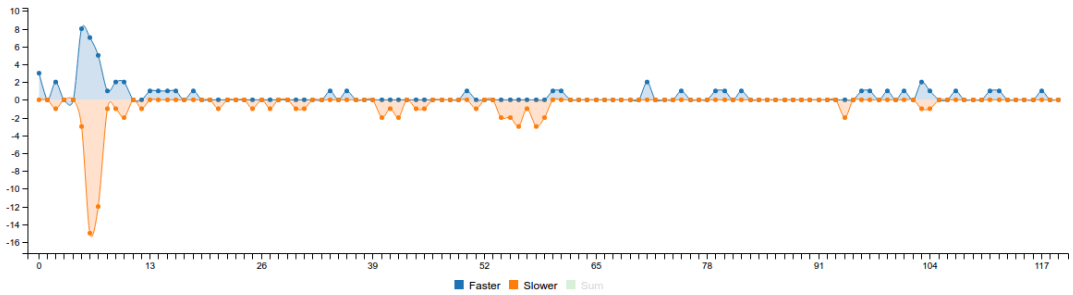


Figure 4.10: Experiment 8: Fourth Years, feedback accessible through lab computers with comments enabled.

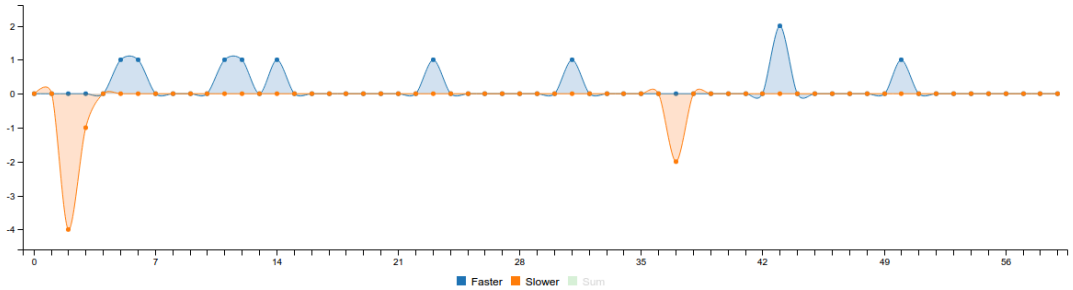


Figure 4.11: Experiment 9: Fourth Years, feedback to lecturer only.*

4.5.2 Student Comments

Students were asked to fill out a brief anonymous survey about their attitudes towards the system.

First Year Students

After using Engage a few times (after Experiment 5), the first year students were presented with the following questionnaire.

Table 4.2: Did you use Engage to report distraction during the lecture?

Answer		Responses	
Yes	I used it a lot	0/9	0 %
Yes	I used it once or twice	6/9	67 %
No	I didn't feel comfortable using it	0/9	0 %
No	I was trying to focus on the lecture	2/9	22 %
Other		1/9	1 %

Table 4.3: Do you think the lecturer was responsive to the information received?

Answer	Response	
Yes	10/10	100 %
No	0/10	0 %

Table 4.4: Would you like to use this app in all your classes?

Answer	Response	
Yes	8/9	89 %
No	1/9	11 %

Table 4.5: When asking short questions do you prefer...

Answer	Response	
... using the comment box on engage	6/9	67 %
... raising my hand to ask the question	3/9	33 %

Table 4.6: Did you find the program distracting?

Answer	Response	
Yes	2/9	22 %
No	7/9	78 %

Table 4.7: What buttons should the program display?

No responses

Table 4.8: Please provide any further comments you might have.

Comments
Worked really well in terms of how quickly the feedback responded
Comments can be completely unrelated to class due to their anonymity
Comment box should be bigger

Fourth Year Students

After their first time using it (Experiment 6), fourth year students were presented with the questionnaire.

Table 4.9: Did you use Engage to report distraction during the lecture?

Answer		Responses	
Yes	I used it a lot	1/10	10 %
Yes	I used it once or twice	7/10	70 %
No	I didn't feel comfortable using it	0/10	0 %
No	I was trying to focus on the lecture	2/10	20 %
Other		0/10	0 %

Table 4.10: Do you think the lecturer was responsive to the information received?

Answer	Response	
Yes	9/10	90 %
No	1/10	10 %

Table 4.11: Would you like to use this app in all your classes?

Answer	Response	
Yes	9/10	90 %
No	1/10	10 %

Table 4.12: When asking short questions do you prefer...

Answer	Response	
... using the comment box on engage	7/10	70 %
... raising my hand to ask the question	3/10	30 %

Table 4.13: Did you find the program distracting?

Answer	Response	
Yes	5/10	50 %
No	5/10	50 %

Table 4.14: What buttons should the program display?

Response
I understand / I don't understand
Vote buttons
Same as it is currently, but better worded.
What is shown now.

Table 4.15: Please provide any further comments you might have.

Comments
The section covered should be represented in the graph (i.e. capture when a section is started and when a section is ended)
I like it's fun

4.6 Conclusion

This chapter presented *Engage* which allows students to report live their engagement levels to the lecturer during class. An engagement graph is presented to the lecturer in real time allowing lecturers to adjust their teaching styles to better cater to the current state of the class.

Unfortunately experiments were interrupted by unrest on campus and lectures were either poorly attended or cancelled altogether. In all the lectures that did take place during the unrest, the uptake of Engage was higher. This suggests a correlation between the students who attend lectures during protests and those that are likely to make use of such a self-reporting system. Unfortunately the cookies identifying a user expire after a few hours so there was no way to track whether the students using Engage in the normal lectures were the same as those using them during the protests. However, this seems like a reasonable assumption and has important ramifications. It suggests that the students using the system are the more committed students. This raises questions about whether the students that are prepared to use a self-reporting system are indeed the one's disengaging and if they are the ones that the lecturer needs to know about.

In all experiments, students used the system significantly more if they were able to see the graph updating in real time. This was true for the first years, where they went from not seeing the graph (3/120 and 12/200) to seeing the graph (21/200, 26/200, 21/200, 10/35), as well as the fourth years, where the graph was available in the first lecture (16/18) not available in the second lecture (7/12).

In general, the mature fourth year students only used the comment section seriously, whereas the first year students appeared more distracted by it. When asked to keep it focused, however, the students took it seriously or did not post. While this was a concern beforehand, no students used the comment section to post any offensive material. In classes where there is breakdown in the lecturer–student relationship or where the comments are all spurious, it may be wise not to show posted comments to the students.

Based on lecturer feedback, students preferred asking short, off-topic or 'silly' questions using the comment box. This meant that the questions could be queued up and the lecturer would deal with them once the current train of thought was finished. This was very convenient for the lecturer as it was easier to work through the queue of questions at the end of a concept. Often if students needed the lecturer to repeat something, they asked using the comment box. Students preferred asking complicated or mathematical questions by raising their hands.

The fourth year students reported higher levels of distraction, although when asked whether they'd like to use the system in other classes the majority said yes. The few first year students that responded to the questionnaire said they did not find the program distracting. However, a

reading of the comment feeds generated in their classes show that they were indeed distracted by the ability to post arbitrary anonymous text to the projector during the lecture. When specifically asked to stay on topic, there was not a single spurious comment. This indicates that the students were largely operating with the boundaries of the student–lecturer relationship. This suggests that if the lecturers encourage meaningful use of the comment section, then as the novelty of being able to post to the projector wears off, and as the students further mature, this will become less of a distraction.

The experiments presented in this chapter showed that in the larger first year class only about 20 of the 200 (10%) students were consistently interested in using the system. While this number may seem fairly low, this was the first time the students were exposed to the system and effect of aspects such as the student–lecturer relationship (approachability, formality, etc), subject difficulty, and cohort should be considered in a longer study.

The system presented to the students only showed the positive and negative sentiment buttons. This could mean that students that were concentrating or engaged in flow were willing to use the system but did not need to. Asking these students to use the system will interrupt that flow, and potentially cause more distractions to those that are legitimately concentrating. However, a voting system has been incorporated into the system to allow the lecturer to use Engage in two modes: reporting engagement or as a traditional clicker for voting. This provides a way to more accurately measure how many students are willing to use the system, but do not feel that the current captions describe them.

As the system is used in more contexts, one may be able to identify whether the uptake will increase through mere exposure over different subjects, with different lecturers, and for longer periods of time. The almost universally positive feedback from the admittedly small number of respondents about the system indicate its utility, especially for students that would otherwise avoid speaking up in lectures.

As the number of students using the system was low it is difficult to take the results as representative of the entire class. There may be significant bias in the type of student that is willing to use such a system and if so, they may be the kind of students that engage well during lectures. It might be the case that the disengaged students that the system is trying to identify are not the students that would report that information. Ultimately it may be that the majority of students are just unwilling or unable to self-report on their engagement during the class and if that is the case, then *non-intrusive* engagement monitoring tools that do not require student intervention should be developed.

Chapter 5

The Non-Intrusive Approach: The Wits Intelligent Teaching System

5.1 Introduction

In the previous chapters the importance of being able to assess the affective state of a class was established. In the preceding chapter, a system was developed to allow students to self-report their affect using their mobile device. The system then displayed the data to the teacher live during class.

It is believed to be the first system designed specifically to provide formative lecturer feedback during the class and its effect on teaching and learning should be considered in the future. The method outlined above however, has a number of drawbacks. The first of which is the reliability of the data. Even if the majority of students bought into the idea and reported their affect, literature suggests that while further study is warranted, the results can be very unreliable (Kapoor *et al.* 2007). In many cases students are unable to reflect on their current state adequately. For example, asking a student to report in each time he is distracted may be a fruitless exercise as the student is not directly aware of the number of times he checks his cellphone messages or talks to a friend. On the other hand, asking another student to report that she is concentrating is self-defeating in that it breaks the train of thought and introduces a high level of distraction.

The low response rate is also a concern. While it is difficult to draw significant conclusions from the data collected thus far, the numbers in the previous chapter suggest that the majority of students simply opt not to use the system at all. A primary argument for this is the numbers of 'test clicks' at the start of a lecture. As only about 10% to 20% of the class even seem to test the system at the beginning, it indicates that the remaining 80% or 90% have no intention of using the system regardless of their affective state. Future work using alternative presentations of the system or encouraging some kind of gamification that ties into the lecture may encourage more students to make use of it and be able to draw in more support.

The majority of students appear either unwilling or unable to report their affect, and therefore at this stage Engage is not able to collect enough data to make informed decisions about

the class as a whole. Because the system is *intrusive* by nature, it means that for it to work properly there must be *buy-in* from the students and it needs to be setup in such a way that it does not cause distractions. In this light, the need for a *non-intrusive* automatic mechanism becomes apparent.

The WITS INTELLIGENT TEACHING SYSTEM (WITS) is proposed as a smart lecture theatre to meet this need. WITS provides an intelligent learning environment that automatically monitors the students in the class during each teaching episode. Using computational vision and machine intelligence approaches, the system monitors students and reports to the lecturer about their affective state. The system presented in this work is based entirely on vision data, but should also be extended to monitor other sensor feedback as well. Currently vision is used to perform inferences about the students based on common actions or gestures, body posture, and facial expression. In the future, physiological, audio, and environmental sensors could ultimately augment such a system and improve the overall accuracy.

Ideally a non-intrusive *smart lecture theatre* such as this should be able to report to the lecturer about the affective state of the class in real-time. It should also allow the lecturer to review the captured data in more detail after the class has finished. By generating feedback live during class, the system enables contingent teaching on an unprecedented level. By allowing review of the data after the class, the system allows self-evaluation and reflection by a teacher where one can assess the overall efficacy of his or her teaching methods and how well certain content was received by the students.

It is already established that expert teachers are able to asses the state of the class to a large degree (Kapoor *et al.* 2007), but this is largely based on intuition. New lecturers are usually left to develop this intuition and experienced lecturers work on the basis that their intuition is correct. WITS can be used to train new staff members and assist them in knowing what to look for and where to focus their attention. On the other hand, experienced lecturers can use the system to objectively confirm their intuition, to identify 'blind spots' and to better understand how the classes behave when one's back is turned.

Assessing the state of large classes in-the-moment is a difficult task, and while the intuition of what makes for an engaged student may be straight forward to some, the experiences in this work show that even just looking back and reviewing a recorded video can yield surprising revelations. Reviewing every student in a video after class is an arduous task that any busy academic would consider laughable if asked to do frequently.

Furthermore, understanding the correct strategy to combat different patterns of disengagement may not be a straight forward task. Some patterns may indicate that there are problems with a specific group of students, while others may indicate that there was a party the night before. By recording these patterns over time and then analysing them, a smart lecture theatre's AUTOMATIC TEACHING ASSISTANT (AUTOTA) may one day be able to actually suggest hints to the lecturer in a pedagogically sound way based on the efficacy of strategies tested over time.

The remainder of this chapter is organised as follows. Section 5.2 presents the architecture of the WITS system. Section 5.3 explains the development of a database to support the machine learning process. Specifically, Sections 5.3.1 to 5.3.4 describe the cross-platform, open-source WITS ANNOTATION TOOL (WITSAT) that was developed to support data annotation and Section 5.3.5 outlines the WITS DATABASE (WITSDB) of labelled video sequences. This

data forms the foundation upon which Chapters 6 and 7 train and test a number of SVMs and CNNs. Section 5.4 outlines the training and testing process and Section 5.5 briefly concludes the chapter.

5.2 Architecture

The ultimate goal of WITS is a smart lecture theatre equipped with video, audio, environmental, and other sensors that can be used to monitor students during lectures. Video cameras should be placed throughout the room at a high enough elevation that occlusions caused by student seating arrangements are minimised. The cameras should directly face the students and provide as frontal a view as possible. The feeds from these cameras should be live streamed to the central servers to allow for real-time processing. Ideally the view and resolution should allow for body posture, classroom actions, and facial expression analysis. These are all modes of expression identified in Chapter 3.

Microphones placed throughout the room should be used to monitor noise levels in various frequency bands. Recording actual voice may prove an invasion of privacy, but by rather tracking the amplitude of only a few wide frequency bands, the system could detect periods where students are whispering to one another (high frequency), where a student is asking a question (voice frequencies in the student group), or where the lecturer is speaking (voice frequencies at the front of the hall). On the other hand, detecting levels of ambient noise itself may prove a useful feature when considering the likelihood of distraction. Infrasonic noise (7–20Hz) is known to cause drowsiness and discomfort (Fecci *et al.* 1971; Landström and Byström 1985; Landstrom *et al.* 1991; Leventhall *et al.* 2003), and severely hinders task performance (Kyriakides and Leventhall 1977; Landstrom *et al.* 1991; Persson Waye and Rylander 2001; Persson Waye *et al.* 1997). Using audio sensors to check for environmental distractions, as well as to establish what is happening in the room may assist the system in monitoring the students.

Other environmental factors such as CO_2 levels, temperature and humidity should also be considered and monitored as these are known to negatively effect students' concentration, performance, and even attendance (Heudorf *et al.* 2009; Mendell and Heath 2005; Shendell *et al.* 2004).

Other specialised hardware such as thermal infra-red cameras have shown success monitoring various physiological phenomena such as the cutaneous temperature on the face or even heart rate. Other sensors such as body pressure monitoring systems or galvanic skin response bands may be useful, but ultimately require significant hardware investment on a per student or seat basis and start to require student participation again.

These sensors should be installed throughout a venue such as that shown in Figure 5.1 to achieve the best coverage possible. The sensors should feed information back to a server that extracts relevant features and classifies students along the various axes of interest. This information should be available to the lecturer in real-time during the lecture and for review afterwards. The presentation and acceptance of this information is discussed in Chapter 8.

In the current study, however, no secure permanent facility was available in which to install the hardware. For this reason, the work focuses on the use of a single camera system placed at

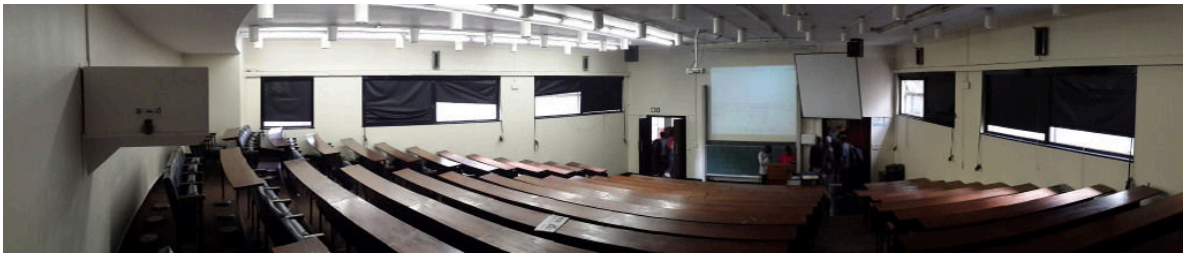


Figure 5.1: Standard Lecture Venue

the front of the class that, if replicated in parallel, could be expanded to properly monitor the entire venue.

A system level schematic of the system is provided in Figure 5.2. The various sensor modules connect through to the main WITS server, which then pre-processes the data, and then performs feature extraction and classification on a per student basis. Once a student is given an *Interest Value* by the classifier, that value is passed to the data presentation system that prepares the data for display. This may be recorded along with video for later review. It may be rendered live as an Interest Map (Chapter 8) or passed to a secondary classification system – termed an Automatic Teaching Assistant (AutoTA) – that can provide the lecturer with hints about how best to re-engage the audience based on previous data.

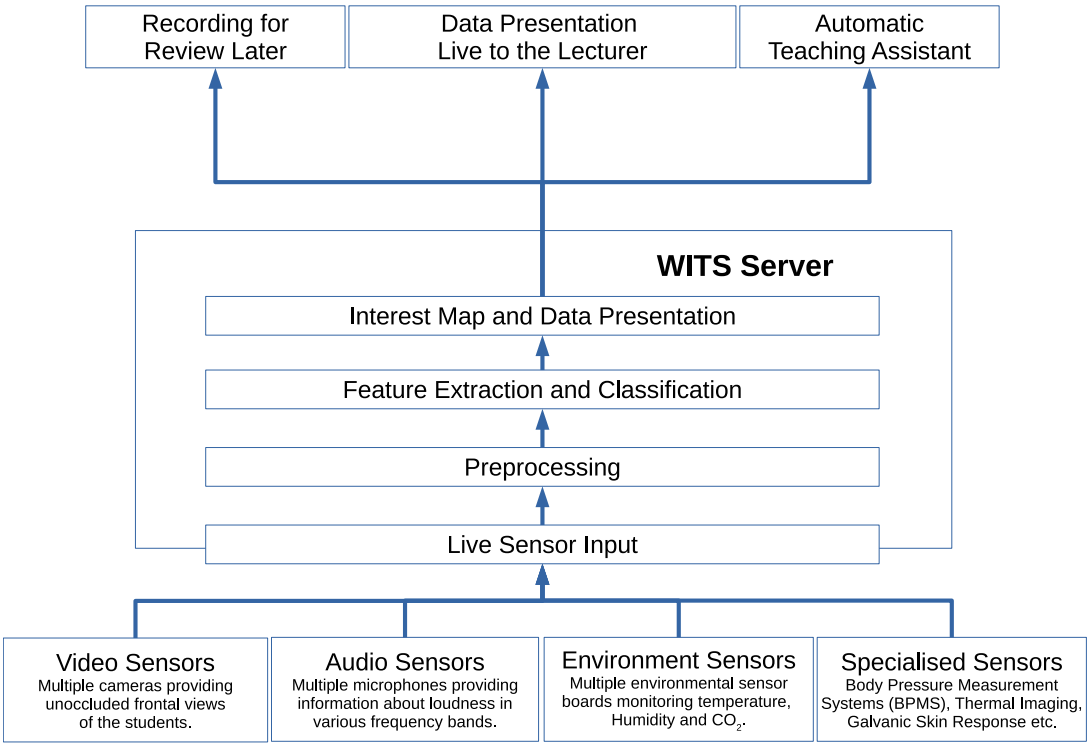


Figure 5.2: System Level View of WITS

5.3 Data Collection

For the development of a supervised learning based computer vision system, a large database of labelled video sequences is required. It was originally hoped that if accepted by students Engage could be used to construct a labelled dataset where the self-reports and the student's position in the lecture venue would be saved alongside a video of the class. This would have allowed the relatively easy construction of a self-labelled dataset. Unfortunately, as the students did not take to the system en masse, this was not an option.

Instead, a number of first year Computer Science lectures were recorded and then manually labelled by untrained assessors. Cross-platform, opensource software was developed to support the data labelling process and is called the WITS ANNOTATION TOOL (WITSAT). WITSAT is presented in Sections 5.3.1 to 5.3.4. The full database of labelled video frames is called the WITS DATABASE (WITSDB) and is described in Section 5.3.5.

5.3.1 WITS Annotation Tool (WITSAT)

To support the creation of a labelled video database, a cross-platform video data labelling program called the WITS ANNOTATION TOOL (WITSAT) was developed. A screenshot is shown in Figure 5.3 on the next page. Although the software was developed for labelling students in this case, it can be used for other object labelling purposes with arbitrary videos. Programmed in C++11, it is based on the Qt (The Qt Company 2016) and OpenCV (Bradski 2000) frameworks and allows the user to label multiple aspects of a captured video frame-by-frame. WITSAT stores both spatial and affect/action labels on a per frame basis which can then be saved to either a text or binary JSON file, or exported to an SQLite database. It can also store some meta-data such as a subject and labeller/rater name.

Spatial labels include the position of a subject's face as well as a rectangular bounding box around the entire subject. These are set by dragging the circle/rectangle markers over the video. The affect/action labels are stored as integer values between -1 and 1 (3 options) or -2 and 2 (5 options) which are adjusted using sliders to the right of the video. In general 0 is treated as unlabelled, which allows one to exclude frames that should be skipped or were unclear to the labeller.

As the user steps through the video, the UI updates with the correct spatial and affect/action labels for the current frame. The user can override any values by repositioning the spatial markers or moving the labelling sliders. There are *record* buttons next to each slider which automatically overwrite the current labels with the previous frame's – this allows the user to play the video forward at normal or fast speeds and only update the labels when there are changes. The software allows for spatial zooming and panning, as well as playback modes at frame rates between 1 and 40 frames per second.

The user can plot the current labels (both spatial and/or affect) underneath the video. For example, in Figure 5.3 the red graph shows where the subject was occluded by the lecturer, and the green graph shows when the subject was writing.

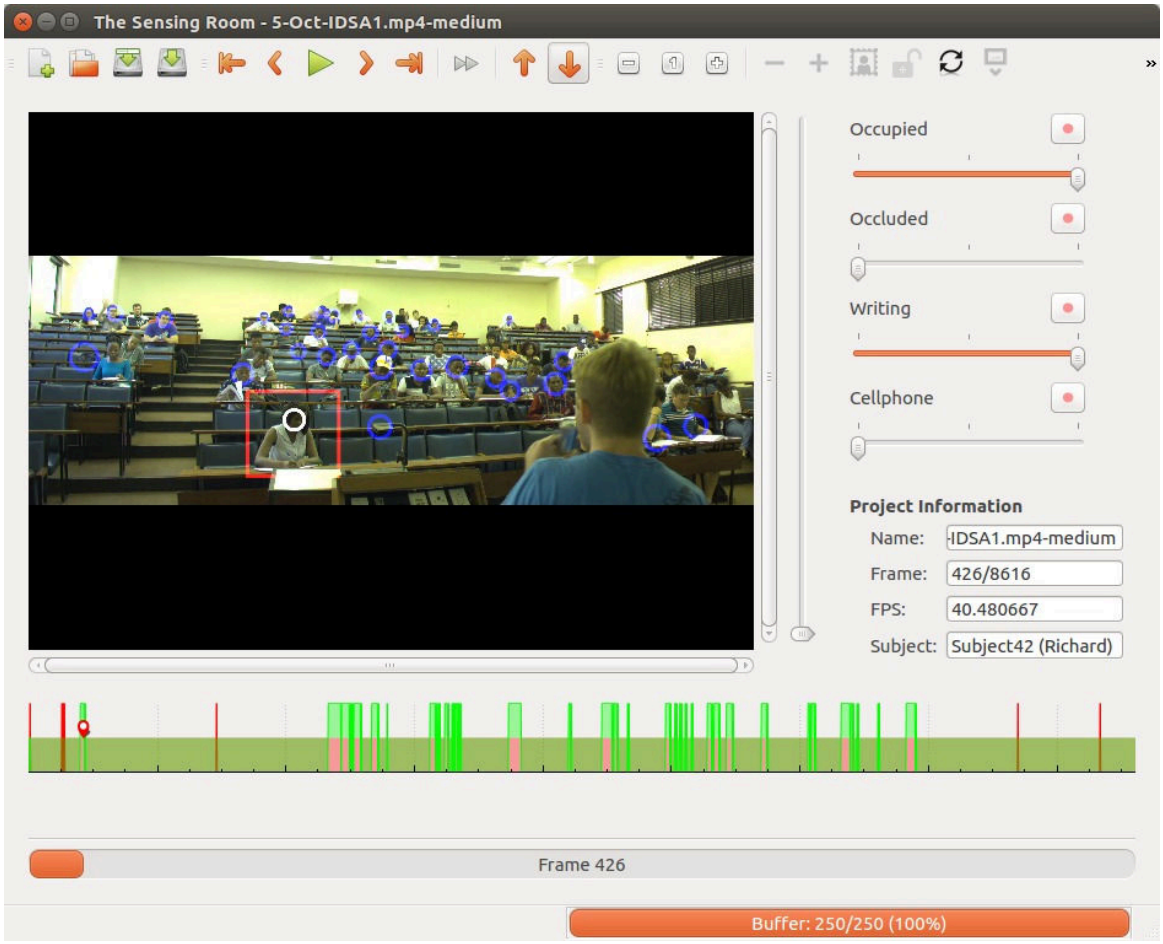


Figure 5.3: Annotation Tool

Robust Semi-Automatic Face Labelling

The first step in the labelling process is to place a face marker, followed by a bounding box. Once these have been placed, the software can automatically track the subject's face within the bounding box. A schematic representation of the algorithm is shown in Figure 5.4. If tracking is turned on, then the software will overwrite the current subject's face label with the detected position inside the bounding box. The user can overwrite the spatial labels if the system makes an error, by dragging the marker to the correct location. WITSAT uses a number of the OpenCV face detectors – based on cascades of Haar-like and LBP features – as well as optical flow. These methods were discussed in Chapter 2.

If the frontal detector returns any faces, the face with a centre closest to that of the previous frame is considered (Figure 5.5a). If the distance between the previous face centre and the new one is below some threshold, θ, then the position on the current frame is updated. If the distance is too large, then the face is rejected and the process is repeated with the left and right face detectors (Figure 5.5b). Finally, if a valid face has not been found, then the previous frame with spatial labels is loaded and interest points within the face area are detected. Using optical flow, the system tracks the interest points to the current frame and uses the mean position with an offset as the new face centre (Figure 5.5c). The threshold can be updated in real time, and

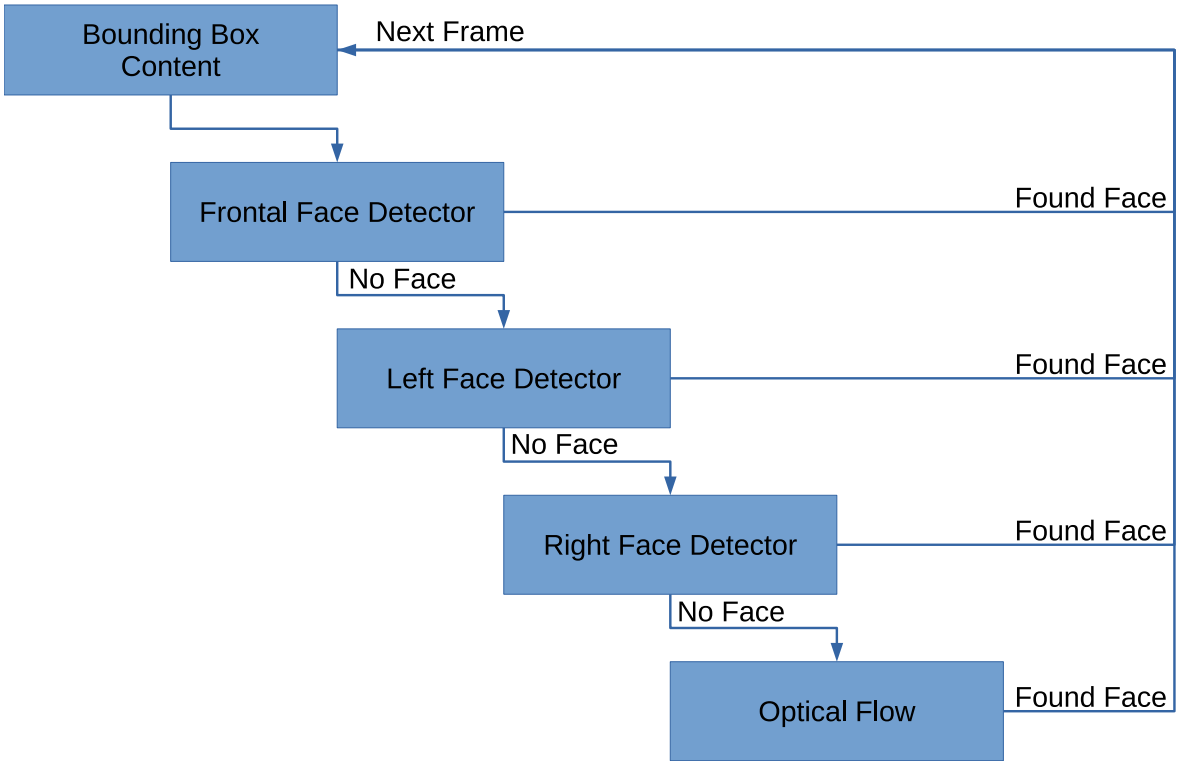


Figure 5.4: Automated Face Labelling



(a) Frontal Face (b) Profile Face (c) Optical Flow

Figure 5.5: Face Detectors

the system uses colour to show whether a face was detected or whether tracking was used. This approach provides robust face labelling that can run at up to 20 frames per second[1] on a single subject and only requires user intervention in the event that the face disappears or is occluded.

5.3.2 Data Export

WITSAT can export labels and image sequences for training. The content of each subject's bounding box is exported as a sequence of png files. Settings such as size and grey scale can be set by the user. The labels are exported to an SQLite file as shown in Figure 5.6. Column names correspond to the number of options on the slider (3 or 5) followed by the label name. This allows the user to easily generate datasets of different labels. For example, the query in Listing 5.1 will generate a list of all unoccluded images with valid labels for Writing.

```
SELECT filename, `3Writing` == 1 as Writing
  FROM data
  WHERE `3Occluded` == -1   /* Must not be occluded */
    AND `3Occupied` ==  1   /* Must be occupied     */
    AND `3Writing`  !=  0;  /* Must be labelled     */
```

Listing 5.1: SQL Statement to Export a Label

The face position labels, $x, y, r \in [0, 1]$ are given as a proportion of the image width, height, and width respectively. So to calculate the pixel position of the face in the image $\tilde{x}, \tilde{y}, \tilde{r}$ where w and h are the image width and height:

$$\tilde{x} = w \cdot x, \tag{5.1}$$
$$\tilde{y} = h \cdot y, \tag{5.2}$$
$$\tilde{r} = w \cdot r. \tag{5.3}$$

5.3.3 Classification

The software interfaces with libSVM (Chang and Lin 2011) and Caffe (Jia *et al.* 2014) to perform live classification. After exporting data and training the SVM (libSVM) or Convolutional Neural Network (Caffe), WITSAT can load the model to perform classification on new image sequences. The predicted labels can be saved into new or old subjects either as new labels or overwriting previous ones.

[1] On an i7 4790K without parallelism or hardware acceleration.

	filename	rater	subject	x	y	r	3Occupied	3Occluded	3Writing	3Cellphone
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Subject1_1617_996/img_0_0001.png	Richa...	Subject1	0.392...	0.242...	0.164...	-1	-1	-1	-1
2	Subject3_1356_746/img_0_0001.png	Richa...	Subject3	0.560...	0.240...	0.216...	1	-1	-1	-1
3	Subject4_352_644/img_0_0001.png	Richa...	Subject4	0.297...	0.248...	0.186...	-1	-1	-1	-1
4	Subject7_1920_682/img_0_0001.png	Richa...	Subject7	0.573...	0.372...	0.136...	1	-1	-1	1
5	Subject8_1664_512/img_0_0001.png	Richa...	Subject8	0.924...	0.829...	0.071...	-1	-1	-1	-1
6	Subject9_1632_467/img_0_0001.png	Richa...	Subject9	0.524...	0.257...	0.180...	1	-1	-1	-1
7	Subject10_1813_469/img_0_0001.png	Richa...	Subject10	0.611...	0.375	0.184...	1	-1	-1	-1
8	Subject11_1474_428/img_0_0001.png	Richa...	Subject11	0.489...	0.319...	0.070...	-1	-1	-1	-1
9	Subject13_2617_712/img_0_0001.png	Richa...	Subject13	0.371...	0.272...	0.196...	1	-1	-1	-1
10	Subject14_2380_486/img_0_0001.png	Richa...	Subject14	0.479...	0.402...	0.170...	1	-1	-1	1
11	Subject16_2075_332/img_0_0001.png	Richa...	Subject16	0.351...	0.300...	0.135...	1	-1	-1	-1
12	Subject17_1644_341/img_0_0001.png	Richa...	Subject17	0.542...	0.328...	0.235...	1	-1	-1	-1

Figure 5.6: Label Export Format as an SQLite Table.

5.3.4 Label Comparisons

The program also allows the user to merge different label files, allowing multiple raters to label a single subject. Figure 5.7 shows how the differences between the labels can be displayed. The user can select two raters in the top left, and the affect/actions in the top right. After clicking the plot button, the first two plots show the graphs for the two raters, while the graph at the bottom shows where they differ. A dialog is displayed that shows the confusion matrices and Cohen's Kappa (Cohen 1960) so that Inter-Rater reliability can be established.

5.3.5 WITS Database (WITSDB)

Recordings

Two cohorts of first year students were recorded in two different years. Students were given two ways to opt out of being recorded - these included sitting in areas not monitored by the camera or by briefly holding up a QR code at the beginning of class that meant they would be automatically blurred from the recordings later on. In the first year, students were recorded using a GoPro camera. Unfortunately due to bad lighting and poor views of the students due to a flat lecture theatre the first set of videos were not usable. However, the recordings were continued to see how the presence of the camera affected the students.

