Facial Action Unit Classification Using Weakly Supervised Learning

School of Computer Science and Applied Mathematics,
University of the Witwatersrand, Johannesburg.

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfillment of the requirements for the degree of Master of Science.

Oseluole Tobi ENABOR
(Student Number: 558039)

Supervised by
Prof. Hima VADAPALLI

June 2023

Declaration

I, Oseluole Tobi Enabor, hereby declare the content of this dissertation to be my own work unless otherwise explicitly referenced. This dissertation is submitted for the degree of Master of Science at the University of the Witwatersrand, Johannesburg. This work has not been submitted to any other university, nor for any other degree.

Signed: _____

Date: June 2023

Abstract

Deep learning has gained popularity because of its supremacy in terms of performance when trained on large datasets. However, collecting and annotating large datasets is laborious, expensive, and time-consuming. Weak supervision learning (WSL) has been at the forefront in exploring solutions to the above limitations. WSL techniques can create accurate classifiers under different scenarios, such as limited sample datasets, inaccurate datasets with noisy labels, and datasets that do not have the desired labels. This work applies WSL to facial Action Unit (AU) recognition, a problem space that relies on subject-matter experts (i.e., certified Facial Action Unit Coders (FACS)) to annotate samples. Two WSL techniques, namely incomplete supervision using a pseudo-labelling mechanism, where one has access to vast amounts of unlabelled data and a limited amount of labelled data, and inaccurate supervision using Large-Loss Rejection (LLR) mechanism, where one has access to only noisy labels, were explored. The pseudo-labelling mechanism involves feeding samples with generated pseudo-labels during the training process. Alternatively, the LLR mechanism prevents model learning noisy labels by rejecting samples that reported large-loss during training. To better evaluate the limitations posed by accurate data and label availability and its impact on training models, the authors trained a baseline emotion recognition model and fine-tuned for AU recognition using transfer learning. This process also helped access the ability to estimate fine-grain labels (AUs) using only coarse-grain labels (facial emotions).

The experimental setup included training and validating a VGG16 Convolutional neural network (CNN) using the Extended Denver Intensity of Spontaneous Facial Action Database (DISFA+) and the use of the Karolinska Directed Emotional Faces (KDEF) dataset as cross-dataset evaluation. Pseudo-labelling approach for AU recognition had three models, the first, PL-1, reported subset accuracy of 68% and 0.56 weighted F1-score, PL-2a reported a subset accuracy 89% and 0.9 weighted F1-score, PL-2b reported a subset accuracy of 66% and a weighted F1-score of 0.44. The LLR approach for AU recognition reported a subset accuracy of 69% and a weighted average F1-score of 0.66. The baseline AU model reported accuracy of 97% and an F1-score of 0.98 for AU recognition, signifying the need for large data sets and transfer learning. However, with an average reported accuracy of 68.5%, WSL mechanisms provide a solution in the right direction and can assist researchers in addressing data annotation challenges.

Keywords: Weakly supervised learning, Machine Learning, Deep Learning, Object detection, Facial Expression Recognition, Facial Action Coding System

Acknowledgements

I wish to express my deepest gratitude to God, my Creator, the Lord, God almighty for the gift of life; who has made today possible. A special gratitude to my supervisor, Prof. Hima Vadapalli for her supervision and guidance throughout this journey of completing my research work. I truly appreciate the time and effort she put into the completion of my research work. My deepest appreciation to my family, my mother (Dr. Onabolu) and brother (Ehiremen), for their never-ending support and faith. Thank you so much. To my partner, Fundile, thank you for being my confidant and just being you. Thank you for your support and care during highs and lows of this journey.

Contents

List of Figures

List of Tables

List of Abbreviations

AD Action Descriptor

ADAM Adaptive Movement Estimation Optimizer

AU Action Units

CACD Cross-Age Celebrity Dataset

CNN Convolutional Neural Network

CL Curriculum Labelling

DAM Defect Activation Map

DISFA Denver Intensity of Spontaneous Facial Action Database

FACS Facial Action Coding System

FBA Facial Behaviour Analysis

FER Facial Expression Recognition

JSON JavaScript Object Notation

KDEF Karolinska Directed Emotional Faces Database

HTL Hidden-Task Learning

IoU Intersection over Union

LLR Large Loss Rejection

ML Machine Learning

MS-MIL Multiple-Segment Multi Instance Learning

MIL Multi-Instance Learning

NMS Non-Maximum Suppression

PL Pseudo Labelling

ReLU Rectified Linear Unit

RMSE Root Mean Square Error

SHTL Semi-Hidden-Task Learning

SSE Semi-Supervised Embedding

SSD Shot Multibox detector

SSL Semi-Supervised Learning

SME Subject Matter Experts

SVM Support Vector Machine

TL Transfer Learning

TSVM Transductive Support Vector Machine

VGG Visual Geometry Group

WSML Weakly Supervised Multi-Label

WSL Weakly Supervised Learning

Chapter 1

Introduction

A strong predictive machine learning (ML) model requires a significant amount of quality data samples. These data samples are used to train and validate the model so that it is capable of making appropriate actions when the model is introduced to new data. This approach to training ML models by using annotated data is called supervised machine learning. In supervised learning, data annotation, also known as labelling, is performed to add metadata to each instance of a raw dataset [Schreiner et al., 2006]. This is a necessary step in various ML workflows as algorithms struggle to infer meaning without human intervention. Data labelling is performed to allow models to learn the underlying context or correlations to the metadata provided.

A common challenge faced by ML researchers is a lack of sufficient quality labelled data to train reliable predictive models [Mahesh, 2020; Granger et al., 2021]. Collecting and annotating large datasets is laborious, expensive, and time-consuming [Mahesh, 2020; Granger et al., 2021]. A recent approach to overcome this challenge is Weakly Supervised Learning (WSL) [Zhu and Goldberg, 2009; Zhou, 2018]. This is a blanket term used to cover a range of techniques that build predictive models through learning with weak supervision [Zhu and Goldberg, 2009; Zou et al., 2019]. In weak supervision, researchers train models by using unlabelled datasets, inaccurately labelled datasets, or datasets without the ideal level of precision in the data labels. Weak supervision is discussed in more detail in Chapter 2.

1.1 Significance of research problem

This research work addresses the data annotation challenge faced by deep learning researchers in the domain of Facial Behaviour Analysis (FBA). The findings of this

research work will explore the ways to enable the creation of robust Facial Emotion Recognition (FER) systems that incorporate Facial Action Coding System (FACS) annotation in their deep learning models. Reduction of the cost, labour, and time required to annotate datasets with FACS annotations will greatly assist FER researchers. Overall, this work takes a data-centric approach to utilise limited annotations to train models for Action Unit (AU) classification tasks.

1.2 Research context

Facial behaviour analysis is an important area of research in computer vision and affective computing because of its potential for many application areas in human-computer interaction such as mental disease diagnosis, human social interaction detection, neuromarketing, and human-computer interaction machines [Wang and Gu, 2018; Samadiani et al., 2019; Sebe et al., 2005]. There have been studies that show that one-third of human communication is through verbal means and the rest is through non-verbal means [Mehrabian, 2017]. Facial behaviour conveys a significant amount of non-verbal communication such as mental state, and this communication can be understood by understanding the relevance of movements of a person's facial muscles.

In [Ekman and Friesen, 1971], the authors conducted cross-cultural research on facial expressions, which revealed that there were six basic facial emotions common across cultures - Angry, Disgust, Fear, Happy, Sad, and Surprise. They defined a standard to cover the wide range of facial muscle movements, this standard is known as the Facial Action Coding System (FACS) [Ekman et al., 2002]. The FACS manual is a classification manual for facial expressions that defines all possible facial movements for every emotion. It consists of 32 action units (AUs) and 14 action descriptors (ADs). These AUs are the fundamental actions of individual muscles or collections of muscles that create specific facial movements. Unfortunately for deep learning applications in FER, there are a few AU annotated datasets and the process of creating them for new datasets requires a certified FACS coder. There is also a requirement that the datasets has low cross-coder variarition. The process is labour-intensive and costly. Hence, WSL can assist FER researchers who want to build robust systems that take advantage of the FACS for automated facial behaviour analysis.

1.2.1 Motivation

By developing FER systems that can generate AU annotations for FER datasets, FER researchers can build more robust FER applications. Currently, FER datasets with emotion labels are readily available, but AU-annotated datasets are scarce.

1.2.2 Hypothesis

It is hypothesised that weak supervision learning mechanisms can be used to train machine learning models to learn from incomplete or inaccurate data labels. This hypothesis is to be tested in the domain of facial emotion and action unit labelling.

1.2.3 Research questions

This section discusses the questions that aim to prove or disprove the research hypothesis stated above.

> R1: Given a limited amount of AU-annotated data samples and a significantly larger amount of unlabelled data samples, can we use WSL to train an accurate CNN that can generate AU-annotations for unlabelled test data?
>
> R2: Given an AU dataset with inaccurate annotation, can we use weak supervision to train a CNN that can generate accurate labels?

1.2.4 Research Objectives

This section discusses the objectives that need to be addressed in order to answer the questions listed above. The objectives will be listed under each question. The research objectives for R1 and R2 are as follows:

> R1: Given a limited amount of AU-annotated data samples and a significantly larger amount of unlabelled data samples, can we use WSL to train an accurate CNN that can generate AU-annotations for unlabelled test data?
>
> - Train an AU recognition model with limited amount of AU-annotated samples
> - Generate pseudo-labels for unlabelled data samples using self-training mechanism

- Re-train model with larger dataset that includes pseudo-labels
- Evaluate the performance of model on test data

R2: Given an AU dataset with inaccurate annotation, can we use weak supervision to train a CNN that can generate accurate labels?

- Generate noisy labels for AU dataset
- Train model with LLR framework to reject labels that have a large loss
- Use LLR model to generate AU-annotations for different unlabelled dataset

1.2.5 Expected contribution to knowledge

The expected contribution to knowledge that this research work aimed to achieve was to test the use of WSL in generating AU-annotations for facial images by using model trained with either a limited amount of AU-annotated samples, and a noisy dataset that contains inaccurate AU-annotations. The research also aims to provide a comparative analysis of WSL to supervised learning with good amount of labelled data.

1.3 Organisation of thesis

The rest of the thesis is organized as follows, Chapter 2 discusses the background knowledge associated with WSL and deep learning research work related to WSL. Chapter 3 discusses the operation of convolutional neural networks, evaluation metrics used, background information on WSL approaches, and transfer learning. In Chapter 4, the methodology used in this research work is presented. Chapter 5 presents the experimental results of the WSL techniques. The conclusion of this research work is discussed in Chapter 6.

Chapter 2

Background and related work

In this chapter, the background knowledge related to weak supervision is discussed. The different types of weakly supervised techniques are discussed. The background knowledge of FER is then discussed. Last, we present research work related to WSL and transfer learning.

2.1 Background

What is Weakly Supervised Learning (WSL)?
Weakly Supervised Learning refers to several machine learning approaches that attempt to create predictive models using weak supervision [Zhou, 2018]. Weak supervision is a branch of machine learning techniques in which models are trained using datasets that are incomplete, noisy, or less accurate. This branch of machine learning has gained popularity due to the high cost and intensive human labour to create manually labelled datasets [Zhu and Goldberg, 2009; Zhou, 2018; Zou et al., 2019]. Weak supervision is grouped into three categories: Inexact Supervision, Incomplete Supervision, and Inaccurate Supervision [Granger et al., 2021]. The three categories of WSL are depicted in Figure 2.1.

2.1.1 Incomplete Supervision

Incomplete supervision involves training a model with a limited number of labelled data samples along with a significantly larger number of unlabelled data samples (Figure 2.2). The two common techniques used in this form of WSL are: Active learning (AL) and Semi-Supervised Learning (SSL) [Settles, 2009; Zhu and Goldberg, 2009]. Active learning works with the assumption that there is an oracle, a subject matter expert

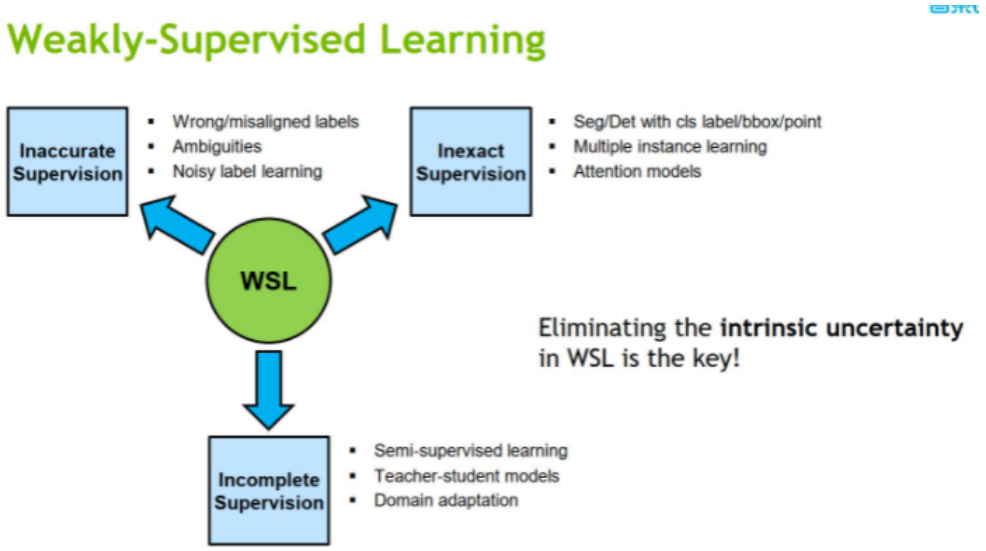


FIGURE 2.1: Three typical types of weakly supervised learning [Programmersought, 2021].

(SME) that can be queried to retrieve ground-truth labels for the selected unlabelled instances [Zhu and Goldberg, 2009]. The labelling cost depends on the number of queries made, so the goal is to minimise the number of queries. In SSL, SMEs provide labelled data, and SMEs are not queried during the training process. Example of SSL approaches include generative models, label propagation, and transductive support vector machines (TSVM). [Zhou, 2018]. Generative methods, use Gaussian mixture dis-

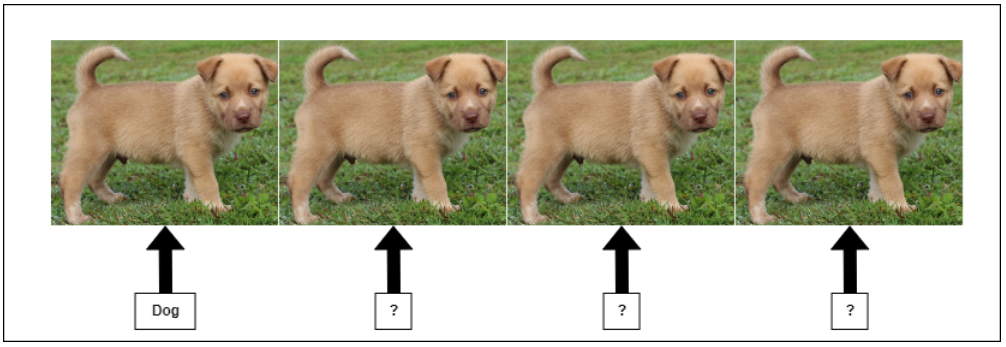


FIGURE 2.2: Incomplete Supervision: Limited labelled data with sufficiently large cohort of unlabelled data.

tribution to learn the probability with which unlabelled instances belong to a specific class of labelled instances [Ben-Yosef and Weinshall, 2018]. Label propagation methods are considered a combination of SSL and active learning in which multiple models are generated and learners teach others to use instances of unlabelled data [Zhou, 2018]. In TSVM, the unlabelled instances are labelled by searching for a decision boundary with a maximal margin throughout the data [Zhou, 2018].

2.1.2 Inexact Supervision

In Inexact supervision, some supervision information is available but not at the level of precision as desired (Figure 2.3). They are used in deep learning approaches, specifically CNNs. A typical form of this involves the use of coarse-grained labels [Zhu and Goldberg, 2009]. Multi-instance learning (MIL) is also another WSL technique where the goal is to predict the labels of unseen bags [Zhou, 2018] and the models are trained using bags. Bags are a collection of training examples that are grouped together. A bag can be labelled either positive or negative whereas the former label refers to the instance where there is at least one point in the bag that belongs to the positive class. A negative bag refers to the instance where all points in the bag belong to the negative class [Dietterich et al., 1997].

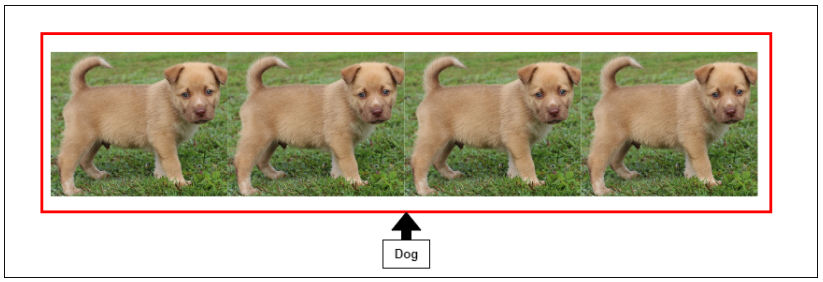


FIGURE 2.3: Inexact Supervision: Data contains high-level information,i.e., dog and not finer details, i.e., the number of instances.

2.1.3 Inaccurate Supervision

The labels in this form of WSL are not always the ground truth i.e., the labels may contain errors [Zhu and Goldberg, 2009] (Figure 2.4). The goal is to develop a model that can learn with noisy labelled data [Frénay and Verleysen, 2013]. Ensemble models are typically used to identify unlabelled instances and to check them against training data to make corrections [Frénay and Verleysen, 2013]. Two common types of inaccurate supervision are the data editing approach, which makes use of a relative neighbourhood graph to detect mislabelled data, and crowd-sourcing, which involves outsourcing the labelling process to a large group of external parties who might or might not be subject matter experts [Zhou, 2018]. Crowd-sourcing is a cheaper and much faster method for annotating datasets compared to using SMEs for annotation [Granger et al., 2021].

In this research work, we apply WSL to facial behaviour analysis where we have readily available expression labels and want to generate finer labels, AUs. Incomplete supervision and Inaccurate supervision approaches are used in this research work. We have a limited number of AU annotated samples and also a noisy AU dataset.

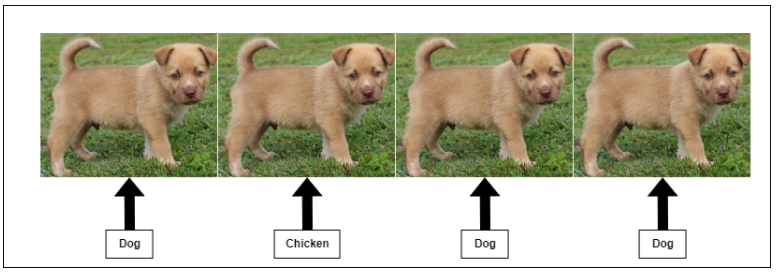


FIGURE 2.4: Inaccurate Supervision: Some of the labels contain errors or noise.

2.2 Facial Expression Recognition

In computer vision and machine learning, the goal of FER systems is to understand the emotional state or intentions of a person based on their facial expression [Li and Deng, 2020]. FER systems perform facial behaviour analysis on static images and videos. This area of research is a subset of affective computing, where models are developed to recognise and understand human emotions and affective states [Li and Deng, 2020]. There are three stages involved in FER: a) face detection, b) facial feature extraction, and c) expression classification to an emotional state.

Face detection is the first stage in FER analysis, as it locates the face in the input image or video. In facial feature extraction, facial features are extracted from the located face region - features such as skin colour, face shape, and facial expression are examples of features that are extracted. Interpreting facial expression involves using the extracted facial features to determine facial muscle movements or emotional states such as happy, sad, and angry, etc. FER is an important area of research, as it plays a significant role in application areas such as diagnosis of mental disease, detection of human social interactions, neuromarketing, and human-computer interaction [Wang and Gu, 2018; Samadiani et al., 2019; Sebe et al., 2005].