Throughout the six weeks of recordings (one session per week), not a single student ever opted out or indicated that they would prefer not to be recorded. However, for the first 3 weeks exactly, the students were visibly aware of the fact the camera was watching them. Particularly in the first recording, when students became tired or were distracted from the lecture, they looked directly into the camera. By counting the number of times students looked at the camera one is able to get a sense of how aware they are of the recordings taking place. Interestingly, by the fourth encounter with the camera the students stopped caring and stopped looking at it completely. At the same time, students went back to playing games, using their phones, sleeping, etc. Students were possibly worried about getting caught doing these things when the camera was first introduced but as they became comfortable with the fact that the lecturer

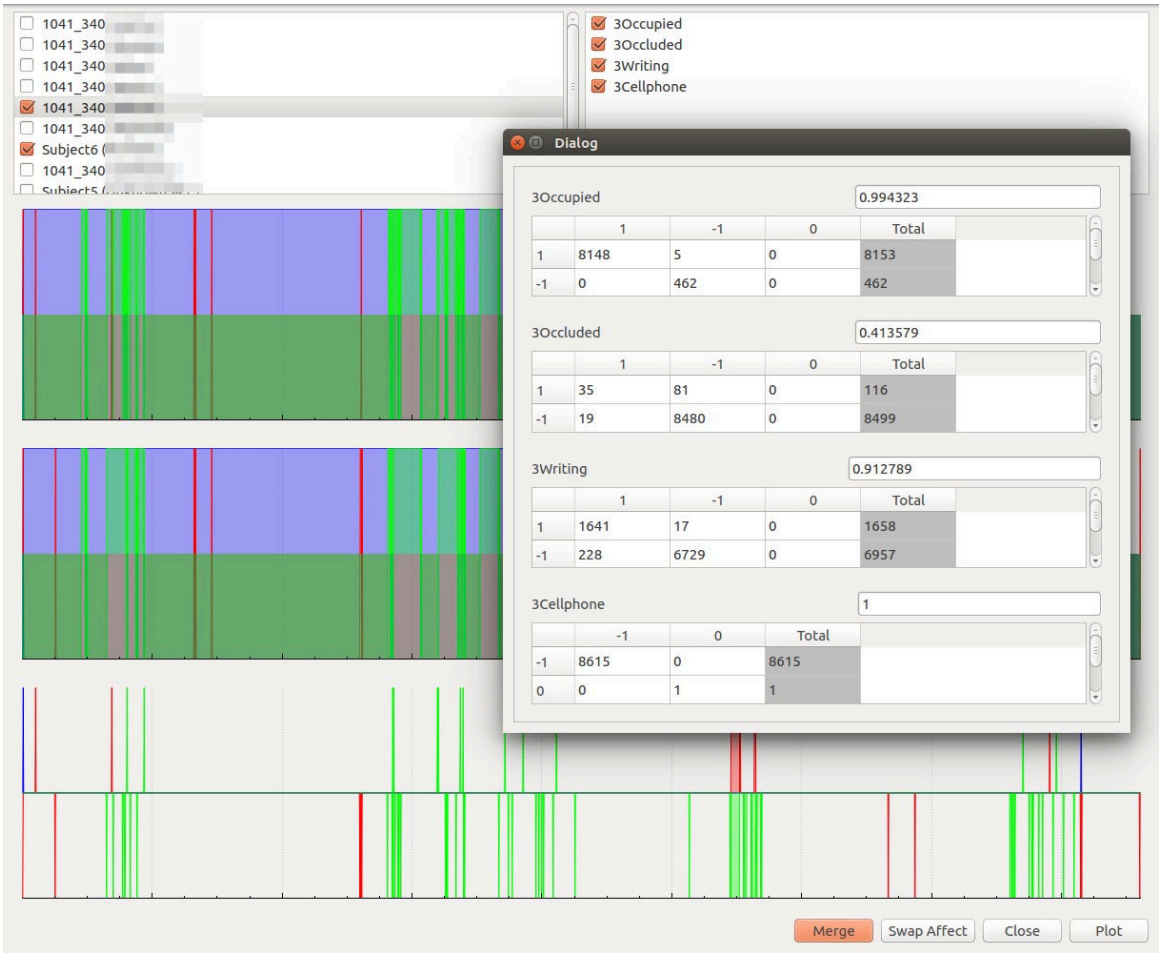


Figure 5.7: Compare Labels

was not reviewing their activity after each session, and they were not getting in trouble, they stopped worrying about it.

The following year, the recordings were performed in a raised lecture theatre where the students were on multiple levels. This class is shown in Figure 5.3. Just like the year before them, not a single student opted out of the recordings, and after the third week of recordings the students were comfortable. This time a Basler acA4600-10uc was placed at the front of the class and captured frames at between 5 and 10 frames per second which were saved as PNG image files. Theses were later merged into 4096x1080 resolution videos. These recordings were higher quality and while there were still issues with lighting and occlusion, these videos were used to build a labelled dataset.

The first three weeks of recordings were excluded so that students natural states in class could be identified. There were then five lectures recorded. The dataset is built of three lectures. Lecture 1 and 2 took place on the same day in consecutive lectures with the same class. Lecture 3 took place in the same class, but on a different day.

Students that were not heavily occluded by those in front of them were labelled spatially using the methods outlined in the previous sections. Each subject in the recordings was labelled by at least three raters. The Cohen's Kappa scores were calculated to identify raters with

poor quality labels. Those with poor labels were discarded. Once the high quality raters were identified, the author manually made decisions about where these raters disagreed. This leaves one set of labels per subject.

Labelling was a time consuming and expensive process, which is why only three of the five lectures were used for the final dataset.

Labels

The labels in the dataset are broken up into four categories: validity, actions, pose, and affect.

There are two labels relating to **validity**:

- Occupied
- Occluded

Labels relating to **actions** include:

- Writing
- Cellphone
- Laptop
- Talking
- Raised Hand
- Yawning
- Head on Desk

Labels relating to **pose** include:

- Horizontal Head Pose
- Vertical Head Pose
- Upper Body Posture

Finally, the **affect** labels are:

- Bored vs Interested
- Distracted vs Attentive
- Tired vs Energetic
- Confused vs Understanding

The labels of validity are illustrated in Figure 5.8, while the action labels are illustrated in Figure 5.9. In both cases there are three possible values: -1 means `false`, 0 means `unlabelled`, and 1 means `true`. When referring to a frame with a false label it is written as ¬Label. The labels relating to pose have five possible values ranging from -2 to 2 where each number corresponds to a specific pose or posture. These are outlined below and shown in Figure 5.11. Finally, the affect labels also take five possible values ranging from -2 to 2 but these values indicate the extent of the affect in question where 0 is neutral.

In general, a frame may be `unlabelled` because the rater was unsure, or because a label did not make sense in that circumstance. For example, when a subject is not present, the rater may have marked Writing as unlabelled. In general unlabelled frames should be excluded from the data for both training and testing.

Validity Labels

An **Occupied** value of true indicates that the subject is present inside the bounding box. This assists in knowing if the subject arrives late to or leaves early from the class. Figure 5.8a



(a) ¬Occupied (b) Occluded (c) Occluded

Figure 5.8: Invalid Frames

shows a frame marked as Not Occupied (or in short ¬Occupied). All frames in Figure 5.9 are Occupied.

An **Occluded** value of false (¬Occluded) means that the subject is visible inside the bounding box. If the subject is marked as Occluded, then the view of the subject is obstructed in some way. This may be because the lecturer stepped between the camera and the subject or another student sits or walks in front. Figures 5.8b and 5.8c show frames marked as Occluded while all frames in Figure 5.9 are ¬Occluded.

When considering the remaining labels, only frames marked as Occupied and ¬Occluded are used. Due to the uncertainty of the occluded region, aspects may be unlabelled (0) if the subject is not visible. These frames are included in the dataset however, so that arbitrary length image sequences can still be extracted to support temporal data processing.

Action Labels

With examples in Figure 5.9 on page 109, the action labels are defined as follows:

- **Writing** – The subject is currently writing. In many cases a subject will write, then stop and look at the lecturer while still holding the pen to paper. If a subject is in this writing position, but the pen is not moving, then the subject is marked as ¬Writing.
- **Cellphone** – The subject's focus is on their cellphone. In some cases the cellphone is not visible as it is either under or flat on the desk, these cases are still marked positive as in Figure 5.10a. In cases where the subject is holding their phone, but are focused on the lecturer, they are labelled with ¬Cellphone.
- **Laptop** – This case is dealt with in the same way as the Cellphone case above. This also includes large tablets or other mobile devices other than phones.
- **Talking** – Students that are communicating with their peers are labelled as talking. This may involve actually talking, or just listening to the peer.
- **RaisedHand** – If a student raises their hand to ask/answer a question then they are

marked with this label. A student touching their face or performing a hand-over-face gesture as seen in Figure 5.10b is marked false (¬RaisedHand).

- **Yawning** – If a student is yawning, with or without covering their mouth they are marked as such.
- **Head on Desk** – If a student puts their head on the desk looking forward (Figure 5.9m) or as if sleeping (Figure 5.9n) they are marked with this label.

Pose and Posture Labels

Illustrated in Figure 5.11, upper body posture was labelled with values between -2 and 2 according to the following categories:

-2. Leaning Left

-1. Leaning Back

0. Sitting Upright

1. Leaning Forward

2. Leaning Right

The head pose was labelled according to horizontal and vertical focus with exemplars in Figure 5.12:

-2. Far Left
-1. Slightly Left
0. Forward
1. Slightly Right
2. Far Right

-2. Below the desk
-1. On the desk
0. Unlabelled
1. Forward
2. Up

Affect Labels

Two attempts were made to label data along affective axes directly. In the first case, 3 students in fourth year Psychology labelled a single lecture. In the second case, around 20 fourth year Computer Science and Applied Mathematics research assistants labelled a single lecture. In both attempts the raters' pairwise κ scores were between 0.03 (extremely poor reliability) and 0.3 (fair).

This result is in contrast to the results of Whitehill *et al.* (2014) that found their raters were able to distinguish between high and low degrees of interest with high reliability ($\kappa = 0.56$ for four-way classification and up to $\kappa = 0.96$ for binary classification). In these experiments, the raters were told to label the students along the axes given and very little guidance was given. Future research may achieve reliable results if the raters are given a detailed checklist.

The initial labelling sprints focused on affect labels, but given the poor inter-rater reliability in both attempts, it was decided to continue to label the students for validity, action, and



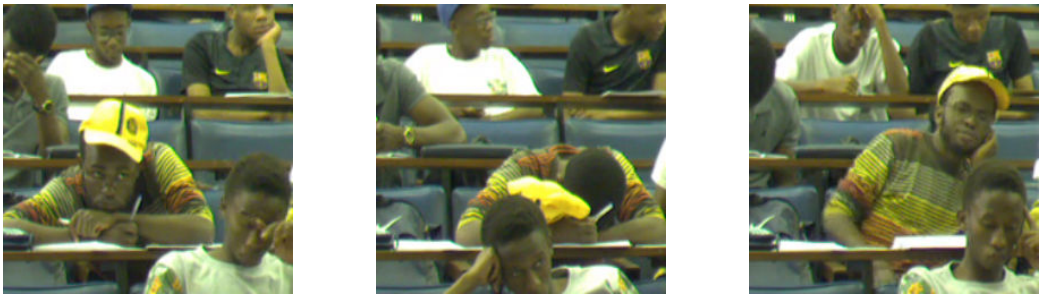
(a) Writing (b) ¬Writing (c) Cellphone (d) ¬Cellphone



(e) Laptop (f) ¬Laptop (g) Talking (h) ¬Talking



(i) RaisedHand (j) ¬RaisedHand (k) Yawning (l) ¬Yawning



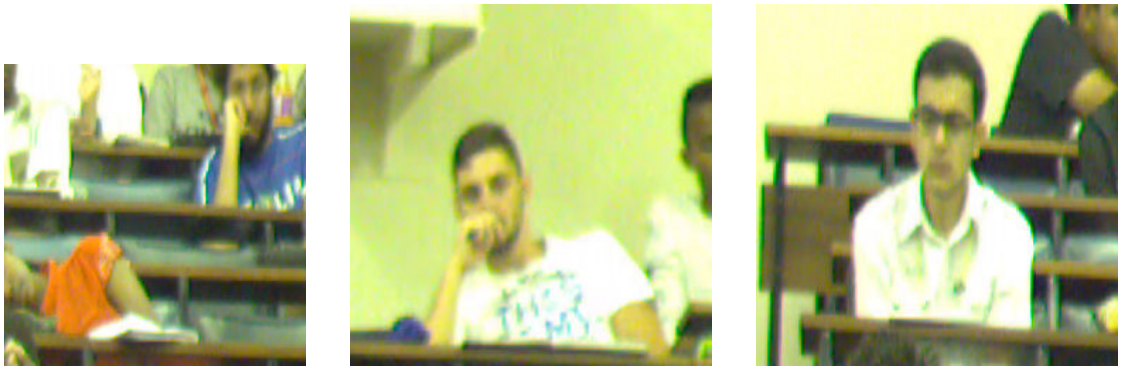
(m) HeadOnDesk (n) HeadOnDesk (o) ¬HeadOnDesk

Figure 5.9: Action Labels



(a) Cellphone (b) ¬RaisedHand

Figure 5.10: Action Labels – Special Cases

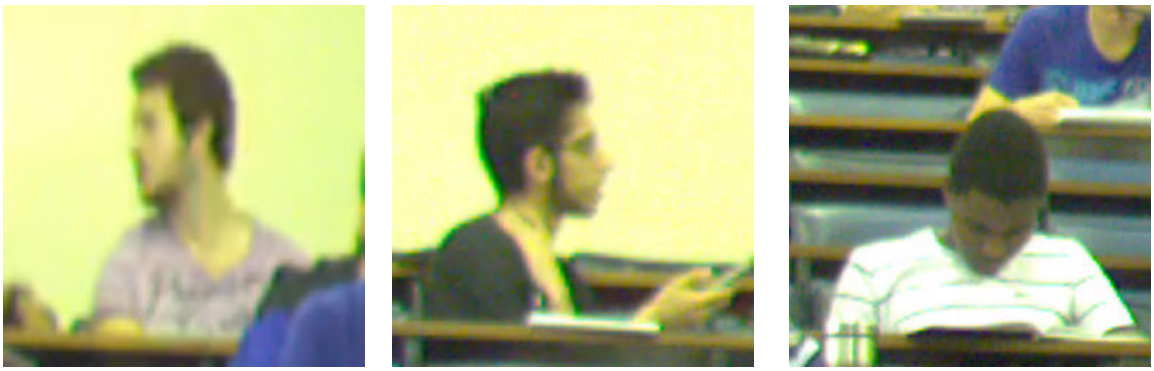


(a) Leaning Left (b) Leaning Back (c) Upright

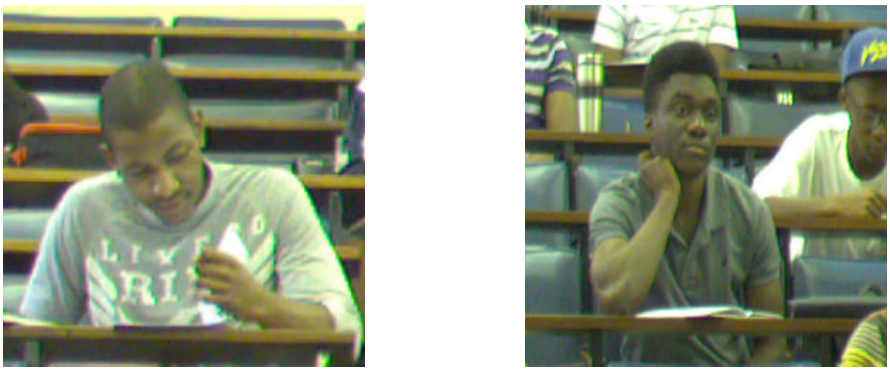


(d) Leaning Forward (e) Leaning Right

Figure 5.11: Subject Postures



(a) Far Left (b) Far Right (c) Under desk



(d) On desk (e) Forward

Figure 5.12: Subject Head Poses

pose rather than affect directly. Later, in Section 8.3, these labels are used in an observational checklist to construct a dataset of interest.

Summary

Table 5.1 shows the number of valid (Occupied and ¬Occluded) frames across all the lectures. Table 5.2 then gives the number of valid frames for each action label. Tables 5.3 and 5.4 give the breakdown of frames by posture and head pose. Note that the number of valid frames are not the same between action labels and postures as the number of students that were labelled differ.

WITSDB presents a number of unique challenges due to the acquisition conditions. There are significant variations in the resolution of the images as students were seated near or far from the camera. There are large differences in angle – both horizontal and vertical – as students were seated around the lecture venue. Variations involving gender, race, posture, clothing, and handedness are present in the dataset. Instances with poor lighting occur, and in many cases, students sitting in front of others partially or entirely occlude each other. Similarly, neighbouring students enter and leave the sides of the frame while other students often appear over a subject's shoulder in the background. Finally, changes in the position of the students' books, stationery and other belongings introduce further complications.

Table 5.1: Number of Valid Frames in WITSDB

	Lecture 1	Lecture 2	Lecture 3	Total
Total Frames	275,712	157,625	44,824	478,161
Occupied	256,512	154,325	44,823	455,660
¬Occluded	244,232	155,459	41,032	440,723
Valid	229,921	153,202	41,031	424,154

Extensions to the dataset should look to balance the representation of gender, race, viewpoint, and pose. Note that there are very few frames for the Laptop, Talking, Raised Hand, Yawning, and Head On Desk labels. There was only one student that used his laptop throughout the recordings. Unfortunately this makes the laptop labels unusable for now. The Raised Hand labels are minimal because students only raise their hand for brief periods during the class. Similarly, there are only about 5 students in this category.

The Yawning, Head On Desk, and Talking categories are also relatively small and with fewer students represented. In Lecture 2, the majority of the positive Head on Desk labels come from the same person. The only categories that were well represented across all students and all lectures were the Writing and Cellphone categories.

It should be a priority of future work to capture enough data for the rarer action and posture labels such as laptop use, raised hands and yawning.

5.4 Computational Pipeline for Training and Testing

The real-time pipeline presented in Figure 5.2 assumes that a classification system is available to monitor students. In order to create such a system a classifier needs to be trained and tested based on the collected data.

A pipeline for training and testing a computer vision based classification system is shown in Figure 5.13. In the training stage, labelled RGB frames from the videos are utilised to extract features upon which to train a classifier.

WITSAT extracts the bounding box content from each subject in a frame and saves it to disk as a 64×64 image. The files are labelled sequentially and each subject is saved to its own folder. The labels and filenames are written to an SQLite database. A file list of valid (Occupied and ¬Occluded) positive and negative samples is exported by using an SQL query based on that from Section 5.3.2. For example, Listing 5.2 shows a script that extracts a list of filenames and labels for Writing and saves them to `files.csv`.

For training purposes, the data is partitioned based on either the cross-subject validation (hold-one-out) or cross frame validation (n-fold cross validation) strategies. If desired, temporal information is included using either data or feature fusion. Features are extracted using the relevant method and if necessary, dimensionality reduction is performed before the data is

Table 5.2: Number of Valid Frames by Action Labels in WITSDB

	Lecture 1	Lecture 2	Lecture 3	Total
Valid Frames	229,921	153,202	41,031	424,154
Writing	26,384	25,022	829	52,235
¬Writing	203,511	128,180	40,202	371,893
Cellphone	42,615	12,711	6,941	62,267
¬Cellphone	187,305	140,491	140,491	361,886
Laptop	-	0	3,788	3,788
¬Laptop	-	153,202	37,243	190,445
Talking	-	5,421	2,169	7,590
¬Talking	-	147,781	38,862	186,643
RaisedHand	-	118	3	121
¬RaisedHand	-	153,084	41,028	194,112
Yawning	2,383	3,356	-	5,739
¬Yawning	315,743	330,607	-	646,350
HeadOnDesk	3,275	25,165	-	28,440
¬HeadOnDesk	298,159	305,741	-	603,900

Table 5.3: Number of Valid Frames by Posture in WITSDB

		Lecture 1	Lecture 2	Lecture 3	Total
	Valid Frames	414,515	359,847	212,430	986,792
-2	Leaning Left	20,087	19,271	13,845	53,203
-1	Leaning Back	64,949	61,971	43,701	170,621
0	Upright	239,666	161,031	68,630	469,327
1	Leaning Forwards	73,595	92,463	64,715	230,773
2	Leaning Right	16,218	25,111	21,539	62,868

Table 5.4: Number of Valid Frames by Head Pose in WITSDB

		Lecture 1	Lecture 2	Lecture 3	Total
Valid Frames		414,515	359,847	212,430	986,792
Horizontal Head Pose					
-2	Looking Far Left	1,457	923	566	2,946
-1	Looking Left	11,467	15,604	10,005	37,076
0	Looking Forward	374,002	294,912	175,125	844,039
1	Looking Right	25,554	46,537	25,752	97,843
2	Leaning Far Right	2,035	1,871	982	4,888
Vertical Head Pose					
-2	Looking Below Desk	36,232	20,870	15,500	72,602
-1	Looking On Desk	97,780	63,920	52,839	214,539
1	Looking Forward	190,777	164,394	117,058	472,229
2	Looking Up	369	5,520	2,681	8,570

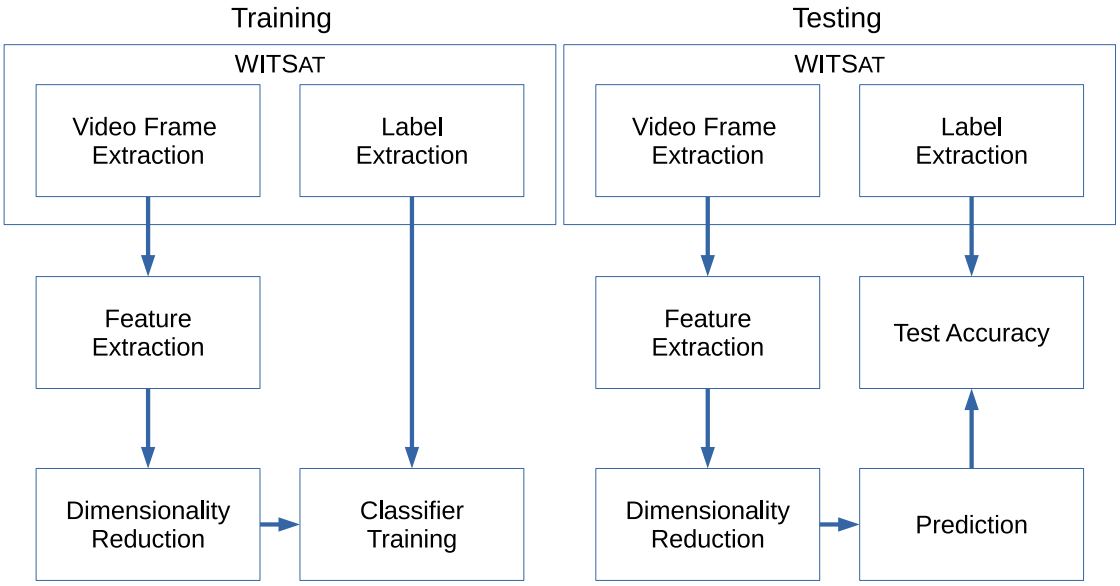


Figure 5.13: Training and Testing Methodology

presented to the classifier for training. Once the classifier is trained, the testing data is presented to the classifier and accuracy is calculated. These steps are repeated according to the validation strategy.

Note that for the CNN based classifier the feature extraction and dimensionality reduction steps are part of the CNN itself and are not performed as separate steps as they are in the HOG SVM method.

```
/* Extract images where subject is present, and not occluded
Extract 1 ⇒ Writing and 0 ⇒ ¬Writing */
.mode csv
.output files.csv
SELECT filename,`3Writing` = 1 as Writing
  FROM data
  WHERE `3Occupied` ==  1
    AND `3Occluded` == -1
    AND `3Writing`  !=  0;
.output stdout
```

Listing 5.2: Extract File List from SQLite

Chapters 6 and 7 present the results of training and testing a machine learning system based on Support Vector Machines and Convolutional Neural Networks respectively. Both are trained on the same sets of data, as presented above. The various normalisation, feature extraction, dimensionality reduction, and different neural network architectures are considered in those chapters.

5.5 Conclusion

While the previous chapter presented an intrusive approach to measuring interest, this chapter proposes a non-intrusive approach based on computer vision and machine learning. An architecture for a real-time system is proposed that uses sensor data from video, audio, environmental, and other specialised sensors as input to a live machine learning based system that is trained to recognise interest.

To facilitate the development of such a system a labelled dataset is required. The WITS ANNOTATION TOOL (WITSAT) was developed using C++11, Qt, and OpenCV. WITSAT allows users to label subjects in a video. Firstly, students are labelled spatially. A subject is given a bounding box and the face is identified. WITSAT uses the Viola-Jones Cascade Classifier based on Haar-like or LBP features (Sections 2.2.4, 2.2.5 and 2.3.5) along with optical flow (Section 2.2.2) to robustly detect and track faces. Following spatial labelling WITSAT allows the user to apply labels using a slider. The bounding box images and corresponding labels can then be exported to image files and an SQLite database.

This process was followed to develop the WITS DATABASE (WITSDB). Recordings of students from three lectures were used to build the dataset. While the students were concious of the camera at first, they were not negatively affected by the presence of the camera after three exposures. The students in the recordings were first labelled by untrained assessors according to affect. These labels included Boredom, Confusion, Frustration, and Arousal. In contrast with Whitehill *et al.* (2014), the labels showed poor reliability with Cohen's Kappa results well below their 0.56.

It was then decided that an approach based on observational checklists should be employed. The assessors were then asked to label the video according to gestures and postures that could

be used as proxies for interest. Writing, cellphone use, laptop use, talking, raised hand, yawning, and head-on-desk actions were identified for use in an observational checklist. Vertical and horizontal head pose, as well as upper body posture were also labelled.

In the two chapters that follow, a Support Vector Machine (Chapter 6) and a Convolutional Neural Network (Chapter 7) are trained and tested on the data extracted from WITSDB. In Chapter 8 these proxies are combined to create a classifier that recognises and reports interest.

Chapter 6

Non-Intrusive Approach using Engineered Visual Features

6.1 Introduction

In Chapter 2, kernel methods and the Support Vector Machine (SVM) were introduced. This chapter uses SVMs along with Histograms of Oriented Gradient (HOG) features to recognise the classroom actions and postures that were identified and labelled in the previous chapter.

In each case the system is trained as a binary classifier that indicates whether the specific label is present in the input image or not. Each label is considered separately in the sections that follow and both cross frame and cross-subject validation are performed. Invalid (¬Occupied or Occluded) frames are excluded from both testing and training when dealing with the classroom actions and postures.

The next section considers whether the validity of frames can be automatically classified. Following that, Section 6.3 considers the classification of the action labels. Section 6.4 considers the classification of the posture and head pose labels. Finally, Section 6.5 concludes the chapter.

6.2 Validity

To recognise if frames are valid, a dataset was extracted using the code shown in Listing 6.3. This yielded 473,786 images of which 424,154 (89.5%) were valid and 49,632 (10.5%) were invalid.

Table 6.1: HOG SVM Validity Results: 5-Fold Cross Validation on 10,000 Frames

Kernel Type	1 Frame			2 Frame (FF)			2 Frame (DF)			4 Frame (FF)		
PCA:	×	0.8	0.9	×	0.8	0.9	×	0.8	0.9	×	0.8	0.9
Features (l):	1,764	160	336	3,528	185	432	15,876	190	427	7,056	214	546
Linear	97	95	96	97	96	96	97	95	96	98	97	96
Polynomial	89	90	89	89	90	90	90	90	90	90	90	90
Radial Basis	89	90	90	90	95	90	90	97	95	90	96	90
Sigmoid	89	90	90	90	93	90	90	94	92	90	95	90

```
.mode csv
.output valid.csv
SELECT filename, (`3Occupied` = 1 AND `3Occluded` = -1) as Valid
   FROM data
  WHERE `3Occupied` != 0 AND `3Occluded` != 0;
.output stdout
```

Listing 6.3: SQL Statement to Export Validity Data

A random sample of 10,000 images were drawn which had the same proportions in each class as the original data. HOG features were extracted for image sequences of length 1, 2, and 4 frames. To merge multiple frames, both feature fusion (FF) and data fusion (DF) were used. 5-fold validation across frames was performed. Linear, polynomial, radial basis, and sigmoid kernel functions were used. The results are presented in Table 6.1.

In all experiments the linear SVM outperformed the other kernels both in terms of training speed, testing speed, and accuracy. The second best performing kernel was the radial basis function. The extremely high accuracies of these tests indicate the ease with which this label can be detected. This result makes sense as the colour gradients and edges of the chairs are relatively simple in comparison to the many varying gradients that are seen when a subject is seated – this accounts for being able to easily recognise unoccupied seats. Similarly, with occlusions the image gradients differ drastically from those that are unoccluded.

Including more temporal frames in the data increased the accuracy in all cases, but also substantially increases the number of features. More features take longer to compute and increase the training and testing time. Of interest is the number of extra features calculated when performing data fusion, but without any accuracy gains over the feature fusion. The feature space increases as there are now features that correspond to blocks that cover the seams where the frames are joined. Comparable accuracy with the feature fusion approach indicates that the SVM is not finding new information in these areas. The data fusion approach yields a five-fold increase in the number of features compared to the feature fusion approach without a related increase in the size.

Performing PCA makes a large difference to the number of features needed. Given that the linear SVM performs well on the original data, it is unsurprising that the linear transformations from PCA are able to successfully capture a significant amount of variation in relatively few parameters. PCA was performed on a random subset of 3,000 image sequences drawn from

Table 6.2: Validity Results: 5-Fold Cross Validation on 50,000 Frames

Frames	Features (l)	Kernel	
		Linear	Radial Basis
1 Frames	1002	95	91
2 Frames	1521	98	94
4 Frames	1959	98	95
8 Frames	2263	98	96

the 10,000 sequences. Principal components were then included to retain 80% or 90% of the variance in the data. All 10,000 HOG feature vectors were projected onto the reduced dimensional space and used for the cross validation tests. While calculating the PCA basis itself took up to 7 minutes due to the numerical inversion of the 3000×3000 matrix, it only needs to be done once. Training the SVM on feature vectors of length 214 takes approximately 8 seconds, while training it on 7,056 took about 1 minute. On a reduced size dataset these numbers are small, but when training on the full set, the computation time and required memory increase substantially. The training time is $O(n^2 l)$, where n is the number of samples and l is the length of the feature vector (Chapelle 2007).

With sequence lengths of four frames and making use of data fusion, the 10,000 feature vectors take up 6.2GB of storage. An increase in data size makes training such sets infeasible. By using PCA, this problem is overcome and the larger datasets can still be considered.

One further experiment was run using a balanced training set of 50,000 samples and PCA was used to retain at least to 99% of the variance. Based on the previous experiments only linear and radial basis function SVMs were trained. The data was presented using feature fusion and sequence lengths of 1, 2, 4, and 8. The results are presented in Table 6.2.

Again the linear SVM outperforms the radial basis function kernel. It appears that the RBF kernel's accuracy increases with more frames, although the linear SVM still outperforms it in all cases.

The increase in the number of PCA bases has caused a decrease in accuracy for both the linear and RBF SVMs in the single frame case. When using longer sequences, the accuracy reaches a maximum of 98%.

Validation across subjects was not performed for these frames as they are not explicitly part of the interest definition presented later in Section 8.3.

6.3 Classroom Actions

6.3.1 Writing

This section focuses on experiments to recognise writing actions. A dataset was extracted using the code shown in Listing 6.4. This yielded 424,128 images of which 371,893 (87.7%) were ¬Writing and 52,235 (12.3%) were Writing.

```
.mode csv
.output writing.csv
SELECT filename, `3Writing` == 1 as Writing
   FROM data
  WHERE `3Occupied` ==  1
    AND `3Occluded` == -1
    AND `3Writing`  !=  0;
.output stdout
```

Listing 6.4: SQL Statement to Export Writing Data

Preliminary tests were run by extracting balanced datasets of 20,000, 40,000, and 80,000 images. These sets each contained positive and negative images in equal proportions. Using data fusion and principal component analysis retaining various levels of variance, a number of experiments were run. The results of 2-fold validation across frames are shown in Table 6.5 on page 122.

These results indicate that the information about writing is encoded temporally and more than one frame is required. Both the number of features and the size of the data are also provided. It indicates a trade-off between the size when using a single frame, and the size when using multiple frames. When using multiple frames with data fusion, the data size grows drastically and again PCA is required.

These results indicate that the larger datasets provide better accuracy. This is a reasonable result as the larger datasets are more likely to contain images that are similar. Also of interest is that in all cases, PCA provided an increase in accuracy. This is indicative of the curse of dimensionality and shows how high dimensional data can make learning more difficult.

These results inform the experiments that follow. The next tests aimed to see how the number of frames and the variance retained affect the results. Tables 6.3 and 6.4 show the results of the different kernels for image sequences of length 1, 2, 4, and 8. Feature fusion is used and PCA retains 90% and 99% of the data's variance. The datasets were stratified and the expected accuracy due to chance is 78%.

Interestingly, the accuracy of the linear kernel with 8 frames and 664 features, is equivalent to that of the single frame with 1,047 features. A further run was performed with PCA retaining 90% variance when using 16 frames, this meant using 1,328 features – roughly similar to the number of features for a single frame in 99% PCA. The 5-fold cross validation yielded an accuracy of 92% (linear kernel) and 91% (RBF kernel). This suggests that for writing, when talking about a similar number of features, it is better to take them from multiple frames rather than a single one – particularly when making use of the RBF kernel.

Due to the similarity of image sequences when looking at a single subject in a single lecture, performing cross validation across the subjects provides more insight into how well the system generalises to new subjects.

The images were separated by subject and a dataset was sampled where a subject had the same number of positive and negative images based on the number available for each subject. A subject with 500 positive images and 5000 negative images would contribute 500 images

Table 6.3: Writing Results: 5-fold Validation on Stratified Datasets, PCA Retains 90% Variance.

Frames	Features (l)	Kernel	
		Linear	Radial Basis
1	333	90	88
2	425	91	88
4	535	91	88
8	664	92	90

Table 6.4: Writing Results: 5-fold Validation on Stratified Datasets, PCA Retains 99% Variance.

Frames	Features (l)	Kernel	
		Linear	Radial Basis
1	1,047	92	88
2	1,514	92	88
4	1,915	93	87
8	2,256	92	87

Table 6.5: Cross Validation Results Over Frames – Writing

	2-Fold Validation Accuracy																
Frames:	1 Frame									4 Frames							
PCA Variance:	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99	No PCA	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99	No PCA
Images	Accuracy (%)																	
20 000	71	75	77	77	76	76	75	72	69	89	92	94	94	95	94	92	86	70
40 000	73	78	79	80	78	77	77	75	72	92	95	96	96	96	96	95	90	-
80 000	75	79	81	82	81	80	79	78	77	93	95	97	97	97	97	96	93	-
	Features per Image																	
	8	15	26	45	75	131	254	675	1 764	8	16	29	51	89	165	388	831	44 100
	Data Size (MB)																	
20 000	2	4	6	11	19	34	69	192	482	2	4	7	12	22	42	89	226	13 GB
40 000	4	7	13	22	38	69	138	383	965	3	7	13	24	43	84	178	452	26 GB
80 000	7	14	25	43	73	131	264	730	1.9 GB	7	14	26	47	85	165	352	903	52 GB

from each set, resulting in 1000 images. This was done for all 65 different subjects resulting in a dataset with an equal number of positive and negative samples. There were 21 subjects with more than 1000 positive frames. These were used for hold-one-out validation.

A subject was selected from the set of 21, the system was trained on the remaining 64 subjects and tested on the excluded one. A number of different pre-processing steps were considered to assist learning. The parameters of these approaches along with their results are presented in Table 6.6. The various parameters include:

- the **number of training images** randomly sampled from the remaining subjects (10,000 or 80,000),
- **mirroring** the image with probability 0.5,
- randomly **cropping** up to 5% off the width and height of the images (jitter),
- treating previous images in the sequence as difference images, where I_n is the nth, concatenating frames would yield: $[I_n; I_n - I_{n-1}; I_{n-1} - I_{n-2}; I_{n-2} - I_{n-3}]$),
- Principal Component Analysis performed over the same set of frames with the above methods applied.

80,000 training images were randomly sampled from the remaining subjects for training. Table 6.6 shows the results. In most cases the prediction accuracy was in the low 60%'s, and as the stets were balanced, an accuracy of 50% is expected by chance alone. Just like the cross validation over frames above, learning with PCA out performed learning without it in terms of speed and almost always in terms of accuracy. Training on a dataset of 80,000 images without PCA took up to 6 hours meaning that 5-fold cross validation took over a day. The same training with PCA took about 35 minutes and 5-fold cross validation took only 3 hours per run. In the experiments run, it appears that applying jitter to the image assists in generalisation. Performing the tests with Feature Fusion yields similar results.

As seen in Figure 6.1 and Table 6.6, the performance of the different configurations is significantly worse when validating across the subjects rather than across the frames. As recordings were taken during lectures, the students were mostly still. Thus, there is not a lot of movement between frames which makes consecutive frames similar. This means that when randomly selecting training and testing sets from the frames, the accuracy rate is inflated due to these similarities. In order to perform well, the system only needs to recognise similar frames.

When validating across subjects, the differences between the training and testing images is much greater. In order to perform well here, the system must generalise to subjects that it has never seen before. This task is more difficult and is expressed through the decreased accuracy.

Of interest is that regardless of the parameters of the test, the performance on a single specific subject was relatively consistent. A subject that performed poorly in the first test, was likely to perform poorly in all the tests. For example, Subject 34's accuracy in the first experiment was 45%, indicating that the system was unable to generalise to this subject and performed worse than chance. Regardless of the parameters in the other experiments the accuracy for that subject remained below 50%. Looking at the raw data, the subject is found at a wide angle and is often partially occluded by his neighbour. This is illustrated in Figure 6.2, which contrasts Subject 51 (Figure 6.2a) with a good view and Subject 34 (Figure 6.2b) with a poor view.

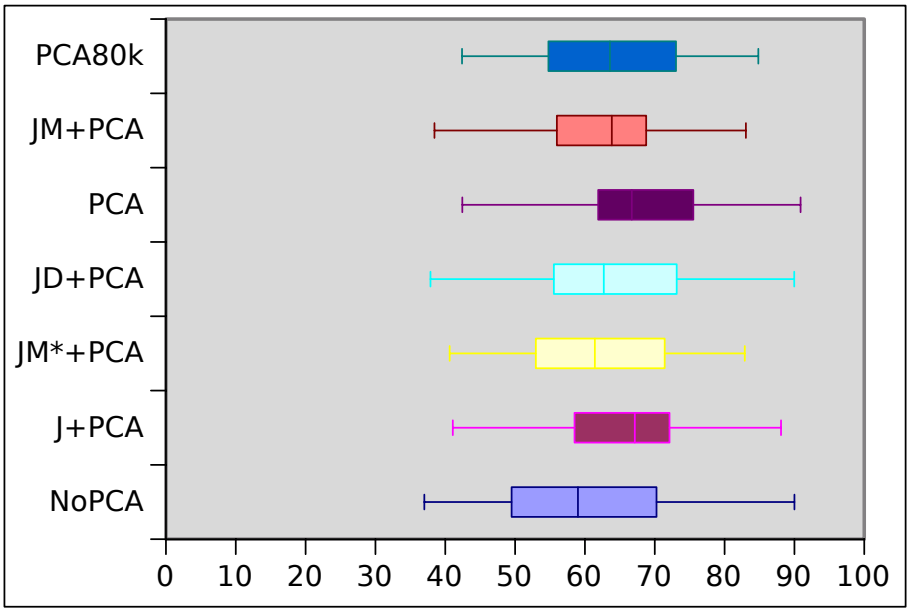


Figure 6.1: Cross Validation Results Over Subjects – Writing

It may be necessary to handle wide angles or cases of partial occlusions separately in another more robust classifier. The better option is probably to use multiple cameras that are installed throughout the room that can provide frontal views of each student. Elevating the hight of the cameras along with these more frontal views can assist avoiding the partial occlusions that are interfering with good results.



(a) Subject 51 Frontal View With No Occlusions and Minimal Interference From Other Students.



(b) Subject 34 view from a wide angle and partial occlusion.

Figure 6.2: Good (a) and Bad (b) Subject Views.

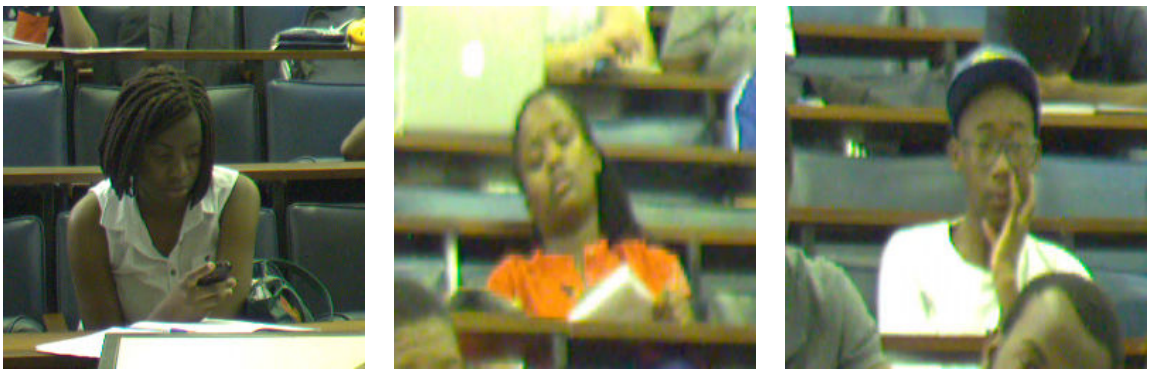
Table 6.6: Cross Validation Results Over Subjects – Writing

		NoPCA	J+PCA	JM*+PCA	PCA	JD+PCA	JM+PCA	PCA80k
Training Images		80,000	10,000	10,000	10,000	10,000	10,000	80,000
Mirror		-	-	0.5*	-	-	0.5	-
Jitter		-	0.05	0.05	0.05	-	0.05	-
Previous as Diff		-	-	-	Y	-	-	-
PCA Retained Variance		-	0.9	0.9	0.9	0.9	0.9	0.9
PCA Items		-	3000	3000	3000	3000	3000	3000
Subject	Testing Images	Accuracy (%)						
18	4816	68	71	63	73	68	64	69
43	4724	78	85	83	78	87	83	85
20	4690	60	68	67	60	65	69	81
15	4392	71	81	80	80	76	77	82
53	3930	69	76	71	74	76	69	58
39	3828	56	72	61	60	76	64	78
09	3506	45	47	47	48	47	47	49
54	3446	47	67	72	52	67	69	71
00	3374	76	67	75	73	76	74	69
34	3300	41	42	52	46	45	50	44
44	3174	72	72	61	73	67	62	64
38	3088	62	72	65	72	74	66	72
13	3068	58	66	68	58	67	68	63
36	3014	70	67	56	69	63	58	63
37	2864	37	46	48	38	42	49	47
51	2852	90	88	75	90	91	82	77
02	2652	54	64	51	59	63	56	55
32	2630	58	66	53	63	70	52	73
19	2566	37	41	59	38	43	63	42
16	2432	50	54	41	56	52	38	50
01	2386	48	59	58	63	62	56	57
Mean		61	65	62	63	65	63	64
Median		59	67	61	63	67	64	64
Min		37	41	41	38	42	38	42
Max		90	88	83	90	91	83	85

* Mirroring applied to training data only.

6.3.2 Cellphone Use

Using the process outlined previously, subjects with at least 1,000 positive frames were selected. This yielded 23 subjects. Cross-subject validation was performed incorporating PCA, jitter and mirroring. The system based on the PCA, HOG and SVM was unable to learn to recognise this label with the mean and median accuracy sitting close to 54% in most configurations. This indicates the classifier accuracy matches that of random chance in most cases. Engineering features that focus on the areas around the hand and desk may produce better results. Subjects where the classifier performed best for cellphone recognition had frontal views without occlusions, generally had clothing that offered high contrast with their skin and background, and the background of the subjects was not cluttered which helps the HOG features capture the subject pose more efficiently - this is illustrated in Figure 6.3. Table 6.7 and Figure 6.4 present the results of validation over the remaining subjects.



(a) Subject 16 (b) Subject 18 (c) Subject 49

Figure 6.3: High Performing Subjects for Cellphone Recognition.

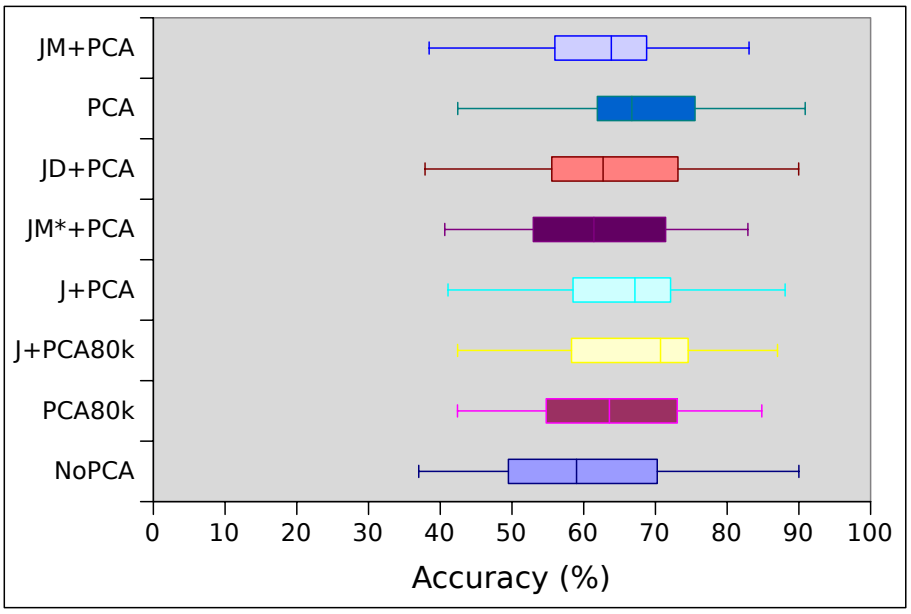


Figure 6.4: Cross Validation Results Over Subjects – Cellphone Use

Table 6.7: Cross Validation Results Over Subjects – Cellphone Use

		J+PCA5k	JM+PCA5k	J+PCA10k	J+PCA10k	J+PCA80k
Training Images		5,000	5,000	10,000	10,000	80,000
Mirror		-	0.5	-	-	-
Jitter		0.05	0.05	0.05	0.05	0.05
Subject	Testing Images	Accuracy (%)				
23	12,494	43	47	44	45	56
29	11,822	63	63	63	62	57
57	7,588	49	50	51	52	58
48	6956	53	60	50	54	65
27	6696	63	58	59	57	52
07	5866	50	50	50	50	50
19	5592	57	68	58	62	61
15	5072	51	62	51	50	50
49	4874	71	68	73	74	74
42	3530	57	59	57	55	52
50	3480	67	62	65	71	66
21	3384	48	46	48	48	51
16	3322	72	57	55	50	50
28	3032	50	50	50	50	50
11	3016	62	50	56	57	54
01	2882	50	49	50	50	50
25	2876	53	55	54	59	53
60	2798	52	50	50	50	50
14	2746	51	46	52	52	60
22	2624	50	47	54	51	54
18	2314	55	56	89	55	53
30	1912	50	41	50	50	50
26	1910	50	50	50	49	49
Mean		55	54	55	54	55
Median		52	50	52	52	53
Min		43	41	44	45	49
Max		72	68	89	74	74

* Mirroring applied to training data only.

6.3.3 Laptop Use

Unfortunately there was only a single subject in the dataset that made use of a laptop in the labelled lectures. Performing 5-fold cross validation over the frames for that subject yields 89% accuracy. Validation across subjects was not possible for this label as there are no others.

6.3.4 Talking

The system was also unable to recognise subjects that were talking. The results are shown in Table 6.8 and Figure 6.5, with most accuracy values close to 54%. These results are comparable with chance indicating that the classifier is unable to learn adequately with the current features.

This is mainly due to the fact that the head occupies a relatively small area within the image. The HOG features are calculated on the entire image, rather than just the head region. The system may perform better in identifying talking action if HOG features localised around the head are used.

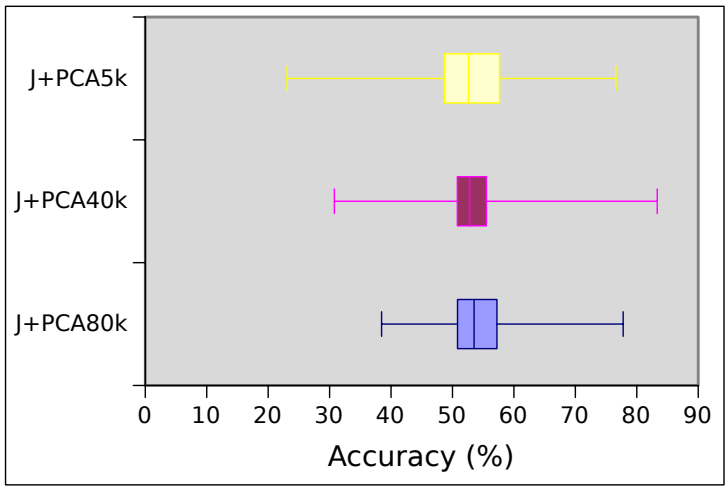


Figure 6.5: Talking Recognition

6.3.5 Raised Hand

There were only 121 frames where the subject was labelled as having a raised hand. This is generally not a large enough number on which to perform machine learning and results should be considered with extreme caution. Results are on par with that of chance, and more data for this label should be collected before making any conclusions. These results are shown in Table 6.9 on page 130.

Table 6.8: Cross Validation Accuracy Over Subjects – Talking

		J+PCA	J+PCA	J+PCA
Training Images		5000	10000	80000
Mirror		-	-	-
Jitter		0.05	0.05	0.05
Subject	Testing Images	Accuracy (%)		
14	1163	55	56	54
30	961	71	73	71
18	952	48	50	48
26	927	58	60	55
08	921	48	51	50
03	437	51	53	57
12	400	49	57	53
20	359	49	51	52
17	259	49	51	50
10	238	59	55	63
21	180	53	56	53
06	153	59	53	56
27	125	53	54	51
15	114	50	52	51
32	90	30	40	39
07	74	55	51	57
05	69	53	54	58
31	66	52	51	51
09	47	77	41	61
16	33	60	55	57
00	13	23	31	38
01	9	44	83	78
Mean		52	54	55
Median		52	53	54
Min		23	31	38
Max		77	83	78

Table 6.9: Cross Validation Accuracy Over Subjects – Raised Hand

		J+PCA	JM+PCA
Training Images		242	242
Mirror		-	0.5
Jitter		0.05	0.05
Subject	Testing Images	Accuracy (%)	
18	90	77	89
08	74	50	51
07	20	50	48
06	12	50	50
13	10	50	50
20	10	60	60
21	8	50	38
10	6	50	50
12	6	100	100
30	6	50	50
Mean		59	59
Median		50	50
Min		50	38
Max		100	100

6.3.6 Yawning

The yawning action suffers from the same problem as the raised hand action above. In total there are 34 subjects that are labelled as ever yawning and each contributes roughly 30 to 100 frames. Cross-subject validation performs no better than chance and more data is needed to train the system to recognise this action.

6.3.7 Head on Desk

The head on desk action is also under-represented in the dataset. Across the entire dataset there are only 16 students that perform the action; all except one contribute fewer than 800 frames. Cross-subject validation performs no better than chance and more data is needed to train the system to recognise this action.

6.4 Posture

A stratified set of 60,000 sequences was built from the valid posture labels in the dataset. Recall that there are five options corresponding to sitting upright or leaning left, back, forwards or right. The set is stratified rather than balanced as the number of samples in the five categories are highly skewed to those sitting upright. 5-fold cross validation over frames yielded an accuracy of 80.8% across the five categories. With the stratified unseen testing set the SVM achieves an accuracy of 81.4% with $\kappa = 0.68$, which indicates significant agreement as accuracy due to chance is approximately 43%.

Cross-subject validation over stratified validation sets yielded 41.6% accuracy with $\kappa = -0.01$ indicating no agreement and the SVM is acting worse than chance which expects approximately 42% accuracy. Equivalent results were found when detecting both vertical and horizontal head pose; indicating that the SVM is completely unable to generalise to postures and head poses of unseen subjects.

6.5 Conclusion

Validation across frames produced accuracies often in the high 90%'s for binary classification and in the 80%'s for five-way classification, but in general the HOG SVM was unable to generalise to unseen subjects. On balanced testing and training sets, it was able to recognise when unseen students were writing with mean and median accuracies of 65% and 67% respectively. When recognising cellphone use of unseen students, the HOG SVM performed poorly overall with mean and median accuracies of 55% and 52% respectively.

The remaining classroom action labels did not have enough exemplars to adequately test. Unsurprisingly, in all cases the system was unable to outperform chance when viewing new subjects which indicates that the system was unable to learn to recognise the gestures based on the current feature set.

The fact that the HOG SVM was able to perform very well on validation across frames but not on validation across subjects indicates that it was not learning the correct patterns in the data. Consecutive frames of the same subject are very similar to each other. The high values for frame validation mean that the SVM is just looking for similar frames, rather than interpreting features within the frame.

Performing feature engineering, focusing on specific areas of the image, explicitly removing the background, and potentially trying a different set of features altogether may improve accuracy and allow the SVM to correctly classify the actions and postures. However, this requires manual intervention without a guarantee of results. The SVM also struggles with issues relating to large datasets as the training time is quadratic in the number of samples presented. This means that it is infeasible, for example, to train on the entire database.

Based on these issues the HOG SVM is not recommended for WITS and in the following chapter, CNNs are used to try automatically learn good feature representations for classification, rather than using the HOG features from this chapter.

Chapter 7

Non-Intrusive Approach using Automated Visual Features

7.1 Introduction

In Chapter 2, Convolutional Neural Networks (CNNs) were introduced as a method whereby a neural network classifier is built upon a number of other convolution, pooling, drop-out, and normalisation layers. The convolution layers learn the weights of a number of linear convolution kernels to produce feature maps that are passed as input into the next layer. Pooling layers summarise neighbouring groups of input neurons. For example, max-pooling takes the maximum value as the summary of some neighbourhood. By using overlapping neighbourhoods, over-fitting in the CNN is often reduced. Normalisation layers can perform global, local, or lateral normalisation while drop-out layers help avoid over-fitting by stochastically deactivating neurons during the training process. These layers are all used in conjunction with the fully connected layers typical of neural networks and training is performed using the usual back propagation algorithm.

AlexNet (Krizhevsky *et al.* 2012) is a CNN architecture that performed exceptionally well on the ImageNet database. AlexNet's architecture is shown in Figure 2.24. The work in this chapter uses AlexNet to perform binary classification on the various labels discussed in the previous sections. The final AlexNet softmax layer of 1,000 neurons is replaced with a two or five neuron softmax layer where each neuron corresponds to one of the categories being learnt. The category corresponding to the neuron with the largest activation is used as the output of the network.

Image sequences are presented to the network as multiple channels of a single image. This makes an image of $3k$ channels, where k is the number of frames in the sequence. So where $\mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i \in \mathbb{R}^{w \times h}$ are the intensities of each channel at frame i, then the input to the network, $\mathbf{I}_i^k \in \mathbb{R}^{3k \times w \times h}$, is

$$\mathbf{I}_i^k = [\mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i, \mathbf{R}_{i-1}, \mathbf{G}_{i-1}, \mathbf{B}_{i-1}, \ldots, \mathbf{R}_{i-(k-1)}, \mathbf{G}_{i-(k-1)}, \mathbf{B}_{i-(k-1)}] \tag{7.1}$$

By concatenating along channels, pixels in the same location are 'above' each other in the image cube and a kernel is able to compare pixels that correspond spatially.

Section 7.2 illustrates the first major benefit of the CNN based approach – speed. Section 7.3 briefly considers the detection of valid and invalid frames. Section 7.4 looks into the detection of the different action labels and Section 7.5 presents the detection of the various postures and head poses. In each case, validation across frames is performed first, followed by validation across subjects, and then across lectures. When performing cross-subject validation, only the frames from Lecture 1 are considered to prevent the accidental inclusion of subjects that were present across multiple lectures. Cross-lecture validation involves training on a held-out lecture and testing on the remaining two for all the combinations.

7.2 Hardware Acceleration

The CNN method lends itself to hardware acceleration. The convolutional and other local region based operations in the CNN are *embarrassingly parallel* and are easily accelerated using General Purpose Graphics Processing Units (GPGPUs). Specifically, the use of NVIDIA's CUDA framework (Nickolls *et al.* 2008) allows the user to offload computation to the GPU and make use of the massively parallel facilities of modern graphics cards. NVIDIA also provide cuDNN (Chetlur *et al.* 2014), a CUDA accelerated library of deep learning primitives.

This work makes use of the Caffe CNN library (Jia *et al.* 2014) that leverages these resources to provide significant performance improvements over the standard CPU approaches. Termed an "irresponsible amount of performance" by the manufacturer, an NVIDIA Titan X with 3,584 CUDA cores, 12GB of on-board GDDR5 RAM, and capable of 11 TFLOPs was used for the training and testing in the rest of this chapter (NVIDIA 2015).

Training a CNN on the CPU can take days or weeks, while training on a GPU takes only a few hours. To briefly illustrate the improvement over the CPU approach, AlexNet was trained on 50,000 image sequences for 5,000 iterations using the CPU and then GPU respectively. The CPU training took 9 hours 41 minutes. In contrast, the GPU training took 4 minutes 20 seconds. This significantly impacts the number of experiments that could be run.

For classification, the CNN was used to classify 50 subjects per frame. The CPU classifier worked at 0.23 frames per second, while the GPU classifier did 7 frames per second. For comparison, the 8-core parallel CPU based HOG SVM runs at 1.6 frames per second. Running the GPU based CNN significantly improves the performance of WITS, making real-time processing a possibility.

7.3 Validity

7.3.1 Cross-Frame Validation

From all three lectures there are 424,154 valid and 49,632 invalid frames. Recall that a frame is valid if the subject is both present and unoccluded. A balanced dataset of 15,000 positive and 15,000 negative image sequences was sampled to build a training set. A balanced test set was sampled that contained 5,000 positive and 5,000 negative image sequences. A stratified

validation set of 100,000 frames was created using 88,532 valid and 11,468 invalid frames of the remaining unused ones.

Table 7.1: CNN Accuracy Detecting Validity

Sequence Length	Test Set Accuracy (%)		
	Peak	10,000 Iterations	20,000 Iterations
1	98.6	98.6	98.6
2	98.7	98.4	98.6
4	98.8	98.6	98.7

Table 7.1 shows the accuracy values for image sequences built from 1, 2, and 4 frames. Figure 7.1 on the next page provides the loss value during each training epoch as well as the accuracy over the test set after every 1,000 training epochs. The results show how the CNN can easily learn to differentiate between frames with unoccluded students and those that are either occluded or empty. The accuracy is not sensitive to the number of input frames and based on the accuracy curves in Figures 7.1a, 7.1c and 7.1e, the network quickly learns to understand the frames regardless of the sequence length.

The network trained on single-frame sequences was then run on the validation set to understand how the classifier would function in unbalanced sets. The classifier achieved an accuracy of 98.9% (chance expects 79.2%) and the confusion matrix is given in Table 7.2. This leads to $\kappa = 0.95$ which is extremely high indicating near-perfect agreement.

Table 7.2: CNN Confusion Matrix for the Validation Set (Validity)

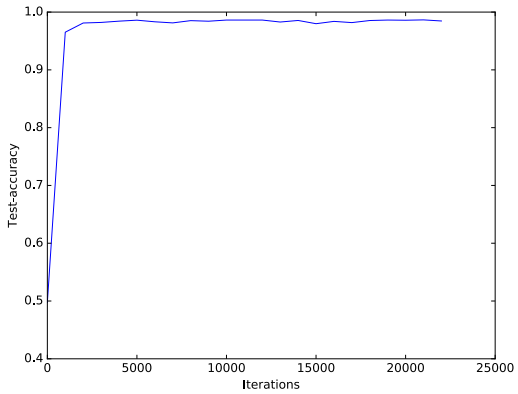
Actual	Predicted	
	Invalid	Valid
Invalid	11,258	210
Valid	871	87,661

These results agree with those in Chapter 6 and indicate that the large spatial changes that occur in the image make it easy to tell whether the current frame is valid or not.

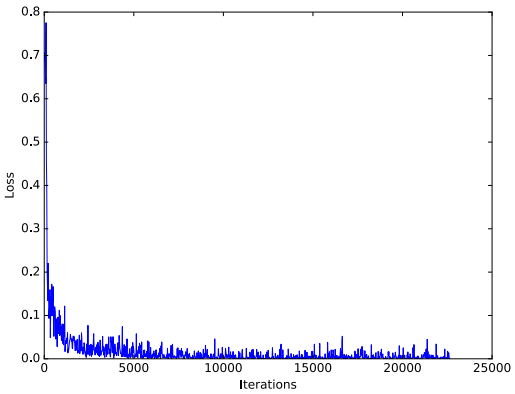
7.3.2 Cross-Subject Validation

There are 50 subjects in lecture 1. Cross-subject validation was performed by constructing a balanced training set of 30,000 valid and 30,000 invalid frames randomly selected from all subjects except for the current holdout subject. An equal number of valid and invalid frames were then randomly selected from the holdout subject for a testing set based on the number available for that subject. A validation set was constructed by using *all* frames from the relevant subject.

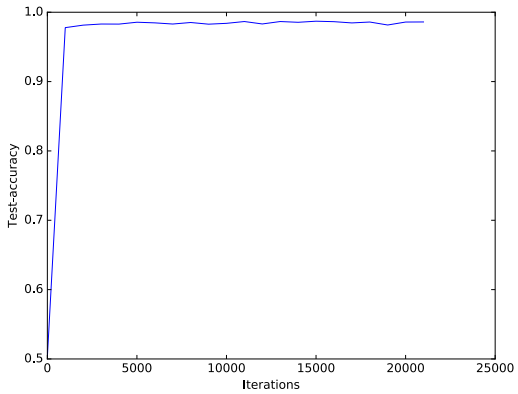
Over the balanced testing sets the minimum and maximum accuracies for different subjects were 53.1% and 99.9%. The mean and median accuracies were 82.4% and 85.7% respectively. Over the validation set, an accuracy of 81.6% (chance gives 54.7%) with $\kappa = 0.6$ was achieved



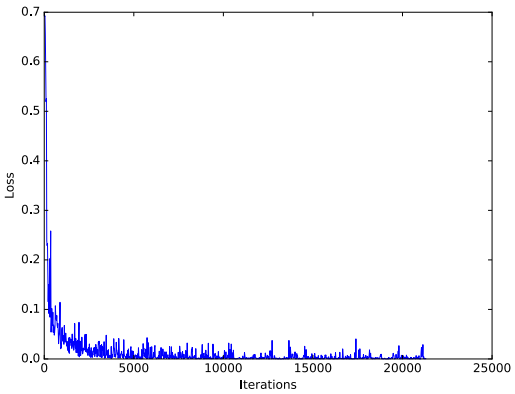
(a) 1 Frame Sequences - CNN Accuracy (Validity)



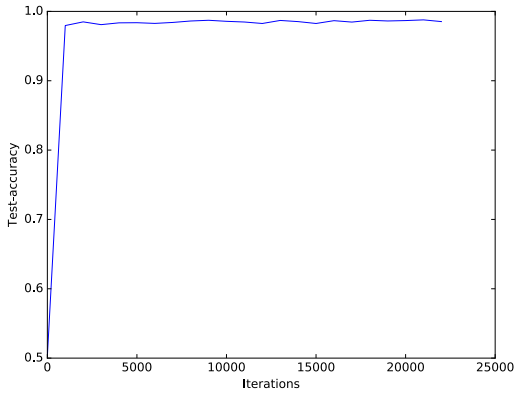
(b) 1 Frame Sequences - CNN Loss (Validity)



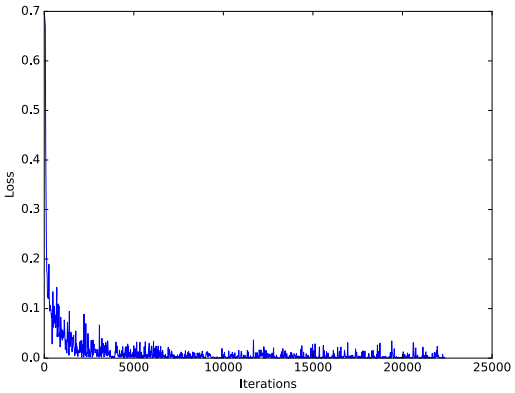
(c) 2 Frame Sequences - CNN Accuracy (Validity)



(d) 2 Frame Sequences - CNN Loss (Validity)



(e) 4 Frame Sequences - CNN Accuracy (Validity)



(f) 4 Frame Sequences - CNN Loss (Validity)

Figure 7.1: CNN Accuracy and Loss (Validity)

over the validation frames. The confusion matrix for the predictions over the validation set is shown in Table 7.3. This confusion matrix is built by summing the relevant items from the confusion matrices for each of the 50 pairs made of a validation set and a trained CNN. This indicates substantial agreement with the labelled data even for unseen subjects.

Table 7.3: CNN Confusion Matrix for Cross-Subject Validation (Validity)

Actual	Predicted	
	Invalid	Valid
Invalid	154,969	42,252
Valid	70,440	344,075

7.3.3 Cross-Lecture Validation

The previous section showed that the CNN was able to generalise well to unseen subjects. This section aims to test how well the CNN generalises to subjects that have potentially been seen before, but in different lectures.

Cross-lecture validation was performed by training on two lectures and testing on the remaining one. Recall that lectures 1 and 2 contained roughly the same students on the same day and therefore had similar views of the students wearing the same clothes. Lecture 3 was a different day and while students usually sit in similar places, clothing and camera placement varied.

Validation was performed by holding out a lecture and selecting a balanced training set of 80,000 images from the other two lectures. A balanced test set was built with 10,000 random images from the held-out lecture. If there were not enough frames, then a balanced set was constructed using what was available. The validation set was all the held-out lecture frames. Accuracy over the balanced test set for holding out lectures 1, 2, and 3 were 70.8%, 85.7%, and 86.5% respectively.

Over the full validation sets, the accuracies were 85.4%, 83.5%, 89.4%, with κ values of 0.41, 0.24, 0.5 respectively. Accuracy expected due to chance is approximately 72%. While accuracies are high, the κ values indicate only moderate agreement – this is due to the large imbalance in the dataset as it has mostly valid frames.

It is interesting to note that the system learns better on different subjects in the same lecture, than on the same subjects over different lectures where the camera placement and lighting may be different, as well as the student positions and clothing.

7.4 Classroom Actions

The following sections consider the accuracy of the CNN over the classroom actions. Only the actions with sufficient training data are considered.

Table 7.4: CNN Accuracy Detecting Writing

Sequence Length	Test Set Accuracy (%)		
	Peak	10,000 Iterations	20,000 Iterations
1	96.9	96.4	96.9
2	97.0	96.2	96.6
4	96.9	96.1	96.4

Table 7.5: CNN Confusion Matrix for the Validation Set (Writing)

Actual	Predicted	
	¬Writing	Writing
¬Writing	84,699	2,986
Writing	361	11,954

7.4.1 Writing

Cross-Frame Validation

From all three lectures, there were 424,128 valid writing labels. These consisted of 52,235 positive labels (Writing) and 371,893 negative labels (¬Writing). Using the same adjustments to AlexNet as in the previous section, a CNN was trained on a balanced dataset containing 15,000 image sequences corresponding to each label. The balanced test set contained 5,000 unseen examples from each category. A stratified validation set of 100,000 frames was created using 87,685 positive and 12,315 negative frames from the remaining unused ones.

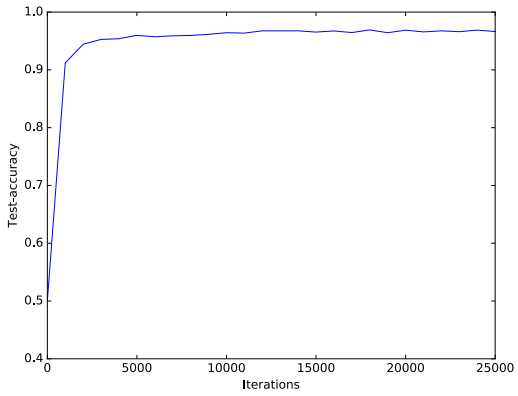
The accuracy and loss graphs are presented in Figure 7.2 and results in Table 7.4. Surprisingly, the CNN is largely unaffected by the sequence length – this is in contrast to the HOG SVM that performed better with more frames even when using the same number of features.

Calculating accuracy over the validation set yielded an accuracy of 96.7% with the confusion matrix shown in Table 7.5. The corresponding $\kappa = 0.858$, which indicates extremely good agreement.