FIGURE 2.5: The facial expressions associated with the universal emotions according to Ekman [Bennett, 2015]

Six emotions were determined by [Ekman and Friesen, 1971] that were perceived the same way across different cultural groups (Figure 2.5). These emotions are anger, disgust, fear, happiness, sadness, and surprise. However, recent research has argued that these emotions are actually culturally specific and should not be considered universal [Barrett, 2006; Coan, 2010]. FER systems that determine the state of emotion by using just these basic emotions are limited in their ability to detect complex and subtle changes in facial movement [Vadapalli, 2011]. Hence, why creating automated FER systems that make use of FACS is an important research area.

The FACS was developed by the Swedish anatomists Carl-Herman Hjortsjo, Paul Ekman, and Wallace Friesen [Ekman and Friesen, 1971], to code facial movements. The FACS codes the movement of the individual facial muscles by noting slight changes in the facial appearance. There are many use cases in facial behaviour analysis that use FACS to categorise facial expressions. Using FACS, facial expressions can be broken down into anatomic units called action units. There are 32 AUs and 14 additional action descriptors (ADs) (Figure 2.7). The AUs are defined as contractions or relaxations of facial muscles to form a specific facial movement [Ekman and Friesen, 1978]. The classification of AUs according to FACS is based on facial location and facial action muscles involved in facial movement. The developers of FACS divided the entire face into separate regions, e.g., the glabella, the root of the nose, the eye cover lid, the lower eyelid furrow, etc. The two dominant groups are upper-face AUs and lower-face AUs [Vadapalli, 2011]. Upper face AUs are made up of eyebrows, forehead, and eyelids, while lower face AUs are made up of the nasolabial, mouth, and chin region (Figure 2.6). The FACS uses intensity scoring to determine the intensity of an AU once it has been detected on the face. The intensities are coded with letters A to E, where A is the minimum intensity and E is the maximum intensity.

The value of FACS, however, depends on the skill level and experience of the FACS coder. Coders must study the FACS manual and then take a test to become certified FACS coders. However, the certification process is time intensive, and prospective FACS coders need around 100 hours of study and practice to achieve the minimum competency level [EIA, 2022]. A FACS coder approximately requires an hour to code for every minute of video. The time requirement and small pool of FACS coders hinder situations in which FACS can be used [EIA, 2022]. The FACS is still valuable as it is considered the most reliable form of facial measurement, as it is an exhaustive classification standard [EIA, 2022].

The advancement of deep learning architectures and computing capacity has made it

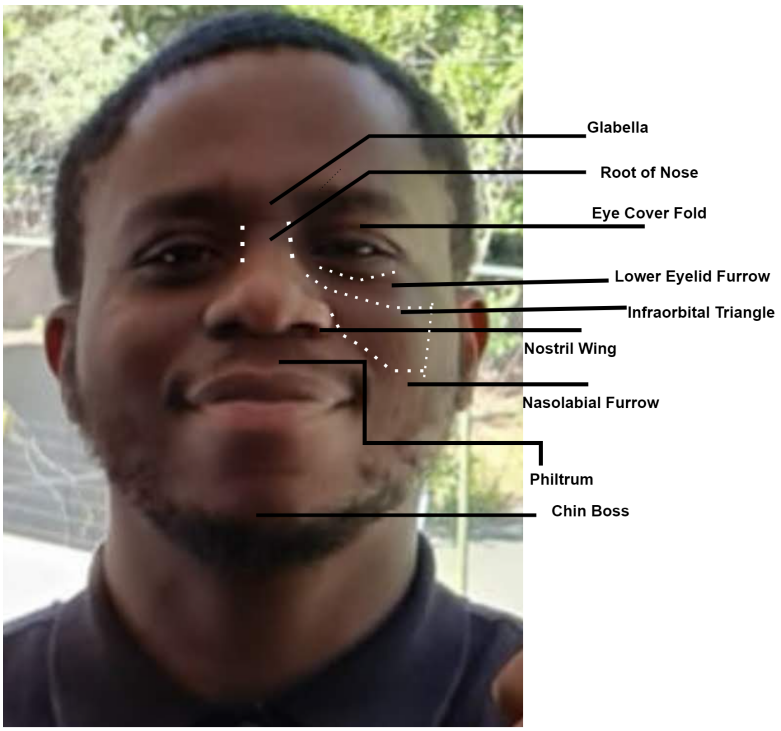


FIGURE 2.6: The different facial areas and their associated names [Ekman et al., 2002].

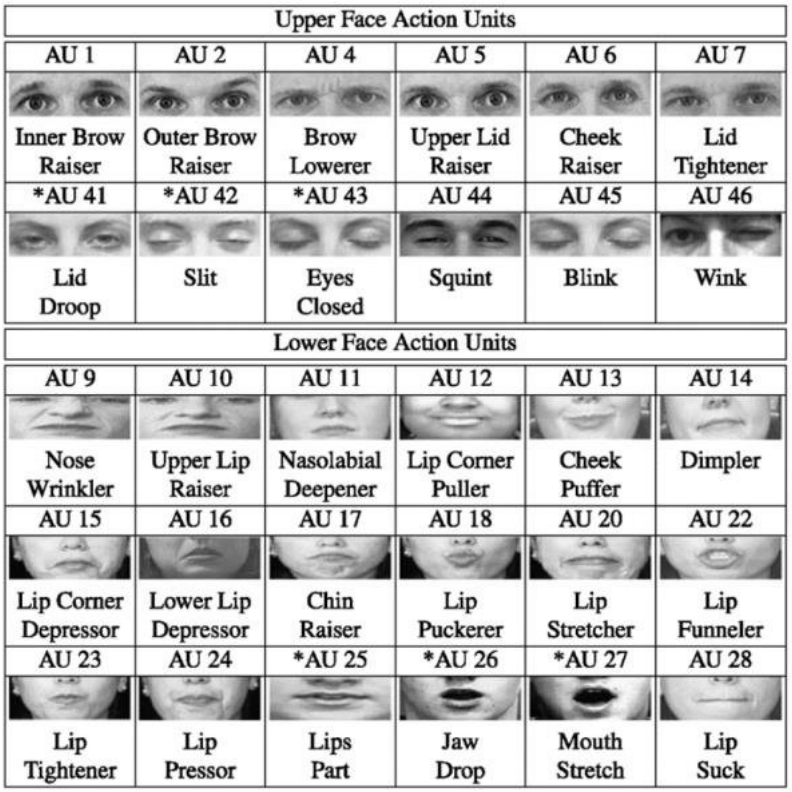


FIGURE 2.7: The upper face action units and lower face action units according to FACS [li Tian et al., 2004]

possible for researchers to create intelligent FER systems. However, deep learning requires large volumes of training data that are accurately labelled for the task. The AU annotation process for deep learning solutions is laborious, time-intensive, and expensive. Hence, this research will explore research into weak annotations is an important area for facial behaviour analysis so researchers can implement deep learning solutions.

2.3 Related Work

The WSL techniques that have been used for FBA are presented in the related works mentioned below. These related works will look at techniques such as Pseudo Labelling (an incomplete supervision technique), Large Loss Rejection for weakly supervised multi-label classification, multi-instance learning, hidden task learning and semi-hidden task learning, and others. Finally, we present research work that has used transfer learning in FBA.

2.3.1 Pseudo Labelling and Self Training

Werner et al.[Werner et al., 2020] proposed an approach for AU recognition in the wild. Their system uses self-training, multitask learning, and a heterogeneous ensemble with three CNN architectures. They use a weighted loss to handle data imbalance, a detailed face alignment for expression AUs, and context face alignment for head pose AUs. The authors implemented multi-task learning using two output neurons per AU, one for each training subset used in their research. Their self-training approach involved using a teacher model trained on a labelled dataset. The trained model was used to predict pseudo-labels on a larger unlabelled (weakly/ partially labelled) dataset. The authors used predicted pseudo-labels and manual labels to train a student model. Noise was introduced into the student model by dropout and data augmentation to learn better than the teacher model. The authors reported an accuracy of 91.24% and an F1 score of 54.78%, placing them second in the EmotioNet 2020 Challenge.

In [Cohen et al., 2003], the authors set out to detect facial expressions using an incomplete supervision technique. They investigated the effect of unlabelled data on classification performance. They used a Bayesian network as a learning algorithm to perform the recognition task. They achieved 75% classification accuracy using unlabelled data and 83% accuracy with labelled data. They concluded that if there were more unlabelled data available, they would have achieved the same performance as the classifier trained on labelled data.

In [Wang et al., 2016], the authors proposed a semi-supervised framework, Cost-Effective Active Learning (CEAL), for deep image classification tasks that was incrementally trained with a limited amount of labelled training data and could capture the optimal representation of features. The framework uses a sample selection strategy where the minority of the most informative samples are selected, and the majority of the high-confidence samples are used as pseudo-labels to update the model learning. The minority of the labelled samples are used to benefit the decision boundary, and most pseudo-labels are used as training data for robust feature learning. The CEAL framework was evaluated using the CACD dataset [Chen et al., 2015] and the Caltech-256 object categorisation dataset [Griffin et al., 2007]. They compared the performance of CEAL with other state-of-the-art active learning methods. The authors could demonstrate that CEAL needed a smaller percentage of labelled data to achieve state-of-the-art object recognition compared to the other methods. In [Zeng et al., 2018], the authors propose a solution to address errors and biases in FER datasets that affect performance and limit the improvement of FER methods. Their proposed framework, Inconsistent Pseudo Annotations to Latent Truth (IPA2LT), used multiple noisy labelled datasets and large-scale unlabelled data to address the errors and biases in FER datasets. They combined human annotations with model predictions to train a FER model, which was used to uncover the latent information in the inconsistent pseudo labels and the facial image inputs. Their proposed method achieved test accuracy of 87.23% on CIFAR-10 noisy label dataset [Krizhevsky et al., 2009].

In Kim et al. [Kim et al., 2022], the authors present three novel methods for weakly supervised multi-label classification (WSML) that reject or correct large loss samples to ensure that noisy labels do not influence the network's output. In the rejection method, they introduce a weight term, λ, to deal with large loss samples by setting the weight term to zero so that the large loss does not influence the model. The rejection threshold rate is gradually increased during the training process, and a hyperparameter that determines the speed of increase of the rejection rate is also incorporated. Two approaches for the correction method namely: temporary correction and permanent correction, were employed. For temporary correction, the label of the large loss sample is temporarily changed, and the new loss is used to update the model weights. For the permanent correction method, the large loss value label is permanently changed and used in the next training epoch. The authors compared the performance of their approach with other state-of-the-art WSML approaches, such as Weak Assume Negative and Curriculum Labelling (CL) [Cascante-Bonilla et al., 2021]. The authors achieved a mean average precision of 0.83 for the three techniques when evaluated using the PASCAL Visual Object Classes Challenge 2012 (VOC 2012) dataset [Dong et al., 2021].

In [Sikka et al., 2014], the authors could tackle the problem of pain classification and localisation using weakly labelled pain videos. Their approach combined MIL with multiple segment representation (MS-MIL) to locate the presence or absence of pain in a given image frame and to determine the time at which the expression occurred and to determine its duration. Their approach achieved state-of-the-art performance with a mean classification accuracy of 84.55%.

In [Ruiz et al., 2015], the authors set out to develop an AU classifier using expression labels by using the relationship between AUs and expressions. With their Hidden Task Learning (HTL) and Semi-Hidden Task Learning (SHTL) frameworks, they map each input sample to an AU, which they mapped to an expression. In their approach, they consider detecting AUs as a hidden task as AU labels are not present. They train a classifier for each expression and use the empirical relationship between expressions and AUs to use gradient descent to learn AU classifiers. They could prove that their technique improved the generalisation capability of AU classifiers. They achieved a mean AUC of 75% and 73.6% with SHTL and HTL, respectively.

In [Mollahosseini et al., 2016b], the authors' research centered on the consistency of inexact annotations of images collected from the internet and how well deep learning networks handled the noisy data labels. They trained AlexNet [Krizhevsky et al., 2012] and WACV-Net [Mollahosseini et al., 2016a] with three strategies, a) training data with true labels only, b) training data with true and noisy labels, and c) true and with noisy labels with noisy modelling. They evaluated the performance of each strategy using a test set with true labels. The results of their experiments showed that deep learning neural networks were capable of recognising wild facial expressions with an accuracy of 82.12%.

2.3.2 Transfer Learning

Zhou and Shi [Zhou et al., 2017] presented two transfer-learning models for AU intensity estimation for the Facial Expression Recognition and Analysis (FERA 2017) sub-challenge. It tasked the participants of the FERA challenge with estimating AU occurrence and intensity for nine different pose angles. The first model was a pose-invariant model, and the second model was a pose-dependent model. In the pose-invariant model, each AU was concatenated with the bottom five stages of VGG16 and a two-layer regression network that has been randomly initialised. The network is used

to learn the pose-invariant AU-specific and region-specific features. To train the pose-dependent model, the authors group the nine poses into three pitch-dependent groups and train three pose-dependent regressors. They trained an auxiliary pose estimator with three softmax outputs that approximate the probability that the input image belongs to each pose group. The regressors and estimator share the same bottom layers. The two tasks use the same transferred bottom layers of a CNN pre-trained on ImageNet. The pose-invariant model achieved a mean of 0.842 root-mean-square error (RMSE) on the validation set, while the pose-dependent model achieved a mean of 0.823 RMSE.

Almaev et al. [Almaev et al., 2015] used transfer learning and multitask learning to address the problem of creating person-specific AU detection models. The authors transfer the information of annotated data for a specific facial expression or AU to build person-specific models of other facial expressions or AUs. They do this using no or little annotated data. This was done using multitask learning and transfer learning framework to use the latent-related information. This methodology is inspired by Grouping and Overlap in Multi-Task Learning (GOMTL). GOMTL is used to find the latent relations in the dataset when organised in matrix form. GOMTL was used to create person-specific models for one specific AU (independent of other AUs). The authors extended GOMTL to consider the relationship between subjects and AU. They related and constrained the latent relations that were learnt for each AU using GOMTL. Their approach achieved a 0.75 F1 score on the DISFA dataset and a 0.43 F1 score on the McMaster dataset.

2.4 Conclusion

From the related works presented, it is evident that CNNs are used extensively in deep learning facial behaviour analysis, and WSL has gained traction within various research areas. This research work will use incomplete supervision and inaccurate supervision techniques to generate AU annotations on an unlabelled dataset. Emotion recognition datasets are readily available, therefore we can use these datasets as unlabelled and combine them with a limited AU annotated dataset. Pseudo-labelling, a semi-supervised technique will be used for incomplete supervision while Large loss rejection will be used for dealing with inaccurate supervision. In Chapter 3, the operation of CNNs, such as what makes them successful in computer vision tasks, and the

parameters that need to be optimised to improve their effectiveness, are discussed in detail.

Chapter 3

Background Methods

In this chapter, we will discuss background knowledge of convolutional neural network architectures along with the hyperparameter optimisation techniques used on these networks. We will then introduce the WSL approaches used to accomplish the research objectives. Finally, we discuss the general evaluation metrics used to evaluate the performance of the model.

3.1 Introduction

The Convolutional Neural Network (CNN) is a type of artificial neural network, with deep feedforward architecture and has notable generalisation capability compared to other networks. They can learn abstract features of objects such as spatial data. The modern framework of a CNN was introduced by Yann Lecunn et al., in the paper [LeCun et al., 1998]. CNNs have achieved state-of-the-art performance in various fields related to pattern detection from natural language processing (NLP) to computer vision. They can process different data formats: 1D for signals and sequences, including text and language; 2D for images or audio; 3D for video or volumetric images. The typical CNN is composed of one or multiple convolutions and pooling layers followed by one or several fully connected layers and finally an output layer. Figure 3.1 shows the structure of the modern CNN.

The input layer receives the input data, which is an image for computer vision applications. The image is represented as a three-dimensional matrix: width, height, and colour channel (Figure 3.2). The convolution layer is the core computational block of CNN; this layer is responsible for learning the representation of features from the input data. The convolutional layer accomplishes this by utilising one or more feature detectors or filters. These feature detectors calculate different feature maps through the

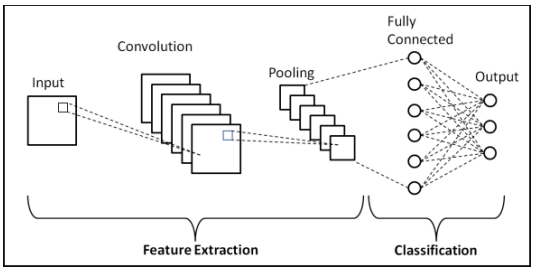


FIGURE 3.1: Schematic diagram of a basic convolutional neural network architecture [Phung and Rhee, 2018].

convolution process. For each unit in the feature map, a connection to the receptive field from the previous layer is present. This sharing of parameters helps reduce the model's complexity. The convolution operation is used in many applications in sci-

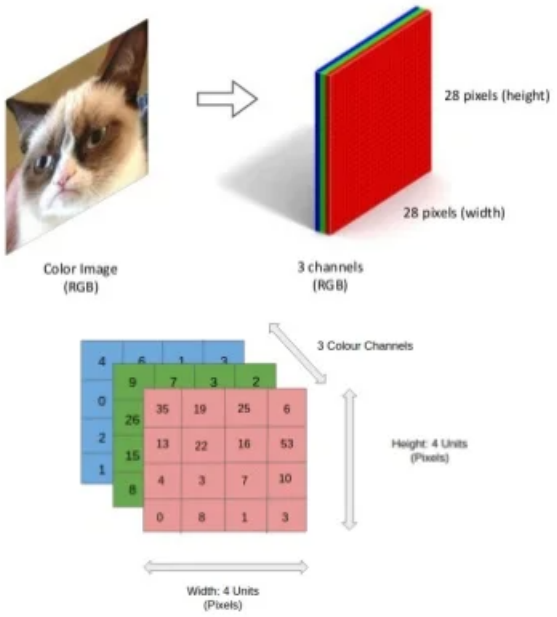


FIGURE 3.2: A colour image represented as a 3D tensor [Earnshaw, 2017].

ence, engineering, and mathematics. The convolution operation blends one function with the kernel function to produce an output that is more interpretable. The convolution operation can be defined formally for the continuous 1D dimension as:

$$l(t) = (f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{3.1}$$

where l is the feature map, g represents the kernel function, f is the input data. The discrete definition of convolutional operation for 1D:

$$[f * g][t] = \sum_{t=-\infty}^{+\infty} g(t)h(n-t) \tag{3.2}$$

where t and n are integers. In CNNs, the discrete 2D convolutional operation is often used:

$$[\mathbf{A} * \mathbf{B}][j_1, j_2] = \sum_{k_1} \sum_{k_2} \mathbf{A}(k_1, k_2)\mathbf{B}(j_1 - k_1, j_2 - k_2) \tag{3.3}$$

The feature detector is a 2-dimensional array of fixed weights that represent different parts of the input image [LeCun et al., 1998]. The feature detector moves across the whole image and extracts features by performing a convolutional operation. The convolutional operation involves the dot product between the weights of the feature detector and the input image pixels (Figure 3.3). The result of this operation is known as the feature map. The following are parameters used to optimise the performance of the feature detector and the convolutional process:

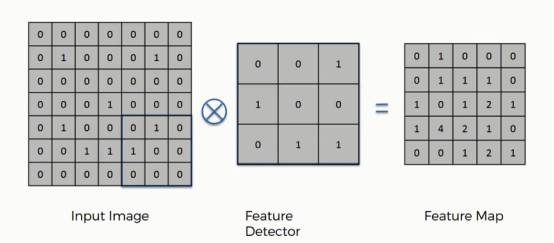


FIGURE 3.3: Visualisation of operating convolutional neural network layers [Chakrabarty and Chatterjee, 2019].

1. Depth: The depth refers to the number of feature detectors used in the convolutional operation and is directly proportional to the volume of the feature map. If three feature detectors are used, there will be three feature maps.

2. Stride: This parameter describes the number of pixels the feature detector will move horizontally (and vertically when convolved over the next row). Figure 3.4 shows a feature detector moving with a stride of 1 over the input image pixels.

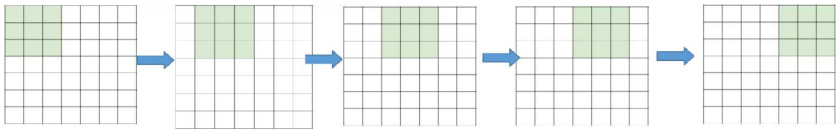


FIGURE 3.4: Feature detector in convolution operation with a stride of 1 [Albawi et al., 2018].