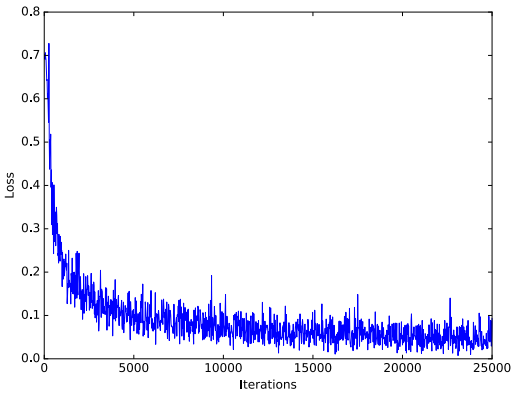
Cross-Subject Validation

Cross-subject validation was performed over lecture 1. There were 50 subjects in the lecture, of which 30 provided more than 1,000 frames from each category. Validation sets were built using all frames from the held-out subject. These were used to perform cross-subject validation.

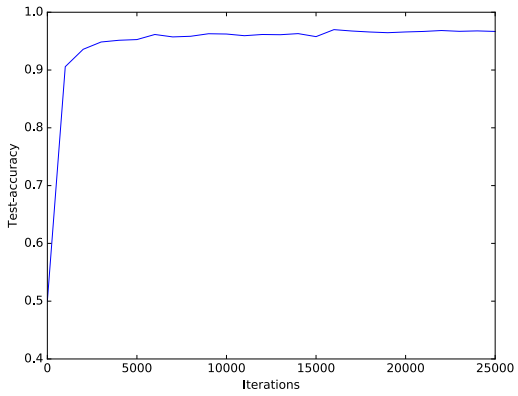
Over the 30 CNNs that were trained, the minimum and maximum accuracies were 50% and 97.9%. The mean and median accuracies were 72.1% and 70.3% respectively. On the validation sets the CNNs scored an accuracy of 82.7% (chance 74.4%) which gives $\kappa = 0.326$. The relevant confusion matrix is shown in Table 7.6 on page 139. Just like the HOG SVM, the CNNs were able to learn to recognise when unseen students were writing. However, where the SVM approach did so with accuracies in the low to mid 60%'s, the CNNs accuracy is approximately 10% higher in the low 70%'s.



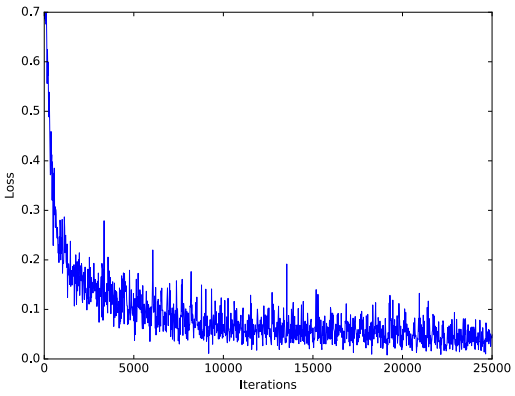
(a) 1 Frame Sequences - CNN Accuracy (writing)



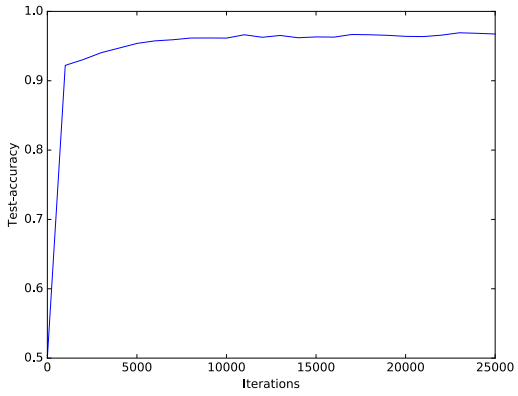
(b) 1 Frame Sequences - CNN Loss (writing)



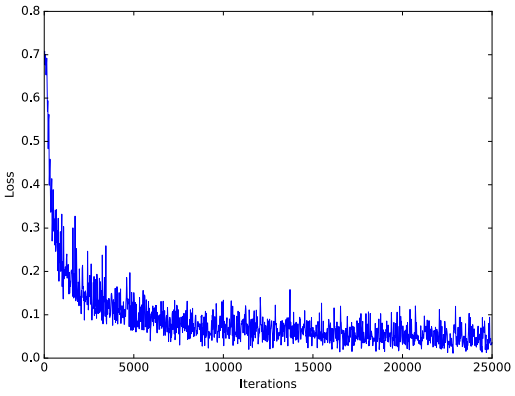
(c) 2 Frame Sequences - CNN Accuracy (writing)



(d) 2 Frame Sequences - CNN Loss (writing)



(e) 4 Frame Sequences - CNN Accuracy (writing)



(f) 4 Frame Sequences - CNN Loss (writing)

Figure 7.2: CNN Accuracy and Loss (Writing)

Table 7.6: CNN Confusion Matrix for the Cross-Subject Validation Sets (Writing)

Actual	Predicted	
	¬Writing	Writing
¬Writing	217,014	20,909
Writing	28,183	18,270

Table 7.7: CNN Confusion Matrix for the Cross-Lecture Validation Sets (Writing)

Actual	Predicted	
	¬Writing	Writing
¬Writing	217,014	20,909
Writing	28,183	18,270

Cross-Lecture Validation

Cross-lecture validation was performed over the three lectures. Balanced training sets of 50,000 to 60,000 images were used based on the availability of frames from the lectures. Balanced test sets of 20,000 images from the held-out lecture were used as a testing set. Validation sets were made from all frames in the held-out lecture.

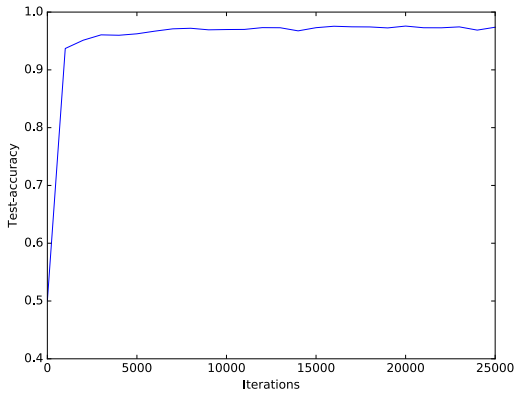
Holding out lectures 1, 2, and 3 yielded accuracies of 70.0%, 69.5%, and 54.0% on the balanced testing sets. On the validation sets 88.4% (chance 82%), 86.5% (chance 79%), and 91% (chance 91%). These give κ scores of 0.35, 0.35, and -0.03 respectively. Over all the validation sets, the accuracy was 88.1% (chance 82.6%) and $\kappa = 0.314$. The confusion matrix is shown in Table 7.7. Lecture 1 and 2 could be classified with fair accuracy, but the CNN did not generalise to lecture 3 at all. Note that the number of labels available in lecture 3 were very limited with fewer than 1,000 positive samples. More data should be collected from a number of different lectures and these tests should be repeated.

7.4.2 Cellphone

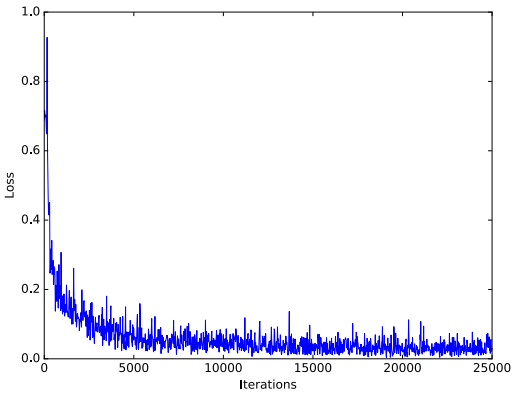
Cross-Frame Validation

The same process was followed for the Cellphone labels for which there are 424,153 frames in total. There are 361,886 negative (¬Cellphone) labels and 62,267 positive (Cellphone) ones. A CNN was trained on a balanced dataset containing 15,000 image sequences corresponding to each label. The balanced test set contained 5,000 unseen examples from each category. A stratified validation set of 100,000 frames was created using 85,320 positive and 14,680 negative frames from the remaining unused ones.

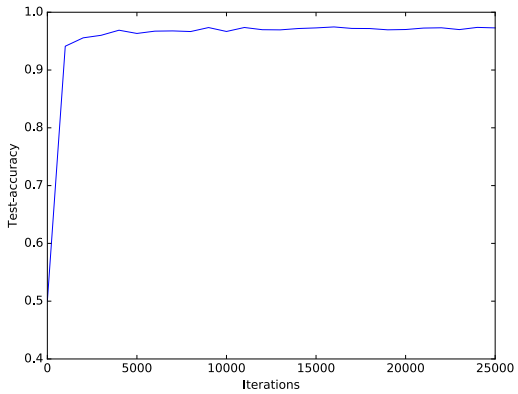
The accuracy and loss graphs are presented in Figure 7.3 and results in Table 7.8. Surprisingly, the CNN is largely unaffected by the sequence length – again this is in contrast to the HOG SVM that performed better with more frames even when using the same number of features.



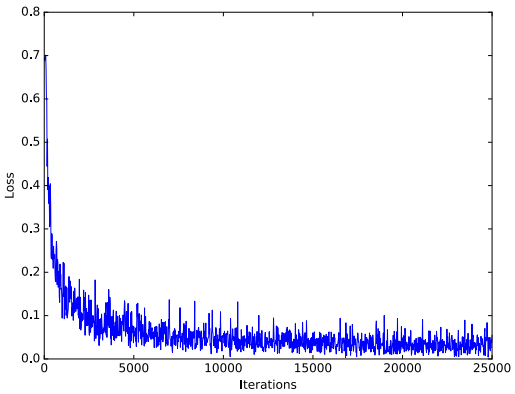
(a) 1 Frame Sequences - CNN Accuracy (cellphone)



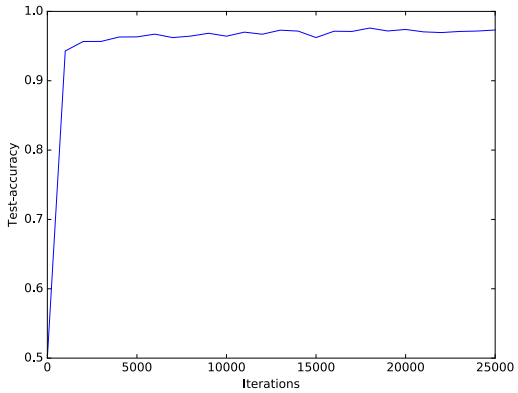
(b) 1 Frame Sequences - CNN Loss (cellphone)



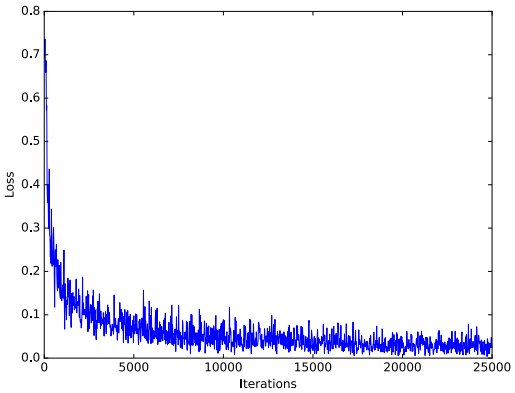
(c) 2 Frame Sequences - CNN Accuracy (cellphone)



(d) 2 Frame Sequences - CNN Loss (cellphone)



(e) 4 Frame Sequences - CNN Accuracy (cellphone)



(f) 4 Frame Sequences - CNN Loss (cellphone)

Figure 7.3: CNN Accuracy and Loss (Cellphone)

Table 7.8: CNN Accuracy Detecting Cellphone

Sequence Length	Test Set Accuracy (%)		
	Peak	10,000 Iterations	20,000 Iterations
1	97.6	97.0	97.6
2	97.5	96.6	97.0
4	97.6	96.4	97.4

Table 7.9: CNN Confusion Matrix for the Validation Set (Cellphone)

Actual	Predicted	
	¬Cellphone	Cellphone
¬Cellphone	82,610	2,710
Cellphone	294	14,386

Calculating accuracy over the validation set yielded an accuracy of 97.0% (chance 73%) with the confusion matrix shown in Table 7.9. The corresponding $\kappa = 0.888$, which indicates extremely good agreement.

Cross-Subject Validation

Cross-subject validation was performed over lecture 1. There were 50 subjects in the lecture, of which 18 provided more than 1,000 frames from each category. All frames from the held-out subject were used to construct a validation set. These were used to perform cross-subject validation.

Over the 18 CNNs that were trained, the minimum and maximum accuracies were 50% and 99.7%. The mean and median accuracies were 76.5% and 75.6% respectively. Over all the validation sets, the CNNs scored an accuracy of 83.2% (chance 65.2%) with $\kappa = 0.516$. The confusion matrix is shown in Table 7.10. Unlike the HOG SVM, the CNNs were able to learn to recognise when unseen students were on their cellphone in spite of the small number of subjects in the training set.

Table 7.10: CNN Confusion Matrix for the Cross-Subject Validation Sets (Cellphone)

Actual	Predicted	
	¬Cellphone	Cellphone
¬Cellphone	134,058	14,667
Cellphone	17,990	27,180

Cross-Lecture Validation

Cross-lecture validation was performed over the three lectures. Balanced training sets of 80,000 images were used when holding out either lecture 2 or 3. Based on the availability of frames when holding out lecture 1, a balanced training set of only 39,900 images was possible. Balanced test sets of 20,000 images were used as a testing set from the held-out lecture, except

when holding out lecture 3 where a balanced set of only 13,944 images was available. Validation sets were made from all frames of the held-out lecture.

Holding out lectures 1, 2, and 3 yielded accuracies of 66.9%, 80.1%, and 71.7% on the balanced testing sets. The validation sets returned accuracies of 87.0% (chance 78%), 89.6% (chance 82.1%), and 77.0% (chance 72.2%). These give κ scores of 0.382, 0.42, and 0.172 respectively. Over all the validation sets, the accuracy was 86.2% (chance 79.1%) with $\kappa = 0.343$. The relevant confusion matrix is shown in Table 7.11. Lecture 1 and 2 could be classified with fair accuracy. While still beating chance, generalisation to lecture 3 was poor.

Table 7.11: CNN Confusion Matrix for the Cross-Lecture Validation Sets (Cellphone)

Actual	Predicted	
	¬Cellphone	Cellphone
¬Cellphone	134,058	14,667
Cellphone	17,990	27,180

7.5 Posture

Cross-Frame Validation

Based on the previous tests, only single frames were tested for the posture. For posture there are five categories, so training was performed using both a balanced training set and a stratified training set.

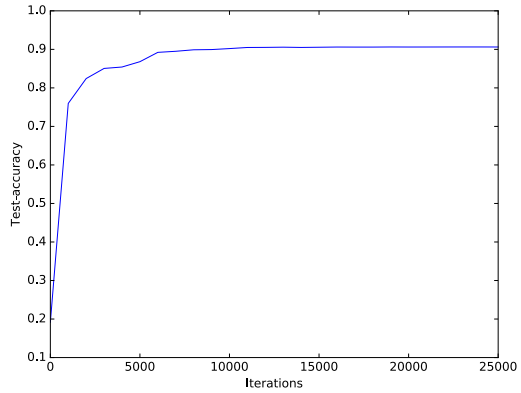
For the stratified test, a training set was built using random images from each respective category in proportions similar to the original data. There were 75,000 images in the training set. Similarly, the stratified testing set was built with 25,000 images and a validation set of 100,000 images. The network was trained and provided an accuracy of 89.5% (chance 31.6%) on the stratified test set. On the stratified validation set, the network achieved an accuracy of 84.8% (chance 31.8%) with $\kappa = 0.777$.

For the balanced test, a training set was built using equal numbers of images from each category. There were 75,000 images in the training set, corresponding to 15,000 from each label – which is the same as the experiments for the previous labels. Similarly, a balanced testing set was built with 25,000 images. Once again, a stratified validation set of 100,000 images is used. The trained network had an accuracy of 90.6% (chance 20%) on the balanced test set. On the stratified validation set, the network achieved an accuracy of 89.9% (chance 25.5%) with $\kappa = 0.798$. The validation set's confusion matrix for the CNN trained on the balanced data is given in Table 7.12. The corresponding accuracy and loss curves are shown in Figure 7.4.

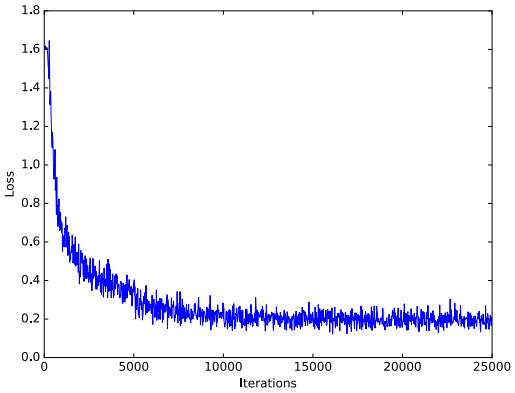
These results indicate that for the five-category data the best course is to continue using balanced datasets although the difference is small.

Table 7.12: CNN Confusion Matrix for the Validation Set (Posture)

Actual	Predicted				
	Left	Back	Upright	Forward	Right
Left	6,197	49	30	24	15
Back	657	18,244	608	446	549
Upright	1,230	3,246	29,449	3,392	2,506
Forward	152	546	1,124	23,695	298
Right	14	64	65	54	7,346



(a) 1 Frame Sequences - CNN Accuracy (posture)



(b) 1 Frame Sequences - CNN Loss (posture)

Figure 7.4: CNN Accuracy and Loss (Posture)

Cross-Subject Validation

Cross-subject validation was performed over lecture 1. There were 50 subjects in the lecture, of which 45 provided more than 1,000 frames from each category. All frames from the held-out subject were used to construct a validation set. These were used to perform cross-subject validation. Both stratified and balanced training sets were tested.

Over the 45 CNNs that were trained on stratified datasets, the minimum and maximum accuracies were 19.3% and 99.9%. The mean and median accuracies were 59.8% and 62.7% respectively. Over the validation sets, the CNNs scored an accuracy of 47.2% (chance 41.9%) with $\kappa = 0.092$ which indicates that while some subjects are being recognised correctly, overall the network is not generalising.

The same tests were run using balanced training sets which used equal proportions of images from each category. Over the 45 CNNs that were trained on balanced datasets, the minimum and maximum accuracies over the balanced testing sets were 17.3% and 94.3%. The mean and median accuracies were 54.6% and 53.5% respectively. Over the stratified validation sets, the overall accuracy was 43.8% (chance 40.6%) with $\kappa = 0.053$, which indicates very poor agreement. The confusion matrix is shown in Table 7.13.

In this case the balanced training data decreased the performance overall – although in both cases the system generalised very poorly to unseen students.

Table 7.13: CNN Confusion Matrix for the Cross-Subject Validation Sets (Posture)

Actual	Predicted				
	Left	Back	Upright	Forward	Right
Left	487	2845	13,279	2,440	723
Back	3,389	11,596	34,079	8,180	2,499
Upright	12,885	25,574	139,746	42,956	9,564
Forward	2,319	6,107	40,646	19,498	2,858
Right	732	3,080	8,503	1,556	1,377

Cross-Lecture Validation

Cross-lecture validation was performed over the three lectures. Stratified training sets of 80,000 images were used when holding out either lecture 2 or 3. Based on the availability of frames when holding out lecture 1, a balanced training set of only 39,900 images was possible. Balanced test sets of 20,000 images were used as a training set from the held-out lecture, except when holding out lecture 3 which was a balanced set of 13,944 images. Validation sets were made from all frames of the held-out lecture.

Holding out lectures 1, 2, and 3 yielded accuracies of 49.3%, 73.3%, and 80.2% on the balanced testing sets. The validation sets returned accuracies of 43.1% (chance 38.3%), 72.7% (chance 26.4%), and 79.8% (chance 26.8%). These give κ scores of 0.126, 0.63, and 0.724 respectively. Over all the validation sets, the accuracy was 63.1% (chance 30.7%) with $\kappa = 0.467$. The relevant confusion matrix is shown in Table 7.14. Lecture 2 and 3 could be classified with fairly good accuracy. While still beating chance, generalisation to lecture 1 was poor – this may be due to the relatively small number of frames in lectures 2 and 3 in comparison to lecture 1.

Table 7.14: CNN Confusion Matrix for the Cross-Lecture Validation Sets (Posture)

Actual	Predicted				
	Left	Back	Upright	Forward	Right
Left	13114	3274	5945	3806	264
Back	2582	49926	22340	7560	2602
Upright	8112	23176	148311	42471	13011
Forward	1357	3617	27187	81979	1364
Right	302	2444	8263	2617	17771

7.6 Horizontal Head Pose

Cross-Frame Validation

Approximately 90% of the data has students looking forward with 9% looking slightly left or right and only 1% looking completely sideways. Through all three lectures, there were only 2,787 frames of students looking far left and a similar amount looking far right.

To construct a balanced training set would result in a set of only 25,000 images, which is too small for training. Therefore, to construct a balanced dataset it was decided to reuse images from the under-represented categories, but with arbitrary cropping and rescaling. 20,000 images in each category were used to create a balanced training set of 100,000 images. Training on the balanced set provided 89.6% accuracy over a stratified validation set of 100,000 images. Removing the repeated images and using the resulting set as a stratified training set resulted in an accuracy of 90.3%. The accuracy of chance over the validation set was 66.6% and $\kappa = 0.709$.

In both cases the network was able to accurately categorise new image frames. The confusion matrix from the stratified test is given in Table 7.15. It appears that even with the different pre-processing, repeating images did not help provide the system with more information.

Table 7.15: CNN Confusion Matrix for the Validation Set (Horizontal Head Pose)

Actual	Predicted				
	Far Left	Left	Forward	Right	Far Right
Far Left	348	20	7	0	1
Left	28	3,741	66	110	9
Forward	225	2,979	76,078	5,072	316
Right	8	344	326	9,562	131
Far Right	2	8	10	47	562

Cross-Subject Validation

Cross-subject validation was performed over lecture 1. All frames from the held-out subject were used to construct a validation set. These were used to perform cross-subject validation. Both stratified training sets were tested.

The minimum, maximum, mean and median accuracies are not reported as they are misleading without comparing each individual result with chance. Over all the validation sets, the CNNs scored an accuracy of 83.91% (chance 83.5%) with $\kappa = 0.025$ which indicates that the network is not generalising to recognise the horizontal head pose of unseen subjects. The confusion matrix is given in Table 7.16. The confusion matrix shows how the CNN is heavily biased towards the forward-facing label and against the left and right extremes, getting none of those correct. Even when combining the moderate and far labels (e.g. far left and left become one category), κ remains low at 0.028.

Table 7.16: CNN Confusion Matrix for the Cross-Subject Validation Sets (H. Head Pose)

Actual	Predicted				
	Far Left	Left	Forward	Right	Far Right
Far Left	0	9	130	5	0
Left	0	76	956	101	0
Forward	46	669	34157	1961	82
Right	3	82	2323	125	17
Far Right	0	15	174	13	0

Cross-Lecture Validation

Cross-lecture validation was performed over the three lectures. Stratified training sets of 200,000 images were used. Validation sets were made from 100,000 frames of the held-out lecture with categories represented proportionally.

Holding out lectures 1, 2, and 3 for training and then testing on them yielded accuracies of 79.9% (chance 79.4%), 74.9% (chance 75.4%), and 66.1% (chance 65.7%). These give κ scores of 0.023, -0.021, and 0.011 respectively. Over all the validation sets, the accuracy was 73.7% (chance 74.1%) with $\kappa = -0.019$. The relevant confusion matrix is shown in Table 7.17. The κ scores indicate that the network is performing poorly and is unable to generalise between lectures.

Table 7.17: CNN Confusion Matrix for the Cross-Lecture Validation Sets (H. Head Pose)

Actual	Predicted				
	Far Left	Left	Forward	Right	Far Right
Far Left	5	180	1014	129	5
Left	41	1008	12823	388	40
Forward	435	6274	218405	17989	959
Right	46	976	35528	1455	92
Far Right	9	31	2023	144	1

7.7 Vertical Head Pose

Cross-Frame Validation

There were four possible categories for vertical head pose. A stratified training set of 100,000 images was used for training and another 100,000 for validation. The network achieved 88.6% (chance 46.7%) accuracy with $\kappa = 0.785$. The confusion matrix from the stratified validation is given in Table 7.18.

Table 7.18: CNN Confusion Matrix for the Validation Set (V. Head Pose)

Actual	Predicted			
	Below Desk	On Desk	Forward	Up
Below Desk	7893	374	826	3
On Desk	575	24254	5514	13
Forward	467	3439	55963	19
Up	18	54	126	462

Cross-Subject Validation

Cross-subject validation was performed over the vertical head pose labels across all 50 subjects in lecture 1. Because of the limited number of upward exemplars, balanced test sets were not

considered – only stratified validation sets. As such the individual accuracies are not provided as each would need to be compared to chance.

Over all validation frames, however, the accuracy was 60.5% (chance 50.9%) with $\kappa = 0.195$, which indicates fairly poor agreement overall, but much better than for horizontal classification. The confusion matrix is provided in Table 7.19. It is evident that there were not enough labels of students looking upwards in the training set and the CNN has learnt to never predict that label in favour of the others.

Table 7.19: CNN Confusion Matrix for the Cross-Subject Validation Sets (V. Head Pose)

Actual	Predicted			
	Below Desk	On Desk	Forward	Up
Below Desk	302	727	1134	0
On Desk	469	3159	4221	0
Forward	793	3469	13131	0
Up	0	3	29	0

Cross-Lecture Validation

Cross-lecture validation was performed over the three lectures. Stratified training sets of 50,000 images were used. Validation sets were made from 100,000 frames of the held-out lecture with categories represented proportionally.

Holding out lectures 1, 2, and 3 for training and then testing on them yielded accuracies of 66.2% (chance 50.0%), 60.5% (chance 47.3%), and 51.4% (chance 44.5%). These give κ scores of 0.325, 0.250, and 0.125 respectively. Over all the validation sets, the accuracy was 59.38% (chance 47.6%) with $\kappa = 0.225$. The relevant confusion matrix is shown in Table 7.20. The κ scores indicate that the network is still not performing well; it is generalising better between lectures than between students.

Table 7.20: CNN Confusion Matrix for the Cross-Lecture Validation Sets (V. Head Pose)

Actual	Predicted			
	Below Desk	On Desk	Forward	Up
Below Desk	6383	9418	12283	35
On Desk	6679	38562	47703	45
Forward	7568	35926	133147	47
Up	267	296	1630	11

7.8 Conclusion

This chapter applied the AlexNet CNN architecture to classify the various labels in WITSDB. In general the CNN out-performed the HOG SVM in terms of absolute accuracy, training time,

classification time[1], and generalisation abilities.

For the various labels cross-frame, cross-subject and cross-lecture validations were performed. In all cases, the cross-frame validation yielded accuracies in the high 80%'s and 90%'s. This indicates low intra-subject variation – that the frames of a single subject in a single lecture are quite similar – and the system is able to easily remember similar frames from training to perform the classification.

In all cases this accuracy drops when dealing with cross-subject and cross-lecture validation, i.e. there is larger inter-subject variation. Cross-subject validation assess how well the system is able to generalise its knowledge to unseen students, while cross-lecture validation assess students that have possibly been seen before but at different lecture times.

Lectures 1 and 2 were recorded on the same day, with the same group of students in the same lecture venue. However, the camera was moved slightly between the lectures and all students stood up and repositioned themselves. Lecture 3 was recorded in the same venue, but on a different day. This means that students were seated differently and wore different clothes. Due to time and cost constraints, there are fewer action labels for lecture 3 than lectures 1 and 2.

The cross-subject validation for all the writing and cellphone labels produced κ values of 0.326 (fair agreement) and 0.516 (moderate agreement) respectively. For writing, this means an accuracy of 82.7% where chance expects an accuracy of 74.4%, and for cellphone, this was an accuracy of 83.2% with a chance accuracy of 65.2%. The HOG SVM completely failed to generalise across subjects for the cellphone labels while the CNN performed relatively well. On the writing labels, the CNN outperformed the SVM by approximately 10%.

Cross-lecture validation showed that in general training on lecture 1 resulted in fair agreement when classifying lecture 2, and vice versa. This is as students were largely seated in the same places, and were wearing similar clothes. Training on lectures 1 and 2, however, generally resulted in very poor classification on lecture 3 where students wore different clothes and had changed seats more than in the other two. In this light – considering a subject on a different day is seemingly equivalent to viewing a completely unseen subject and previous training on that specific person does not translate.

Similar results were seen in the cases where four- and five-way classification was performed for posture and head pose. Generally the system performed better when trained on stratified data sets, although large balanced datasets were impossible in situations where some labels were severely under-represented. By training on the stratified sets, the networks learned to bias their results, and were more likely to misclassify minority labels into the majority ones. This is evidenced by the distribution of predictions shown in the confusion matrices above.

In cross-frame validation, posture was classified with an accuracy of approximately 90%, while vertical and horizontal head pose were classified at 88.6% and 90% accuracy with κ scores indicating very high to almost perfect agreement. Again, the cross-frame validations are high because frames containing the same subject are generally fairly similar.

[1] Using GPU hardware acceleration

The generalisation to unseen subjects and different lectures gives more insight into the system's ability to actually recognise the different postures and head poses. In all three cases the CNN outperformed the HOG SVM which failed to generalise entirely. Even though it outperformed the HOG SVM, the CNN generally yielded low κ scores indicating only poor to fair generalisation.

This low generalisation may stem from limitations in the dataset and is considered further in Chapter 9.

Chapter 8

Interest and Visualisation: The Interest Map

The CNN aspects of this chapter are published in Klein and Celik (2017b), and the proposed interest map and lecturer acceptance are published in Klein and Celik (2017c).

8.1 Introduction

Throughout this work, the focus has been on generating the interest information for the lecturer. In Chapters 4 and 5, it was found that gathering interest labels directly from subjects or external raters was unreliable and those approaches were abandoned in favour of observational checklists. The idea behind observational checklists is to focus on aspects that are more measurable than the less easily defined emotions. By recognising when students are distracted or when they are showing common postural traits, an *Interest Value* can be constructed based on predefined rules.

Once an interest value is calculated for each visible student in the class, this information should be available to the presenter in a way that one can gauge both the mood of the class as

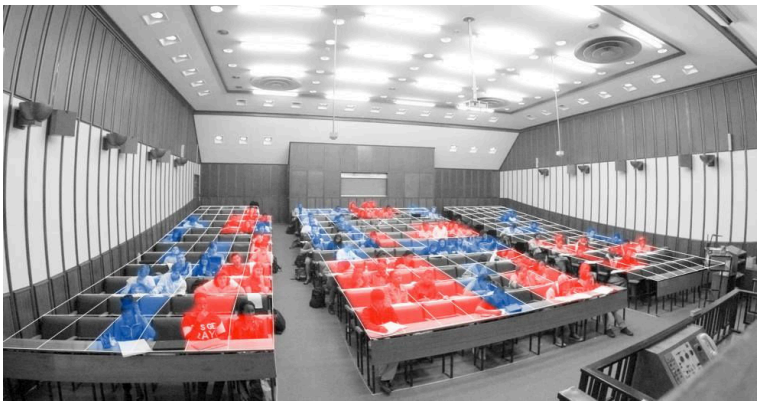


Figure 8.1: Interest Map

a whole and identify specific issues. This chapter proposes an Interest Map where the system overlays engagement information on either a static image or a live video feed of the venue. Figure 8.1 provides an example of this. An interest map is a heat map that displays the relevant metric in a way that the presenter can gauge the overall level and spatial spread of engagement at a glance. It is updated during the lecture to indicate which students are disengaged at different times throughout. For example, Figure 8.2 on the next page shows 4 frames from different times during a lecture where the red highlights indicate disengagement.

The colour scheme can be customised, but the feedback needs to be clear and concise so that a presenter can glance at the map and immediately know where attention should be focused.

A general spread of red over the map may indicate that a concept needs to be covered again or that the audience needs a break. On the other hand, a group of students that consistently show up red over time or even across lectures may indicate that some other specific intervention is needed. While this work focuses on creating the interest map, future work should also focus on identifying patterns within the interest map and suggest interventions that are known to help in specific instances.

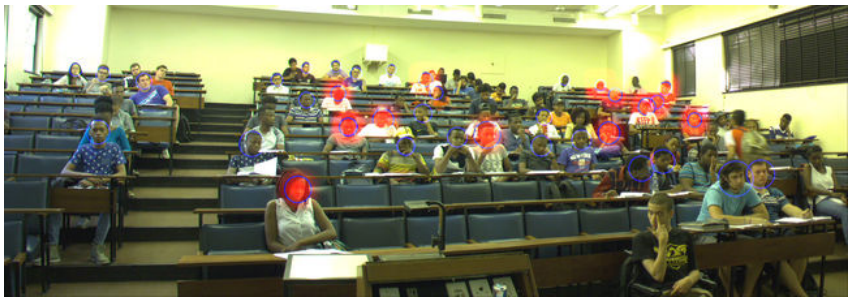
8.2 The Interest Map Computational Pipeline

There are two possible ways to build such a system based on observational checklists. On the one hand, the system can be built to independently recognise each of the previously identified behaviours and then merge them together using predefined rules. On the other, the labelled data can be used to construct a dataset of engaged and disengaged students. This dataset can then be used to train a single classifier. The system itself is never actually shown the observational checklist, but instead must learn the characteristics from the dataset by itself.

The previous chapters focused on individually recognising each of the identified actions and postures. Using the rules developed in Section 8.3 this work can be extended easily to return an interest value based on the individual observations. This chapter instead focuses on using the observations to construct a single dataset of interest that can then be used to train and test a single classifier. A benefit of this approach is that the overall computational cost of a prediction is limited to a single classifier rather than having to run multiple classifiers at each stage – although that could surely be parallelised to increase speed.

Based on the single classifier design, Figure 8.3 on page 153 shows a pipeline for creating and testing an interest map. There are two main concerns for the process: training the classifier, and displaying the interest map to the user. The pipeline for training resembles those in the previous sections. The image sequences are first extracted from the video and saved as image files alongside a database of labels.

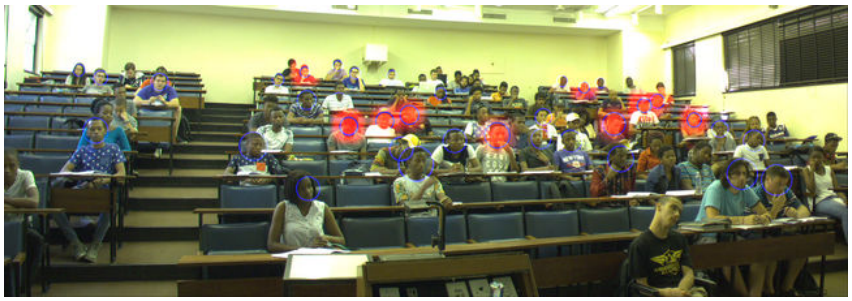
If using an SVM, the HOG features are extracted from the relevant frames and dimensionality reduction is performed. The SVM is then trained as a binary classifier. Using Platt's method, one can calculate the probability of an observation belonging to a specific class based on its distance to the support vectors (Chang and Lin 2011; Lin *et al.* 2007; Platt and others 1999). In this case, the SVM provides the probability that a subject is interested or not, which is termed the interest value, $I(s,t) \in [0,1]$, for subject, s at frame, t.



(a) Frame 2000



(b) Frame 2500



(c) Frame 3000



(d) Frame 3500

Figure 8.2: Temporal Interest Map

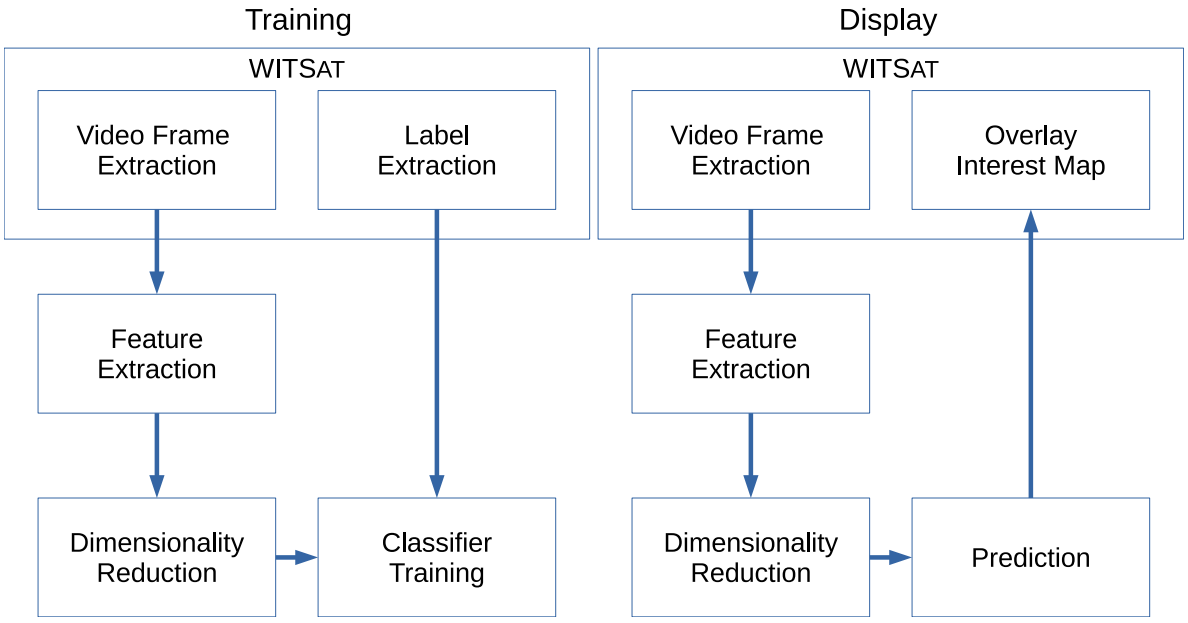


Figure 8.3: Computational Pipeline for Interest Map Creation

Alternately, if a CNN is used as the classification agent, then the feature extraction and dimensionality reduction steps are encoded into the network itself and the training process occurs over the images directly. The output of the CNN should use a softmax layer as the output which has two neurons. One neuron signals interest and the other disinterest. As the sum of the neurons must be one, the values of the outputs can be interpreted as the probability of the subject belonging to each class. Once again, let the interest value be $I(s,t) \in [0,1]$, for subject, s at frame, t where the interest value is the probability of the subject being interested.

WITSAT was extended to display an Interest Map as seen in Figure 8.2. Circles tracking subjects' faces are rendered along with Gaussian shaped highlights. The shape, colour blending, and transparency can be customised. The colour and transparency are adjusted according to each subject's interest value. In Figure 8.2, the highlight is opaque red when interest is low at 0, with the transparency increasing linearly from [0,1].

This means that disinterested students are highlighted in red while interested students are not highlighted at all. This makes its easier for the presenter to quickly assess where and to what extent their attention is required. The software can also vary the display between two colours, blending them so that an interested student is highlighted in blue as seen in Figure 8.1, for example.

8.3 Defining Interest: The Observational Checklist

To actually define interest, the actions, posture, and head pose of the students from Chapter 5 are all considered in the definition that follows. Images where a subject is present and not occluded are labelled as valid (1) – only these valid images are considered and images that classified as invalid (0) are automatically excluded. Let the label 'Interest' indicate that a

student is interested or engaged, and '¬Interest' indicate that the student is not interested or disengaged. The assumptions listed below are based on the literature surveyed in Chapter 3 as well the experience developed during data labelling and lecturing in general. Limitations are discussed where appropriate. The assumptions that follow form a sieve – the rule either causes a student to be marked (¬)Interested or it has no effect. If a student passes through all the rules without being explicitly marked one way or another, then the student is considered to be engaged.

There are seven actions identified in the database: writing, cellphone use, laptop use, talking, raised hand, yawning, and head-on-desk. These are defined in Section 5.3.5 and illustrated in Figure 5.9.

The following rules about actions are considered:

$$\begin{aligned} \text{Writing} &\Rightarrow \quad \text{Interest} \\ \text{Cellphone} &\Rightarrow \neg\text{Interest} \\ \text{Talking} &\Rightarrow \neg\text{Interest} \\ \text{Raised Hand} &\Rightarrow \quad \text{Interest} \\ \text{Yawning} &\Rightarrow \neg\text{Interest} \\ \text{Head on Desk} &\Rightarrow \neg\text{Interest} \end{aligned}$$

The fundamental assumption is that a student that is writing is actually taking notes and not just drawing or playing 'Dots and Boxes' with a friend. A student with a raised hand is either asking a question or responding to one and therefore engaging with the lecturer. On the other hand, a student that is on a cellphone or that has put their head on the desk is considered to have disengaged and lost interest. Yawning is considered to be a symptom of fatigue that ultimately leads to disinterest. A student talking to a friend may be discussing work, asking a question, or talking about an upcoming party. WITS is unable to tell the difference between these cases, so the assumption is that they are just *chatting* and therefore disengaging. It is unclear whether laptop use implies interest or not. In many real-life cases the students are taking notes, and in many others the students are playing games or browsing the internet. As only one subject in the dataset used a laptop, for the purposes of this research the debate can be avoided and laptop use is ignored completely.

There are five labelled upper body postures that correspond to the images shown in Figure 5.11. Specifically, these postures correspond to leaning left, right, backwards, forwards or sitting upright.

The following rules about posture are considered:

$$\begin{aligned} \text{Left} &\Rightarrow \neg\text{Interest} \\ \text{Right} &\Rightarrow \neg\text{Interest} \end{aligned}$$

A student that is leaning far to either side is always disengaged. This behaviour co-occurs with talking to friends, sleeping and cellphone use. Those leaning only moderately to the side still exhibit a closed body posture that is identified in literature to occur with combative (resistance and aggression) or fugitive (withdrawal, defensive, boredom) stances. In both cases, this implies disinterest or disengagement on the part of the student. Those students with an

open posture are usually sitting upright, leaning forwards or backwards. Whether this indicates interest or not, largely depends on a student's head pose and actions. Therefore, there are no rules that specifically deal with these poses, but if the student is not excluded in the steps that follow, they will still be marked as interested.

For example, in Figure 5.11b on page 110, the student is leaning back while watching the lecturer with an open body position and a frontal head pose – this student is evidently concentrating on the class. Figure 5.12c, on the other hand, shows a student leaning back so that he can see the cellphone under the desk – in this case the student is obviously disengaged from the class and trying to hide it. In the latter case, the student will pass through the rules relating to body posture, but will be excluded by either the cellphone use or because is head pose focuses his attention under the desk.

The head pose labels are illustrated in Figure 5.12 and mostly corresponds to a student's focus of attention – unfortunately the resolution of the images does not allow for eye tracking. The horizontal labels correspond to far left or right, moderately left or right, or forwards. Vertical labels correspond to looking below the desk, on the desk, forwards, or upwards. These labels lead to the following rules involving head pose:

$$\begin{aligned} \text{Far Left} &\Rightarrow \neg\text{Interest} \\ \text{Far Right} &\Rightarrow \neg\text{Interest} \\ \text{Up} &\Rightarrow \neg\text{Interest} \\ \text{Below Desk} &\Rightarrow \neg\text{Interest} \end{aligned}$$

Students looking to the far left, far right, or up at the roof are not focusing on the lecturer or their notes. They are either looking at a friend, or looking around the lecture venue. Sometimes it is the case that they are focusing on another student that is asking a question. Detecting this would require a global consideration of the students in the video and should be the subject of future work. Those that are looking below the desk are almost always distracted by something that they are trying to hide from the lecturer – usually a cellphone as illustrated in Figure 5.12c.

Altogether these rules can be considered in a manner similar to the cascade classifier. Firstly, the Writing and Raised Hand labels are checked – if either is true, the student is marked interested. The remaining rules are all used to reject students and mark them as ¬Interested. If a student successfully passes through all 'weak rules' without being rejected then the student is labelled as Interested.

8.4 Recognising Interest

8.4.1 Direct and Indirect Interest Classification

Based on the rules above, two primary approaches to interest recognition are apparent. In the first approach, a number of individual proxy classifiers can be used first and using the rules above for decision fusion they can be merged into final interest labels. The previous two chapters examined the feasibility of recognising the individual proxy labels and while the CNN is seen to vastly outperform the HOG SVM on all metrics, there are still generalisation issues

	filename	subject	frame	x	y	r	Valid	Writing	Cellphone	HeadOnDesk	ntalHea	calHead	Posture
1	Subject15_885_169/img_...	Subject15	1	0.378...	0.246...	0.207...	0	0	0	-1	-2	0	0
2	Subject17_772_161/img_...	Subject17	1	0.534...	0.323...	0.246...	0	0	0	-1	0	-1	0
3	Subject18_843_151/img_...	Subject18	1	0.469...	0.245...	0.185...	0	0	0	-1	0	1	0
4	Subject19_248_175/img_...	Subject19	1	0.561...	0.330...	0.228...	0	0	0	0	0	0	0
5	Subject20_195_172/img_...	Subject20	1	0.670...	0.369...	0.25	0	0	0	-1	0	-1	0
6	Subject21_1382_350/im...	Subject21	1	0.470...	0.398...	0.187...	0	0	0	-1	1	1	0
7	Subject22_1562_367/im...	Subject22	1	0.883...	0.278...	0.246...	0	0	0	-1	0	1	0
8	Subject23_1650_391/im...	Subject23	1	0.510...	0.239...	0.226...	0	0	0	-1	0	1	0
9	Subject24_1471_341/im...	Subject24	1	0.557...	0.462...	0.275...	0	0	0	-1	0	-1	0
10	Subject41_1637_193/im...	Subject41	1	0.5625	0.301...	0.125	0	0	0	-1	1	1	0
11	Subject44_2004_463/im...	Subject44	1	0.637...	0.299...	0.239...	0	0	0	-1	0	-1	1

Figure 8.4: Extract of SQLite Database of Labels

Table 8.1: Interest Frames per Lecture

Lecture	¬Interested		Interested		Total
1	102,402	(25%)	312,113	(75%)	414,515
2	66,219	(31%)	146,339	(69%)	212,558
3	68,705	(27%)	187,083	(73%)	255,788
	237,326	(27%)	645,535	(73%)	882,861

that should be addressed. Once the proxy classifiers have adequate accuracy on unseen subjects, then they could be used to build a system to recognise interest using the rules above. The second approach is to use these rules to build a dataset of students that are labelled according to interest and then train a classifier on this data without ever explicitly presenting the other labels.

The benefit of the first approach is that the system can report to the users in detail about why the system believes the student is engaged or not, based on the individual proxy responses. On the other hand, it would require that a number of separate classifiers are running simultaneously which has implications for run-time performance. The first approach also suffers from issues where classification errors are potentially compounded by the accept/reject nature of the interest rules.

8.4.2 Data Export

Subjects that were missing either posture or action labels were given the missing labels using the CNNs from cross-frame validations in the previous chapters. WITSAT was used to export all the data labels to an SQLite database, such as the one in Figure 8.4, along with the corresponding images. Based on the rules discussed in the previous section, the SQL query in Listing 8.5 allows the construction of an interest dataset.

The query produced 882,861 valid frames of which 237,326 (27%) are ¬Interested and 645,535 (73%) Interested. The number of frames in each lecture is shown in Table 8.1.

3,000 image sequences were randomly sampled to have the same proportions of each class as the original data. These 3,000 sequences were used to construct a PCA basis that retains 90% of the variance.

```
SELECT filename,
   `3Writing` == 1 /* Both actions mean the subject is interested */
    OR `3RaisedHand` == 1      /* regardless of the other labels  */
    OR
   (    `3Cellphone` != 1   /* Not Cellphone                       */
    AND `3Talking`   != 1   /* Not Talking                         */
    AND `3Yawning`   != 1   /* Not Yawning                         */
    AND `3HeadOnDesk`!= 1   /* Not Head on Desk                    */
    AND `5Posture`   NOT IN (-2, 2) /* Neither left nor right      */
    AND `5HHeadPose` NOT IN (-2, 2) /* Neither far left nor right */
    AND `5VHeadPose` NOT IN (-2, 2) /* Neither below desk nor up  */
   ) AS Interest
  FROM data
 WHERE `3Valid` == 1;               /* Occupied and Not Occluded  */
```

Listing 8.5: SQL Statement to Export a Label

A HOG SVM (with PCA) and a CNN are trained as binary classifiers on the input image sequences. Based on the results of previous chapters, the focus is primarily on the CNN based approach although HOG SVM experiments are also presented for completeness.

8.4.3 Cross-Frame Validation

From all available data, a stratified training set of 80,000 images was sampled. A separate validation set of 100,000 images was sampled from the remaining one. The network was able to classify the validation set with 93.0% accuracy (60.8% chance) with $\kappa = 0.820$, which indicates extremely good agreement. The confusion matrix is given in Table 8.2. Training on a balanced set of the same size yielded similar results.

Table 8.2: CNN Confusion Matrix for the Validation Set

Actual	Predicted	
	¬Interested	Interested
¬Interested	23,199	3,807
Interested	3,227	69,767

For comparison, a HOG SVM was trained on the same dataset. Performing 5-fold cross validation over the training set provided an accuracy of 72.6%. When learning on the entire training set, the SVM yielded 73.0% accuracy on the stratified validation set. Again, the CNN significantly outperforms the HOG SVM.

8.4.4 Cross-Subject Validation

Just as in the earlier tests, to avoid inadvertently repeating subjects, cross-subject validation was performed on lecture 1 only. There are 50 labelled subjects in lecture 1. For a selected test subject, the CNN was trained on the remaining 49 subjects. A balanced training set would be built from 60,000 images selected at random from the 49 subjects so that there were 30,000 images in each category. A balanced testing set was then built using an equal number of positive and negative images based on the number that were available for the current test subject. A validation set was built using *all* frames from the test subject.

50 CNNs (one for each subject holdout) were trained in this way to test how well the system generalised to unseen subjects. On the balanced test sets, the minimum accuracy was 50.0% (equivalent to chance), while the maximum was 91.5%. The mean and median accuracies were 65.6% and 63.6% respectively. These accuracies are similar to the 61.86% - 65.72% accuracies attained in Raca *et al.* (2015), although it should be noted that the ground truth labels in that work are not necessarily comparable to those used here as the labelling methodology was different.

Classification on the unbalanced validation sets resulted in an accuracy of 59% (chance 50%, $\kappa = 0.163$) over all frames with the confusion matrix shown in Table 8.3.

Table 8.3: CNN Confusion Matrix for cross-subject validation on all lecture 1 frames

Actual	Predicted	
	¬Interested	Interested
¬Interested	100,535	83,135
Interested	88,095	141,141

The HOG SVM trained and tested on the same data gives minimum and maximum accuracies of 22.5% and 77.8%. The mean and median accuracies are 49.4% and 51.3% respectively. Accuracy over the entire validation set is 53.5%. This is about 6% lower than the CNN approach.

8.4.5 Sequence Length

The previous two experiments used single images of students as the input to the classifier. The experiment in Section 8.4.3 is repeated for sequences of 2 and 4 frames. The results shown in Table 8.4 show no significant improvement over the single frame case, indicating that the CNN is able to extract the relevant information from a single frame already or that the kernels are unable to capture adequately the information from multiple frames.

8.4.6 Image Size

The previous experiments were all performed on 64×64 images. Using single frame image sequences, a number of larger image sizes were considered for the training, testing, and vali-

Table 8.4: CNN Test Set Accuracy by Sequence Length

Number of Frames	Testing Accuracy	Validation Accuracy
1	89.8	89.6
2	89.7	89.4
4	90.0	89.7

Table 8.5: CNN Test Set Accuracy by Image Size

Image Size	Testing Accuracy	Validation Accuracy
64×64	89.8	89.6
96×96	88.0	87.9
128×128	85.0	85.4

dation sets from Section 8.4.3. The results are shown in Table 8.5. Interestingly, the increased image size causes a decrease in accuracy. The cause is likely to be two-fold. On the one hand, the kernel size remains the same, so it can now only find smaller features relative to the whole image. On the other hand, as the size of the image increases the size of the feature maps does too. This means the number of parameters in the model increases. Even with larger training sets of up to 240,000 images, the accuracy using larger images remains equivalent to those above.

8.4.7 Recognising New Subjects

The results in the previous section indicate that in its current form the system does not generalise well to new subjects. Further experiments were run to understand how much information the CNN needed about a subject before it would achieve acceptable accuracies.

Adding 1,000 Random Frames – Cross-Subject Validation

The cross-subject validation experiment from Section 8.4.4 was repeated, but this time 1,000 random frames were moved from the validation set to the training set. This meant that the training set now contained 1,000 random frames of the new subject. The CNNs were trained and validated in the same way as before, although subjects with fewer than 1,000 validation frames left were now excluded from the analysis – leaving behind 45 students for validation.

By including the 1,000 frames of each holdout, the accuracy over the validation set increased from 59.0% to 89.6%. The size of the validation set was 391,769. This is comparable to the results when just randomly selecting frames over the entire set – as in Section 8.4.3. This strongly supports the idea that relatively few frames are actually needed per subject in order to accurately detect a subject's interest. It is particularly salient when considering that 1,000 frames is only 2–5 minutes in total.

These minutes, however, were spread out over the entire lecture, which makes it more representative than if only taken from the start of a lecture.

Training on the start of a lecture – Cross-Subject Validation

The previous finding suggests that a relatively small proportion of frames for a subject are actually needed to achieve good accuracy. This experiment repeats the previous one, but takes the 1,000 frames from the start of the lecture specifically instead of using random frames.

This is comparable to performing some kind of calibration step for an unseen student, but then monitoring that student for the rest of the lecture. This time the network achieves 88.8% accuracy (chance 50.36%) with $\kappa = 0.774$, which still indicates an extremely high agreement. The confusion matrix is shown in Table 8.6.

Table 8.6: CNN Confusion Matrix for Cross-Subject Validation with 1,000 Frame Bootstrap

Actual	Predicted	
	¬Interested	Interested
¬Interested	146,780	18,787
Interested	22,168	178,020

8.4.8 Cross-Lecture Validation

Section 8.4.4 aimed to assess how well the CNN generalised to new, unseen subjects. Section 8.4.7 then investigated how well the CNN handled new subjects with minimal training data. These tests were run using only a single lecture so that students were not inadvertently repeated.

This section aims to understand how well the CNN generalises to subjects that it has potentially seen before, but in different lectures.

As mentioned previously, lectures 1 and 2 took place on the same day and captured more or less the same group of students, sitting in similar places, wearing the same clothes. Lecture 3 took place on another day, but again with the same class of students.

A balanced testing set of 20,000 images was built from the current hold-out lecture. A stratified training set was built using 60,000 images from the remaining two lectures. A stratified validation set was built using 100,000 images from the hold-out lecture.

On the balanced test sets, the networks achieved accuracies of 63.4%, 64.6%, and 61.4% for each hold-out lecture. On the validation sets the network achieved 72.7% (chance 65.9%), 69.12% (chance 59.4%), 61.1% (chance 55.4%) with κ values of 0.199, 0.239, and 0.126 respectively. Over all the validation sets, this yields an accuracy of 67.6% (chance 60.2%) with $\kappa = 0.187$. The confusion matrix is provided in Table 8.7.

Table 8.7: CNN Confusion Matrix for Cross-Lecture Validation (Interest)

Actual	Predicted	
	¬Interested	Interested
¬Interested	33,719	49,202
Interested	47,964	169,115

8.5 Acceptance

8.5.1 Respondents

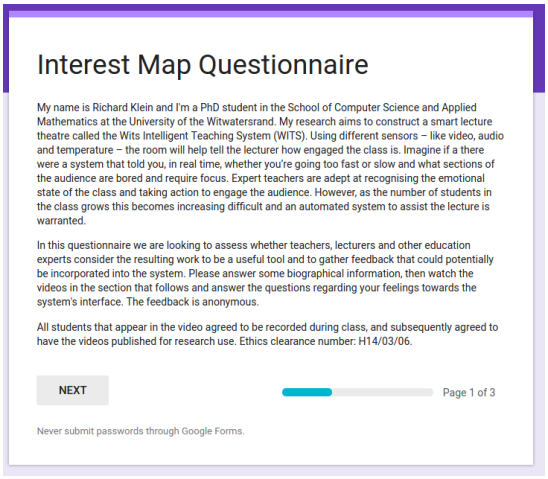
To assess the acceptance of this technology, two videos[1] displaying an Interest Map were generated. The CNN from Section 8.4.3 was used to generate the Interest Map. The accuracy of the classifier is inflated from what it would be on an unseen class, but it was done this way so that an accurate output could be generated to show to lecturers and other teaching experts at the institution. The intention was to get feedback from lecturers and professors regarding the use of an interest map, not whether they thought the interest map was accurate.

A questionnaire was created using Google Forms and a link sent out to staff across the five faculties at the university. The first page of the questionnaire had an introduction of 233 words that introduced the author and briefly explained the idea of the system. This introduction is shown in Figure 8.5a.

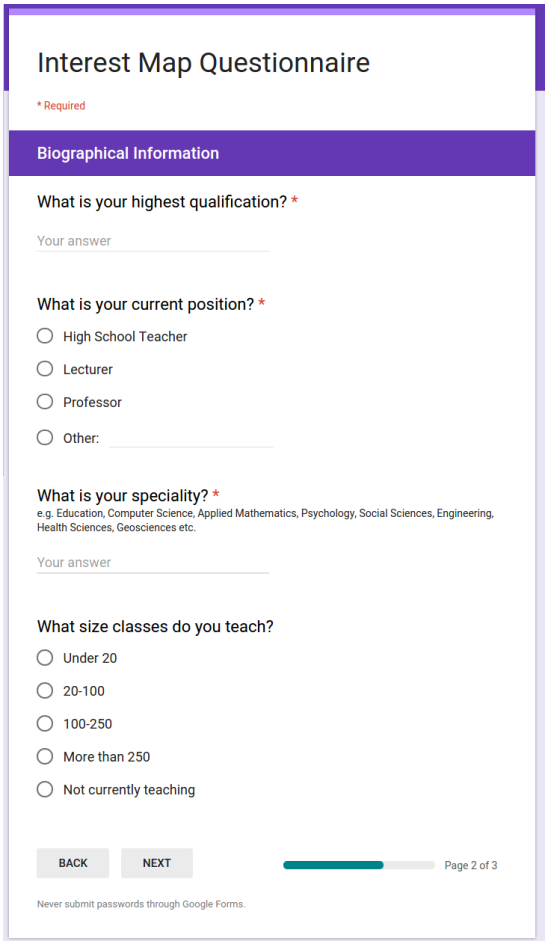
After the introduction, the respondents were asked about their qualifications and what class sizes they taught (Figure 8.5b). Following that they were shown the videos and then asked about their impression of the system (Figure 8.5c). The full set of anonymised results is provided in Appendix A.

Overall, there were 131 respondents across the five faculties at the institution. The respondent disciplines per faculty are broken down in Table 8.8. Of the respondents, 74 (56%) hold specialist or doctoral degrees (PhD), 49 (37%) hold Masters degrees (MSc, MA, MCom, LLM), 7 (5%) respondents had 4-year undergraduate degrees (BSc, BA, BCom) and 1 (0.7%) had a diploma (Figure 8.6a). There were 43 professors (33%) and 82 lecturers (63%), while the remaining 6 (5%) respondents were registrars that have teaching experience, academic developers with teaching experience, or computer technicians involved with e-learning (Figure 8.6b). 50% of respondents teach classes that are larger than 100 students, with 25% of all respondents teaching classes larger than 250 (Figure 8.6c). This information is summarised in Figure 8.6 on page 163.

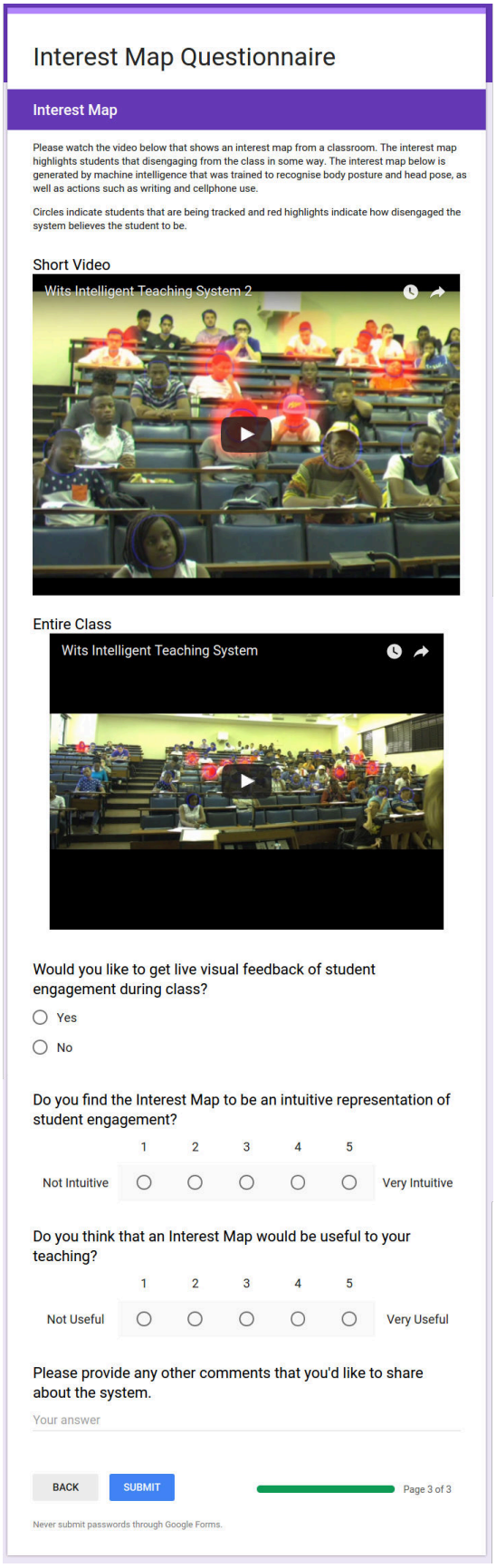
[1] https://youtu.be/rV_Ax8Q6yL0
https://youtu.be/7m5vyxKK1Oc



(a) Introduction

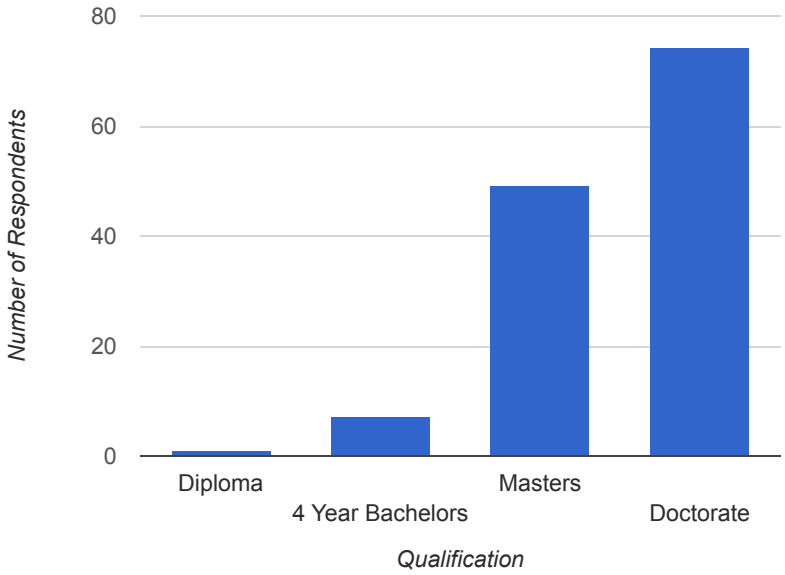


(b) Biographical Questions

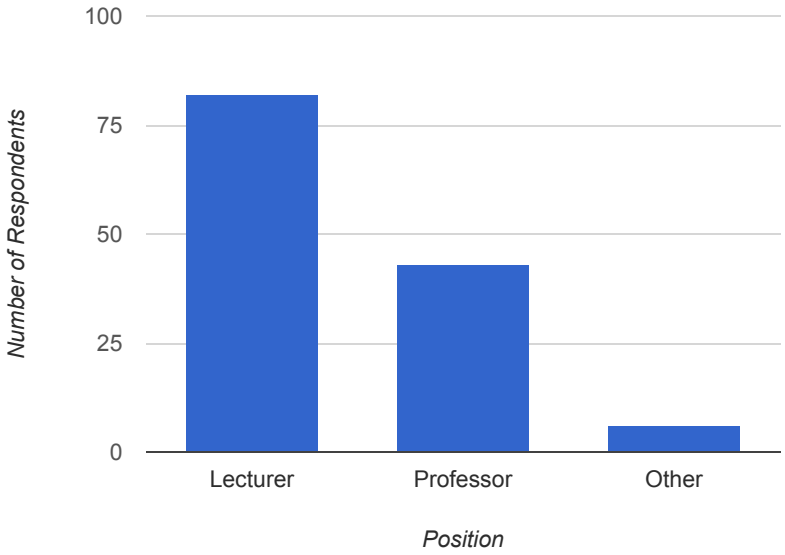


(c) Interest Map Response

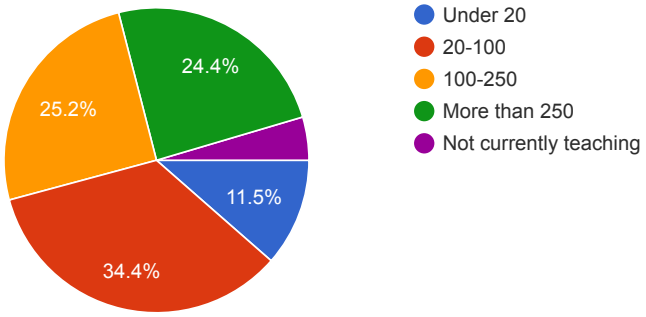
Figure 8.5: Interest Map Questionnaire



(a) Respondent Qualifications



(b) Respondent Position



(c) Largest Class Size

Figure 8.6: Respondent Information

Table 8.8: Respondent Disciplines

Humanities (14)	**Health Science (50)**	**Science (48)**
Arts	Biological Anthropology	Actuarial Science
Education	Clinical Coding	Applied Mathematics
English	Clinical Medicine	Archaeobotany
Film	Dentistry	Archaeology
International Relations	Epidemiology	Astronomy
Italian	Family Medicine	Biological Sciences
Linguistics	Nursing	Chemistry
Literary Studies	Occupational Therapy	Computer Science
Performance	Pathology	Environmental Science
Psychology	Physiotherapy	Geosciences
Education	Psychiatry	Mathematics
Social Sciences	Public Health	Physics
Sociology	Surgery	Pure Mathematics
Theatre	Unspecified Health Science	Statistics
Engineering (10)	**Commerce (9)**	Science Education
Aeronautical Engineering	Accounting	
Electrical Engineering	Information Systems	
Mechanical Engineering	Insurance & Risk Management	
Industrial Engineering	Law	
Unspecified Engineering	Management	

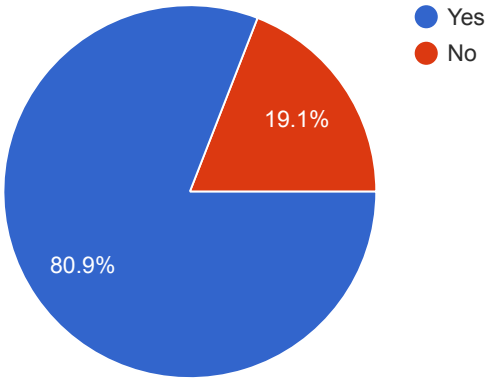
8.5.2 Response to the Interest Map

After watching the videos of an interest map of a class, the respondents were asked to indicate their attitude towards such a system. When asked if they would like to get live visual feedback of student engagement during class, 106 (81%) said yes and 25 (19%) said no.

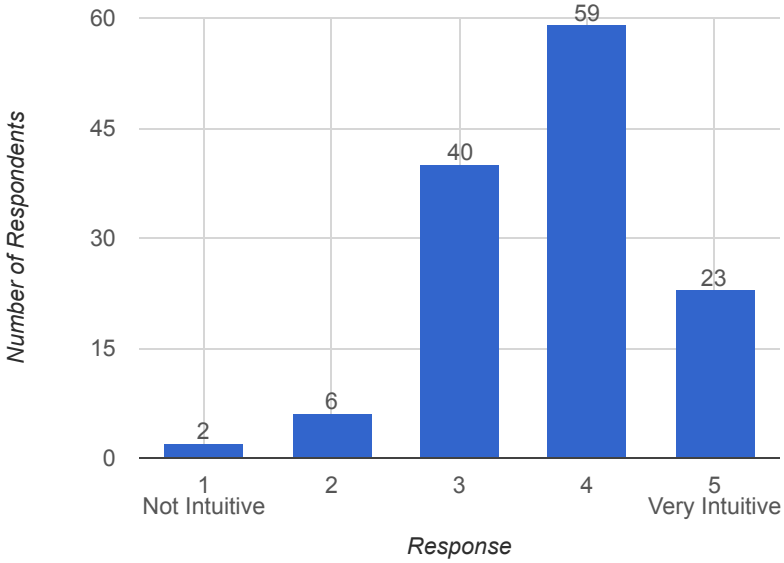
82 (63%) respondents indicated that they found the Interest Map to be 'Intuitive' or 'Very Intuitive' with only 8 (6%) reporting that they did not. 86 (66%) respondents indicated that they believed such an interest map would be useful to their teaching while 16 (12%) said that it would not be useful. The rest of the responses were neutral. These responses are summarised in Figure 8.7 on the next page.