3. Zero-padding: This occurs when the feature detector dimensions do not match the input dimensions. When this happens, zero padding sets all pixels outside the input dimensions to zero. There are other types of padding: valid padding, same padding, and full padding.

The feature map (or activation map) is the output from the convolutional process, and the dimensions of the feature map depend on the stride and padding used. The

padding enables information at the edges of the image to be preserved with the same quality as those in the middle. The dimensions of a feature map can be determined using the following formulas:

$$w = \frac{w_{in} - K + 2P}{S} + 1 \tag{3.4}$$

$$h = \frac{h_{in} - K + 2P}{s} + 1 \tag{3.5}$$

where h_{in}, w_{in} are the input height and width, K is the kernel size, P is padding, and S is the stride.

Once a convolution operation is completed, the feature map is sent to the Rectified Linear Unit (ReLU) activation function. The activation function adds nonlinearity to the network and decides whether or not a neuron should be activated. Activation functions can be categorised into: Binary step function, linear activation function, and nonlinear activation functions. Nonlinear activation functions are more commonly used in computer vision applications. The commonly used activation functions are shown in Figure 3.5. The pooling layer, also known as the subsampling layer, performs dimensional reduction by performing aggregation on the feature map generated by the convolutional layer. The height and width of the feature maps are reduced but the depth or number of maps remains unchanged. The possible aggregation functions include max pooling, min pooling, average pooling, etc. By performing this dimension reduction on the feature map, the computational complexity is reduced as there are fewer parameters to be calculated. The reduction allows for better generalisation ability by reducing the risk of over-fitting. Figure 3.6 shows both average and max pooling operations.

The fully connected layer is structured similar to a traditional neural network. Each node connects directly to every node in the previous and next layer [Albawi et al., 2017]. The fully connected layer handles classification tasks using the features extracted from the previous pooling and convolutions layers. The final fully connected layer is where the final classification occurs, different loss functions are used to calculate the model's prediction error. The prediction error shows how well the model's prediction compares to the actual output. This prediction error is optimised during the learning process. The calculated error, or loss, is the difference between the model prediction and the actual output, which is the label assigned to the instance. There are different loss functions, and their use is problem dependent. In classification tasks, a common activation function used in the last layer is the Softmax activation function. A CNN is basically a stack of multiple layers, which include convolution layers, pooling layers, and fully connected layers.

FIGURE 3.5: Neural network activation functions [Baheti, 2023].

3.2 Hyperparameter Optimisation

Hyperparameter optimisation is the process of adjusting various factors that influence the training results of the deep learning network. This is an iterative process, which required us to run each experiment several times in order to get the most optimal

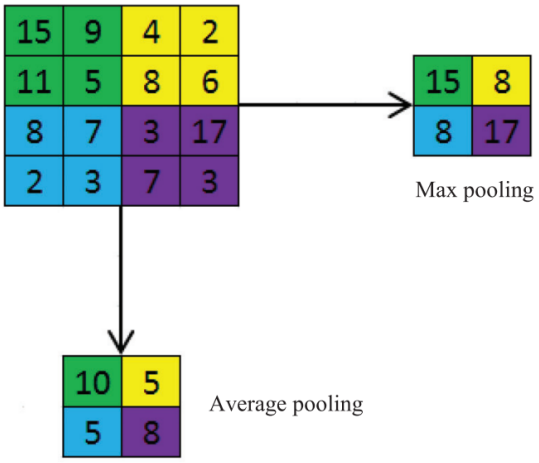


FIGURE 3.6: Visualisation of average versus max pooling [Rawat and Wang, 2017].

solutions for the network. The hyperparameters optimised in this research work are discussed below.

3.2.1 K-fold Cross Validation

Cross-validation is a machine learning technique used to evaluate and test the performance of an ML model on unseen data. In K-fold cross-validation, the data set is shuffled and split into k equal sample sets. One of the sample set is used as the test set and the other sample sets are used as training the model. This process is repeated until all k-parts have been used as a test sample. The factor k is determined during hyperparameter tuning.

3.2.2 Epochs

An epoch refers to a full cycle that a CNN performs through the data set. In a single epoch, the error of the model prediction on each training sample is used to update the model's weights. An epoch consists of several mini-batches, the number of batches is dependent on the data size and batch size. The number of epochs used influences whether a model underfits or overfits the training dataset. An underfiting scenario occurs when a model was unable to adequately learn the features in the training data. An overfit scenario occurs when a model memorises the training data but does not have good generalisation capability and performs poorly when introduced to unseen datasets.

3.2.3 Batch Size

The batch size hyperparameter is the number of training samples that a model examines before updating the model weights. It is crucial that each batch includes samples from each class. This prevents large changes in the metric between iterations. The batch size influences the training performance. A large batch size can cause the GPU/CPU to crash because of insufficient memory. The training time decreases with increasing batch size. However, if too large a batch size, this will negatively affect the model's generalisation ability.

3.2.4 Loss Function

The loss function is used to penalise the model for incorrect predictions during the training process. The goal is to minimise the value of the chosen loss function. Common loss functions used in classification problems include various forms of the Cross Entropy Loss.

3.2.5 Early Stopping

Early stopping is a neural network regularisation technique that monitors the performance of the model during training. This stops the training process when the model performance no longer yields significant improvement with more training.

3.2.6 Learning Rate

The learning rate hyperparameter controls the change of the model's weights based on the prediction error at the end of each epoch. A learning rate that is too high will cause the learning to exceed the minimum, while a learning rate too low will increase the training time or cause the learning to get stuck at an unwanted local minimum. It is common for learning rates to remain constant during the training process, but typically a learning rate schedule is used to update the learning rate value during the training at each epoch or iteration.

3.2.7 Optimiser

In machine/deep learning, an optimiser is a mathematical optimisation algorithm that is used to minimise the prediction error of a model. The most common optimisation algorithms include stochastic gradient descent (SGD) [Bottou, 2012], Root mean squared propagation (RMSProp) [Nugroho and Yuniarti, 2022], Adagrad [Lydia and Francis, 2019], Adadelta [Zeiler, 2012], Adam [Kingma and Ba, 2014], Adamax [Kingma and Ba, 2014], and many more.

3.3 VGG Network

There are several variants of CNNs in use today that have reached state-of-the-art performance in computer vision applications. The Visual Geometry Group Network (VGGNet) is a standard deep CNN architecture that has multiple layers. There are two standard configurations of the VGG network: VGG-16 (Figure 3.7) and VGG-19, which refers to the number of convolutional layers in the network. Researchers, Karen Simonyan and Andrew Zisserman, from the University of Oxford, proposed the model, in their paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition" [Simonyan and Zisserman, 2014]. The authors investigated the effect of the depth of a convolutional network on its accuracy in image recognition. They found that increased depths of a network with small (3x3) convolutional filters improve the accuracy of the network in image recognition. The network secured first and second place in the localisation and classification tracks of the 2014 ImageNet Challenge [Deng et al., 2012]. The VGG16 network has thirteen convolutional layers and three fully con-

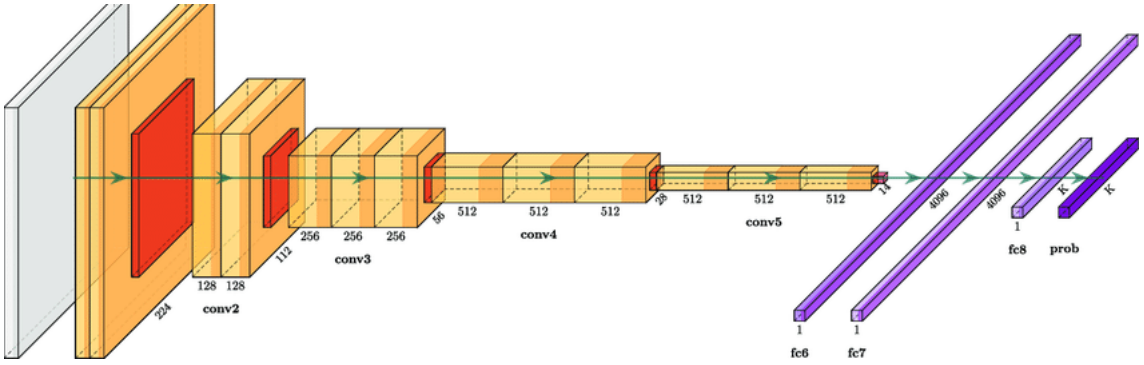


FIGURE 3.7: The structure of VGG16 network architecture [Blauch et al., 2021]

nected layers. The description and operation of these layers are discussed below.

1. **Convolutional Layers 1 and 2:** The dimensions of the input data, an image x_i, is 224x224x3. This block has 2 convolutional layers both with 64 feature detectors

(kernels) with size 3x3. This layer uses same padding and is both activated with the **ReLU activation function**. The output of these layers is a feature map with a volumetric size of 224x224x64. This volume is fed to the Maximum pooling layer. The maximum pooling layer has a kernel size of 2x2 with a stride of 2. The dimensions of the volume are reduced to 112x112x64.

2. **Convolutional Layers 3 and 4:** The input dimension is 112x112x64. This block has 2 layers that both have 128 feature detectors (kernels) which have a size of 3x3. These layers both use same padding and are also activated with the **ReLU** activation function. The feature map has a volume of 112x112x128. This volume is reduced by the maximum pooling layer with size 2x2 and stride of 2. The volume is reduced to 56x56x128.

3. **Convolutional Layers 4, 6, and 7**: The dimensions of the input data to this block is 56x56x128. This block has 3 convolutional layers with 256 kernels of size 3x3. The layers are activated with the ReLU activation function. The feature map volume is 56x56x256 which is reduced by a maximum pooling layer with a kernel size of 2x2 and stride of 2. The volume is reduced to 28x28x256

4. **Convolutional Layers 8, 9, and 10**: The input dimension to this layer is 28x28x256. The layers all have 512 kernels of size 3x3 with same padding. The layers are activated with the ReLU activation function. The feature map of volume 28x28x512 is reduced by a maximum pooling layer with a kernel size of 2x2 and a stride of 2. The output volume dimension is reduced to 14x14x5112.

5. **Convolutional Layers 11, 12, and 13**: The input dimension to this layer is 14x14x512. The layers all have 512 kernels with size 3x3 and all have same padding. The layers are all activated with the ReLU activation function. The feature map of volume 14x14x512 is reduced by a maximum pooling layer with size 2x2 and a stride of 2. The volume is reduced to 7x7x512.

6. **Fully Connected layers**: The input volume of 7x7x512 is flattened to a fully connected layer with 7x7x512 = 25088 units. The 2 layers with 40966 units are added and both are activated using the ReLu activation function.

7. **Output Layer**: The output layer has 1000 units which was the number of class categories in the ImageNet Challenge [Russakovsky et al., 2015]. This layer is activated using the SoftMax activation function.

The following layers can also be added to the standard VGG-16 Network.

1. **Batch Normalisation Layer**: Szegedy et al. [Szegedy et al., 2015] proposed this technique to address the internal covariate shift in the training of neural networks

(Figure 3.8). During the training of neural networks, every layer of a neural network has inputs with a comparable distribution and the random initialisation of parameters and input data affects these inputs. The internal covariate shift is the term that is used to describe the change in a layer's inputs and distribution during training. The internal covariate shift makes the training of neural networks slow due to lower learning rates and parameter initialisation. Szegedy et al. empirically proved that batch normalisation improves the speed of training neural networks and allows for higher learning rates and less care on parameter initialisation [Szegedy et al., 2015].

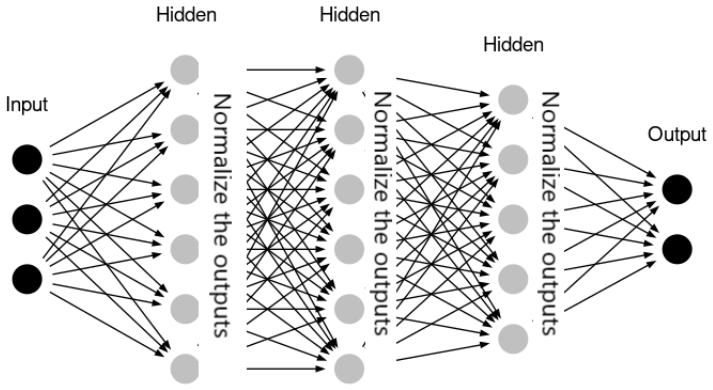
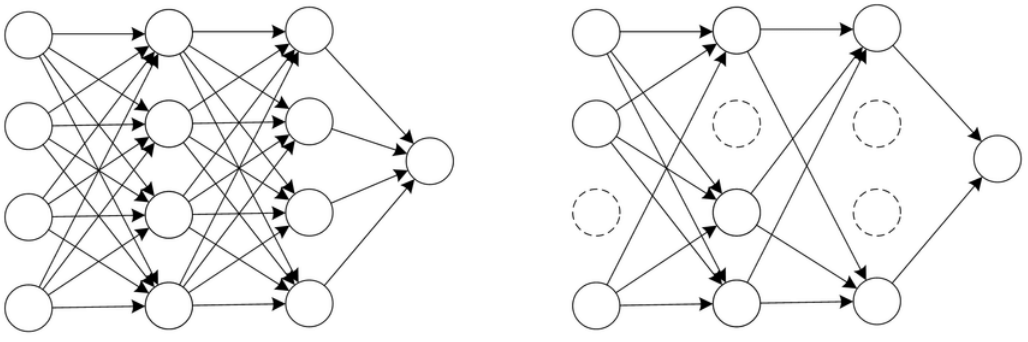


FIGURE 3.8: Batch normalisation applied to a standard neural network [DeepAI, 2019].

2. **Dropout Layer**: The Dropout technique was introduced by [Hinton et al., 2012] in 2012 and is used to reduce the likelihood of the deep learning model over-fit to data. It does this by selecting neurons at random to be ignored during training (Figure 3.9). Dropout prevents the co-adaptation of feature detectors [Baldi and Sadowski, 2013].



(a) Standard Neural Network

(b) Network after Dropout

FIGURE 3.9: Implementation of dropout technique. a)standard neural network before dropout. b) neural network after applying dropout of random nodes [Khalifa and Frigui, 2016]

3.4 Facial Action Unit Recognition Approaches

In the following sections, we discuss the two WSL approaches used for AU recognition in this research work. The first approach is Pseudo Labelling, which is used in the incomplete supervision scenario where we have a limited amount of labelled data and larger amount of unlabelled data available to use. The next WSL approach will be Large Loss Rejection which is used in the inaccurate supervision scenario where we only have a dataset with partially correct labels to use. Lastly, we discuss the transfer learning approach used for action unit recognition.

3.4.1 Pseudo Labelling

Authors in [Lee et al., 2013] proposed a simple and efficient semi-supervised learning method for deep learning networks. This semi-supervised technique uses a small sample set of labelled data together with a significantly larger unlabelled sample set to improve the model's classification performance. Semi-supervised learning is based on the cluster assumption which states that the decision boundary is located in low-density areas to improve the generalisation performance [Chapelle and Zien, 2005]. Techniques such as Semi-Supervised Embedding (SSE) [Weston et al., 2008] and Manifold Tangent Classifier (MTC) [Rifai et al., 2011] work on this assumption. The initial model trained with the labelled data helps the model generalise the structure of the data. However, if the model performs poorly in learning the structure of the labelled data, incorrect labels can be assigned to the unlabelled data. Pseudo labelling method performs poorly when the labelled data is not a true representation of the entire dataset. This can occur when their are outliers, classes of the dataset are missing or the amount is too small for the model to learn the characteristics of these classes.

Pseudo-label is advantageous because it does not require a complex training strategy and similarity matrices which are computationally expensive. This technique has been proven to achieve state-of-the-art performance.The process (Figure 3.10) includes the following stages:

1. A model is trained on a batch of labelled samples
2. The trained model is used to predict labels (pseudo labels) on a batch of unlabelled samples. The predictions with the highest confidence are selected.
3. The predicted labels from the previous step are used to calculate the loss on the new set (labelled samples + pseudo labelled samples) is calculated using Equation 3.7

4. The calculated loss is back propagated to update the model's weight

$$y_i^{'} = \begin{cases} 1, \text{ if} i = argmax_{i'} f_{i'}(x) \\ 0, \text{otherwise} \end{cases} \tag{3.6}$$

For the unlabelled sample set, the pseudo-labels weights (recalculated at every update) are used in the loss function used for the supervised learning task. There is not an even balance of labelled and unlabelled samples. α is a weight used to balance the effect of unlabelled samples on the overall loss function. The overall loss function for this method is defined as

$$\mathbf{L} = \frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{C} \mathbf{L}(y_i^m, f_i^m) + \alpha(t) \frac{1}{n'} \sum_{m=1}^{n'} \sum_{i=1}^{C} \mathbf{L}(y_i'^m, f_i'^m) \tag{3.7}$$

where n is the labelled data batch size during stochastic gradient descent, n' for unlabelled batch size, f_i^m is the model's batch prediction for the labelled batch, y_i^m is the labels for the batch, $f_i'^m$ unlabelled batch, $y_i'^m$ pseudo-label for unlabelled batch. The

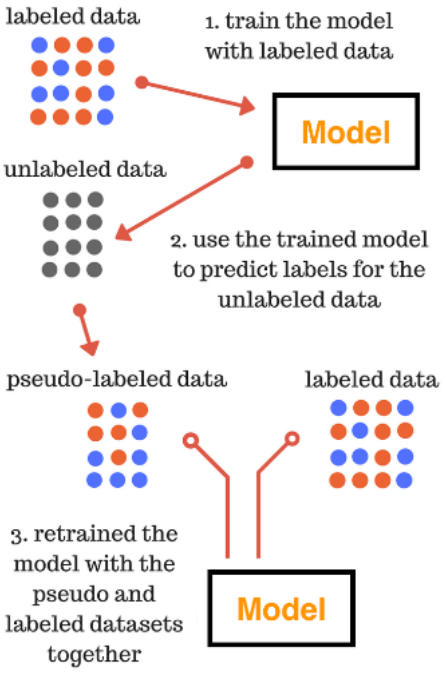


FIGURE 3.10: The pseudo-labelling algorithm for training AU detection model [Kodzoman, 2017].

scheduling of α is an important hyperparameter as it affects the model performance. Authors in [Lee et al., 2013] proposes that α should be gradually increased using the deterministic annealing process. This gradual increase in α assists the optimisation process to avoid bad local minima and allows the pseudo-labels of the unlabelled data

to be similar to true labels. $\alpha(t)$ is adjusted as given below:

$$\alpha(t) = \begin{cases} 0, \text{ if } t < T_1 \\ \frac{t-T_1}{T_2-T_1}\alpha_f, \text{ if } T_1 \leq T_2 \\ \alpha_f, \text{ if } T_2 \leq t \end{cases} \tag{3.8}$$

3.4.2 Large Loss Rejection (LLR)

Large loss rejection is a weakly supervised multi-label classification approach proposed in [Kim et al., 2022]. This approach is used to handle noisy datasets where their erroneous labels are present. In this label correction approach, the labels that have a large loss are rejected to prevent the model from learning noisy information. The loss used for this approach is the binary cross-entropy given as:

$$L = \frac{1}{|D|}\sum \frac{1}{K}\sum_{i=1}^{K} l_i \times \lambda_i \tag{3.9}$$

where l_i is defined as the binary cross entropy loss per label. λ_i determines how much the loss for this label contributes to the overall loss for the sample. K denotes the number of samples used. In large loss rejection, a rejection rate threshold $R(t)$ is used to determine which noisy labels should be rejected. In a sample prediction, if l_i is greater than $R(t)$, then l_i is set to zero by setting the weight term λ_i to 0 so it does not influence the calculation of the model gradients.

$$\lambda_i = \begin{cases} 0, \text{if } l_i > R(t) \\ 1, \text{if otherwise} \end{cases} \tag{3.10}$$

where t is the current epoch in training and $R(t)$ is the rejection threshold which is R(t)% of the largest loss value.

$$R(t) = [(t-1).\Delta_{rel}] \tag{3.11}$$

The Δ_{rel} hyperparameter controls the speed at which the rejection rate increases. The rejection of loss values only occurs after the first training epoch, so the model learns the correct patterns.

3.4.3 Transfer Learning

Transfer learning leverages the features (knowledge) learnt from a previous task for a new but related problem. The aim is to make ML models that can mimic how humans can use the knowledge gained from another task and use it for a different but related task [Torrey and Shavlik, 2010]. This approach is popular in computer vision and is used when there is not enough data available to train a model from scratch. Improvement in learning achieved through transfer learning is measured in three com-

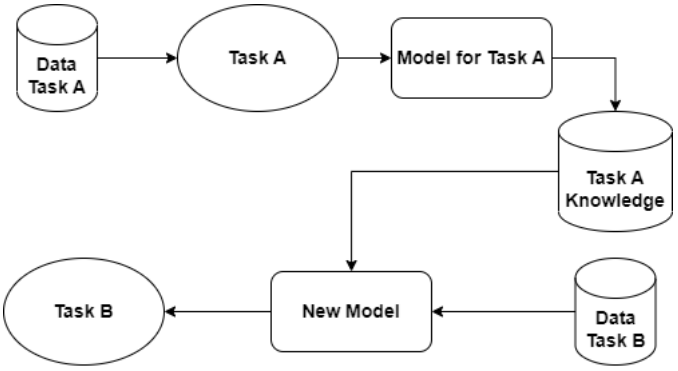


FIGURE 3.11: Transfer learning involves learning with an additional source of information related to the standard training data: knowledge from one or more related tasks.

mon ways. The first is the improvement in initial performance in the target task using transferred knowledge, before more learning is done, compared to the initial performance without the transferred knowledge. Second is the reduction in time that it takes to learn the target tasks when given transferred knowledge compared to the time taken to learn the task from scratch. The third is the final level of performance achievable in the target task compared to the performance without the transferred knowledge [Torrey and Shavlik, 2010].

Transfer learning is a useful technique for ML applications where training data is expensive or difficult to get [Weiss et al., 2016]. In transfer learning, the training data and target data are sub-domains that are related to a high-level common domain. For this research work, the training data is expression images and the target data is AUs, these two are related as both are a subset of facial behaviour analysis. Various ML applications have used TL successfully, such as text sentiment classification [Cao et al., 2021; Chen and Qian, 2019], human activity classification [Du et al., 2018], and facial behaviour analysis [Romera-Paredes et al., 2013]. Emotion datasets are available which can be utilised for transfer learning. Here, a model can take advantage of the information gained from emotion recognition and use the weights learned to recognise AUs

enabling transfer of knowledge[Lewinski et al., 2014]. This process is illustrated in Figure 3.12.

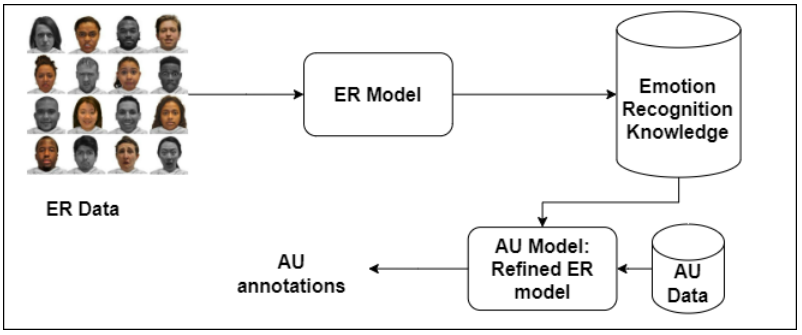


FIGURE 3.12: Transfer learning in FER where emotion recognition is the source task and action Unit detection is the target task.

3.5 Face Detection with Haar Cascades

The Haar cascade face detection technique was used to accomplish face detection. Viola and Jones introduced this technique in 2001 in their paper, Rapid Object Detection, using a Boosted Cascade of Simple Features [Viola and Jones, 2001] as it can process images rapidly and achieve high detection rates. The algorithm makes use of edge or line detection features and is trained with positive samples (target task) and negative samples (without target task). Haar cascades are not only used in face detection but also eye detection, upper and lower body detection [Kasinski and Schmidt, 2010], automatic number plate detection [Atikuzzaman et al., 2019], and other object detection tasks [Soo, 2014]. The features are used to detect sudden differences in the intensities of the

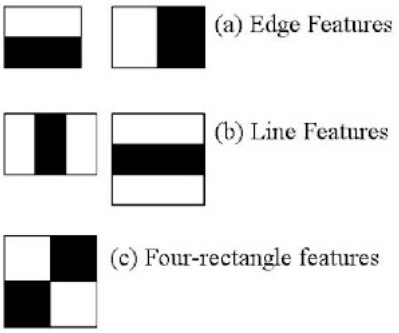


FIGURE 3.13: Haar features used in the Haar cascades algorithm [Ahuja et al., 2018].

pixels and to detect the edges/lines in an image. The edge features in Figure 3.13 are

responsible for detecting a vertical and horizontal edge by calculating the difference of sum of all the image pixels in the dark area of the Haar feature and the sum of all the image pixels in the light area of the Haar feature (Equation 3.12). The flowchart of the Haar cascade classifier is presented in Figure 3.14.

$$Pixel\ value = \frac{\sum Dark\ pixels}{number\ of\ dark\ pixels} - \frac{\sum light\ pixels}{number\ of\ light\ pixels} \tag{3.12}$$

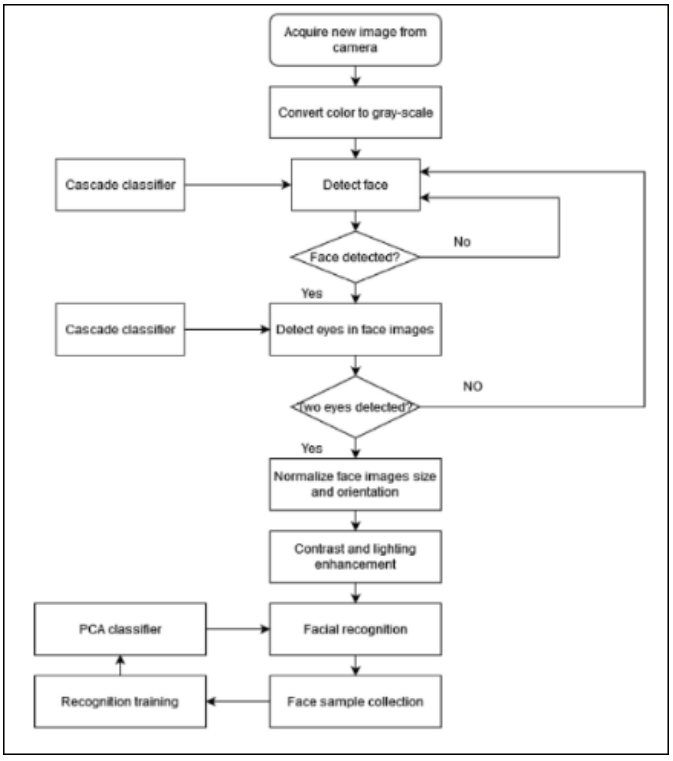


FIGURE 3.14: A flowchart of face detection using Haar cascade classifier [Shetty et al., 2021].

3.6 Evaluation Metrics

Evaluation metrics play a key role in estimating the effectiveness of the proposed frameworks. General evaluation metrics include:

1. **Confusion Matrix**: The confusion matrix is a cross table that shows the number of occurrences between the actual values versus the predicted values of the model (Figure 3.15). There are four terms in the confusion matrix: True Positive, False Positive, False Negative, and True Negative. These terms are discussed below.

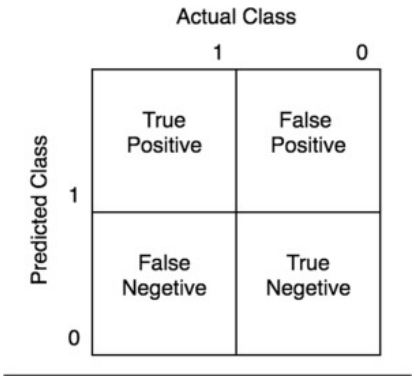


FIGURE 3.15: An example of a confusion matrix [Kulkarni et al., 2020].

- True Positive (TP): True positive is the number of predictions that the model positively classified correctly. This metric is an indication of how many predictions the model was correctly identified as belonging to the correct class.
- False Positive (FP): False Positive is the number of predictions that the model incorrectly positively classified. This metric shows the number of predictions that the model misidentified as being the positive class, even though the sample did not belong to the positive class. .
- False Negative (FN): False Negative is the number of predictions that are incorrectly negatively classified. This metric shows the number of predictions that the model misidentified as not belonging to the positive class, even though the samples belonged to the positive class.
- True Negative (TN): True Negative is the number of predictions that the model correctly identified as negative. This metric shows the number of predictions the model correctly identified as not belonging to the positive class.

2. **Accuracy**: The accuracy of a model is an overall measure of how well the model predicts the entire dataset [Grandini et al., 2020]. Accuracy is the measure of the correct predictions divided by the number of predictions made. It is a popular metric used for multi class classification and is computed from the confusion matrix. Accuracy is a poor metric for classifications with imbalanced datasets. This is because it does not represent the classification errors for classes with fewer samples. In multi label classification, the exact match ratio (subset accuracy) is the strictest metric, and it represents the ratio of samples that have all their labels classified correctly.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.13}$$

(a) **Subset Accuracy (Exact-Match Ratio)**: This accuracy metric represents the percentage of samples with all labels classified correctly.

$$Exact\ Match\ Ratio,\ MR = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i) \tag{3.14}$$

m Where Y_i is the target values and Z_i are the predicted values.

3. **Precision**: Precision is the fraction of True Positive elements divided by the total number of positively predicted units. The precision metric is used to determine the proportion of correct positive identifications [Grandini et al., 2020]. This metric shows the reliability of a model when it predicts an individual sample as positive.

$$Precision = \frac{TP}{TP + FP} \tag{3.15}$$

4. **Recall**: Recall is the ratio of True Positive elements to the total number of positively classified units. The recall metric is used to determine the ability of the model to get all the positive units in a dataset.

$$Recall = \frac{TP}{TP + FN} \tag{3.16}$$

5. **F1-Score**: The F1-Score metric is used to assess a model's classification and is the aggregated harmonic mean of Precision and Recall metrics [Grandini et al., 2020]. The F1-Score value ranges from 0 to 1, where 1 is the best scenario and 0 is the worst scenario.

$$F1 - Score = \frac{2 * precision * recall}{precision + recall} \tag{3.17}$$

The aforementioned evaluation metrics will be used to compare the results obtained by each model on the two datasets. Both of these datasets are posed datasets, where one, DISFA+, has expression labels and AU annotations, and the other, KDEF, only has expression labels. For DISFA+, there are state-of-the-art models that have been created for the task of AU detection. These models are Deep Relational Metric Learning (DRML) Zheng et al. [2021], AU R-CNN [Ma et al., 2019], and JAA-Net [Shao et al., 2021].

3.7 Conclusion

In this chapter, we discussed the background knowledge related to CNN, such as the components that make the CNN work and the hyperparameters that are tuned to improve the classification performance. The structure of the VGG16 network, the network

used in this research work, was also discussed. The WSL approaches used in this research work for AU recognition were discussed. Next, the face detection classifier, Haar cascade, was discussed. Finally, the metrics used to evaluate the performance of each model in this research work are presented. In Chapter 4, the methodology used for the WSML approaches is discussed in detail. This includes the training and validation results for each approach.

Chapter 4

Research Methodology

This chapter presents the methodology for the different WSML approaches used to train a model for AU recognition. The first approach is an incomplete supervision framework called pseudo-labelling. This is followed by large loss rejection mechanism which is based on inaccurate supervision framework. Lastly, the transfer learning approach which was used to evaluate the WSML based frameworks for AU recognition is presented. This chapter is organised as follows. In Section 4.1, the datasets used in this research work and the preprocessing steps performed to use the samples are presented. In Section 4.2, the approaches used in this research work to recognise AUs are presented. We start the methodology with the pseudo-labelling approach, where we train a model with a small amount of AU annotated data along with a larger amount of unlabelled data. This is followed by the large loss rejection method. This approach involves training a dataset with noisy labels. Lastly, we present the transfer learning approach. The TL model will be used to compare the performance of both WSL approaches. The training and validation results for all the approaches.

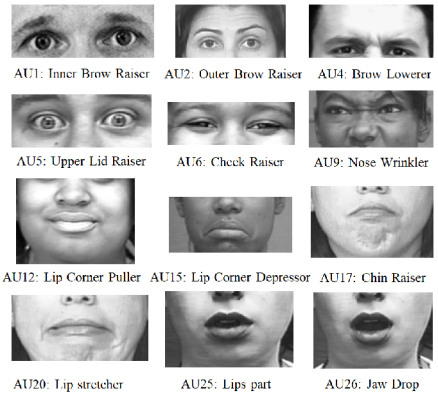
4.1 Datasets and Pre-processing

In this section, the two datasets used in this research work are discussed. The DISFA+ dataset [Mavadati et al., 2016] was used to train and validate the deep neural networks for emotion and AU recognition while KDEF dataset [Lundqvist et al., 1998] was mainly used to test the performance of each approach to generate AUs for an unlabelled dataset. Unlike DISFA+, KDEF does not contain AU annotations for each sample in the dataset and thus samples from KDEF was considered as unlabelled data in our work.

4.1.1 Extended Denver Intensity of Spontaneous Facial Action Database

The Extended Denver Intensity of Spontaneous Facial Action Database [Mavadati et al., 2016] provided expert coded AU-annotations in this research work. DISFA+ dataset contains the six basic emotion-related expressions [Ekman and Friesen, 1971], namely *Anger, Disgust, Fear, Happiness, Sadness, Surprise* and *neutral* (Figure 4.1a). The dataset contains posed and non-posed facial expression data. A FACS coder coded the dataset with AU annotations for each facial expression of the nine participants. The intensity of the AU annotations have intensity scores between 1 to 5 for 12 AUs, i.e., AU 1, 2, 4, 5, 6, 9, 12, 15, 17, 20, 25, and 26 (Figure 4.1b). The participants come from diverse ethnic groups and their posed expressions were captured using software that instructed and guided the participants to imitate facial actions. Figure 4.2a presents the number of

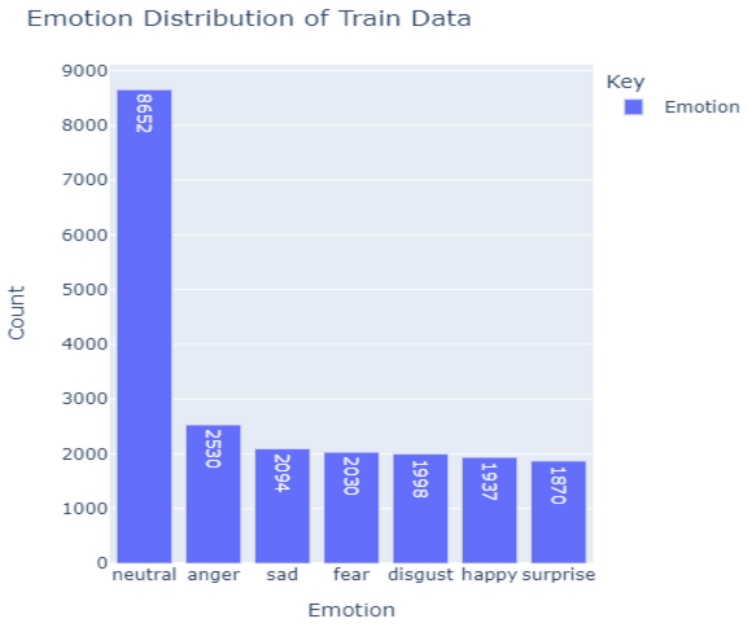
(A) Posed and non-posed facial expressions from the subjects in DISFA+ [Mavadati et al., 2016].



(B) The AUs present in DISFA+ dataset [Ekman and Friesen, 1971].

FIGURE 4.1: Samples from the DISFA+ dataset; Representation of AU in DISFA+.

emotion-labelled samples, from DISFA+, for each of the seven emotions used in this research work. Apart from samples depicting neutral expressions, other samples are relatively balanced. Figure 4.2b presents the distribution of each AU present in the DISFA+ dataset. AU15 and AU17 have the least number of samples. The imbalance in the distribution of AUs is to be noted which might impact the performance of trained models in estimating the presence of specific AUs.



(A) The distribution of facial expression images in the DISFA+ dataset [Mavadati et al., 2016].

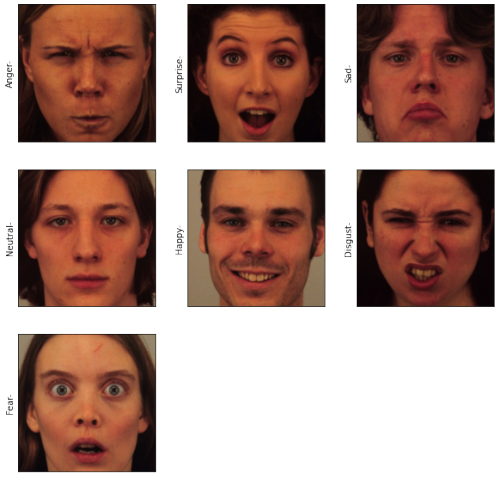
(B) The distribution of facial action units in the DISFA+ dataset [Mavadati et al., 2016].

FIGURE 4.2: The emotion and AU distribution of the DISFA+ dataset.

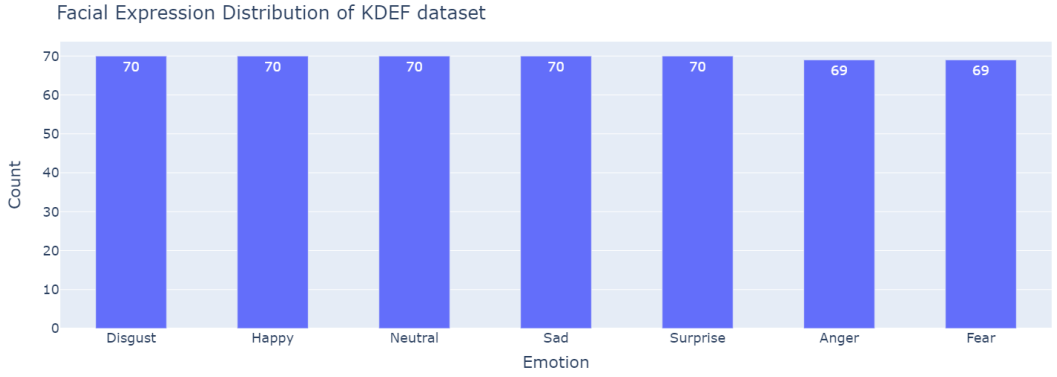
4.1.2 Karolinska Directed Emotional Faces

Karolinska Directed Emotional Faces dataset [Lundqvist et al., 1998] contains 4900 images of facial expressions of emotions. Researchers from Karolinska Institutet, Department of Clinical Neuroscience in Stockholm, Sweden developed the dataset. This dataset was created for psychological and medical research purposes. The images were captured for perception, attention, memory, and backward masking experiments. Images were pre-processed to ensure that the positioning of the eyes and mouths was fixed during scanning. The dataset contains 70 individuals, each displaying 7 different

emotional expressions (Figure 4.3a and Figure 4.3b). This is a posed dataset as the individuals are asked to express each emotion [Lundqvist et al., 1998]. The dataset does not have AU annotations and is considered unlabelled for the AU detection task. The dataset is balanced with respect to the number of samples for each emotion. Therefore the model should not be biased towards an emotion.



(A) Sample images from KDEF dataset [Lundqvist et al., 1998].



(B) The distribution of facial expression images in the KDEF [Lundqvist et al., 1998].

FIGURE 4.3: Samples from KDEF dataset; Sample distribution of KDEF dataset.

4.1.3 Data Labels

The facial expression recognition task is a multi-class classification problem as there are seven emotion classes present in the DISFA+ dataset. Each image input to the model will belong to one of the seven emotion classes namely anger, disgust, fear, happiness, sadness, surprise, and neutral. The DISFA+ training set consists of n feature pairs $\{(\mathbf{X}_i, Y_i)\}_{i=1}^{n}$ where $\mathbf{X}_i \in \mathbb{R}^d$ represent the features and $Y \in \{1, 2, \ldots, k\}$ represent the associated labels for one of the k classes. The labels are represented as

one-hot encoded vectors $\mathbf{y}_i \in \mathbb{R}^k$ which represent on of k classes with one-hot encoding for the convenience of the model. The matrix of input features is represented as $\mathbf{X} = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d*n}$ and the label matrix is $\mathbf{Y} = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^{k*n}$. Label encoder from the Python library, Scikit-learn is used for one-hot encoding of the samples from DISFA+ for multi-class emotion representation.