8.6 Respondent Comments

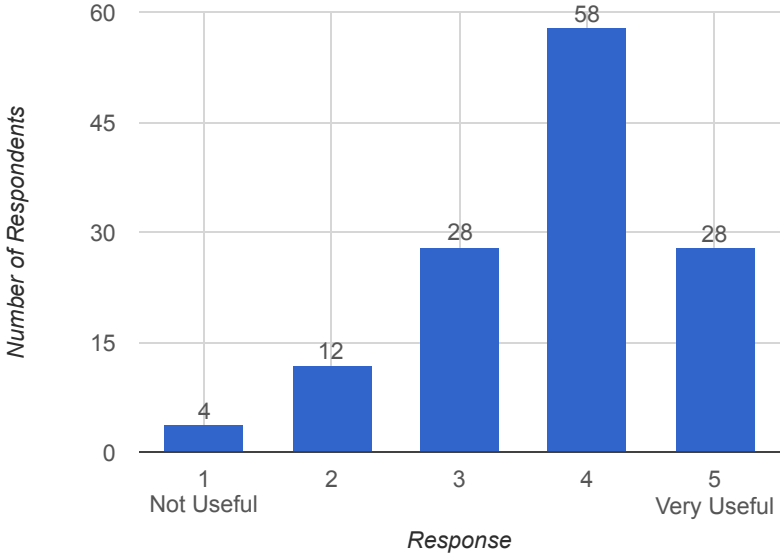
The final part of the questionnaire was open-ended and asked the respondent to provide any other feedback they might have about the system. Some representative comments from respondents that addressed various aspects of the system are provided in the sections that follow. The full set of comments are provided in Appendix A.



(a) Would you like to get visual feedback of student engagement during class?



(b) Do you find the Interest Map to be an intuitive representation of student engagement?



(c) Do you think that an Interest Map would be useful to your teaching?

Figure 8.7: Responses to the Interest Map

> “Richard, this is truly a fascinating opportunity to get “feedback” from students over which they have little “control”, which can highlight for lecturers how what they are doing is genuinely being received. Some comment / questions that spring to mind which may be useful for you to consider are: this ‘pilot’ has been done with a small cohort of students, yet your intention is that it should be used in large classes. If it operates in real time for a lecturer, how would you plan to give supportive feedback for someone who is just dismayed by the levels of disengagement in large classes, for example chemistry 1 where the disparity of prior knowledge and ability in students is known to be huge? How would you be able to use this to support the development of teaching proficiency, and keep it out of the hands of managers who are on the prowl for evidence of “poor teaching” - recognising that you can take a horse to water but you can’t make it drink? There is HUGE research potential in this, in tracking what is going on between student and lecturer and what it is that engages contemporary students . Happy to chat further and keep in touch. [NameRemoved]

Figure 8.8: Comment from a Teaching & Learning Expert

8.6.1 Comments from a Teaching & Learning Expert

This first comment considered is shown in Figure 8.8 and was provided by a teaching and learning expert that works in an academic support unit. The respondent holds a Ph.D in science education and touches upon a number of aspects that are now briefly discussed.

The respondent notes that this is a good way to get objective feedback about the quality of lectures. Often when students fill out lecturer evaluations, they are seen to be a comment on how much the students like the lecturer rather than the quality of the lectures themselves. A system like WITS can be used to objectively track and analyse trends in one’s teaching without using student feedback that is often biased by their opinions of the lecturer.

Next, the respondent notes how the project introduction refers to potential use in large classes while the videos show only a relatively small section of a class. As each student is monitored independently, scaling up to a large class only requires having enough correctly placed cameras and enough computational power in the room. In its current form, the system lends itself to data parallelism where each compute node monitors a portion of the class. The calculated interest values could then be merged into a single interest map for display. As the number of students that are tracked increases, the relative importance of errors for individual students decreases. So the more students the system tracks, the better the system-wide feedback to the lecturer will be.

Another important point is raised by the respondent relating to demoralising lecturers and performance management. Consider some potential causes of disengagement:

1. The lecturer is boring.
2. The content is boring.
3. The student understands the work but the lecturer is moving slowly to cater for other

students in the class.

4. The student does not understand the work and has given up listening to the lecturer for this topic.

5. The student is tired, has a test coming up in another subject, has personal problems that are distracting.

6. The lecture is at an inopportune time like first thing in the morning or last thing on a Friday afternoon.

7. The lecture venue is uncomfortable, the chair might be skew, and the lighting, airflow, or acoustics might be wrong.

There are many reasons that a student may be disengaged, and the system is unable to tell these apart. In some cases it may well be the lecturer's fault in that the presentation is not well put together, but in others the distraction may have nothing to do with the lecturer's presentation itself and rather be a symptom of something else. Those using the system should consider its purpose not to show that so many students are disengaged, but rather to provide information with which the lecturer can scaffold their next move.

A lecturer presenting to disengaged students is not necessarily a bad lecturer, but consider that a lecturer that does not do anything about it might be. As such, the data reported by the system should be explained to and carefully considered by those using it. Providing carefully worded hints to lecturers rather than showing the raw interest information may ultimately provided a better experience and avoid causing feelings of hopelessness for lecturers. Such an Automatic Teaching Assistant (AutoTA) is indeed the ultimate goal, but requires a working classification system first.

Finally, the respondent alludes to the potential for future research into student behaviour supported by the system. WITS provides large opportunity to study how students interact with lecturers, how they engage and disengage, and in general how they behave during lectures. This should indeed be the focus of future work.

8.6.2 Performance Monitoring

A number of other respondents touched on the fact that the system could be used for performance monitoring as well. It was generally seen in a positive light, although this should be approached with extreme caution. As mentioned in Section 8.6.1, there are many factors that affect engagement that are beyond the control of lecturers. For example, a student thinking about personal problems, or one that was at last night's vodka party is unlikely to engage regardless of what the lecturer does. An analysis of the system's output should always consider the class as a whole or at least groups of students together, and even then, the changes in interest values over time are probably more important than the actual values themselves.

It is recommended that the system is not used for performance appraisals at least until common patterns and responses are properly understood.

> “ Excellent idea - this could be incorporated into lecturer performance appraisals as well to support how well some lecturers get students to engage.

> “ The tech seems more useful for performance monitoring of teaching, or to remote monitor classroom settings. It might also be useful for new lecturers who are only starting out and wanting to review their teaching performance in class sessions...

Figure 8.9: Comments Regarding Performance Monitoring

8.6.3 Summative Feedback

A number of respondents – such as those in Figure 8.10 – said that they would like to be able to see summary statistics to get an overall impression of the lecture. This would be useful to see how the engagement levels have changed over some time period. The graph could be generated in a way similar to those created by Engage in Chapter 4. This allows the lecturer to compare current engagement levels with levels earlier on in class, or even from other classes. The lecturer should be able to annotate the graph so that one can understand how some topic was received.

Even with this information, it is useful to still have the 2D interest map so that the lecturer can get a sense of the spatial spread of engagement as this helps make informed decisions about what the best course of action is. For example, if half the class is disengaged it is still important to be able to distinguish between the case where it is the whole left side of the class, or if it is every second person distributed evenly. The former might indicate a venue problem or that the lecturer ignores the one side, while the latter may indicate that generally the students are getting tired regardless of where they sit and who their neighbour is.

The specific summative measures that lecturers find useful should be investigated and implemented into the production system.

8.6.4 Definition of Engagement and Accuracy

A number of respondents commented on the accuracy of the system. Sample comments are presented in Figure 8.11. Specifically, a number of respondents disagreed with the definition of interest that the system had learnt. Some pointed to misclassifications, but most focused on aspects that the system does not currently monitor or students that the system considered to be occluded (for example the student with the white t-shirt from the last comment in Figure 8.11). The definition of engagement presented in this work is provided as first step towards designing and learning more general rules. The definition is limited by the data that is currently available to train the system.

Lecturers should be surveyed to establish an agreed upon definition and those using the

> “ Is there no way to condense the videos so an overall impression can be made?

> “ I would like if it has a quantitative measure of responsiveness from the students.

> “ Need to summarise the system. E.g., proportion of class currently disengaged, particularly dense areas of the class disengaged, what part of the lecture timewise had the highest disengagement. As it stands, the red hot spots are difficult to interpret from a total class perspective.

> “ Incredible idea! This technology could be a methodological tool in determining ideal break times or lecture lengths, even cohort specifically! I would be very interested to view a percentage/graph of ”attention” alongside a lecture itself, possibly allowing an accurate reference to where most were engaged or not.

Figure 8.10: Comments Regarding Summative Feedback

system should be aware of what the system can and cannot measure. One respondent noted that it would be useful to see, perhaps using colours, why the system was labelling a student in a certain way. Using more colours allows the interest map to differentiate between disengaged students that are talking and those that are sleeping.

In future work the definition should be expanded by including more proxies that can contribute to a final interest value and generally to build a more reliable dataset of interest.

8.6.5 Teachers Do This Already

A number of respondents felt that they were already able to accurately assess the level of engagement in the class. Most such responses were on smaller classes, although even when the class sizes were large there were lecturers who felt they did not need the technology. Consider the comments shown in Figure 8.12.

A large difficulty here is that we all think that our intuition is correct, but have no real way to test this *in-the-moment* intuition until one goes back and reviews the students after the class. From experience manually labelling the dataset, students are always less engaged than originally anticipated – they try not to get caught when being mischievous and sometimes they’re quite good at it. Indeed, there are probably many teachers that can keep track of everything, but when training new teachers or testing and improving one’s own skills, it is important to be able to look back and analyse data objectively.

“ System misses some students who are patently disengaged - needs to be tuned up somewhat for better accuracy.

“ It would be useful to get the definition of ”engagement in class” as far as this system is concerned. I observed a student who was fanning himself almost throughout, not taking down any notes, and yet the system identified him as ”engaged”. On the other hand, a student sitting at the front of the class, who seemed to be engaged, was marked as ”not engaged” for a few seconds when she turned away to look at something/someone in the audience - maybe listening to an answer or a question from a fellow student? From my experience, I have had a student who would be looking at the front of the class (correct posture, eyes open, etc.) but could be miles away, and upon asking them a question, only then do you find out that they were not following anything in the lecture at all. I found the system however very good in picking out those who were asleep, or otherwise not looking at the front of the class/had their earphones on.

“ For me, the greatest weakness of this system, is that it seems to record a lack of engagement when students look down. This is particularly noticeable in students taking notes. I agree that eye contact often implies interest, but there was also one candidate, who seemed to be asleep looking at the lecturer. the system identified the individual as interested throughout, but I am not convinced that he actually was.

I think there is an amount of resolution loss towards the back of the room (smaller heads).

My comments notwithstanding, I think the system is very cleverly coded and even if some entropy exists on the individual level, as a system, I think it gives an amazing insight into the systemic dynamic in a class.

“ The student with white T shirt - left in frame 3rd row was also disengaged and system did not identify him. A lecture identify these students. this system is however good for feedback to the student if one follows that up with the student in order to identify problems early on

Figure 8.11: Comments Regarding Accuracy

> “I do feel that I can tell how engaged a class is on my own assessment even though as many as 350. But I may need to see a different view! I might be surprised.

> “I think the system is very helpful in helping me with my blind spots or selective observations and selective recollections.

Figure 8.12: Comments Regarding Teachers' Abilities

> “It would be useful as a 'self-evaluation' tool after lecture, but it would be distracting to be getting the information live while lecturing.

Figure 8.13: Comment Regarding Lecturer Distraction

8.6.6 Lecturer Distraction

In contrast to some of those asking for more colour and detail, some respondents felt that live feedback during the class would be too distracting for the lecturer. One such a comment is shown in Figure 8.13.

This may indeed be the case, particularly with inexperienced and nervous lecturers. Investigations into the effect of this information on lecturers should be established. Once a live system is working, pilot studies should be done to assess how much or how little information should be provided during the class. There should be enough information to inform decisions during class, but not so much that the lecturer is overwhelmed.

8.6.7 Student Privacy, Discomfort and the Hawthorne Effect

While some comments related to lecturer distraction and discomfort, a number of comments considered student discomfort as shown in Figure 8.14. The classes that took part in the re-

> “. . . it could however give ethical problems and 'violate' students privacy . . .

> “. . . based on the Hawthorne effect, just monitoring it will also change student behaviour.

Figure 8.14: Comments Regarding Student Discomfort

search were given multiple ways to opt out of being recorded. In early recordings, a substantial section of the class was not covered by the camera, or if a student still wanted to sit in their normal place they could opt out by lifting their book or a QR code over their face briefly at the start after which they could be blurred from the recordings. The students agreed to the terms before recording was started and throughout a semester of recordings, not a single student ever opted out. In many circumstances the students pointed out that there were already CCTV cameras in some venues.

The Hawthorne effect (Adair 1984) is the change in a subject's behaviour caused by the knowledge that they're part of an experiment. As the students know they are being recorded, it potentially alters their behaviour during class. Once the students had agreed to be recorded, the camera was placed at the front of the class. Initially the students were very aware of the camera, indicated by the number of times a student looked at the camera. By the fourth lecture with a camera, students were visibly more relaxed and stopped looking at the camera completely. At this stage, students were completely comfortable with the camera: glances in its direction had stopped, and cellphone use, nose scratching, and sleeping resumed. Anecdotally, when asked about how they felt about the recordings during consultations, the students reported they were neither directly aware nor uncomfortable any more. In anonymous lecture evaluations at the end of the course the camera was not mentioned at all.

This initial work indicates that the recording does not adversely affect the students after a few encounters, particularly when they realise they will not get in trouble and that the lecturer is not specifically looking through the video to catch them doing something bad. However, the full impact of the cameras on the students should be formally studied in future work.

8.6.8 Resolution

Resolution was mentioned a number of times, often in conjunction with accuracy. Poor resolution of students seated at the back of the class is indeed a problem, but it could be solved through the use of multiple consumer-grade cameras focused on smaller sections of the class. This will also provide more frontal views of the students and if mounted correctly could minimise occlusions as well.

8.6.9 Display Colours

> “ Instead of shading the face of the students, change the colour of the ring from blue (I prefer green) to red and thickens the border. 2. Use additional colours for measuring the intensity of the engagement i.e. colour between blue (green) and red...

Figure 8.15: Comment Regarding Display

A number of respondents indicated that they would prefer multiple colours, like green and red, to indicate interest and disinterest. The software already supports this and it can be set before displaying the Interest Map.

8.6.10 Temporal Considerations

> “During a 45-minute [lecture] you expect every student, even the most attentive, to disengage for short periods. I think a disengagement time threshold (possibly user set) would be useful.

Figure 8.16: Comments Regarding Interest Over Time

Currently, the system shows an interest value based on the last 1 to 4 frames. This means that the display at any time is effectively live. This has two main implications: 1) small insignificant lapses in engagement are always displayed, and 2) any disengagements when the lecturer is not watching the screen will be missed. The user should be able to adjust how much time is considered in the calculation of the interest value – short lapses in attention are expected and should be ignored, while longer ones are then highlighted more predominantly. This could be easily achieved by incorporating some type of moving average such as those in Equations (8.1) and (8.2).

$$V(s,t) = \alpha \cdot V(s,t-1) + (1-\alpha) \cdot I(s,t), \tag{8.1}$$

$$V(s,t) = \frac{1}{n} \sum_{i=t-n}^{t} I(s,t), \tag{8.2}$$

where $V(s,t)$ is the value used for the interest map display, $I(s,t)$ is the interest value predicted by the classification system, s is a subject, t is the frame number, $\alpha \in [0,1]$ is a parameter that sets the importance of previous frames for the exponential average and n is the number of previous frames to consider in the average of a sliding window. Using such an approach, the system can be forced to vary more slowly over time, ignore brief disengagements, and rather focus on those that show disengagement over time.

8.7 Conclusion

This chapter used the proxy behaviours labelled in WITSDB (Chapter 5) to define an approximation of interest. WITSAT was used to extract labelled image sequences and students that exhibited the relevant characteristics were labelled as Not Interested while the others were considered Interested. Using the computer vision and machine learning technologies considered in

Chapter 2 and tested in Chapters 6 and 7, a classifier was built to detect students that are interested and to produce a *Interest Value* score. This interest value is then displayed as a heat map or *Interest Map* that can be used during class by presenters to better understand the affective state of their audience.

Again, the CNN out-performed the HOG SVM in both absolute accuracy and generalisation. The CNN performed very well over cross-frame validation. When considering the cross-subject and cross-lecture validation, the system shows similar results to those of the previous chapters – the CNN performs worse on students it has not seen before, but still out-performs chance and the HOG SVM.

This section also showed that by bootstrapping the classifier with even a small proportion of frames from a new subject, the accuracy rises substantially. The system is able to classify new subjects very well after seeing only a few frames from that subject. This means that the introduction of a short calibration step could be introduced to drastically improve performance.

The current definition of interest is limited by the labels available in the dataset. The dataset should be expanded with new labels to better represent the behaviours of students as they (dis)engage with lecturers.

A video of the interest map was provided to lecturers and professors at the university. The lecturers overwhelmingly supported the idea of such a system with the vast majority (81%) of respondents saying that they would like to have live feedback about the state of their audience during class. The respondents found the interest map visualisation scheme intuitive and effective, but many noted that they would like to be able to access summary data as well. A number of lecturers felt that it might be distracting to get the feedback live during class, but noted that they would make use of a system that provided review capabilities afterwards.

Chapter 9

Conclusion

9.1 Summary

Engagement is a central issue in education and impacts student learning significantly. 'Good teachers' perform contingent teaching and are able to scaffold their classes around a theme, but go off-script in order to cater to the current state of the students. This type of teaching is difficult in ideal circumstances, but becomes progressively harder as class sizes and venues grow. Large class lecturing is often criticised as lecturers become less responsive to individual student needs.

While expert teachers develop an intuition regarding the state of the class, this takes time to establish, is highly subjective, and does not allow for systematic self-evaluation. At the same time, new lecturers lack experience and therefore the intuition necessary to effectively track what is happening in the class. Over and above the issues relating to intuition itself, practically, even experienced lecturers cannot keep track of the class while their back is turned or when students actively hide negative sentiments due to social display rules.

Overall the problem is two-fold: 1) as a lecturer it is difficult to keep track of all the students in a large class, and 2) different groups of students emerge with competing needs.

This thesis proposes the WITS INTELLIGENT TEACHING SYSTEM (WITS) to help deal with the former, and proposes that WITS be used to assess strategies to help deal with the latter. A system like WITS provides lecturers and presenters with feedback about the affective state of their audience. With this information, the lecturer can then moderate their teaching style, speed, content, venue, or schedule to help the audience better engage with the concepts under discussion.

In Chapter 4, ENGAGE, was developed to allow students to self-report their engagement levels during class for the purpose of live, formative, lecturer feedback. This information would be presented to the lecturer using simple graphs that updated in real-time. In spite of reporting that students found the lecturers responsive to their feedback, the uptake of the system was extremely low. When the system allowed students to see the graphs updating the uptake increased, but remained only a fraction of the entire class. It is unclear to what extent the students using the system provided accurate feedback, but at 10% uptake it is unlikely to be representative of

the entire class. It appears that there is also a bias in the type of student that would actually consider using the system. The group of students that are prepared to self-report their engagement overlaps substantially with the group of students that are prepared to come to lectures in the middle of violent student protests. These traits indicate a high level of commitment to their classes and if the students are reporting engagement they are at least aware of it and may self-regulate to some extent. This raises the question of whether these are the students that require re-engagement in the first place, and whether it is pedagogically sound to make decisions based on their feedback alone. The remaining students are either unwilling or unable to report this information. In either case, the intrusive approach fails to gather sufficient data to be valuable to the presenter.

A non-intrusive approach is therefore needed. WITS aims to detect and provide engagement information non-intrusively through the use of multiple sensors around the lecture venue. These sensors provide data to a central system that, based on machine learning techniques, monitors audience interest. A computational pipeline for live audience monitoring and lecture feedback is proposed and a computer vision based subset thereof is tested in this work. The system and the pipeline are discussed in Chapter 5.

The first step in the creation of such a system is the collection of data. No such dataset existed, so one was created. Originally it was hoped that Engage could be used to help build a self-labelled dataset, but due to the low uptake by students this was not possible. The WITS ANNOTATION TOOL (WITSAT) was created and released to facilitate the video annotation process. Psychology, Computer Science, and Applied Mathematics honours and masters students assisted in the labelling process over two separate labelling sprints of one to two weeks each. At first, raters were asked to label students along affect axes involving boredom, interest, confusion, and frustration. Minor guidance was given, but not in the form of an observational checklist. The data returned by both the Psychology students in the first sprint, and the Mathematical Science students in the second sprint had extremely low inter-rater reliability scores. It was decided to abandon the process of training directly on interest labels and rather focus on the construction of an observational checklist that could be used to infer interest and engagement, or lack thereof.

Undergraduate students across 3 lectures were labelled with a number of classroom actions, postures, and head poses. These labels make up the WITS DATABASE (WITSDB), which was released for research purposes. WITSDB consists of about 65–80 unique subjects labelled across three lectures each with about 10 different labels. Its creation was an arduous process and it is hoped that its release will support future research in this area.

Chapter 6 focuses on the application of Support Vector Machines (SVM) and Histogram of Oriented Gradients (HOG) to classify the labels in WITSDB. The experiments found that while the HOG SVM was good at recognising similar frames, it was unable to generalise to unseen students. This is evidenced by the vast difference in the accuracies of cross-frame and cross-subject validations. Generally the HOG SVM performed equivalent to or worse than chance when performing cross-subject validations. More careful feature engineering or different features entirely may improve the accuracy of the SVM approach. Background removal and image segmentation may also provide large performance gains. In spite of this, the SVM's memory requirements and computational complexity make it difficult to train on very large datasets. In general, the HOG SVM approach appears unsuited to the task at hand.

Chapter 7 presents the application of Deep Convolutional Neural Networks (CNNs) to accomplish the same task but with significant accuracy, generalisation, and performance improvements. WITSAT was extended to interface with both libSVM and Caffe to allow classification using SVMs and CNNs. The performance boost is entirely due to GPU hardware acceleration using CUDA. A parallel CPU HOG SVM is able to classify 50 subjects at 1.6 frames per second while the CPU CNN did the same in 0.23 frames per second. However, once accelerated by CUDA, the CNN was able to classify the same 50 subjects at 7 frames per second which is almost real-time for the current recordings. The current system does not optimise the transfer of data over the hardware bus that connects to the GPU – this is a known bottleneck and preliminary tests indicate that with sufficient parallelism and general memory optimisation, a real-time system is well within reach.

Computational performance aside, the CNN outperformed the HOG SVM on all tests that were run. While it is able to scale to significantly larger datasets, even on similar size data sets the CNN outperforms the SVM. The CNN was able to classify the writing and cellphone actions at about 10% better accuracy than chance and the HOG SVM when examining generalisation to unseen subjects.

Finally, in Chapter 8 the proxy labels in WITSDB were put together in an observational checklist to build a dataset of student interest. Both the SVM and CNN based systems were trained and achieved peak accuracies of 71% and 93% respectively in cross-frame validation. When considering generalisation to unseen subjects the SVM performed at 54% and the CNN at 60%.

Further tests also indicate the CNN requires only a small number of frames before it is able to generalise to previously unseen subjects. This means that the introduction of a calibration step at the start of the process can drastically improve the system's performance. When bootstrapping the system by including a few frames from the start of the lecture, the accuracy of the CNN increased to 88.8% which is an extremely high accuracy.

Using Platt's method for SVMs and Softmax for CNNs, an interest probability or *interest value* is generated. The interest value is overlaid on the video as a heat map or *interest map* for visualisation. The interest map for a lecture was generated and shown to lecturers and professors at the university. The overwhelming majority, 81% of the 131 respondents, said they would like to have live interest feedback during class and that they would make use of such a system.

9.2 Accuracy and Generalisation

Aspects such as race, gender, clothing, handedness, and viewing angle play an important role, as does their representation and variation in the dataset. High cross-frame validation accuracies, even when trained on relatively small proportions of the overall data, indicate that only a few frames of a specific subject in a specific lecture are required to perform well, indicating low intra-subject variation. This conclusion is also supported by the fact that in general, lecture 1 data could be used to accurately predict lecture 2 data, but often failed with lecture 3.

This information, coupled with the poor to fair cross-subject validation scores, means that

when building future datasets for this purpose, it is more important to record a larger number of different students with different clothing and backgrounds for short periods of time, than to record fewer students for longer periods (as was done in this work). Doing so forces the system to focus more on inter- rather than intra-subject variation.

Other than the need for an increased number of different students and backgrounds in the recordings, the under-representation of a number of useful labels is a problem. For example, when a student is looking up at the roof, they are certainly disengaging from the lecture. However, in natural circumstances, this occurs infrequently and surprisingly few students perform this action. This means that there were not enough instances of students looking up to be able to adequately classify it, even though it appears to be an extremely valid feature to use in an observational checklist.

Literature indicates that for the purpose of direct affect labelling, it is best to use natural data rather than posed data. For this reason, the students were recorded naturally in lectures. However, once the raters failed to provide sufficiently reliable affect labels, the students were rather labelled according to the actions and poses identified. For this reason the natural videos were used as training data. Based on the low intra-subject variance, it likely that for postures and actions, posed data will provide a meaningful way to collect more balanced and reliable data.

In light of these findings, students should be recorded and asked to write, use their cellphone above and below the desk, to look at each other and specific areas of the room, and to assume a number of common upper body postures. It appears that collecting lots of short sequences over many days, venues, and subjects will provide better data than considering natural videos taken during lectures. With more subjects in different venues (different backgrounds), and more balanced datasets, generalisation to new subjects should improve as the networks are forced to learn features that work across subjects rather than just the few in the current dataset.

9.3 Computational Performance

As it stands currently, the system is able to handle up to 50 students per frame at 7 frames per second when using an NVIDIA Titan X graphics card. With a restructuring of the code, it is believed that this number can be improved as the code does not currently optimise the transfer of data between the main memory and the graphics card. Preliminary tests indicate that by doing so the expected classification time of the system can be decreased by 50 times.

However, even if this were not the case, the problem in general is embarrassingly parallel – different subjects can be classified completely independently. Once each subject is classified, the data can be merged to form a single interest map to show to the lecturer. In either case, the path to real-time performance is clear.

9.4 Future Work

9.4.1 Engage

Overall there was a very low percentage of the total class reporting back with the system. Future work can focus on methods to increase uptake during classes, and to analyse the impact on concentration that the program has. Specifically, using the system as a traditional clicker for voting can help identify students that are willing to use the system, but do not feel the current captions describe them. At the same time, using the system for voting before trying to get the students to report engagement may get them more excited about trying out the system. The system is currently capable of handling voting, but this was implemented based on feedback from the original system and has not been tested with students. At that time it was infeasible to run more experiments with the students due to ongoing protests followed by exams. The effect of voting on the system's uptake should be examined in depth.

Another avenue of future research using Engage should analyse the bias inherent in the system based on the type of student that would use it. The numbers in Chapter 4 hinted that the students using the system originally were probably the same students that used it in the lectures that took place during the protests. It would be useful to know whether a general trend emerges that may indicate that the students that use the system are not actually the students that the lecturer should be primarily focusing on engaging. These students may already have enough self-awareness to regulate themselves.

There is scope for future work in automatically analysing the textual comments using natural language processing. This kind of automated analysis would allow the system to build its own model of the kind of comments posted by the students (off topic, questions, comments, etc). Textual analysis might even be able to monitor the sentiment based on the comments alone.

A button allowing students to report *concentration and flow* may be useful to asses uptake, but this will certainly distract more students in the long run. Research into the accuracy of the reported data should be performed.

9.4.2 WITS Database

Currently the dataset is limited by a number of factors. There are only three labelled lectures, all with roughly the same group of students. To improve generalisability across new students, the system should be able to train on as many different students as possible. This work has also shown that relatively few representative samples of a subject are required. In this light, the database should be expanded to include many shorter labelled segments of lots of different students, rather than labelling single subjects across entire lectures. There should also be a focus on balancing the representation of subject demographics across gender, race, and handedness.

New recordings should limit the horizontal angle from which subjects are viewed by placing multiple cameras that each monitor smaller areas, rather than a single camera at the front. Cameras should be mounted in a way that minimises occlusions.

The collection of less common classroom actions and postures should be a priority. Including yawn and raised hand information in the system may provide significantly more information about the student, even though it may be a challenge to collect images of these less common actions. Posed data can be considered, but generalisation to real data should be verified.

Collecting affect labels directly would be an interesting way to verify the correctness of the observational checklist. However, work needs to be done to establish how to get raters to accurately label these aspects without necessarily resorting to the extremely expensive process using expert Facial Action Coding System (FACS) or Baker-Rodrigo observation method protocol (BROMP) coders. Presenting two subjects at a time and asking raters which one is more engaged may be a better way to establish a ranking among short video segments. This information could then be merged to try build a 'ranked scoreboard' of interest that could be used for training. This process may provide more reliable data and should be considered.

Including other labels in the dataset means that the observational checklist used to 'define' interest can be updated to better reflect what experts consider to be the signs of engagement. Work should asses what expert lecturers look for when assessing students and whether these aspects can be measured by the system.

9.4.3 Classifiers

HOG SVM

The HOG SVM approach could not generalise to unseen students during cross-subject validation. Compared to the high validation accuracies across frames, this indicates that the system was learning to recognise similar frames rather than learning the structure of the action or posture in question.

This may be due to complicated backgrounds with too much noise in the gradient features. Future work should consider emphasising foreground information by first applying a background removal or segmentation processes. Focusing on more relevant parts of the image may also help. For example, writing may be better recognised by looking only at the desk where books are placed. Using a different feature descriptor entirely may improve the results as well.

CNN

Overall the CNN performed significantly better than the HOG SVM and should be the focus of future work. Trying architectures other than AlexNet should be a focus in order to see if there is room for improvement. Currently, all the networks were trained on single labels – writing, cellphone, posture etc. In a true deep learning style, an architecture that incorporates all of these aspects should be considered. For example, if a common set of convolutions feed into a number of layers that allow the detection of posture, head pose and actions, these layers can then form the basis of a layer specifically designed to assess interest. In doing so, a single network is able to provide both the interest score and underlying reasoning to the lecturer.

9.4.4 Interest Map

Currently the definition of interest used in the interest map is based on the available labels in WITSDB. Work should be done to establish whether expert lecturers agree with the current definition and if not, what aspects should be added or removed. This work showed that lecturers are generally supportive of using a system to produce live feedback about the audience. The system should be run in a number of live classes with lecturers to establish whether this enthusiasm translates into lecturers actually making use of the system. Aspects relating to the lecturer's state of mind should also be considered to assess whether WITS distracts and demoralises lecturers, or whether it legitimately provides insight that they start to incorporate into their classes.

As mentioned previously, the ultimate goal of WITS is to have a real-time system to provide feedback during lectures. The current processing pipeline is very close to real-time and with an efficient, parallel re-implementation it should be possible to build a system that runs fast enough for live lecture feedback. Finally, once the system is able to perform in a live environment with acceptable accuracy, its effect on both students, lecturers and ultimately teaching should be investigated.

9.4.5 Automatic Teaching Assistant (AutoTA)

Work should be done to generate a number of interest maps across multiple classes, disciplines, venues, lecturers, and students. Different strategies should be tested and the resulting changes in the interest map recorded. With a large enough dataset, it may be possible to build an Automatic Teaching Assistant that recommends specific strategies to the lecturer that are known to re-engage students based on patterns identified in the interest map.

9.4.6 Educational Research

Further to the questions already raised, there are a number of applications to general educational research. Understanding how students react to different stimuli and teaching styles is important when considering what teaching strategies work best in the classroom. Monitoring actions, postures, and affect in relation to timing and style may provide useful information to the education community and teachers specifically.

9.5 Conclusion

This thesis provides the foundational work upon which to build a live, student affect monitoring system. It shows that with sufficiently representative data such a system is indeed possible and its realisation is close at hand. It is hoped that in time this work leads to the development of useful tools to improve teaching and learning in areas where it has traditionally been difficult due to practical constraints. The system should be used to produce better teachers, benefit students, and ultimately improve education.

Appendix A

Questionnaire Results

This appendix provides the responses from the questionnaire in Chapter 8. To maintain respondent anonymity the demographic information is presented separately to the Interest Map response questions. In the previous analysis, respondents were categorised into Professor, Lecturer, or Other. For that analysis, equivalent academic ranks were merged – for example, a senior tutor is considered to be a lecturer as the ranks are equivalent but on different career tracks in the university.