In contrast, action unit recognition is a multi-label classification problem because there can be multiple AUs present in a sample image. For multi-label classification, let $\mathbf{X} = \mathbb{R}^d$ be the input space of d-dimensional image instances and $\mathbf{Y} = \{y_1, y_2, \ldots, y_q\}$ is the output label space with $|\mathbf{Y}| = q$ possible labels which can be assigned to each image instance. The feature pair $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ where $x_i \in \mathbf{X}$ is an image instance and $y_i \subseteq \mathbf{Y}$ is a set of true labels associated with the image instance. The label set y_i is represented as a q dimensional binary vector $y_i \in \{0, 1\}^{12}$ where the labels relevant to the image instance x are equal to 1 and irrelevant labels are represented by 0. The superscript 12 represents the number of AUs present in the dataset.

4.1.4 Pre-processing: Face Detection

Figure 4.4 presents the flow used to preprocess the DISFA+ and KDEF images before they are used as inputs to the emotion or AU recognition model. To prevent the net-

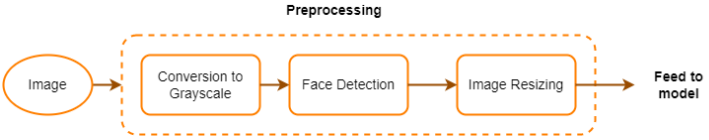


FIGURE 4.4: Preprocessing input samples.

work from learning noise in the images, faces are cropped in all images and extracted using a face detection algorithm. OpenCV [Bradski, 2000] was used to convert the images to grayscale, Haar cascade was used to detect the faces in each image. The image were then resized from 1280x720 to 224x224 with OpenCV. The images were resized to the input size of VGG16 model. An example of this preprocessing stage is shown in Figure 4.5.

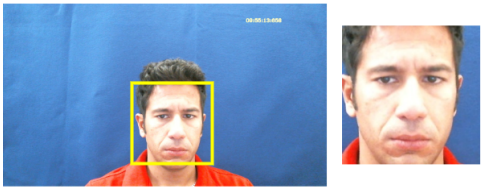


FIGURE 4.5: The result of implementing Haar cascade using OpenCV for face detection.

4.2 Facial Action Units Recognition

This section presents the methodology used for AU recognition using incomplete supervision and inaccurate supervision. Further, AU recognition using transfer learning is undertaken to analyse the effectiveness of WSL methods. First, we present the methodology for incomplete supervision pseudo-labelling mechanism, where we have access to a limited sample set of labelled data and a significantly larger amount of unlabelled samples. The goal here is to use the unlabelled data together with the limited labelled sample to improve the performance of the model in AU recognition. Second, the methodology used for inaccurate supervision using LLR mechanism is presented. Here, we access a noisy dataset that contains inaccurate labels. The goal is to train a model with this inaccurate dataset. Finally, we compare the AU recognition performance of these two methods, with a model trained with transfer-learning where there was no limitation to the amount of AU-annotated training samples. The model structure used to recognise the AUs present in the DISFA+ dataset is depicted in Figure 4.6.

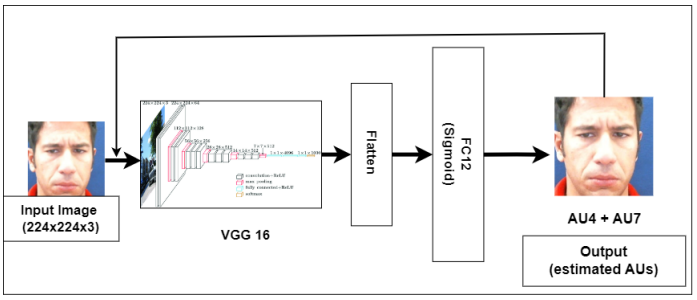


FIGURE 4.6: Structure of the AU model

4.2.1 Pseudo-Labelling

The structure of the VGG16 model used in the pseudo-labelling approach is identical to the structure depicted in Figure 4.6. However, in this approach, the weights learned from the emotion recognition task were not used. The model was trained from scratch to detect the AUs present in the DISFA+ dataset. The DISFA+ dataset was split into the following ratio: 20% as labelled samples, 10% as labelled test samples, and 70% as unlabelled samples. There were two pseudo-label approaches used in this research work. The approaches are listed as follows:

1. Pseudo-Labelling version one (Figure 4.7)

   (a) Train an initial model with the limited labelled data

(b) Use the trained model to generate pseudo-labels for the entire cohort of unlabelled data

(c) combine limited labelled data with unlabelled data (with pseudo-labels) and retrain the model

(d) The model trained on labelled data and pseudo-labelled data is then validated on validation data

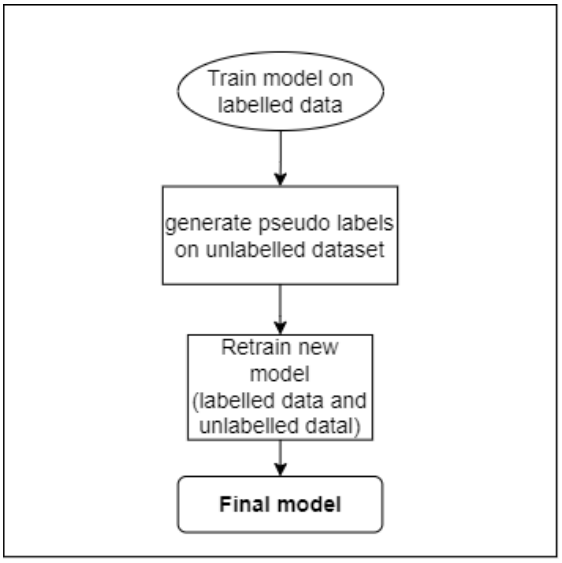


FIGURE 4.7: Methodology for Pseudo-Labelling version one.

2. Pseudo-Labelling version two (Figure 4.8)

(a) Train an initial model with the limited labelled data

(b) Use trained model to generate pseudo labels for unlabelled data

(c) Select the most confident label predictions and use as pseudo labels

(d) Train the model again on labelled data plus confident pseudo labels

(e) Use model to predict labels on the remaining cohort of unlabelled data

(f) Evaluate the model on validation data

Initial Model Training

This is the initial model trained with limited labelled data that is used for both pseudo labelling versions. The model was trained using three-fold cross validation with 20 epochs per fold, a batch size of 32, with the Adaptive Movement Estimation (ADAM) optimiser, and a step-decay learning rate scheduler with an initial learning rate of 0.001. Binary cross entropy was used as the loss function. The early stopping criterion was used to stop the training process. The model achieved a validation subset accuracy of 99.2%. The training results and plot for this model is presented in Table 4.1 and Figure

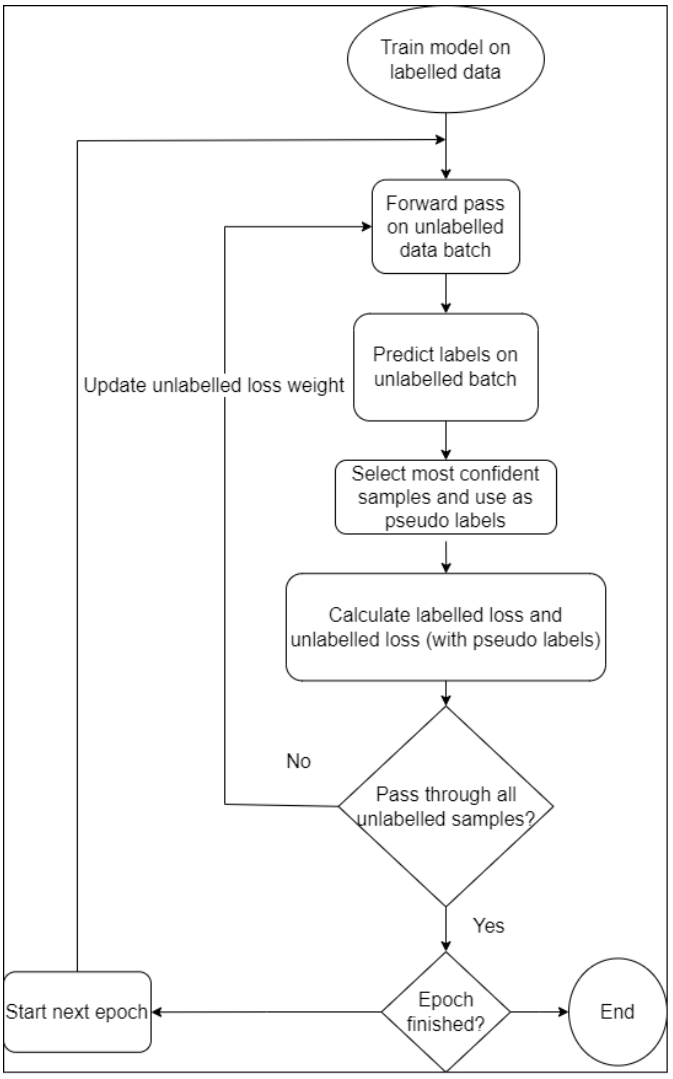


FIGURE 4.8: Methodology for Pseudo-Labelling version two.

4.9, respectively.

TABLE 4.1: Training result of the initial model trained with 20% of DISFA+.

Val loss	Val accuracy	Training loss	Training accuracy
0.00613	99.2%	0.00787	97.8%

Pseudo version one

The previous model was used to generate pseudo-AU annotations for the unlabelled DISFA+ samples. A new model, pseudo labelling version one (PL-1) was then trained with the combination of pseudo-AU-annotated samples and expert-coded AU-annotated samples. The model plateaued after it reached a validation subset accuracy of 99.7%.

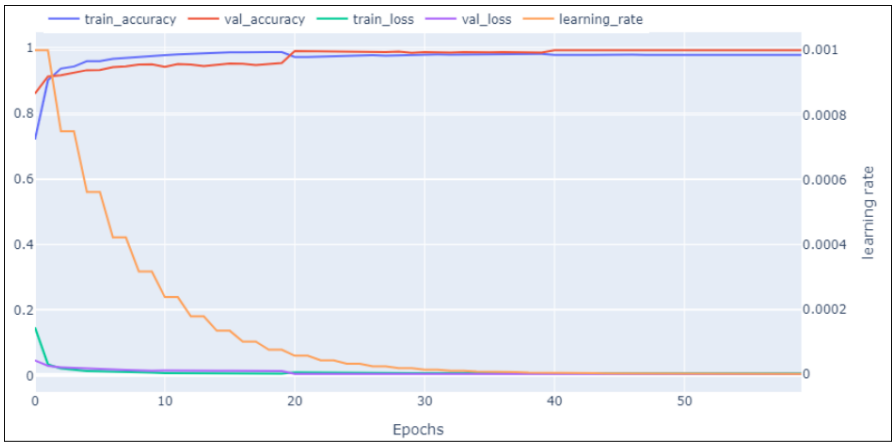


FIGURE 4.9: Training and validation accuracy using the labelled DISFA+ sample set for Pseudo-Labelling version one and two.

The results are presented in Table 4.2. The evaluation results of the pseudo-label version one on the DISFA+ test set is presented in Chapter 5.

TABLE 4.2: Training result of version-one on unlabelled data with pseudo-labels and 20% of DISFA+.

Val loss	Val accuracy	Training loss	Training accuracy
0.00247	99.7%	0.00441	98.4%

Pseudo version two

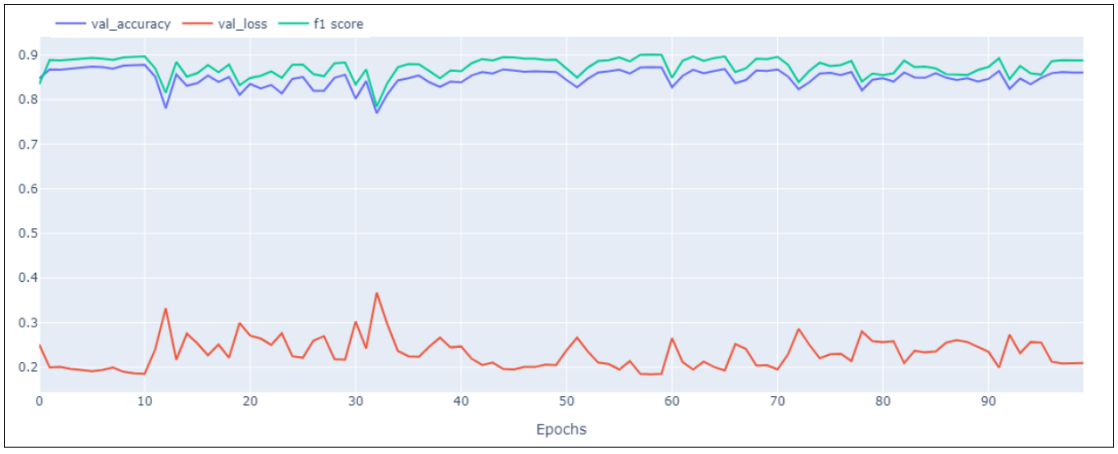
Unlike in version one where the default training and evaluation loops provided by Keras [Chollet, 2019] are used, a custom training loop was written from scratch for version two because we need low-level control over training and evaluation. The Keras context manager GradientTape [Chollet, 2019] was used to update the gradients of the model's weights with respect to the loss value. Adam optimiser was used with a constant learning rate of 0.001. The binary cross-entropy loss function was applied according to Equation 3.7. For labels which had a prediction value above 0.9 were set to 1 (indicating an AU) while labels that had a value less than 0.1 were set to 0 (showing the absence of an AU). This was done to reduce the bias caused by poor predictions that affect the model weights. The F1 score was used as the validation metric. As the training progressed, if there was an increase in the F1-score at each epoch, the model would be saved.

There were two pseudo labelling version two (PL-2) models, both were trained with a batch size of 50 and with the ADAM optimiser. The first model, PL-2a, was trained for 100 epochs, and with a constant learning rate of 0.001. The second model, PL-2b, was trained for 150 epochs and with a learning rate of 0.0001. As mentioned above, both models were saved at each epoch if there was an increase in the F1-score. The

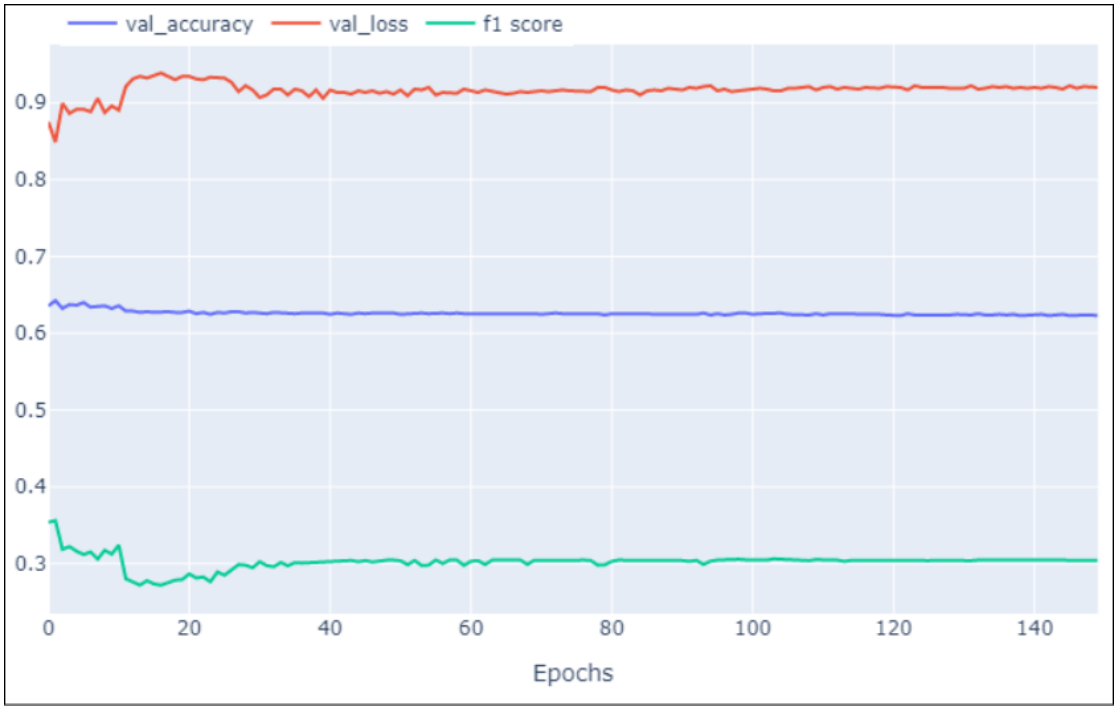
training plots for PL-2a and PL-2b are presented in Figure 4.10a and Figure 4.10b. PL2-a was trained with a random sample of DISFA+ dataset and PL-2b was trained with the same data distribution used for version one. The results of both models are reported in Table 4.3.

TABLE 4.3: Pseudo labelling model validation results.

Configuration	Val Acc	Val Loss	Val F1
PL-2a	87%	0.184	0.90
PL-2b	64%	0.84	0.36



(A) Pseudo version two-a model performance on the validation set during training.



(B) Pseudo version two-b model performance on the validation set during training.

FIGURE 4.10: The training plot of PL-2a and PL-2b.

4.2.2 Large Loss Rejection

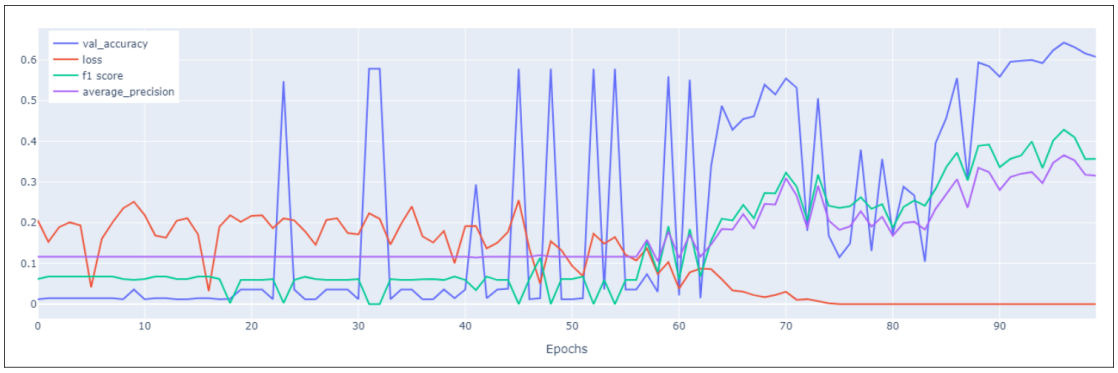
The structure of the VGG16 model used for this approach is depicted in Figure 4.6 and is initialised with ImageNet weights. This was done because the model would be trained with inaccurate labels and therefore the feature representations from ImageNet would be useful for better error rejection. The noisy dataset used in this approach is the combined dataset in pseudo-labelling version one (with pseudo and expert annotated labels). For training and predictions, the samples which have an intensity value greater than 0.5 are set to 1. The model is trained with the ADAM optimiser. Different batch sizes, Δ_{rel}, and the number of epochs are experimented with to optimise the model performance.

At every epoch, if there was an increase in the F1-score, the model would be saved. The best-performing model would then be used for testing. Each model was trained with a constant learning rate of 0.0001 with Δ_{rel} changed to see the effect on performance. The number of epochs used during the training process ranged from 100 to 900 epochs. Table 4.4 presents the different hyperparameter configurations used to determine the best-performing AU recognition model trained with the LLR mechanism (LLR model). The first LLR model was trained with a $\Delta_{rel} = 0.01$ for 100 epochs [Kim et al., 2022]. This model achieved a F1-score of 0.43 (Figure 4.11a). There was a 32% increase in the F1-score (0.57) when the model was trained for 200 epochs (Figure 4.11b). Δ_{rel} was reduced to 0.001 for a stricter rejection threshold, to determine if there would be an increase in the classification results. The model was further trained until the 300th epoch which resulted in a 3% increase in the F1-score (0.59) from the previous F1-score mentioned above. The model was trained longer for 450 epochs, resulting in a 1.7% reduction in the F1-score. The model was then trained for 637 epochs (Figure 4.12b). This further increase in training epochs resulted in the reduction of F1-score by 10% (0.52).

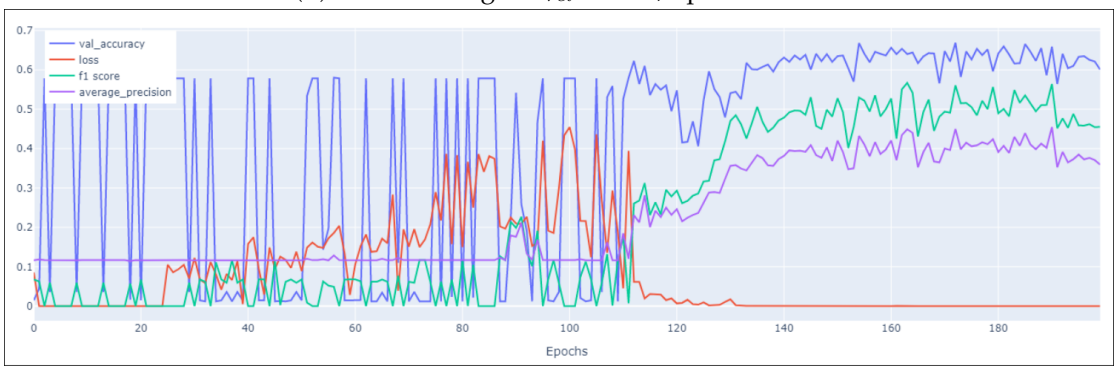
TABLE 4.4: Hyperparameter network tuning for optimal AU recognition by the LLR model.

Configuration	Epochs	delta rel	Batch Size	Learning Rate
1	100	0.01	32	0.0001
2	200	0.01	32	0.0001
3	300	0.001	32	0.0001
4	450	0.001	32	0.0001
5	498	0.002	32	0.0001
6	637	0.001	50	0.0001

The Δ_{rel} was increased to 0.002 to determine the performance of the model when the strictness is reduced by 100%. The model was trained for 498 epochs and achieved an



(A) WSML config1: $\Delta_{rel} = 0.01$, Epochs= 100



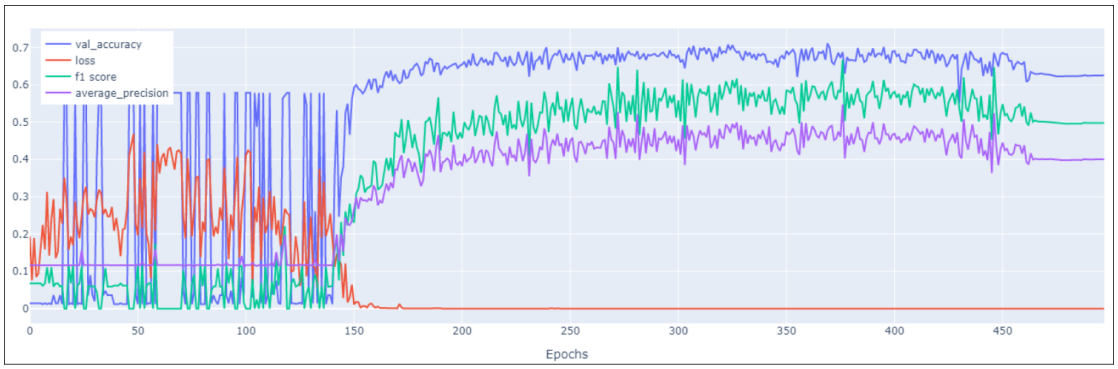
(B) WSML config2: $\Delta_{rel} = 0.01$, Epochs= 200

FIGURE 4.11: Performance plot of VGG16 using WSML with $\Delta_{rel} = 0.01$.

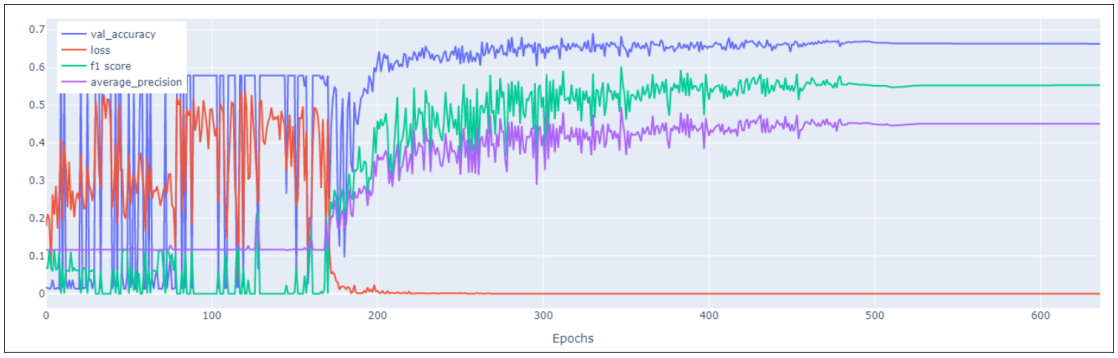
F1 score of 0.67 (Figure 4.12a). Configuration-5 had the best overall performance and was trained with a batch size of 32, $\Delta_{rel} = 0.001$, and a maximum of 498 epochs. When evaluated on DISFA+ validation set, the model achieved a subset accuracy of 69% and a 0.67 F1 score. From Figure 4.12a and Figure 4.12b, the model took longer to converge when the rejection threshold,Δ_{rel}, was stricter. Table 4.5 presents the experimental results of each LLR model configuration evaluated on DISFA+ validation set.

TABLE 4.5: Experimental results of each LLR model configuration on DISFA+ validation set.

	Validation		
Configuration	**Subset Accuracy**	**F1-Score**	**Average Precision**
1	64.2%	0.43	0.37
2	64%	0.57	0.45
3	66.4%	0.59	0.48
4	66.1%	0.584	0.484
5	68.7%	0.667	0.548
6	67.2%	0.521	0.423



(A) WSML config 5:$\Delta_{rel} = 0.002$, Epochs= 498



(B) WSML config 6: $\Delta_{rel} = 0.001$, Epochs= 637

FIGURE 4.12: WSML performance plots for configuration 5 and configuration 6.

4.2.3 Transfer Learning

To evaluate the success of pseudo-labelling and large loss rejection methods, a baseline transfer learning model was created where data samples were not restricted. As emotion datasets are relatively easy to access compared to FACS AU annotated datasets, we followed the approach of transfer learning. A modified VGG16 network is trained to recognise the seven emotions present in the DISFA+ and KDEF datasets. Figure 4.13 depicts the structure of the model. The VGG16 model was altered by adding several layers to reduce the chance of overfitting. Soft-max activation function was used to

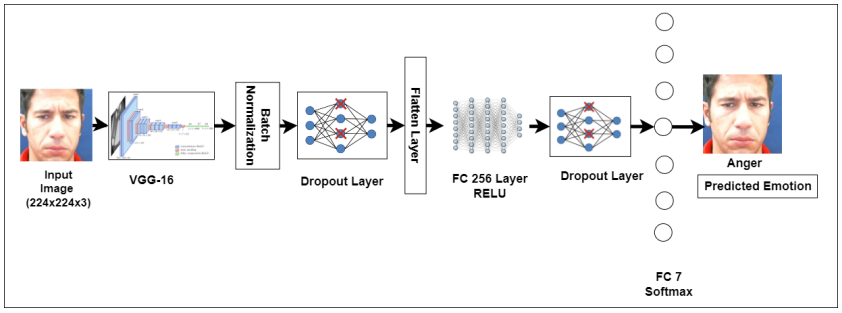


FIGURE 4.13: Model architecture to detect emotions from facial expressions.

transform the model outputs into a probability vector, where each entry in the vector is

the probability of the input image belonging to one of the seven emotions. The model was compiled using the ADAM optimiser, a categorical cross-entropy loss function, and accuracy as the performance metric. Three-fold cross-validation was used to train the model with early stopping. To set the optimal hyperparameters for the network, four configurations were run, and the training parameters are presented in Table 4.6. The first stage of transfer learning involved training a modified VGG16 network to

TABLE 4.6: Hyperparameter network tuning for optimal emotion recognition performance.

Configuration	Number of folds	Epochs/ fold	Epochs total	Min_delta	Batch Size	Learning Rate
1	3	15	45	0.01	32	0.0001
2	3	20	60	0.001	16	0.001
3	3	20	60	0.001	16	0.001
4	3	40	120	0.001	32	0.001

classify the emotions present in the DISFA+ dataset (within dataset evaluation). The emotion recognition model was trained and validated with four different configurations (Table 4.6) to determine the optimal hyperparameters for the task. The result of each configuration is presented in Table 4.7 and configuration-1, the configuration with the lowest number of epochs, achieved the highest validation accuracy and validation F1-score reaching a training and validation accuracy of 96.3% and 98.5%, respectively. The training plot for configuration-1 is presented in Figure 4.14. The architecture of the

TABLE 4.7: Performance of each configuration on the DISFA+ validation set.

Config	Val Loss	Val Accuracy	Training Loss	Training Accuracy
1	0.0488	98.5%	0.112	96.3%
2	0.0591	97.7%	0.178	94.4%
3	0.0604	97.9%	0.19	93.7%
4	0.0543	98%	0.2	93.8%

best-performing ER model (configuration-1) was updated to accommodate the task of recognising 12 AUs. The last layer of the ER model was replaced with a fully connected layer with 12 units (one for each AU in the DISFA+ dataset) activated by the sigmoid activation function. For training and predictions, the AU intensity of the DISFA+ participants is set to 1 if greater than 0.5 and set to 0 if lower than 0.5. The AU model was then compiled using the ADAM optimiser with a binary cross-entropy loss function. The updated model was trained using two approaches. Approach one involved fine-tuning the last FC layer and the second approach involved fine-tuning the entire network. Figure 4.6 depicts the general structure of the model used for AU detection. After initial experimentation, three-fold cross-validation and early stopping were employed. Early stopping was initiated after no change in the validation accuracy for 1/3 of epochs per fold.

To set the optimal hyperparameters for the network, four models were trained using

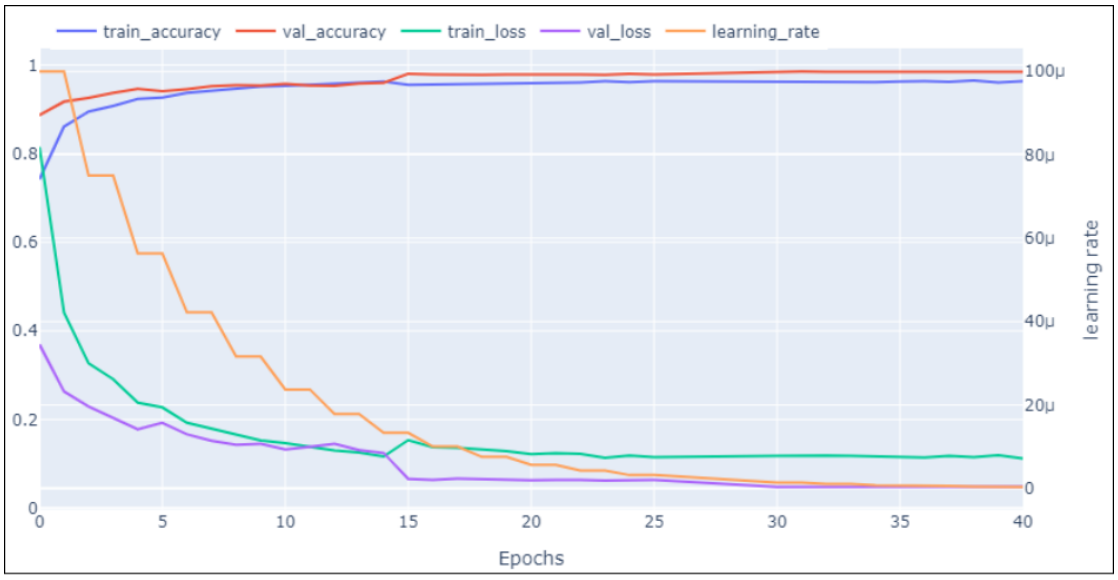


FIGURE 4.14: Configuration 1: epochs=45, initial LR=0.001, early-stopping min delta = 0.001, patience = 10 epochs.

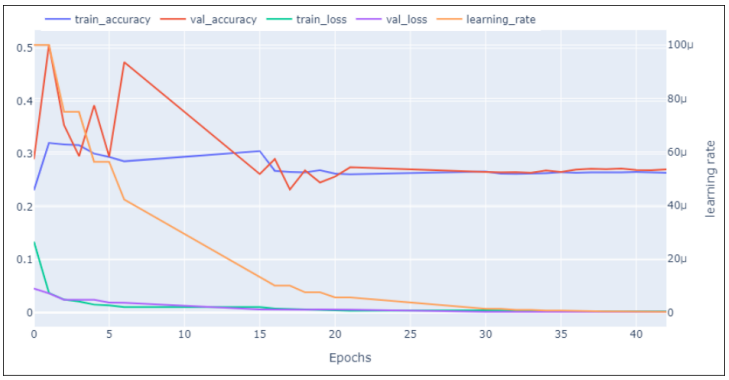
different parameters. The fourth model was selected because it had the highest validation accuracy and loss. This model was trained using three-fold cross-validation, with 40 epochs per fold, a batch size of 32, and a step-decay learning rate scheduler with an initial learning rate of 0.001. The hyperparameters used for the AU model are presented in Table 4.8. The first three configurations were trained with approach one, and the last two with approach two. Approach one involved fine-tuning the last FC layer, and the second involved fine-tuning the entire network. The training plots for the best-performing models are presented in Figure 4.15a and Figure 4.15b. In Table 4.9, the results of training and evaluation on validation and test sets are presented. The training and validation accuracy values are based on the generic accuracy metric used in Keras. The subset accuracy matrix was used to evaluate performance on the evaluation sets (Chapter 5).

TABLE 4.8: Hyperparameter network tuning for optimal AU detection performance.

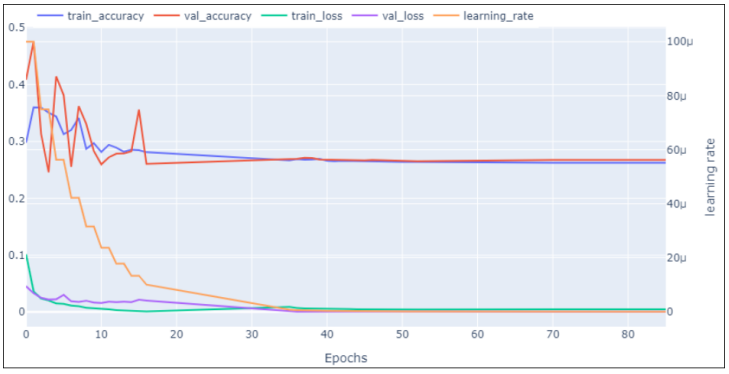
Configuration	Threshold value	No. of folds	Epochs/ fold	Epochs total	Min_delta	Batch Size	Learning Rate
1	0.5	3	15	45	0.001	32	0.0001
2	0.5	3	20	60	0.001	32	0.001
3	0.5	3	30	90	0.001	32	0.01
4	0.5	3	15	45	0.001	32	0.0001
5	0.5	3	35	105	0.001	32	0.0001

4.3 Hardware and Software Details

The experiments in this research work were performed mainly with Python and the associated Python libraries. For machine learning computations, Keras and Tensorflow



(A) Configuration-4: epochs=45, initial LR=0.0001, early-stopping (min-delta)=0.001, patience=10 epochs.



(B) Configuration-5: epochs=105, initial LR=0.0001, early-stopping (min-delta)=0.001, patience=20 epochs

FIGURE 4.15: The training plots of configuration 1 and configuration 5.

TABLE 4.9: Results of the training model for AU detection task using DISFA+ dataset and k-fold cross validation

Config	Val loss	Val accuracy	Training loss	Training accuracy
1	0.0813	25.2%	0.0831	25.9%
2	0.0332	33.1%	0.0351	33.4%
3	0.0191	29.6%	0.0232	30.1%
4	0.0019	27%	0.00246	26.5%
5	0.000887	26.7%	0.00491	26.3%

were used. The computations were performed using the high-performance computing clusters provided by the School of Computer Science and Applied Mathematics at the University of the Witwatersrand.

4.4 Conclusion

In this chapter, we looked at the datasets used in this research work and the preprocessing done on each image in the dataset. The methodology used for the two WSL approaches was discussed. This was then followed by the methodology for the transfer learning approach which is used to compare the performance of the WSL approaches.

In Chapter 5, the evaluation results on the DISFA+ test and KDEF sets are presented for each approach mentioned in this chapter.

Chapter 5

Results and Discussions

In Chapter 4, the procedure used to train the model for each approach was presented. The training results and validation results were also presented. In this chapter, each model is evaluated on the test set for each approach. Each model was used to generate AUs for the KDEF dataset. The KDEF dataset is considered unlabelled as it does not contain AU annotations. The AUs generated by each model were compared to the FACS mapping of AU to emotion for further understanding of the results obtained.

5.1 AU Recognition Model with Pseudo-labelling

5.1.1 Pseudo Labelling Test Results on DISFA+ Test Set

The DISFA+ test split for the pseudo-labelling approach was used to evaluate each version's performance in detecting AUs. The PL-1 model achieved a subset accuracy of 68% and an F1-score of 0.56. For PL-2a, the model achieved a subset accuracy of 89% and an F1-score of 0.9 on the test set. PL-2b achieved a subset accuracy of 66% and an F1-score of 0.44 on the same DISFA+ set. The AU recognition performance of PL-1, PL-2a, and PL-2b are presented in Table 5.1.

The PL-1 model did not detect AU15, which is present in the facial expression of sadness, and AU17, which is present in the facial expression of disgust. These two AUs have the lowest count in the DISFA+ dataset. The PL-2a model did not detect AU17. In PL-2b, AU15 and AU17 were not detected by the model. All three models struggled with recall on various AUs. An emotion-to-AU mapping analysis will be performed to determine which AUs each model would detect when presented with images from each emotion present in the DISFA+ dataset. The ground truth AU distribution for a

AU	Precision	Recall	F1-Score
1	41%	29%	34%
2	99%	52%	68%
4	95%	37%	53%
5	94%	74%	83%
6	100%	22%	36%
9	98%	50%	66%
12	100%	21%	35%
15	0%	0%	0%
17	0%	0%	0%
20	29%	27%	28%
25	97%	41%	58%
26	98%	35%	51%

(A) PL-1

AU	Precision	Recall	F1-Score
1	95%	87%	91%
2	76%	53%	63%
4	97%	93%	94%
5	99%	89%	94%
6	100%	84%	91%
9	95%	90%	93%
12	97%	100%	98%
15	100%	20%	33%
17	0%	0%	0%
20	97%	93%	95%
25	97%	93%	95%
26	96%	99%	97%

(B) PL-2a

AU	Precision	Recall	F1-Score
1	100%	13%	23%
2	97%	53%	69%
4	94%	42%	58%
5	94%	91%	93%
6	100%	19%	32%
9	100%	45%	62%
12	100%	24%	38%
15	0%	0%	0%
17	0%	0%	0%
20	23%	23%	23%
25	100%	5%	9%
26	100%	5%	10%

(C) PL-2b

TABLE 5.1: Pseudo Labelling classification performance on DISFA+ test set.

test sample set of 100 from DISFA+ and the predictions made by PL-1 are presented in Figure 5.1. The similarity between the true and predicted distributions shows the effectiveness of the PL approach for AU recognition on DISFA+ samples.

Further analysis of the results depicted show that, for DISFA+ samples annotated with the anger emotion, the PL-1 model could register the same dominant AUs present in the true distribution (Figure 5.1). The dominant AUs were in line with the AUs used to code anger emotion according to the FACS. The model was unable to register AU10, AU22, and AU23 as they are not present in the DISFA+ dataset. The PL-2 models had a similar distribution for anger emotion. Unlike PL-1, these models were able to detect more of AU25 and AU26 in the anger samples. When PL-1 was tested on the samples annotated with the disgust emotion, only 2 of the dominant AUs (AU9 and AU25) were in line with the FACS coding for the disgust emotion. However, the AUs detected were similar to the AUs coded by the expert FACS coder. All three models had a similar distribution for the disgust emotion. However, PL-2b and PL-1 were identical in their distribution.