Table A.1: Raw Demographic Responses

Highest Qualification	Position	Discipline
BA Hons	Senior Tutor	Applied maths course for first year Commerce
Bachelor of medicine and surgery	Lecturer	Health sciences
BCur (Nursing Ed & Admin et al)	Lecturer	Nursing
BSc	Lecturer	Industrial Engineering
BSc Honours	Lecturer	Actuarial Science
BSc Hons	Lecturer	Mathematics
BSc(Eng)	Lecturer	Aeronautical Engineering
BSc(Eng) - elec	Lecturer	Engineering
Chartered Accountant	Lecturer	Financial Accounting
Diploma	IT Technician	Computer Science
Doctorate	Lecturer	Nursing Education
doctorate	Professor	Physics & Astronomy
FC Path	Professor	Anatomical Pathology
FCP (SA)	Specialist Physician	Health Sciences
FCPath (SA) Anat	Anatomical Pathologist	Health Sciences
FCPath Micro	Specialist Microbiologist	Health Sciences & School of pathology
FCPHM	Lecturer	Health Sciences
FCPSYCH PGDHSE	Lecturer	Psychiatry / neuropsychiatry
FCS (SA)	Lecturer	Surgery

Highest Qualification	Position	Discipline
FCS (SA)	Professor	Health Sciences
Hons - Currently doin MA	Lecturer	Art - Film - Post production
LLM	Professor	Law
M Sc	Lecturer	Health Sciences
Master of Arts	Lecturer	Language; Linguistics
Master of Dramatic Art	Lecturer	Theatre and Performance
Master of Science in Nursing	Lecturer	Nursing Education
Masters	Data Scientist, Reseaercher, Lecturer,	Data science, Demography, GIS, Statistics
Masters	Lecturer	Applied Mathematics
Masters	Lecturer	Health Sciences
masters	Lecturer	theatre and performance
Masters	Professor	Health Sciences
masters	registrar	Dentistry
Masters Degree	Professor	Oral Health Sciences
Masters Degree	Registrar	Public Health
Masters in Italian	Lecturer	Italian Studies
Masters in Law	Lecturer	Law
MBA	Lecturer	Clinical Coding
MBChB	Lecturer	Health Sciences
MCom	Lecturer	Insurance and Risk Management
MMed	Professor	Internal Medicine
MMed (Fam Med) & MBA	Lecturer	Family Medicine
MMed (Family Medicine)	Lecturer	Family Medicine and Primary Care
MMed and PG Dip	Lecturer	Health sciences
MSC	Lecturer	Applied Mathematics
MSc	Lecturer	Computer Science
MSc	Lecturer	Computer Science
MSc	Lecturer	Dentistry
MSc	Lecturer	Electrical Engineering
MSc	Lecturer	Engineering
MSc	Lecturer	Engineering
MSc	Lecturer	Engineering
MSc	Lecturer	Engineering
MSc	Lecturer	Health Sciences
MSc	Lecturer	Mathematics
MSc	Lecturer	Nursing education
MSc	Lecturer	Statistics
MSc	Senior Tutor	Physics Education
MSc	Senior Tutor	Statistics

Highest Qualification	Position	Discipline
MSC (Computer Science)	Lecturer	Computer Science
MSc Degree	Lecturer	Education English Academic Reading/Writing
MSc OT	Lecturer	Health Sciences
MSc Psyc	Lecturer	Social science in medicine
MSc. Eng	Lecturer	Industrial Engineering
PhD	Academic Developer	T&L in Higher Education
PhD	Associate Professor	Information Systems
PhD	Lecturer	Anatomy and Forensic Anthropology
PhD	Lecturer	Applied Mathematics
PhD	Lecturer	Archaeobotany
PhD	Lecturer	Arts
PhD	Lecturer	Biochemistry
PhD	Lecturer	Biological Sciences
PhD	Lecturer	Chemistry
PhD	Lecturer	Chemistry
PhD	Lecturer	Computer Science
PhD	Lecturer	Geosciences
PhD	Lecturer	Health Sciences
PhD	Lecturer	Health Sciences
PhD	Lecturer	Health Sciences
PhD	Lecturer	Health Sciences
PhD	Lecturer	Health Sciences - Molecular Biology
PhD	Lecturer	Health Sciences (basic)
PhD	Lecturer	Health Sciences & Cell Biology
PhD	Lecturer	Humanities and Social Sciences
PhD	Lecturer	Linguistics
PhD	Lecturer	Mathematics
PhD	Lecturer	Physics
PhD	Lecturer	Physics
PhD	Lecturer	Physiology
PhD	Lecturer	Public Health
PhD	Lecturer	Public Health
PhD	Lecturer	sciences
PhD	Lecturer	Sociolgy
PhD	Professor	Accounting
PhD	Professor	Applied Mathematics
PhD	Professor	Applied Mathematics
PhD	Professor	archaeology
PhD	Professor	Biology
PhD	Professor	Biology
PhD	Professor	Chemistry
PhD	Professor	Clinical Medivine
PhD	Professor	Computer Science
PhD	Professor	Computer Science

Highest Qualification	Position	Discipline
PhD	Professor	Computer Science
PhD	Professor	Engineering
PhD	Professor	Environmental Science
PhD	Professor	Epidemiology
PhD	Professor	health sciences
PhD	Professor	Health Sciences
PhD	Professor	International Relations
PHD	Professor	LAW
PhD	Professor	Literary studies
PhD	Professor	Management education; research methodology
PhD	Professor	Mathemaitcs
PhD	Professor	Medical
PhD	Professor	Musculoskeletal physiotherapy
PhD	Professor	Occupational Therapy
PhD	Professor	Physics
PhD	Professor	Physics
PhD	Professor	Physics and Physcis Education
PhD	Professor	Physiology
PhD	Professor	Psychology
PhD	Professor	Pure Mathematics
PhD	Professor	Science Education
PhD	Professor	Social Sciences
PhD	Senior Researcher/lecturer	Law
PhD	Software Engineer	Computer Science
PhD	Researcher	Health Sciences
Professional Actuarial Qualification: Fellow	Lecturer	Actuarial Science
Professional Master	Professor	Health Sciences
Specialist in anaesthesia	Lecturer	Health sciences
Sub-specialist Certificate	Lecturer	Health sciences

Table A.2: Raw Interest Map Responses

Largest Class Size	Do you want live feedback?	Intuitive (1-5)	Useful to me (1-5)	Comments
100-250	No	2	1	System misses some students who are patently disengaged - needs to be tuned up somewhat for better accuracy.
100-250	No	2	2	Not very accurate

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
100-250	No	3	2	I worry about the level of distraction of the lecturer using the system. Rather than focusing on the lecture, he/she would be concentrating on the students who aren't paying attention, rather than on those who are. In a large class (of say more than 200 students), there will always be students who aren't interested, who are easily distracted etc. In my opinion, it is not possible to get a whole class to concentrate, no matter how hard you try. Perhaps that is why the world is moving to online lectures. I think the coding you have done to get this system to work is amazing. I'm just not sure how many lecturers would find this useful in a real world situation.
100-250	No	3	3	The system appears too sensitive to minor disengagements. During a 45-minute you expect every student, even the most attentive, to disengage for short periods. I think a disengagement time threshold (possibly user set) would be useful. Also, as class size increases, monitoring individual students becomes meaningless. The system would be me useful if it gave real-time summative feedback (e.g. proportion of class disengaged, regions of the lecture theatre with the greatest level of disengagement).
100-250	No	3	3	
100-250	No	4	2	Teaching in amphitheatres shows the lecturer which students are engaged or not. The interest Map is very accurate but would not be useful to me as I already can detect this with my large classes.
100-250	No	4	2	
100-250	No	4	3	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
00-250	No	5	1	I do not have much time in the lecture to pay attention to every student. There is usually a lot of content to get through and I cannot stop to make sure every single person is engaged. Students are constantly being distracted by cellphones, friends etc. Would this not be just another distraction? It would also be a distraction for me as a lecturer to have to constantly be paying attention to how many red dots are in my class. I think intuitively lectures are already aware that many students do not stay fully engaged throughout the whole lecture and have already factored that into their lecturing. This study will perhaps serve to reinforce that intuition. However it does not provide suggestions or recommendations on ways to re-engage the student. What I would find beneficial is to also see which students are engaging - perhaps they can be shown in green?"
100-250	Yes	2	2	Does writing show engagement? Arguably it is not related - easy to be tired and still writing
100-250	Yes	3	3	It would be useful to get the definition of "engagement in class" as far as this system is concerned. I observed a student who was fanning himself almost throughout, not taking down any notes, and yet the system identified him as "engaged". On the other hand, a student sitting at the front of the class, who seemed to be engaged, was marked as "not engaged" for a few seconds when she turned away to look at something/someone in the audience - maybe listening to an answer or a question from a fellow student? From my experience, I have had a student who would be looking at the front of the class (correct posture, eyes open, etc.) but could be miles away, and upon asking them a question, only then do you find out that they were not following anything in the lecture at all. I found the system however very good in picking out those who were asleep, or otherwise not looking at the front of the class/had their earphones on.
100-250	Yes	3	4	Facial expression alone may not be a strong indicator of interest.
100-250	Yes	3	4	I think this is something which certainly bears closer scrutiny
100-250	Yes	3	4	
100-250	Yes	3	4	
100-250	Yes	3	5	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
100-250	Yes	4	4	”For the system indicates what I observe each time I teach. I can ‘see’ and ‘feel’ when students are engaged or disengaged. I also make it a point that I approach students who seem to be disengaged for more than 3 lecturers. However, because of big class sizes, I think the system is very helpful in helping me with my blind spots or selective observations and selective recollections.”
100-250	Yes	4	4	I would be interested in a few things. First, the algorithm appears to favor frontward facing students. This assumes that the lecturer is the one imparting knowledge and the student is a (rather) passive listener to what the lecturer says. A student who spends too long looking down or looking away from the front seems to be identified as disengaged. For many of the students/instances I saw in the videos this seems a fair assumption (one student appeared to be looking at their phone for example). In other instances this could be misleading - for example if the lecturer moves around the classroom and up the aisles, if a student asks a question then other engaged students would turn round to face them. Students who are taking notes, comparing notes with another student, or simply thinking about a problem posed by the lecturer will be coded as disengaged. Second, I noticed that some students never seemed to be tagged although they seemed obviously disengaged (e.g. sitting behind a laptop screen, standing up, walking around, etc.). Third, I would be interested to understand how the live feedback would influence my own attention while trying to lecture. Anyway, a very interesting idea. All the best.
100-250	Yes	4	4	presumably if the live feed comes to a computer that I can see, if there are several students who appear disengaged, it might alert me to the possibility of changing tack to increase interest or actively engage the disinterested students.

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
100-250	Yes	4	4	Sometimes it looks like students are disengaged when they seem only to be writing notes. Also sometimes when a student does not move much, it appears that they are engaged (no red highlight) but they may well be "off somewhere in their mind"? In the second longer version the red highlights went up considerably at about 11:30 - some of the student postures after that were "similar" to when students were engaged before then but the highlight was now red? Body language? Crossed arms in front row at 9:24 - no red highlight? Writing with head up - no red highlight but when students write with their heads down - red highlight what is the difference? It appears as if students in light or white shirts were more prevalent in the red highlights - noticeably so in the second longer piece but also in the short piece. Male to female ratio? Were the students aware that the Interest Map was running? "
100-250	Yes	4	4	Very helpful to see how engaged students are during lectures
100-250	Yes	4	4	
100-250	Yes	4	4	
100-250	Yes	4	4	
100-250	Yes	4	4	
100-250	Yes	4	5	Incredible idea! This technology could be a methodological tool in determining ideal break times or lecture lengths, even cohort specifically! I would be very interested to view a percentage/graph of "attention" alongside a lecture itself, possibly allowing an accurate reference to where most were engaged or not.
100-250	Yes	4	5	It seems interesting. However, I worry about the fact that the system highlights in red all students whose focus are off the teacher. It could be that a student is listening to the lecturer while also solving a problem on a paper. It does not mean that a students is distracted and not following when they have their attention off the lecturer
100-250	Yes	5	4	Looks fantastic Rich! I think working on a laptop also counts at not engaged. Also, perhaps add a low pass filter / delay of some sort to prevent the red flashing on and off. Finally, some sort of time/moving average would be pretty handy as a lecturer only really has time to glimpse at the screen. [NameRemoved]
100-250	Yes	5	4	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
100-250	Yes	5	5	I would like if it has a quantitative measure of responsiveness from the students.
100-250	Yes	5	5	The first focus group all are male participants I personally felt the female students tended to pay more attention.
100-250	Yes	5	5	What have you recommended thus far in your lectures to ensure more engagement? Has this been tested and are the results any different from the Interest Map above (many factors will influence this though). Further where the students highlighted interviewed to ensured that they were not engaged?
100-250	Yes	5	5	
20-100	No	2	3	It would be a steep learning curve using this
20-100	No	3	2	In my opinion, not really relevant to classes of 70 or less, since you could very quickly assess this without technology (especially if you know students by name). Could see the use for very big classes.
20-100	No	3	2	Need to summarise the system. E.g. proportion of class currently disengaged, particularly dense areas of the class disengaged, what part of the lecture timewise had the highest disengagement. As it stands, the red hot spots are difficult to interpret from a total class perspective.
20-100	No	3	3	
20-100	No	3	3	
20-100	No	4	4	
20-100	No	4	4	
20-100	No	5	4	Obtaining feedback after a session would be helpful rather than during a session. How useful would this be if other teaching methods are being employed e.g. small group discussions?
20-100	Yes	2	3	The videos are out of context to the viewer so it is really difficult to judge if what you are looking at is good engagement or bad engagement relative to what is happening: recapping versus teaching new material. teaching is hard work while this kind it may be valuable you don't want to have a tool that interferes with the teaching process that you have to concentrate on (divide your attention)
20-100	Yes	3	1	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
20-100	Yes	3	2	The student with white T shirt - left in frame 3r row was also disengae and system did not identify him. A lecture identify these students. this system is however good for feedback to the student if one follows that up with the student in order to identify problems early on
20-100	Yes	3	3	I feel like I need more information to explore the pros and cons of using such a system and how it could / would affect / assist my development as a teacher.
20-100	Yes	3	3	I love the application of the technology, and the idea to use it in the lecture context is really innovative. However, we already get our own live (low-tech) visual feedback of student engagement while teaching. You get a sense of the room, and adjust interaction with students accordingly. The relative advantage of this technology solution over our existing low-tech approach to looking around the room and getting our own feel for student engagement is unclear. Would this technology beat an experienced lecturer in detecting student (dis)engagement? That would be a cool experiment. The tech seems more useful for performance monitoring of teaching, or to remote monitor classroom settings. It might also be useful for new lecturers who are only starting out and wanting to review their teaching performance in class sessions, or the extent to which their students become more or less engaged in different areas of the curriculum. But for experienced teachers who do this all the time anyway, I'm not sure that I'm sold yet on the usefulness of this high-tech approach. And what are the unintended consequences of lecturers looking at screens and video feedback instead of at their class?
20-100	Yes	3	3	I would like to pilot this in one of my class
20-100	Yes	3	3	
20-100	Yes	3	3	
20-100	Yes	3	4	Has the interest map been validated with anonymous student feedback from the lecture?
20-100	Yes	3	4	Is there no way to condense the videos so an overall impression can be made?
20-100	Yes	3	4	It is quite dramatic.
20-100	Yes	3	4	
20-100	Yes	3	4	
20-100	Yes	3	4	
20-100	Yes	3	4	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
20-100	Yes	4	2	For me generally you know your if you class is engaged or not, its what you do about it that's a problem. would be cool to try/experiment and see. based on the Hawthorne effect, just monitoring it will also change student behavior.
20-100	Yes	4	3	I do interactive class work in groups during which I move around the class. Not sure whether this system works in that type of environment but would be interested to try
20-100	Yes	4	3	I don't think it will be worth spending thousands on such a system, if you just take a video and watch afterwards you will be clearly able to see who you are engaging and who not.
20-100	Yes	4	3	
20-100	Yes	4	3	
20-100	Yes	4	4	I can imagine the system to give valuable data when more developed, it could however give ethical problems and 'violate' students privacy. E.g. when they are not looking up it does not mean they are not paying attention etc So, there one should be careful.
20-100	Yes	4	4	Include detection of late coming more than 2min (implication this lecture was not important enough to them to be on time...), include detection of eye movement - even though head is up there is one or two droopy eyes not being highlighted as red
20-100	Yes	4	4	The back left student was constantly chatting to and distracting his neighbour but wasnt picked up by the system. Interesting idea - good luck with the PhD
20-100	Yes	4	4	
20-100	Yes	4	4	
20-100	Yes	4	4	
20-100	Yes	4	4	
20-100	Yes	4	4	
20-100	Yes	4	4	
20-100	Yes	4	4	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
20-100	Yes	4	5	For me, the greatest weakness of this system, is that it seems to record a lack of engagement when students look down. This is particularly noticeable in students taking notes. I agree that eye contact often implies interest, but there was also one candidate, who seemed to be asleep looking at the lecturer. the system identified the individual as interested throughout, but I am not convinced that he actually was. I think there is an amount of resolution loss towards the back of the room (smaller heads). My comments notwithstanding & I think the system is very cleverly coded and even if some entropy exists on the individual level, as a system, I think it gives an amazing insight into the systemic dynamic in a class.
20-100	Yes	4	5	
20-100	Yes	5	4	
20-100	Yes	5	5	
20-100	Yes	5	5	
20-100	Yes	5	5	
20-100	Yes	5	5	
More than 250	No	1	2	While novel, I think that at the university level this system will have little impact since engagement can't be forced and lecturers differ too much in their strategies.
More than 250	No	2	3	
More than 250	No	3	2	
More than 250	No	4	3	In some cases students appear to be "flagged" as not engaged however they appear to be taking notes... Maybe students can be "flagged" if they are not engaged for an extended period of time so that is will be easier to identify students not engaged at all
More than 250	No	4	4	Would be useful as a "self-evaluation" tool after lecture, but it would be distracting to be getting the information live while lecturing.

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
More than 250	Yes	1	4	Your class size question only allows one answer. Many of us teach from one person to over 250, depending on which class we are teaching. The analysis seems to use face posture to make the attention decision? This is highly variable. Sometimes head down means writing and taking notes - an ancient and disappearing art. Distance from the camera also seems to affect the "attention" decision. Your analysis didn't highlight the male student, top left back, in the close-up, and top left at 11 o'clock, next to the aisle, in the whole calss view, who is chatting to and disturbing his buddies to his left, in the back row. I'm not sure the red dots will outdo my current, real time sensor - the basic human ability to engage with each other, face-to-face. Also the red dots completely obscure the miscreants faces. How can I really verify the programmes analysis if I can't see what it is seeing? Nevertheless. Nice idea."
More than 250	Yes	3	3	For your question on class size, allow an option to select more - our sizes are variable depending on course. End result - how could we engage students more in larger classes. In the classes I hold, of the size in your video, I have never noticed this level of disengagement - is this a lecture or tutorial?
More than 250	Yes	3	3	There are students that are not identified because their use of gesturing etc. seems to flag them as active. That said, a reduction of the visualisation to a smaller map for larger classes would be useful as it could be used to assess overall percentage engagement. Regarding the quick assessment of the map, use of a two-colour or multi-colour scheme (in which engaged students are e.g. green) would also make such an assessment much easier especially in allowing new lecturers to train themselves to recognise patterns of engagement or disengagement. Fascinating work overall and perhaps more useful once lecturers start using wearable augmentation technologies.
More than 250	Yes	3	4	I have never heard of this until now

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
More than 250	Yes	3	4	”This is a helpful idea in assisting lecturers in understanding their teaching methods and efficacy. Although, the challenge with such a system is that it provides presumed information. That is, disengagement is presumed by physical characteristics. What if one of the teaching strategies is to get students to utilise technology in the class. According to WITS, cell phone use presumes disengagement. Such a system/study would benefit from including some qualitative feedback about how students perceived their engagement - to do some cross referencing to see how effective such technology really is. Well done on this research, really fascinating work.”
More than 250	Yes	3	4	
More than 250	Yes	3	4	
More than 250	Yes	4	3	I am not sure if students would like to be observed permanently during the lecture.
More than 250	Yes	4	3	I think it’s fairly easy to get a feeling whether many students aren’t following the class without technology. But if they know, they are tracked, it might motivate them to pay more attention.
More than 250	Yes	4	3	
More than 250	Yes	4	4	I am ambivalent about the live feedback during class because it could be very distracting, perhaps better would be on demand, e.g. when I am having trouble with restlessness and/or noise.
More than 250	Yes	4	4	I do feel that I can tell how engaged a class is on my own assessment even though as many as 350. But I may need to see a different view! I might be surprised. I realise that big classes have a percentage of fall-out in terms of engagement that is unavoidable but I try to minimise this.
More than 250	Yes	4	4	i think it would be interesting to see if there is a correlation with where people wit in large venues so one could target these for class input, questions etc
More than 250	Yes	4	4	
More than 250	Yes	4	4	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
More than 250	Yes	4	5	After a difficult 2016, to be made aware of this innovative, interesting and exciting system is a welcome boost! At first glance, the Interest Map appears to be quite formulaic. I realise, however that the level of analysis is appropriately complex. Interest Map differentiates between looking down (only) and looking down with head in hand, and looking down at a cellphone! This is great work!
More than 250	Yes	4	5	Excellent idea - this could be incorporated into lecturer performance appraisals as well to support how well some lecturers get students to engage.
More than 250	Yes	4	5	Would it be possible to have this 'live', as I teach the class? That would be really useful
More than 250	Yes	5	5	Great concept!
More than 250	Yes	5	5	
More than 250	Yes	5	5	
More than 250	Yes	5	5	
More than 250	Yes	5	5	
More than 250	Yes	5	5	
More than 250	Yes	5	5	
More than 250	Yes			The videos didn't work but I'd be interested in seeing how the system works.
Not currently teaching	Yes	3	4	
Not currently teaching	Yes	4	4	I am impressed that you have moved so far
Not currently teaching	Yes	4	4	I like the idea. It will take some practice to use it in real-time, especially if you use other interactive teaching techniques, such as clickers.

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
Not currently teaching	Yes	4	4	If students are aware of the recording, does that impact the numbers?
Not currently teaching	Yes	4	4	Richard, this is truly a fascinating opportunity to get "feedback" from students over which they have little "control", which can highlight for lecturers how what they are doing is genuinely being received. Some comment / questions that spring to mind which may be useful for you to consider are: this 'pilot' has been done with a small cohort of students, yet your intention is that it should be used in large classes. If it operates in real time for a lecturer, how would you plan to give supportive feedback for someone who is just dismayed by the levels of disengagement in large classes, for example chemistry 1 where the disparity of prior knowledge and ability in students is known to be huge? How would you be able to use this to support the development of teaching proficiency, and keep it out of the hands of managers who are on the prowl for evidence of "poor teaching" - recognising that you can take a horse to water but you can't make it drink? There is HUGE research potential in this, in tracking what is going on between student and lecturer and what it is that engages contemporary students . Happy to chat further and keep in touch. [NameRemoved]"
Not currently teaching	Yes	4	4	
Under 20	No	3	2	In small classes such as mine, the system is not necessary because the lecturer can gauge the situation and adapt. I can imagine that in bigger classes, the system would be useful. A second point is that I could not always clearly see the expressions on the students' faces, so I was not sure how accurate the system was, hence my score of three on the intuitive question.
Under 20	No	5	1	
Under 20	No	5	4	I think it's a great system and idea. Not likely to be useful in small group setting where I do most of my teaching. Will be great in the large class, my only concern is that it may be a bit of a distraction to the lecturer, something one would need to get used to.
Under 20	Yes	3	3	

Largest Class Size	Do you want live feed-back?	Intuitive (1-5)	Useful to me (1-5)	Comments
Under 20	Yes	3	3	
Under 20	Yes	3	4	The Interest Map would be useful to chart students engagement with particular material and courses - so would be beneficial to have a series of maps to analyse
Under 20	Yes	3	5	
Under 20	Yes	4	3	I teach a small group so this system not very helpful. Is there a link between the disengaged students and their exam marks, i.e. low marks as compared to the engaged students?
Under 20	Yes	4	3	Less useful in small groups
Under 20	Yes	4	4	
Under 20	Yes	4	4	
Under 20	Yes	4	4	
Under 20	Yes	4	5	1. Instead of shading the face of the students, change the colour of the ring from blue (I prefer green) to red and thickens the border. 2. Use additional colours for measuring the intensity of the engagement i.e. colour between blue (green) and red 4. Make provision for time in terms of disengagement intensity 3. Motivate for big TV screen so that the students can monitor themselves i.e. it will be too tedious for the lecturer. In general, this is a nice project. You can Get in touch with me for more comments [NameRemoved].
Under 20	Yes	4	5	Looks very impressive, Richard!
Under 20	Yes	4	5	

References

[Adair 1984] John G Adair. The Hawthorne effect: A reconsideration of the methodological artifact. *Journal of applied psychology*, 69(2):334, 1984.

[Adiv 1985] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE transactions on pattern analysis and machine intelligence*, (4):384–401, 1985.

[Afzal and Robinson 2009] Shazia Afzal and Peter Robinson. Natural affect data—collection & annotation in a learning context. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–7. IEEE, 2009.

[Agrawal *et al.* 1993] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *International Conference on Foundations of Data Organization and Algorithms*, pages 69–84. Springer, 1993.

[Ahonen *et al.* 2006] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, Dec 2006.

[Alyuz *et al.* 2016] Nese Alyuz, Eda Okur, Ece Oktay, Utku Genc, Sinem Aslan, Sinem Emine Mete, David Stanhill, Bert Arnrich, and Asli Arslan Esme. Towards an emotional engagement model: Can affective states of a learner be automatically detected in a 1: 1 learning scenario. In *Proceedings of the 6th Workshop on Personalization Approaches in Learning Environments (PALE 2016)*. 24th conference on User Modeling, Adaptation, and Personalization (UMAP 2016), CEUR workshop proceedings, this volume, 2016.

[Anderson *et al.* 2004] Amy R Anderson, Sandra L Christenson, Mary F Sinclair, and Camilla A Lehr. Check & Connect: The importance of relationships for promoting engagement with school. *Journal of School Psychology*, 42(2):95–113, 2004.

[Andrade 2010] Jackie Andrade. What Does Doodling do? *Applied Cognitive Psychology*, 24:100–106, February 2010.

[Arroyo *et al.* 2004] Ivon Arroyo, Carole Beal, Tom Murray, Rena Walles, and Beverly P Woolf. Web-based intelligent multimedia tutoring for high stakes achievement tests. In *Intelligent Tutoring Systems*, pages 468–477. Springer, 2004.

[Arroyo *et al.* 2007] Ivon Arroyo, Kimberly Ferguson, Jeffrey Johns, Toby Dragon, Hasmik Meheranian, Don Fisher, Andrew Barto, Sridhar Mahadevan, and Beverly Park Woolf. Repairing disengagement with non-invasive interventions. In *AIED*, volume 2007, pages 195–202, 2007.

[Arroyo *et al.* 2009] Ivon Arroyo, David G Cooper, Winslow Burleson, Beverly Park Woolf, Kasia Muldner, and Robert Christopherson. Emotion Sensors Go To School. In *AIED*, volume 200, pages 17–24, 2009.

[Asthana *et al.* 2009] Akshay Asthana, Jason Saragih, Michael Wagner, and Roland Goecke. Evaluating aam fitting methods for facial expression recognition. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–8. IEEE, 2009.

[Astin 1984] Alexander W Astin. Student involvement: A developmental theory for higher education. *Journal of college student personnel*, 25(4):297–308, 1984.

[Averill 1980] J.R Averill. A constructivist view of emotion. In R. Plutchik and H. Kellerman, editors, *Emotion: Theory, research and experience*, pages 305–339. Academic Press, 1980.

[Ba and Odobez 2009] Sileye O Ba and J-M Odobez. Recognizing visual focus of attention from head pose in natural meetings. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):16–33, 2009.

[Ba and Odobez 2011] Sileye O Ba and J Odobez. Multiperson visual focus of attention from head pose and meeting contextual cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):101–116, 2011.

[Ba 2007] S. Ba. *Joint Head Tracking and Pose Estimation for Visual Focus of Attention Recognition,*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), February 2007.

[Barrett 2006] Lisa Feldman Barrett. Are emotions natural kinds? *Perspectives on psychological science*, 1(1):28–58, 2006.

[Benzaid and Dewan 2010] Sami Benzaid and Prasun Dewan. Semantic Awareness Through Computer Vision. In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '10, pages 205–210, New York, NY, USA, 2010. ACM.

[Bianchi-Berthouze and Lisetti 2002] Nadia Bianchi-Berthouze and Christine L Lisetti. Modeling multimodal expression of user's affective subjective experience. *User Modeling and User-Adapted Interaction*, 12(1):49–84, 2002.

[Blaszczynski *et al.* 1990] Alex Blaszczynski, Neil McConaghy, and Anna Frankova. Boredom Proneness in Pathological Gambling. *Psychological Reports*, 67(1):35–42, 1990.

[Bosch *et al.* 2015a] Nigel Bosch, Huili Chen, Sidney D'Mello, Ryan Baker, and Valerie Shute. Accuracy vs. availability heuristic in multimodal affect detection in the wild. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 267–274. ACM, 2015.

[Bosch *et al.* 2015b] Nigel Bosch, Sidney D'Mello, Ryan Baker, Jaclyn Ocumpaugh, Valerie Shute, Matthew Ventura, Lubin Wang, and Weinan Zhao. Automatic Detection of Learning-Centered Affective States in the Wild. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 379–388, New York, NY, USA, 2015. ACM.

[Boser *et al.* 1992] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[Bouguet 2001] Jean-Yves Bouguet. Pyramidal implementation of the affine Lucas-Kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.

[Boyle and Nicol 2003] James T Boyle and David J Nicol. Using classroom communication systems to support interaction and discussion in large class settings. *Research in Learning Technology*, 11(3), 2003.

[Bradski 2000] Gary Bradski. The OpenCV library. *Doctor Dobbs Journal*, 25(11):120–126, 2000.

[Brett 2011] Paul Brett. Students' experiences and engagement with SMS for learning in higher education. *Innovations in Education and Teaching International*, 48(2):137–147, 2011.

[Brick *et al.* 2009] Timothy R Brick, Michael D Hunter, and Jeffrey F Cohn. Get The FACS Fast: Automated FACS face analysis benefits from the addition of velocity. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–7. IEEE, 2009.

[Bull 1987] Peter E Bull. *Posture and gesture.* Pergamon press, 1987.

[Calvo and D'Mello 2010] Rafael A Calvo and Sidney D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *Affective Computing, IEEE Transactions on*, 1(1):18–37, 2010.

[Calvo and Nummenmaa 2007] Manuel G Calvo and Lauri Nummenmaa. Processing of unattended emotional visual scenes. *Journal of Experimental Psychology: General*, 136(3):347, 2007.

[Card *et al.* 1974] WI Card, Mary Nicholson, GP Crean, Geoffrey Watkinson, CR Evans, Jackie Wilson, and Daphne Russell. A comparison of doctor and computer interrogation of patients. *International journal of bio-medical computing*, 5(3):175–187, 1974.

[Castellano *et al.* 2008a] Ginevra Castellano, Loic Kessous, and George Caridakis. Emotion recognition through multiple modalities: face, body gesture, speech. In *Affect and emotion in human-computer interaction*, pages 92–103. Springer, 2008.

[Castellano *et al.* 2008b] Ginevra Castellano, Marcello Mortillaro, Antonio Camurri, Gualtiero Volpe, and Klaus Scherer. Automated analysis of body movement in emotionally expressive piano performances. *Music Perception: An Interdisciplinary Journal*, 26(2):103–119, 2008.

[Chan and Fu 1999] Kin-Pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 126–133. IEEE, 1999.

[Chang and Lin 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[Chang *et al.* 2002] Jyh-Yeong Chang, Wen-Feng Hu, Mu-Huo Cheng, and Bo-Sen Chang. Digital image translational and rotational motion stabilization using optical flow technique. *IEEE Transactions on Consumer Electronics*, 48(1):108–115, 2002.

[Chapelle 2007] Olivier Chapelle. Training a support vector machine in the primal. *Neural computation*, 19(5):1155–1178, 2007.

[Chen and Chung 2008] Chih-Ming Chen and Ching-Ju Chung. Personalized mobile English vocabulary learning system based on item response theory and learning memory cycle. *Computers & Education*, 51(2):624–645, 2008.

[Chetlur *et al.* 2014] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *CoRR*, abs/1410.0759, 2014.

[Cohen 1960] Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, (20):37–46, 1960.

[Cohn and Schmidt 2004] Jeffrey F Cohn and Karen L Schmidt. The timing of facial motion in posed and spontaneous smiles. *International Journal of Wavelets, Multiresolution and Information Processing*, 2(2):121–132, 2004.

[Conati and Maclare 2004] Cristina Conati and Heather Maclare. Evaluating a probabilistic model of student affect. In *Intelligent tutoring systems*, pages 55–66. Springer, 2004.

[Cooper 1960] L. Cooper. *Aristotle, The Rhetoric of Aristotle*. Appleton-Century-Crofts, 1960. An expanded translation with supplementary examples for students of composition and public speaking.

[Coulson 2004] Mark Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Journal of nonverbal behavior*, 28(2):117–139, 2004.

[Craig *et al.* 2008] Scotty D. Craig, Sidney D'Mello, Amy Witherspoon, and Art Graesser. Emote aloud during learning with AutoTutor: Applying the Facial Action Coding System to cognitive–affective states during learning. *Cognition & Emotion*, 22(5):777–788, 2008.

[Cytowic 1996] Richard E Cytowic. *The neurological side of neuropsychology*. MIT Press, 1996.

[Dalal and Triggs 2005] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[Dalgleish *et al.* 2009] Tim Dalgleish, Barnaby D Dunn, and Dean Mobbs. Affective neuroscience: Past, present, and future. *Emotion Review*, 1(4):355–368, 2009.

[Dalgleish 2004] Tim Dalgleish. The emotional brain. *Nature Reviews Neuroscience*, 5(7):583–589, 2004.

[Damasio 1994] Antonio R Damasio. *Descartes' error: Emotion, reason, and the human brain*, 1994.

[Damasio 2000] Antonio Damasio. *Looking for Spinoza. Joy, Sorrow, and the Feeling Brain*, 2000.

[Darwin 1859] Charles Darwin. On the origins of species by means of natural selection. *London: Murray*, 1859.

[Darwin 1872] Charles Darwin. *The Expression of the Emotions in Man and Animals*. John Murray, 1872.

[Davidson *et al.* 2003] Richard J Davidson, Klaus R Scherer, and Hill Goldsmith. *Handbook of affective sciences.* Oxford University Press, 2003.

[De Meijer 1989] Marco De Meijer. The contribution of general features of body movement to the attribution of emotions. *Journal of Nonverbal behavior*, 13(4):247–268, 1989.

[Dearnley *et al.* 2008] Chris Dearnley, Jackie Haigh, and John Fairhall. Using mobile technologies for assessment and learning in practice settings: a case study. *Nurse education in practice*, 8(3):197–204, 2008.

[Dennerlein *et al.* 2003] Jack Dennerlein, Theodore Becker, Peter Johnson, Carson Reynolds, and Rosalind W Picard. Frustrating computer users increases exposure to physical factors. In *Proceedings of the International Ergonomics Association, Seoul, Korea*, 2003.

[Devi and Bajaj 2008] M.S. Devi and P.R. Bajaj. Driver Fatigue Detection Based on Eye Tracking. In *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on*, pages 649–652, July 2008.

[D'mello and Graesser 2007] Sidney D'mello and Arthur Graesser. Mind and Body: Dialogue and posture for affect detection in learning environments. *Frontiers in Artificial Intelligence and Applications*, 158:161, 2007.

[D'Mello and Graesser 2009] Sidney D'Mello and Art Graesser. Automatic Detection of Learner's Affect From Gross Body Language. *Applied Artificial Intelligence*, 23(2):123–150, February 2009.

[D'Mello and Graesser 2010] Sidney K D'Mello and Arthur Graesser. Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-Adapted Interaction*, 20(2):147–187, 2010.

[D'Mello *et al.* 2006] Sidney K D'Mello, Scotty D Craig, Jeremiah Sullins, and Arthur C Graesser. Predicting affective states expressed through an emote-aloud procedure from AutoTutor's mixed-initiative dialogue. *International Journal of Artificial Intelligence in Education*, 16(1):3–28, 2006.

[D'Mello *et al.* 2007] Sidney D'Mello, Rosalind Picard, and Arthur Graesser. Towards an affect-sensitive autotutor. *IEEE Intelligent Systems*, 22(4):53–61, 2007.

[D'Mello *et al.* 2010] Sidney D'Mello, Blair Lehman, Jeremiah Sullins, Rosaire Daigle, Rebekah Combs, Kimberly Vogt, Lydia Perkins, and Art Graesser. A time for emoting: When affect-sensitivity is and isn't effective at promoting deep learning. In *Intelligent tutoring systems*, pages 245–254. Springer, 2010.

[Donato *et al.* 1999] Gianluca Donato, Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski. Classifying facial actions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(10):974–989, 1999.