The PL-1 model performed well with the samples annotated with fear emotion and the AUs detected by the model, which were in line with the FACS coding manual. When PL-1 was tested on the samples annotated with the happy emotion, the model detected the presence of other AUs and not just AU6 and AU12. AU6 and AU12 are used in the FACS manual to code the happy emotion. The AUs detected by the model were also present in the expert annotated distribution for happy. For samples annotated with the sad emotion, the AUs detected by the model were in line with the FACS manual coding for sad emotion. The predicted distribution was almost identical to the

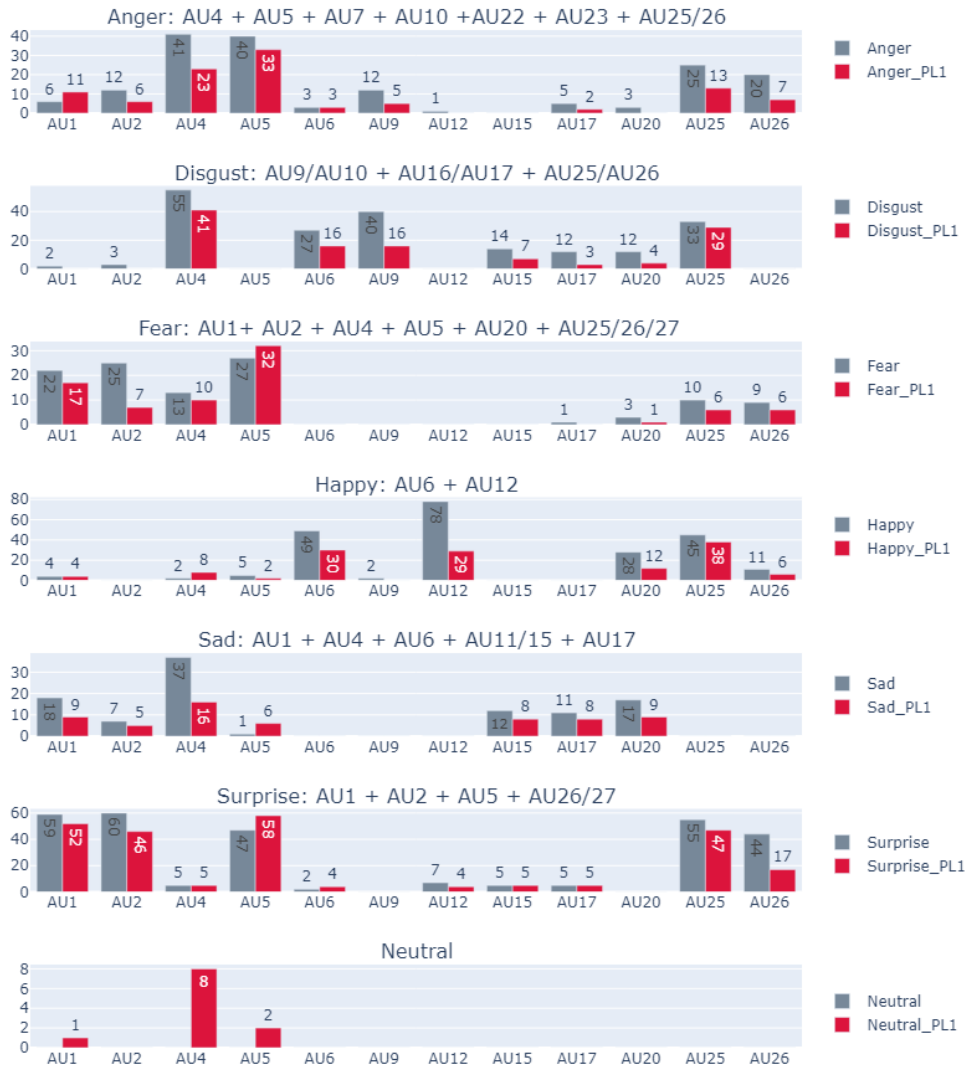
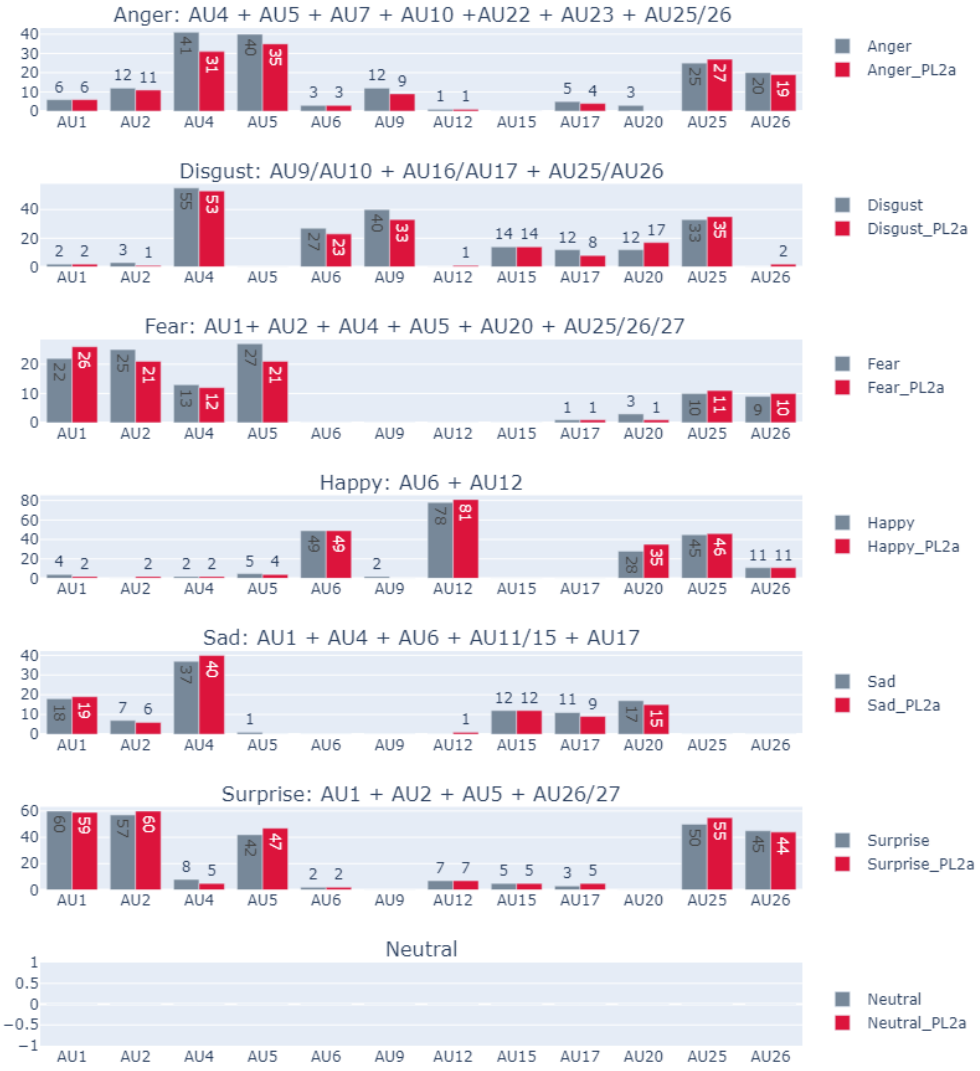


FIGURE 5.1: The ground truth AU distribution compared with the distribution predicted by the PL-1 model on the DISFA+ sample set.

expert-annotated distribution.

When tested on the samples annotated with the surprise emotion, the AUs detected by the three models were in line with the FACS manual. The distribution was identical to the expert annotated distribution. When tested on the samples expressing the neutral emotion, the PL-1 model detected 1 sample with AU2, 8 samples with AU4, and 2 samples with AU5. These are false positives, as the neutral emotion should not contain any AUs. The PL-2a model did not detect any AUs in the neutral samples while PL-2b detected AU4 and AU9. The PL-2b model detected 13 neutral samples depicting AU4.



*

FIGURE 5.2: The ground truth AU distribution compared with the distribution predicted by the PL-2a model on the DISFA+ sample set.

On further investigation, the neutral samples for which the model registered AU4, all belonged to the same subject. The shape of the subject's eyebrows might have caused the misclassification (Figure 5.4).

5.1.2 Evaluation of Pseudo-Labelling on KDEF Dataset

The three models were evaluated on the KDEF dataset to validate its cross-dataset performance. A successful PL model can produce a similar AU distribution per emotion similar to the FACS manual. Figure 5.5 presents the different combinations of AU used

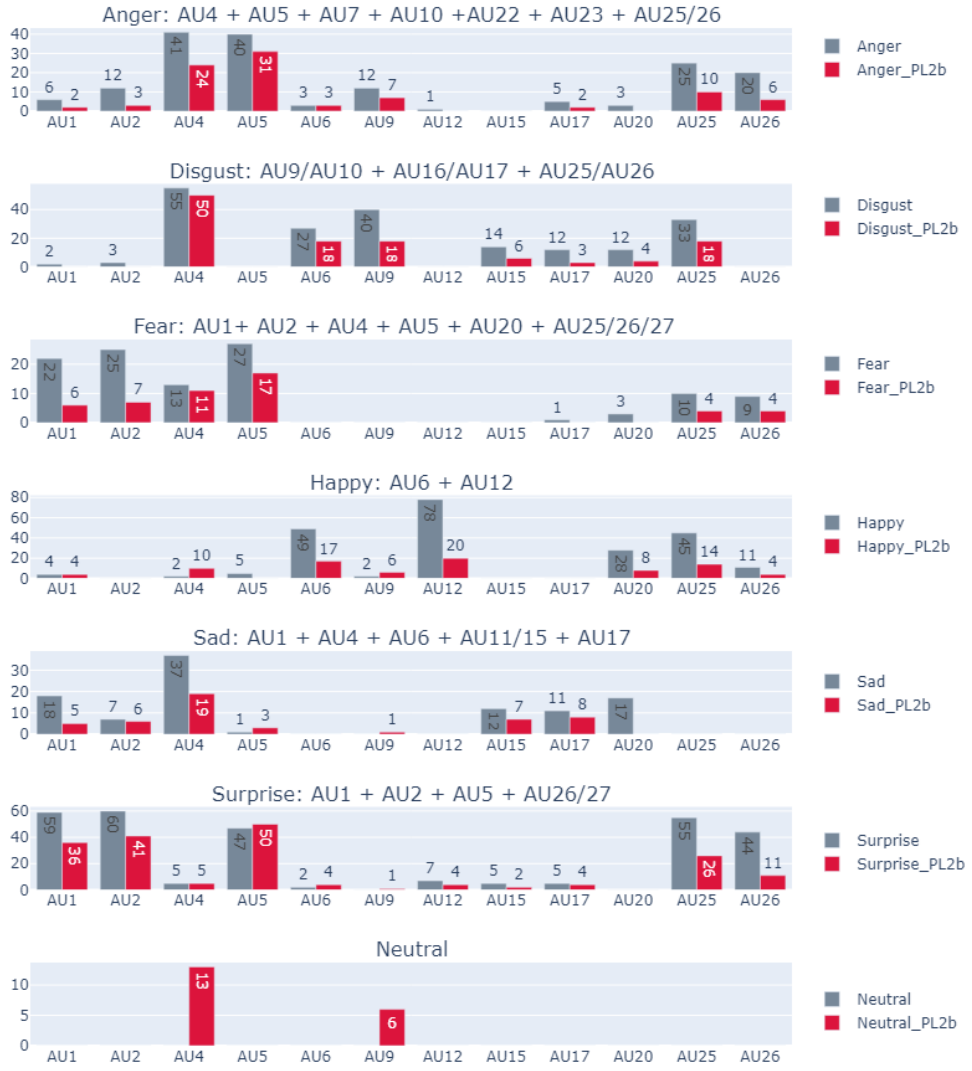


FIGURE 5.3: The AU distribution predicted by PL-2b model on the DISFA+ sample set.

to code each of the universal emotions. Figure 5.6 presents the AU performance of each PL model on the KDEF samples. When tested on KDEF samples annotated with anger emotion, the PL-1 model detected three AUs, AU4, AU6, and AU25. Of these three, AU4 and AU25 are present in the different combinations used by the FACS manual to code anger. However, the PL-1 model did not detect AU5 in any of the anger samples. AU5 is present in the different AU combinations used to code anger. There was an anomaly with PL-2a for AU4 the model detected 60 instances of AU4 compared to 3 and 8 detected by PL-1 and PL-2b, respectively. For the KDEF samples annotated with disgust emotion, the PL-1 model registered five AUs and out of these five only

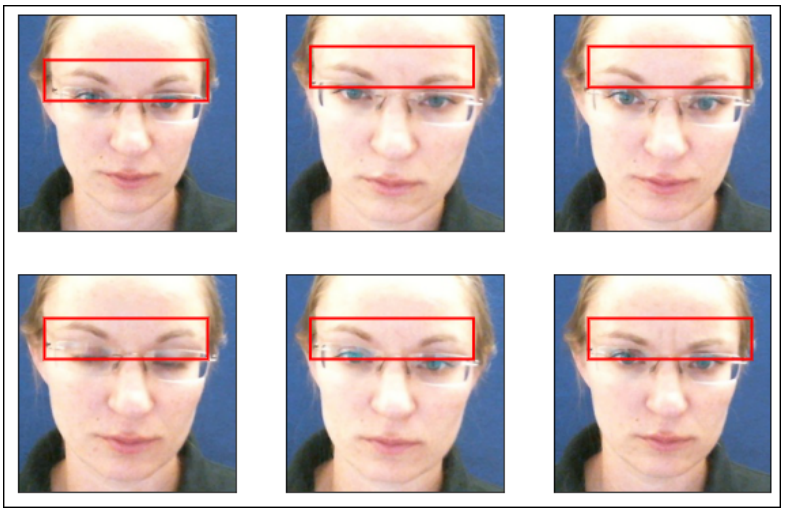


FIGURE 5.4: The neutral samples misclassified as depicting AU4 by PL-1 and PL-2b models [Mavadati et al., 2016].

AU9 and AU25 are present in the combinations used by the FACS manual to code the disgust emotion. AU4 and AU6, although not present in the FACS manual mapping for disgust, were dominant. These two AUs were dominant in the expert annotated distribution for disgust (Figure 5.1). To determine whether these were false positives, samples where the model detected the presence of AU4 and AU6 were selected. Figure 5.7 shows the subjects depicting what appears to be AU4 and AU6. Therefore, the PL-1 model was accurate in registering these AUs in the disgust emotion.

For both PL-2 models, AU9 was a dominant AU alongside AU4, however just like PL-1 model, both did not detect the presence of AU26 in any of the disgust samples. When tested on KDEF samples annotated with the fear emotion, every AU registered by the PL-1 model is in line with the different combinations used by the FACS manual to code the emotion fear. This was also true for both PL-2 models however PL-2a detected more samples per AUs than PL-2b. For the KDEF samples annotated with happy emotion, the PL-1 model detected the presence of AU12, which is present in the FACS combinations used to code happy emotion. The PL-1 model detected the presence of AU6 and AU25. AU6 is present in the combinations used in the FACS manual to code the happy emotion. AU25 was the most dominant AU in the distribution even though it is not included in the FACS manual coding for the happy emotion. However, after further investigation, most of the subjects have their lips parted when expressing happy emotion. The PL-2a model had a similar distribution to PL-1 while in PL-2b, AU6 and AU12 were the dominant AUs for the happy emotion.

When tested on KDEF samples annotated with sad emotion, the dominant AUs detected by the PL-1 model were in line with the AUs used by the FACS manual to code

Emotion	AUs
Anger	4+5+7+10+22+23+25/26
	4+5+7+10+23+25/26
	4+5+7+17+23/24
	4+5+7+23/24
	4+5/7
	17+24
Disgust	9/10+17
	9/10+16+25/26
	9/10
Fear	1+2+4
	1+2+4+5+20+25/26/27
	1+2+4+5+25/26/27
	1+2+4+5
	1+2+5+25/26/27
	5+20+25/26/27
	5+20
	20
Happy	12
	6+12
Sadness	1+4
	1+4+11/15
	1+4+15+17
	6+15
	11+17
	1
Surprise	1+2+5+26/27
	1+2+5
	1+2+26/27
	5+26/27

FIGURE 5.5: The AU mappings used by the FACS manual to code the six universal emotions.

the sad emotion. However, PL-2a detected more AUs per sample than PL-1 and PL-2b in the sad samples. For the KDEF samples annotated with the surprise emotion, the AUs detected by the PL-1 model were in line with the FACS manual coding. When tested on KDEF samples annotated with the neutral emotion, the PL-1 model registered AU5 in 8 samples. The PL-2a model registered AU4 in 22 samples. These false positives are presented in Figure 5.8a and Figure 5.8b.

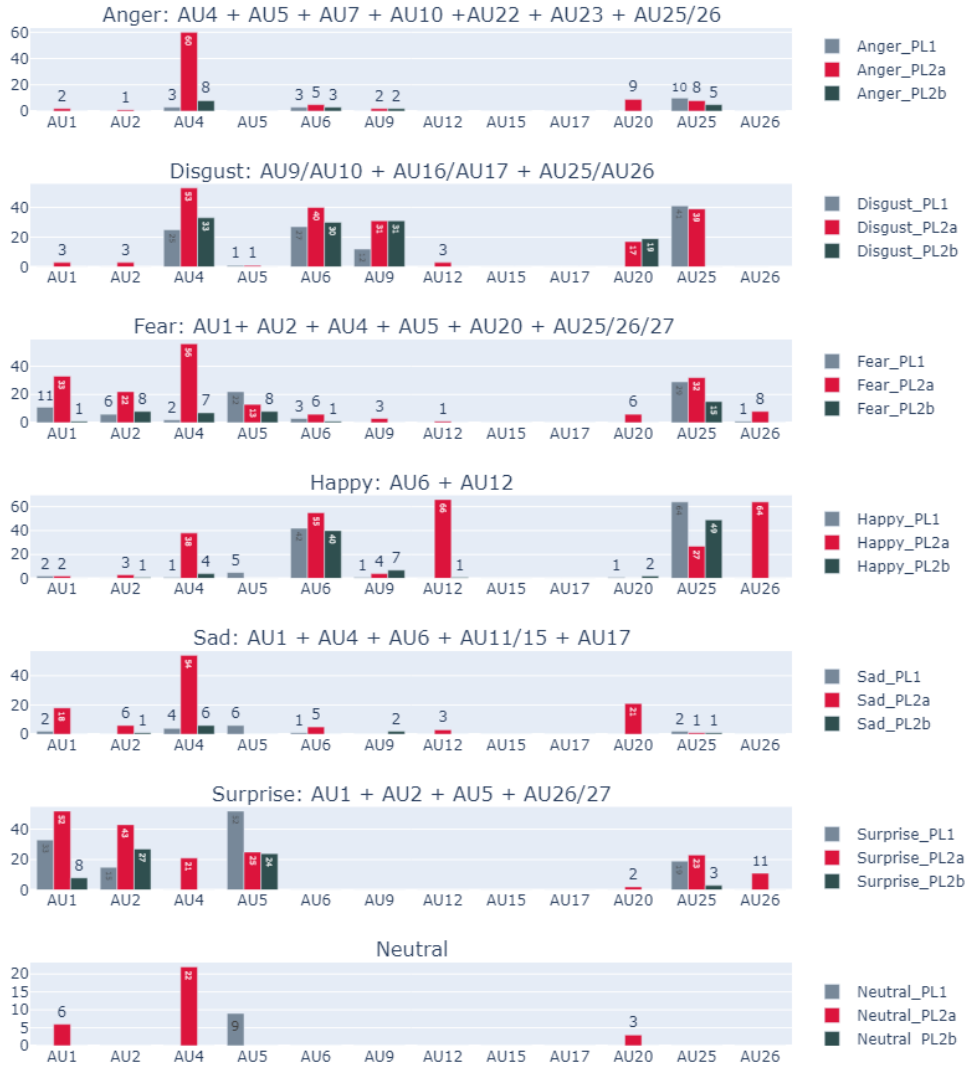
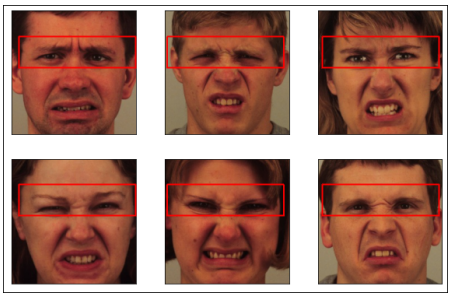


FIGURE 5.6: PL-2a and PL-2b predicted AU distribution for KDEF dataset.