[Dong and Wu 2005] Wenhui Dong and Xiaojuan Wu. Fatigue detection based on the distance of eyelid. In *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop on*, pages 365–368, May 2005.

[Draper and Brown 2004] Stephen W Draper and Margaret I Brown. Increasing interactivity in lectures using an electronic voting system. *Journal of computer assisted learning*, 20(2):81–94, 2004.

[DMello 2013] Sidney DMello. A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *Journal of Educational Psychology*, 105(4):1082, 2013.

[Ekman and Friesen 1969] Paul Ekman and Wallace V Friesen. *Nonverbal leakage and clues to deception*. Technical report, DTIC Document, 1969.

[Ekman and Friesen 1978] Paul Ekman and Wallace V Friesen. Facial action coding system: A technique for the Measurement of Facial Movement: Investigators Guide 2 Parts. *Consulting Psychologists Press*, 1978.

[Ekman and Friesen 2003] Paul Ekman and Wallace V Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Ishk, 2003.

[Ekman *et al.* 1980] Paul Ekman, Wallace V Freisen, and Sonia Ancoli. Facial signs of emotional experience. *Journal of personality and social psychology*, 39(6):1125, 1980.

[Ekman *et al.* 1990] Paul Ekman, Richard J Davidson, and Wallace V Friesen. The Duchenne smile: Emotional expression and brain physiology: II. *Journal of personality and social psychology*, 58(2):342, 1990.

[Ekman 1971] Paul Ekman. Universals and cultural differences in facial expressions of emotion. In *Nebraska symposium on motivation*. University of Nebraska Press, 1971.

[Ekman 1992] Paul Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200, 1992.

[El Kaliouby and Robinson 2005a] Rana El Kaliouby and Peter Robinson. Generalization of a vision-based computational model of mind-reading. In *Affective computing and intelligent interaction*, pages 582–589. Springer, 2005.

[El Kaliouby and Robinson 2005b] Rana El Kaliouby and Peter Robinson. Real-time inference of complex mental states from facial expressions and head gestures. In *Real-time vision for human-computer interaction*, pages 181–200. Springer, 2005.

[Faloutsos *et al.* 1994] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. *Fast subsequence matching in time-series databases*, volume 23. ACM, 1994.

[Faure and Orthober 2011] Caroline Faure and Corrie Orthober. Using text-messaging in the secondary classroom. *American Secondary Education*, 39(2):55, 2011.

[Fecci *et al.* 1971] R Fecci, R Bartelemy, J Bourgoin, A Mathia, H Eberle, A Moutel, and G Jullien. Effects of infrasound on the organism. *La Medicina del Lavoro*, 62:130–150, 1971.

[Freund and Schapire 1995] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[Friedman *et al.* 2001] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[Frijda 1987] Nico H Frijda. Emotion, cognitive structure, and action tendency. *Cognition and emotion*, 1(2):115–143, 1987.

[Gavrila 1999] Dariu M Gavrila. The visual analysis of human movement: A survey. *Computer vision and image understanding*, 73(1):82–98, 1999.

[Gendron *et al.* 2012] Maria Gendron, Kristen A Lindquist, Lawrence Barsalou, and Lisa Feldman Barrett. Emotion words shape emotion percepts. *Emotion*, 12(2):314, 2012.

[Glenberg *et al.* 2005] A.M. Glenberg, D Havas, R Becker, and M Rinck. Grounding language in bodily states. In D Pecher and R.A. Zwaan, editors, *Grounding cognition: The role of perception and action in memory, language, and thinking*, pages 115–128. Cambridge University Press, Cambridge, 2005.

[Go *et al.* 2009] Alec Go, Lei Huang, and Richa Bhayani. Twitter sentiment analysis. *Entropy*, 17, 2009.

[Gogia *et al.* 2016] Yash Gogia, Eejya Singh, Shreyash Mohatta, and V Sreejith. Multi-modal affect detection for learning applications. In *Region 10 Conference (TENCON), 2016 IEEE*, pages 3743–3747. IEEE, 2016.

[Goodfellow *et al.* 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[Graesser *et al.* 2006] AC Graesser, Bethany McDaniel, Patrick Chipman, Amy Witherspoon, Sidney D'Mello, and Barry Gholson. Detection of emotions during learning with AutoTutor. In *Proceedings of the 28th Annual Meetings of the Cognitive Science Society*, pages 285–290. Citeseer, 2006.

[Graesser *et al.* 2007] Arthur Graesser, Patrick Chipman, Brandon King, Bethany McDaniel, and Sidney D'Mello. Emotions and learning with auto tutor. *Frontiers in Artificial Intelligence and Applications*, 158:569, 2007.

[Graham and Weiner 1996] Sandra Graham and Bernard Weiner. Theories and principles of motivation. *Handbook of educational psychology*, 4:63–84, 1996.

[Haro *et al.* 2000] Antonio Haro, Myron Flickner, and Irfan Essa. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 163–168. IEEE, 2000.

[Healey and Picard 2000] Jennifer Healey and Rosalind Picard. Smartcar: detecting driver stress. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 218–221. IEEE, 2000.

[Heisele *et al.* 2001] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: global versus component-based approach. In *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV2001)*, volume 2, pages 688–694, 2001.

[Heudorf *et al.* 2009] U. Heudorf, V. Neitzert, and J. Spark. Particulate matter and carbon dioxide in classrooms – The impact of cleaning and ventilation. *International Journal of Hygiene and Environmental Health*, 212(1):45–55, 2009.

[Hinton *et al.* 2012] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[Holley and Oliver 2010] Debbie Holley and Martin Oliver. Student engagement and blended learning: Portraits of risk. *Computers & Education*, 54(3):693–700, 2010.

[Hoque *et al.* 2009] Mohammed E Hoque, Rana El Kaliouby, and Rosalind W Picard. When human coders (and machines) disagree on the meaning of facial affect in spontaneous videos. In *Intelligent Virtual Agents*, pages 337–343. Springer, 2009.

[Horn and Schunck 1981] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[Horng *et al.* 2004] Wen-Bing Horng, Chih-Yuan Chen, Yi Chang, and Chun-Hai Fan. Driver fatigue detection based on eye tracking and dynamk, template matching. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 1, pages 7–12, March 2004.

[Hsu 1999] Feng-hsiung Hsu. IBM's Deep Blue chess grandmaster chips. *IEEE Micro*, 19(2):70–81, 1999.

[Huang *et al.* 2008] Yueh-Min Huang, Yen-Hung Kuo, Yen-Ting Lin, and Shu-Chen Cheng. Toward interactive mobile synchronous learning environment with context-awareness service. *Computers & Education*, 51(3):1205–1226, 2008.

[Iwasawa *et al.* 1997] Shoichiro Iwasawa, Kazuyuki Ebihara, Jun Ohya, and Shigeo Morishima. Real-time estimation of human body posture from monocular thermal images. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 15–20. IEEE, 1997.

[Izard 1971] Carroll E. Izard. *The face of emotion*. East Norwalk, CT, US: Appleton-Century-Crofts, 1971.

[Izard 1994] Carroll E Izard. Innate and universal facial expressions: evidence from developmental and cross-cultural research. *Psychological Bulletin*, 115(2):288–299, Mar 1994.

[Jaimes and Sebe 2007] Alejandro Jaimes and Nicu Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134, 2007.

[James 1884] William James. What is an emotion? *Mind*, 9(34):188–205, 1884.

[Jeong *et al.* 2008] Jae-chan Jeong, Ho-chul Shin, and Dae-hwan Hwang. Real-time Upper Body Pose Detection using Stereo Vision ASIC. In *Proceedings of the 18th International Conference on Artificial Reality and Telexistence*, pages 238–241, 2008.

[Jia *et al.* 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[Jin *et al.* 2007] Shanshan Jin, So-Youn Park, and Ju-Jang Lee. Driver fatigue detection using a genetic algorithm. *Artificial Life and Robotics*, 11(1):87–90, 2007.

[Johns and Woolf 2006] Jeffrey Johns and Beverly Woolf. A dynamic mixture model to detect student motivation and proficiency. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 163. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[Johnstone and Scherer 2000] Tom Johnstone and KR Scherer. Vocal communication of emotion. *Handbook of Emotion*, pages 220–235, 2000.

[Jones *et al.* 2009] Geraldine Jones, Gabriele Edwards, and Alan Reid. How Can Mobile SMS Communication Support and Enhance a First Year Undergraduate Learning Environment?. *ALT-J: Research in Learning Technology*, 17(3):201–218, 2009.

[Juang *et al.* 2009] Chia-Feng Juang, Chia-Ming Chang, Jiuh-Rou Wu, and Demei Lee. Computer vision-based human body segmentation and posture estimation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(1):119–133, 2009.

[Juslin and Scherer 2005] Patrick N Juslin and Klaus R Scherer. Vocal expression of affect. *The new handbook of methods in nonverbal behavior research*, pages 65–135, 2005.

[Kagan 1984] J. Kagan. *The Nature of the Child*. Gosset/Putnam Press, New York, NY, 1984.

[Kai *et al.* 2015] Shiming Kai, Luc Paquette, Ryan S Baker, Nigel Bosch, Sidney D'Mello, Jaclyn Ocumpaugh, Valerie Shute, and Matthew Ventura. A comparison of video-based and interaction-based affect detectors in physics playground. *International Educational Data Mining Society*, 2015.

[Kapoor and Picard 2001] Ashish Kapoor and Rosalind W. Picard. A Real-time Head Nod and Shake Detector. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, PUI '01, pages 1–5, New York, NY, USA, 2001. ACM.

[Kapoor and Picard 2002] Ashish Kapoor and Rosalind W Picard. Real-time, fully automatic upper facial feature tracking. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 8–13. IEEE, 2002.

[Kapoor and Picard 2005] Ashish Kapoor and Rosalind W Picard. Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 677–682. ACM, 2005.

[Kapoor *et al.* 2007] Ashish Kapoor, Winslow Burleson, and Rosalind W Picard. Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65(8):724–736, August 2007.

[Kiyoshi Kawaguchi 2000] Kiyoshi Kawaguchi. *Linear Separability and the XOR Problem*. http://www.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node19.html, 2000. Online; accessed 2017-01-20.

[Klein and Celik 2017a] Richard Klein and Turgay Celik. Engage: Live Self Reported Engagement for Large Classes. *IEEE Transactions on Learning Technologies*, 2017. Under review.

[Klein and Celik 2017b] Richard Klein and Turgay Celik. The Wits Intelligent Teaching System: Detecting Student Engagement During Lectures Using Convolutional Neural Networks. In *Image Processing, 2017. ICIP'17. 2017 International Conference on*. IEEE, 2017. Accepted.

[Klein and Celik 2017c] Richard Klein and Turgay Celik. Visualization of Audience Interest: An Interest Map for Reporting Live Audience Engagement. *IEEE Transactions on Learning Technologies*, 2017. Under review.

[Klein and Celik 2017d] Richard Klein and Turgay Celik. Wits Intelligent Teaching System: A video dataset and computer vision system for student action recognition in the classroom. *IEEE Transactions on Affective Computing*, 2017. Under review.

[Koenderink 1986] Jan J Koenderink. Optic flow. *Vision research*, 26(1):161–179, 1986.

[Kort *et al.* 2001] Barry Kort, Rob Reilly, and Rosalind W Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Advanced Learning Technologies, IEEE International Conference on*, pages 0043–0043. IEEE Computer Society, 2001.

[Kouloumpis *et al.* 2011] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11:538–541, 2011.

[Krizhevsky *et al.* 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[Kroes 2007] Stefan Kroes. Detecting Boredom in Meetings. *Enschede, Netherlands, University of Twente*, pages 1–5, 2007.

[Kumar *et al.* 2009] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009.

[Kyriakides and Leventhall 1977] K Kyriakides and HG Leventhall. Some effects of infrasound on task performance. *Journal of Sound and Vibration*, 50(3):369–388, 1977.

[Lan and Sie 2010] Yu-Feng Lan and Yang-Siang Sie. Using RSS to support mobile learning based on media richness theory. *Computers & Education*, 55(2):723–732, 2010.

[Landis and Koch 1977] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[Landström and Byström 1984] U. Landström and M Byström. Infrasonic threshold levels of physiological effects effects. *Journal of Low Frequency Noise and Vibration*, 3:167–173, 1984.

[Landström and Byström 1985] U. Landström and M Byström. Changes in wakefulness during exposure to noise at 42Hz, 1000Hz and individual EEG frequencies. *Journal of Low Frequency Noise and Vibration*, 4:27–33, 1985.

[Landstrom *et al.* 1991] U Landstrom, A Kjellberg, L Söderberg, and B Nordström. The effects of broadband, tonal and masked ventilation noise on performance, wakefulness and annoyance. *Journal of low frequency noise & vibration*, 10(4):112–122, 1991.

[Lange 1885] Carl Georg Lange. The mechanism of the emotions. *The Classical Psychologists. Boston: Houghton Mifflin*, 1912, 1885.

[Langley 1996] Pat Langley. *Elements of machine learning*. Morgan Kaufmann, 1996.

[Larson and Richards 1991] Reed W Larson and Maryse H Richards. Boredom in the middle school years: Blaming schools versus blaming students. *American Journal of Education*, 1991.

[Lau *et al.* 2014] Rynson WH Lau, Neil Y Yen, Frederick Li, and Benjamin Wah. Recent development in multimedia e-learning technologies. *World Wide Web*, 17(2):189–198, 2014.

[LeCun *et al.* 1989] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[Leventhall *et al.* 2003] Geoff Leventhall, Peter Pelmear, and Stephen Benton. *A review of published research on low frequency noise and its effects*. Technical report, Department for Environment, Food and Rural Affairs, 2003.

[Lienhart *et al.* 2003] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Joint Pattern Recognition Symposium*, pages 297–304. Springer, 2003.

[Lin *et al.* 2007] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007.

[Lindquist *et al.* 2006] Kristen A Lindquist, Lisa Feldman Barrett, Eliza Bliss-Moreau, and James A Russell. Language and the perception of emotion. *Emotion*, 6(1):125, 2006.

[Lindquist *et al.* 2007] David Lindquist, Tamara Denning, Michael Kelly, Roshni Malani, William G Griswold, and Beth Simon. Exploring the potential of mobile phones for active learning in the classroom. In *ACM SIGCSE Bulletin*, volume 39, pages 384–388. ACM, 2007.

[Liu *et al.* 2014] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4209–4216, 2014.

[Lowe and Laffey 2011] Ben Lowe and Des Laffey. Is Twitter for the birds? Using Twitter to enhance student learning in a marketing course. *Journal of Marketing Education*, page 0273475311410851, 2011.

[Lu and Viehland 2008] Xu Lu and Dennis Viehland. Factors influencing the adoption of mobile learning. *ACIS 2008 Proceedings*, page 56, 2008.

[Lucas *et al.* 1977] RW Lucas, PJ Mullin, CB Luna, and DC McInroy. Psychiatrists and a computer as interrogators of patients with alcohol-related illnesses: a comparison. *The British Journal of Psychiatry*, 131(2):160–167, 1977.

[Lucas *et al.* 1981] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[MacHardy *et al.* 2012] Zachary M MacHardy, Kenneth Syharath, and Prasun Dewan. Engagement analysis through computer vision. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, pages 535–539. IEEE, 2012.

[Maclay *et al.* 1938] W.S. Maclay, E. Guttmann, and W. Mayer-Gross. Spontaneous Drawings as an Approach to some Problems of Psychopathology. *Proceedings of the Royal Society of Medicine*, 31(11):1337–1350, September 1938.

[Mallat 1990] SG Mallat. Multiresolution approach to wavelets in computer vision. In *Wavelets*, pages 313–327. Springer, 1990.

[Manning *et al.* 2008] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[Markett *et al.* 2006] Carina Markett, I Arnedillo Sánchez, Stefan Weber, and Brendan Tangney. Using short message service to encourage interactivity in the classroom. *Computers & Education*, 46(3):280–293, 2006.

[Marsland 2015] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2015.

[McCulloch and Pitts 1943] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[McDaniel *et al.* 2007] B. McDaniel, Sidney D'Mello, B. King, P. Chipman, K. Tapp, and A. Graesser. Facial Features for Affective State Detection in Learning Environments. In *Proceedings of the 29th. Annual Meeting of the Cognitive Science Society*, 2007.

[McQuiggan and Lester 2006] Scott W McQuiggan and James C Lester. Diagnosing self-efficacy in intelligent tutoring systems: An empirical study. In *Intelligent Tutoring Systems*, pages 565–574. Springer, 2006.

[Mehrabian 1977] Albert Mehrabian. *Nonverbal communication*. Transaction Publishers, 1977.

[Mendell and Heath 2005] M. J. Mendell and G. A. Heath. Do indoor pollutants and thermal conditions in schools influence student performance? A critical review of the literature. *Indoor Air*, 15(1):27–52, 2005.

[Merla and Romani 2007] A. Merla and G.L. Romani. Thermal Signatures of Emotional Arousal: A Functional Infrared Imaging Study. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 247–249, Aug 2007.

[Mikulas and Vodanovich 1993] William L Mikulas and Stephen J Vodanovich. The essence of boredom. *The Psychological Record*, 1993.

[Minsky 1961] M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, Jan 1961.

[Minsky 2007] Marvin Minsky. *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster, 2007.

[Moeslund *et al.* 2006] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006.

[Mohri *et al.* 2012] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[Montepare *et al.* 1999] Joann Montepare, Elissa Koff, Deborah Zaitchik, and Marilyn Albert. The use of body movements and gestures as cues to emotions in younger and older adults. *Journal of Nonverbal Behavior*, 23(2):133–152, 1999.

[Morency *et al.* 2007] Louis-Philippe Morency, Candace Sidner, Christopher Lee, and Trevor Darrell. Head gestures for perceptual interfaces: The role of context in improving recognition. *Artificial Intelligence*, 171(8-9):568–585, 2007.

[Mota and Picard 2003] Selene Mota and Rosalind W Picard. Automated posture analysis for detecting learner's interest level. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, volume 5, pages 49–49. IEEE, 2003.

[Motiwalla 2007] Luvai F Motiwalla. Mobile learning: A framework and evaluation. *Computers & education*, 49(3):581–596, 2007.

[Murphy-Chutorian and Trivedi 2009] Erik Murphy-Chutorian and Mohan M Trivedi. Head pose estimation in computer vision: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):607–626, 2009.

[Nickolls *et al.* 2008] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, March 2008.

[Nicol and Boyle 2003] David J Nicol and James T Boyle. Peer instruction versus class-wide discussion in large classes: a comparison of two interaction methods in the wired classroom. *Studies in Higher Education*, 28(4):457–473, 2003.

[NVIDIA 2015] NVIDIA. *NVIDIA TITAN X Graphics Card with Pascal*. https://www.nvidia.com/en-us/geforce/products/10series/titan-x-pascal/, 2015. Online; accessed 2017-02-18.

[Ocumpaugh 2012] Jaclyn Ocumpaugh. Baker-Rodrigo observation method protocol (BROMP) 1.0. Training manual version 1.0. *Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual version 1.0, Tech. Rep*, 2012.

[Öhman and Soares 1998] Arne Öhman and Joaquim JF Soares. Emotional conditioning to masked stimuli: expectancies for aversive outcomes following nonrecognized fear-relevant stimuli. *Journal of Experimental Psychology: General*, 127(1):69, 1998.

[Ojala and Pietikäinen 1999] Timo Ojala and Matti Pietikäinen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, 32(3):477–486, 1999.

[Ojala *et al.* 1996] Timo Ojala, Matti Pietikinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):5159, 1996.

[Ojala *et al.* 2002] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[OpenCV 2014] OpenCV. *Introduction to Support Vector Machines*. http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html, 2014. Online; accessed 2017-01-20.

[OpenCV 2015] OpenCV. *Optical Flow*. http://docs.opencv.org/3.2.0/d7/d8b/tutorial_py_lucas_kanade.html, 2015. Online; accessed 2017-01-20.

[Ortony and Turner 1990] Andrew Ortony and Terence J Turner. What's basic about basic emotions? *Psychological review*, 97(3):315, 1990.

[Ortony 1990] Andrew Ortony. *The cognitive structure of emotions*. Cambridge university press, 1990.

[Pak and Paroubek 2010] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[Palloff and Pratt 1999] R.M. Palloff and K. Pratt. *Building learning communities in cyberspace: effective strategies for the online classroom* . Jossey-Bass, San Francisco, 1999.

[Panksepp 1998] Jaak Panksepp. *Affective neuroscience: The foundations of human and animal emotions*. Oxford university press, 1998.

[Pantic and Patras 2006] Maja Pantic and Ioannis Patras. Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(2):433–449, 2006.

[Pantic and Rothkrantz 2003] Maja Pantic and Leon JM Rothkrantz. Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, 91(9):1370–1390, 2003.

[Paquette *et al.* 2016] Luc Paquette, Jonathan Rowe, Ryan Baker, Bradford Mott, James Lester, Jeanine DeFalco, Keith Brawner, Robert Sottilare, and Vasiliki Georgoulas. Sensor-free or sensor-full: A comparison of data modalities in multi-channel affect detection. *International Educational Data Mining Society*, 2016.

[Patten *et al.* 2006] Bryan Patten, Inmaculada Arnedillo Sánchez, and Brendan Tangney. Designing collaborative, constructionist and contextual applications for handheld devices. *Computers & education*, 46(3):294–308, 2006.

[Persson Waye and Rylander 2001] K Persson Waye and R Rylander. The prevalence of annoyance and effects after long-term exposure to low-frequency noise. *Journal of sound and vibration*, 240(3):483–497, 2001.

[Persson Waye *et al.* 1997] Kerstin Persson Waye, R Rylander, S Benton, and HG Leventhall. Effects on performance and work quality due to low frequency ventilation noise. *Journal of Sound and Vibration*, 205(4):467–474, 1997.

[Phillips *et al.* 2011] P Jonathon Phillips, J Ross Beveridge, Bruce A Draper, Geof Givens, Alice J O'Toole, David S Bolme, Joseph Dunlop, Yui Man Lui, Hassan Sahibzada, and Samuel Weimer. An introduction to the good, the bad, & the ugly face recognition challenge problem. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 346–353. IEEE, 2011.

[Picard 1995] Rosalind W. Picard. *Affective Computing*. Technical report, MIT Media Laboratory, 1995. http://www.media.mit.edu/ picard/.

[Picard 1997] Rosalind W. Picard. *Affective Computing*. MIT Press, 1997.

[Picard 2003a] Rosalind W. Picard. Affective computing: challenges. *International Journal of Human-Computer Studies*, 59(1–2):55–64, 2003. Applications of Affective Computing in Human-Computer Interaction.

[Picard 2003b] Rosalind W Picard. What does it mean for a computer to "have" emotions. *Emotions in humans and artifacts*, pages 87–102, 2003.

[Platt and others 1999] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[Poulsen *et al.* 2017] Andreas Trier Poulsen, Simon Kamronn, Jacek Dmochowski, Lucas C Parra, and Lars Kai Hansen. EEG in the classroom: Synchronised neural recordings during video presentation. *Scientific Reports*, 7, 2017.

[Puri *et al.* 2005] Colin Puri, Leslie Olson, Ioannis Pavlidis, James Levine, and Justin Starren. StressCam: Non-contact Measurement of Users' Emotional States Through Thermal Imaging. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1725–1728, Portland, OR, USA, 2005. ACM.

[Raca and Dillenbourg 2013] Mirko Raca and Pierre Dillenbourg. System for assessing classroom attention. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 265–269. ACM, 2013.

[Raca *et al.* 2015] Mirko Raca, Lukasz Kidzinski, and Pierre Dillenbourg. Translating head motion into attention-towards processing of student's body-language. In *Proceedings of the 8th International Conference on Educational Data Mining*, number EPFL-CONF-207803, 2015.

[Ramanan 2008] Deva Ramanan. *ICS 273A Machine Learning, Lecture 11*. http://www.ics.uci.edu/~dramanan/teaching/ics273a_winter08/lectures/lecture11.pdf, 2008. Online; accessed 2017-01-26.

[Rau *et al.* 2008] Pei-Luen Patrick Rau, Qin Gao, and Li-Mei Wu. Using mobile communication technology in high school education: Motivation, pressure, and learning performance. *Computers & Education*, 50(1):1–22, 2008.

[Reisenzein *et al.* 2013] Rainer Reisenzein, Markus Studtmann, and Gernot Horstmann. Coherence between emotion and facial expression: Evidence from laboratory experiments. *Emotion Review*, 5(1):16–23, 2013.

[Richardson 2004] Iain E Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[Robinson and West 1992] Rachael Robinson and Robert West. A comparison of computer and questionnaire methods of history-taking in a genito-urinary clinic. *Psychology and Health*, 6(1-2):77–84, 1992.

[Roseman *et al.* 1990] Ira J Roseman, Martin S Spindel, and Paul E Jose. Appraisals of emotion-eliciting events: Testing a theory of discrete emotions. *Journal of Personality and Social Psychology*, 59(5):899, 1990.

[Roseman 1984] Ira J Roseman. Cognitive determinants of emotion: A structural theory. *Review of Personality & Social Psychology*, 1984.

[Rosenfeld 1988] Azriel Rosenfeld. Computer vision: basic principles. *Proceedings of the IEEE*, 76(8):863–868, 1988.

[Royer and Walles 2007] James M Royer and Rena Walles. Influences of gender, motivation and socioeconomic status on mathematics performance. *Why is math so hard for some children. Baltimore, MD: Paul H. Brookes Publishing Co*, pages 349–368, 2007.

[Russell *et al.* 2003a] James A Russell, Jo-Anne Bachorowski, and José-Miguel Fernández-Dols. Facial and vocal expressions of emotion. *Annual review of psychology*, 54(1):329–349, 2003.

[Russell *et al.* 2003b] James A Russell, Jo-Anne Bachorowski, and José-Miguel Fernández-Dols. Facial and vocal expressions of emotion. *Annual review of psychology*, 54(1):329–349, 2003.

[Russell 1994] James A Russell. Is there universal recognition of emotion from facial expressions? A review of the cross-cultural studies. *Psychological bulletin*, 115(1):102, 1994.

[Russell 2003] James A Russell. Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145, 2003.

[Schachter and Singer 1962] Stanley Schachter and Jerome Singer. Cognitive, social, and physiological determinants of emotional state. *Psychological review*, 69(5):379, 1962.

[Schaeffer *et al.* 2007] Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen. Checkers is solved. *science*, 317(5844):1518–1522, 2007.

[Scherer and Ellgring 2007a] Klaus R Scherer and Heiner Ellgring. Are facial expressions of emotion produced by categorical affect programs or dynamically driven by appraisal? *Emotion*, 7(1):113, 2007.

[Scherer and Ellgring 2007b] Klaus R Scherer and Heiner Ellgring. Multimodal expression of emotion: Affect programs or componential appraisal patterns? *Emotion*, 7(1):158, 2007.

[Scherer *et al.* 2001] Klaus R Scherer, Angela Ed Schorr, and Tom Ed Johnstone. *Appraisal processes in emotion: Theory, methods, research.* Oxford University Press, 2001.

[Schneider and Josephs 1991] Klaus Schneider and Ingrid Josephs. The expressive and communicative functions of preschool children's smiles in an achievement-situation. *Journal of nonverbal behavior*, 15(3):185–198, 1991.

[Scott 2011] G.D. Scott. Doodling and the default network of the brain. *The Lancet*, 378(9797):1133 – 1134, September 2011.

[Sharma *et al.* 1998] Rajeev Sharma, Vladimir I Pavlovic, and Thomas S Huang. Toward multimodal human-computer interface. *Proceedings of the IEEE*, 86(5):853–869, 1998.

[Sharples and others 2006] Mike Sharples et al. Big issues in mobile learning. In *Report of a workshop by the Kaleidoscope Network of Excellence Mobile Learning Initiative*. LSRI, University of Nottingham Nottingham, 2006.

[Sharples *et al.* 2005] Mike Sharples, Josie Taylor, and Giasemi Vavoula. Towards a theory of mobile learning. In *Proceedings of mLearn*, volume 1, pages 1–9, 2005.

[Shastri *et al.* 2009] D. Shastri, A. Merla, P. Tsiamyrtzis, and I. Pavlidis. Imaging Facial Signs of Neurophysiological Responses. *Biomedical Engineering, IEEE Transactions on*, 56(2):477–484, Feb 2009.

[Shavlik and Dietterich 1990] Jude W Shavlik and Thomas Glen Dietterich. *Readings in machine learning*. Morgan Kaufmann, 1990.

[Shendell *et al.* 2004] D.G. Shendell, R. Prill, W.J. Fisk, M.G. Apte, D. Blake, and D. Faulkner. Associations between classroom CO2 concentrations and student attendance in Washington and Idaho. *Indoor Air*, 14(5):333–341, 2004.

[Shernoff *et al.* 2003] David J Shernoff, Mihaly Csikszentmihalyi, Barbara Shneider, and Elisa Steele Shernoff. Student engagement in high school classrooms from the perspective of flow theory. *School Psychology Quarterly*, 18(2):158, 2003.

[Shute *et al.* 2015] Valerie J Shute, Sidney D'Mello, Ryan Baker, Kyunghwa Cho, Nigel Bosch, Jaclyn Ocumpaugh, Matthew Ventura, and Victoria Almeda. Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Computers & Education*, 86:224–235, 2015.

[Silver *et al.* 2016] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, van den Driessche George, Julian Schrittwieser, Loannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016.

[Simon 1983] Herbert A Simon. Why should machines learn? In *Machine learning*, pages 25–37. Springer, 1983.

[Smith and Ellsworth 1985] Craig A Smith and Phoebe C Ellsworth. Patterns of cognitive appraisal in emotion. *Journal of personality and social psychology*, 48(4):813, 1985.

[Smith *et al.* 2008] Kevin Smith, Sileye O Ba, Jean-Marc Odobez, and Daniel Gatica-Perez. Tracking the visual focus of attention for a varying number of wandering people. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1212–1229, 2008.

[South African Reserve Bank 2017] South African Reserve Bank. *Banknotes*. http://banknotes.resbank.co.za/banknotes, 2017. Online; accessed 2017-01-20.

[Stets and Turner 2008] Jan E Stets and Jonathan H Turner. The sociology of emotions. *Handbook of emotions*, pages 32–46, 2008.

[Strang *et al.* 1993] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[Sun *et al.* 2005] Nanfei Sun, M. Garbey, A. Merla, and I. Pavlidis. Imaging the cardiovascular pulse. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 416–421 vol. 2, June 2005.

[Sutton and Barto 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[Szeliski 2010] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[Taigman *et al.* 2014] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[Tarhini *et al.* 2015] Ali Tarhini, Mohammad Hassouna, Muhammad Sharif Abbasi, and Jorge Orozco. Towards the Acceptance of RSS to Support Learning: An Empirical Study to Validate the Technology Acceptance Model in Lebanon. *Electronic Journal of e-Learning*, 13(1):30–41, 2015.

[The Qt Company 2016] The Qt Company. *Qt – Cross-platform software development for embedded & desktop*. https://www.qt.io/, 2016. Online; accessed 2016-12-12.

[Tomkins 1962] Silvan S Tomkins. *Affect, imagery, consciousness: Vol. I. The positive affects.* Springer, 1962.

[Turing 1950] Alan. M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 1950.

[Turk and Pentland 1991] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, Jun 1991.

[van de Pol *et al.* 2011] Janneke van de Pol, Monique Volman, and Jos Beishuizen. Patterns of contingent teaching in teacher–student interaction. *Learning and Instruction*, 21(1):46–57, 2011.

[Van den Bergh *et al.* 2008] Michael Van den Bergh, Esther Koller-Meier, and Luc Van Gool. Fast body posture estimation using volumetric features. In *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*, pages 1–8. IEEE, 2008.

[Vapnik 1995] V Vapnik. *The nature of statistical learning*, 1995.

[Viola and Jones 2001] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[Viola and Jones 2004] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[Vlachos *et al.* 2002] Michail Vlachos, Carlotta Domeniconi, Dimitrios Gunopulos, George Kollios, and Nick Koudas. Non-linear dimensionality reduction techniques for classification and visualization. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 645–651, New York, NY, USA, 2002. ACM.

[Vonderwell 2003] S Vonderwell. An examination of asynchronous communication experiences and perspectives of students in an online course: a case study. *The Internet and Higher Education*, 6(1):77–90, 2003.

[Walk and Walters 1988] RD Walk and KL Walters. Perception of the smile and other emotions of the body and face at different distances. In *Bulletin of the Psychonomic Society*, volume 26, pages 510–510. Psychonomic Soc Int 1710 Fortview Rd, Austin, TX 78704, 1988.

[Wallbott 1998] Harald G Wallbott. Bodily expression of emotion. *European journal of social psychology*, 28(6):879–896, 1998.

[Wang and Shi 2005] Tiesheng Wang and Pengfei Shi. Yawning detection for determining driver drowsiness. In *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop on*, pages 373–376, May 2005.

[Wang *et al.* 2006] Qiong Wang, Jingyu Yang, Mingwu Ren, and Yujie Zheng. Driver Fatigue Detection: A Survey. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 2, pages 8587–8591, 2006.

[Wang *et al.* 2011] H. Wang, A. Klser, C. Schmid, and C. L. Liu. Action recognition by dense trajectories. In *CVPR 2011*, pages 3169–3176, June 2011.

[Weinberg 2012] Zack Weinberg. *Svm separating hyperplanes (SVG)*. https://en.wikipedia.org/wiki/File:Svm_separating_hyperplanes_(SVG).svg, 2012. Online; accessed 2017-01-20.

[Wentzel and Asher 1995] Kathryn R Wentzel and Steven R Asher. The academic lives of neglected, rejected, popular, and controversial children. *Child development*, 66(3):754–763, 1995.

[Whitehill *et al.* 2014] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J.R. Movellan. The Faces of Engagement: Automatic Recognition of Student Engagementfrom Facial Expressions. *IEEE Transactions on Affective Computing*, 5(1):86–98, March 2014.

[Witten *et al.* 2016] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[Woolf *et al.* 2009] Beverly Woolf, Winslow Burleson, Ivon Arroyo, Toby Dragon, David Cooper, and Rosalind Picard. Affect-aware tutors: recognising and responding to student affect. *International Journal of Learning Technology*, 4(3):129–164, 2009.

[Wu *et al.* 2012] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John V Guttag, Frédo Durand, and William T Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, 31(4):65, 2012.

[Yeasin *et al.* 2006] M. Yeasin, B. Bullot, and R. Sharma. Recognition of facial expressions and measurement of levels of interest from video. *IEEE Transactions on Multimedia*, 8(3):500–508, June 2006.

[Zeng *et al.* 2009] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):39–58, 2009.

[Zimmerman 2000] Barry J Zimmerman. Self-efficacy: An essential motive to learn. *Contemporary educational psychology*, 25(1):82–91, 2000.