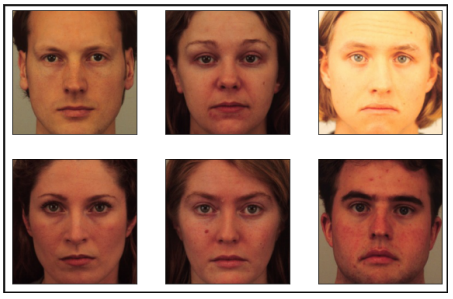
With incomplete supervision, this model has been used to generate relevant AUs for an unlabelled dataset. This effort is considerably less laborious, expensive, and time-intensive than manually annotating the entire dataset with the assistance of a FACS coder. However, the model did not detect the presence of AU12, AU15, AU17, AU20, and AU26 in any of the KDEF samples. A FACS coder could be used to focus on annotating these AUs. This is less effort than focusing on 12 AUs.



(A) KDEF subjects depicting the presence of AU4 (brow lowerer) in their expression of disgust.

(B) KDEF subjects depicting the presence of AU6 (cheek raiser) in their expression of disgust.

FIGURE 5.7: Samples taken from KDEF where the PL-1 model detected the presence of AU4 and AU6 [Lundqvist et al., 1998].



(A) The neutral KDEF samples in which the model detected the presence of AU5 (upper lid raiser) [Lundqvist et al., 1998].

(B) KDEF subjects depicting the presence of AU4 (cheek raiser) in their expression of neutral.

FIGURE 5.8: Samples taken from KDEF where the PL-1 and PL-2a models detected the presence of AU5 and AU4 in neutral samples, respectively [Lundqvist et al., 1998].

5.2 AU Recognition using Noisy Labels with Large-Loss Rejection

This section presents the experimental results using the large-loss rejection method. The LLR method was used for AU recognition to address the challenge of acquiring expert AU-annotated datasets. LLR was evaluated on both DISFA+ and KDEF datasets using a grid search for optimal hyperparameters. The best performing AU model trained with the LLR mechanism (configuration-5 LLR model) when evaluated on the DISFA+ test set achieved a subset accuracy of 69% and a 0.65 F1 score (Table 5.2). This model was then used to generate AUs for each emotion class in the DISFA+ and KDEF test datasets. The result of this AU mapping analysis is presented below. The classification results per AU class are presented in Table 5.3. The model achieved a weighted average precision, recall, and F1-score across all the classes was 85%, 56%, and 0.65, respectively. The model did not detect two AUs with the lowest count; AU15

TABLE 5.2: Results of LLR models when evaluated on the DISFA+ test set.

Test			
Configuration	**Subset Accuracy**	**F1-Score**	**Average Precision**
5	68.7%	0.649	0.532

and AU17, in any of the test samples. Figure 5.9 shows the ground truth AU distribu-

TABLE 5.3: Classification performance of the LLR model per class on DISFA+ test set.

AU	**Precision**	**Recall**	**F1-Score**
1	19%	40%	26%
2	86%	45%	59%
4	94%	50%	66%
5	87%	82%	84%
6	97%	33%	49%
9	65%	22%	33%
12	100%	50%	67%
15	0%	0%	0%
17	0%	0%	0%
20	15%	60%	24%
25	89%	81%	85%
26	93%	47%	49%

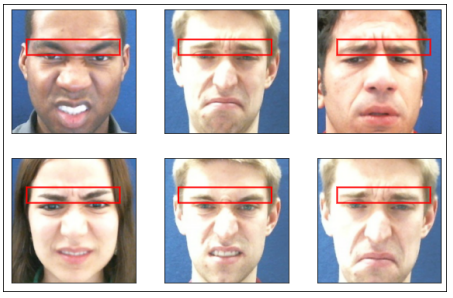
tion for a sample set of 100 from DISFA+ compared with the predictions obtained by the LLR model. The similarity between the distribution indicates the potential use of the LLR technique.

For the DISFA+ anger distribution, the dominant AUs registered by the model are included in the FACS coding for anger (Figure 5.9). The model did not register AU7, AU10, AU22, and AU23 as they were not part of the distribution when finalising the model's output layer. In terms of disgust-annotated samples, AU4, AU9, and AU25 were the dominant AUs in the ground truth distribution (Figure 5.9). In these dominant AUs, AU4 and AU6 are the only missing AUsin the FACS coding for the disgust emotion. In Figure 5.9, we see that AU4 is still dominant and is consistent with the initial distribution. The presence of AU4 and AU20 was investigated in Figure 5.10, these AUs were not false positives. The LLR model performed well in predicting the AUs used in the FACS manual to code the fear emotion. The AUs registered by the model for the fear emotion were AU1, AU2, AU4, AU6, AU25, and AU26.

The FACS manual codes the emotion happy with either AU12 or the combination of AU6 and AU12. In Figure 5.9, we see that AU20, AU25, and AU26 are included in the expert annotated distribution for the emotion happy. AU4 was not present in the expert-annotated distribution for happiness, however, the model detected AU4 in 10 samples. On further investigation, the eyebrow shape of the subject (Figure 5.9) is lowered which is the reason for this misclassification. The ground truth distribution and

FIGURE 5.9: The ground truth AU distribution compared with the distribution predicted by the LLR model on the DISFA+ sample test set.

(A) DISFA+ disgust samples in which the LLR model detected AU20 (lip stretcher).



(B) DISFA+ disgust samples in which the LLR model detected AU4 (brow lowerer).

FIGURE 5.10: Samples taken from DISFA+ where the model detected AU4 and AU20, both of which are not present in the FACS coding for disgust [Mavadati et al., 2016].

the predicted distribution for the sadness emotion are similar. The ground truth distribution and predicted distribution for the surprise emotion are identical. The dominant

AUs in both distributions were listed in the FACS manual to code the surprise emotion.

Overall for the DISFA+ test samples, the LLR model was able to recognise the relevant AUs for each of the seven emotions. The model however detected the presence of AUs in the samples annotated with the neutral expression. For AU4, more than ten neutral samples were false positives, and upon further investigation, the subject's eyebrow shape could be the cause of the classification. Figure 5.11 shows these misclassifications.

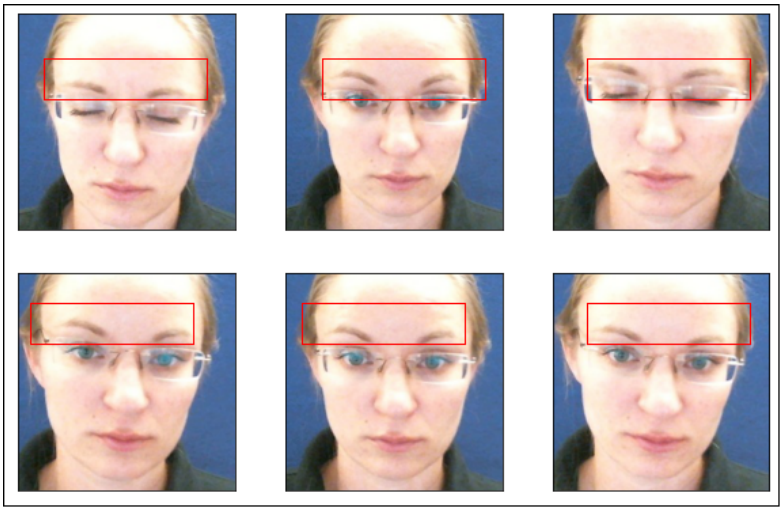


FIGURE 5.11: Neutral samples from DISFA+ where the LLR model detected AU4 (brow lowerer). The shape of the eyebrows are arched downwards [Mavadati et al., 2016].

5.2.1 Evaluation of AU Recognition using LLR Model on KDEF Dataset

The proposed LLR-based model was evaluated on the KDEF dataset to validate its performance on cross-datasets. A successful LLR model should be able to produce similar AU distributions per emotion, similar to the FACS manual. Figure 5.12 presents the AU distribution registered by the LLR model when given the KDEF dataset.

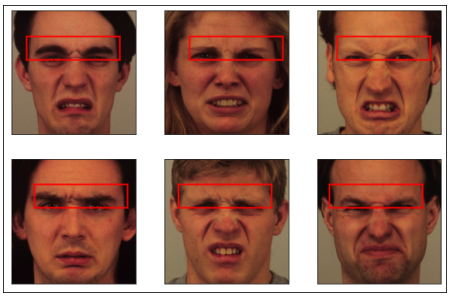
In Figure 5.12, the dominant AUs for anger picked up by the LLR model are AU4 and AU25. These two AUs are present in the FACS manual coding for anger. The model did not register AU5 in any of the anger-annotated KDEF samples. AU5 is the only AU in the FACS coding for anger, present in the DISFA+ annotations, missed by the model. The model also registered five samples with AU6. Only one sample displayed AU6 (cheek raiser), the remaining can be considered false positives or would need a FACS coder to code the intensity of the AU (Figure 5.13a). When tested on KDEF samples annotated with the disgust emotion, the dominant AUs picked up by the model were

FIGURE 5.12: AU annotations generated per emotion by the LLR model for the samples in KDEF dataset.

AU4, AU6, and AU25. AU25 is the only AU that is present in the FACS coding for disgust. AU4 is not in line with the FACS manual mapping for disgust; however, after further investigation, the model detected this AU correctly (Figure 5.13b).

For KDEF samples annotated with fear emotion, the LLR model detects the existence of AU1, AU2, AU4, AU5, AU6, and AU25 from samples annotated as depicting fear emotion. This was in line with the mapping provided by FACS and DISFA+ distributions. The LLR model performed well in this regard, as it picked the relevant AUs to map to the fear emotion. When tested on KDEF samples annotated with the happy emotion, the dominant AUs picked up by the model were AU6 and AU25. The model did not register the presence of AU12. The FACS manual uses AU6 and AU12 to code the happy emotion. AU25 is present in the ground-truth DISFA + distribution for the happy emotion. Figure 5.14 shows samples in which the model detected the presence of AU25.

(A) KDEF anger samples where model detected AU6 (cheek raiser).



(B) KDEF disgust samples where AU4 was detected by the model.

FIGURE 5.13: The KDEF anger samples and disgust samples in which AU recognition model utilising LLR mechanism detected AU6 and AU4, respectively Lundqvist et al. [1998].

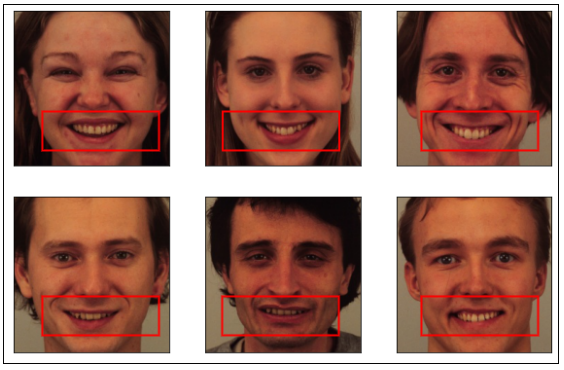
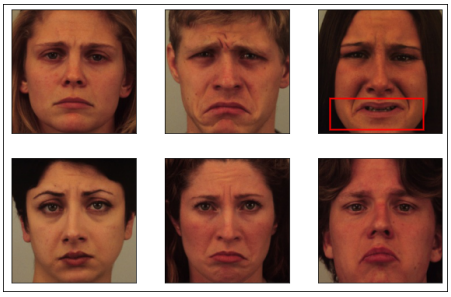


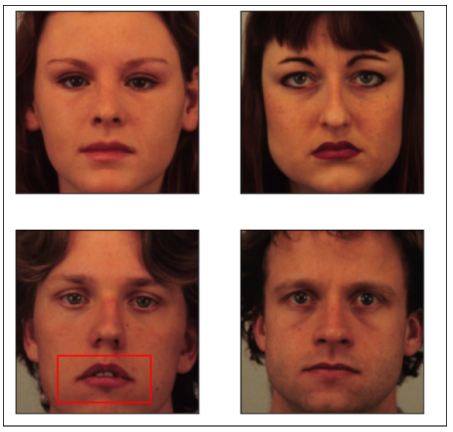
FIGURE 5.14: The model detected the presence of AU25 (lips part) in these images annotated with the happy emotion [Lundqvist et al., 1998].

When tested on the KDEF samples annotated with the sad emotion, the model registered AU1, AU4, AU5, AU6, and AU25. The dominant AUs were AU1, AU4, and AU5. AU5 and AU25 are not present in the FACS coding for the sad emotion. AU5 is not as dominant in the expert annotated distribution in Figure 5.9. The LLR model did not reject AU5 as a large loss label. The presence of AU25 in this distribution was investigated. The samples in which the model detected the presence of AU25 were selected. Only one of the subjects had their lips parted to express the sadness emotion. Figure 5.15a presents this misclassification by the model. When the LLR model was tested on KDEF samples annotated with the surprise emotion, the majority of AUs registered were in the FACS coding for surprise emotion. AU25 was the only AU picked up by the model that is not in the FACS coding. However, AU25 is a dominant AU in the expert annotated distribution for surprise in Figure 5.9. The majority of the KDEF samples annotated with the neutral expression were regarded as not depicting any AUs by the LLR model. There were nine samples which were false positives that received annotations of AU5. There were 6 neutral samples in which the model detected the

presence of AU25. On further investigation, only one of these samples has their lips parted (slightly) while expressing the neutral expression (Figure 5.15b).



(A) KDEF sad samples in which the model detected the presence of AU25.



(B) KDEF neutral samples in which the detected the presence of AU25.

FIGURE 5.15: The samples from Neutral and Sad class in which the model detected the presence of AU25 [Lundqvist et al., 1998].

5.3 AU Recognition Using Transfer Learning

5.3.1 Evaluation of Baseline Emotion Recognition Model on DISFA+ Dataset

Each ER model (different hyperparameter configurations) was evaluated on the DISFA+ test set. The results are presented in Table 5.4. Configuration-1 was selected as the best performing ER model due to its performance on validation and test sets. Configuration-1 reported an overall accuracy of 96.9% with an error of 0.113 and an F1 score of 0.969. Figure 5.16 presents the confusion matrix that depicts the result of the configuration-1 classification on the test set, while Table 5.5 presents the prediction performance per class. Configuration-1 achieved an average per class accuracy, precision, and recall of 99%, 98%, and 96%, respectively.

TABLE 5.4: Results on the DISFA+ test set.

Configuration	Test loss	Test Accuracy	Test F1-Score
1	0.113	96.9%	0.969

5.3.2 Evaluation of Baseline Emotion Recognition Model on KDEF Dataset

The best-performing ER model (Configuration-1) was further evaluated on the KDEF dataset (cross-dataset evaluation). The model achieved an overall accuracy of 90%,

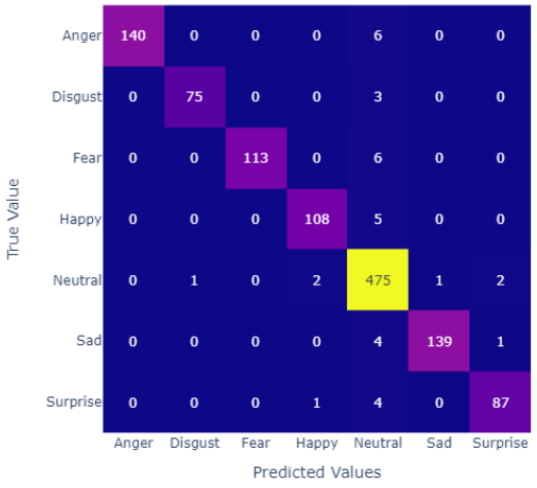


FIGURE 5.16: Confusion matrix depicting config-1 model's performance on DISFA+ test set

TABLE 5.5: Config-1 Results per Emotions on DISFA+ test set

Emotion	No. of Samples	Accuracy	Precision	Recall	F1-score
Anger	146	99%	100%	96%	0.98
Disgust	78	100%	96%	99%	0.97
Fear	119	99%	100%	95%	0.97
Happy	113	99%	97%	96%	0.96
Neutral	481	97%	94%	99%	0.97
Sad	144	99%	99%	97%	0.98
Surprise	92	99%	97%	95%	0.96

with a precision, recall, and F1 score of 63%, 60%, and 0.58, respectively, for the seven emotions (including neutral). Figure 5.17 presents the confusion matrix depicting the classification results, while Table 5.6 presents the prediction performance per class.

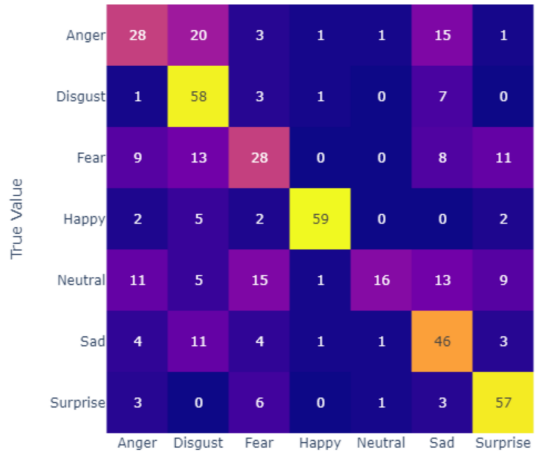


FIGURE 5.17: Confusion matrix depicting config-1 model's performance on KDEF dataset

TABLE 5.6: ER Results per Emotions on KDEF (Cross-Dataset)

Emotion	Samples	Accuracy	Precision	Recall	F1-score
Anger	69	85%	48%	41%	0.44
Disgust	70	91%	52%	83%	0.64
Fear	69	85%	46%	41%	0.43
Happy	70	97%	94%	84%	0.89
Neutral	70	88%	84%	23%	0.36
Sad	70	86%	50%	66%	0.57
Surprise	70	95%	69%	81%	0.75

5.3.3 Evaluation of Transfer Learning Based AU model on DISFA+ Dataset

Each AU model (different hyperparameter configuration) was evaluated on the DISFA+ test set. The evaluation results are presented in Table 5.7. The model trained with configuration-4 hyperparameters was selected as the best performing model. This was because the model achieved the highest subset accuracy on the development and test

sets. When tested on DISFA+ samples for recognition of 12 AUs, the configuration-4 model reported a subset accuracy and F1-score of 97% and 0.98, respectively. The model reports good generalisation ability with weighted recall and precision both at 98%. The classification report per AU class for the TL AU model is presented in Table 5.8.

TABLE 5.7: Results of configuration-4 on DISFA+ test set.

Configuration	**Test loss**	**Test Accuracy**	**Test F1-Score**
4	0.0118	96.7%	0.98

TABLE 5.8: Configuration-4 model classification performance per class on DISFA+ test set.

AU	Precision	Recall	F1-score
1	98%	95%	0.97
2	99%	99%	0.99
4	99%	99%	0.99
5	98%	97%	0.97
6	96%	95%	0.96
9	100%	98%	0.99
12	98%	99%	0.98
15	96%	100%	0.98
17	96%	100%	0.98
20	97%	100%	0.99
25	99%	98%	0.98
26	94%	99%	0.97

An AU mapping analysis was done to determine which AUs were detected by this model per emotion class. For this mapping analysis, the AU distribution generated by the AU model per emotion is compared with the AUs which are used by the FACS manual to code an emotion. A sample of 100 images from each emotion was randomly selected, but meeting the subject-independent condition with respect to train and validation sets, from the DISFA+ test set and the AU model was then used to detect the AUs present in each image per emotion. Figure 5.18 shows the true AU distribution of the randomly selected images and the AU distribution for each emotion detected by the TL model. We can see that the distributions are almost identical, indicating how well the TL model could learn the relevant AUs present in each emotion.

5.3.4 Evaluation of AU Recognition with Transfer Learning Model on KDEF Dataset

The KDEF dataset does not contain AU annotations; therefore, this phase aimed to evaluate if the estimated AU labels for samples from the KDEF dataset are in line with the respective emotion class labels provided. A successful TL model should be able

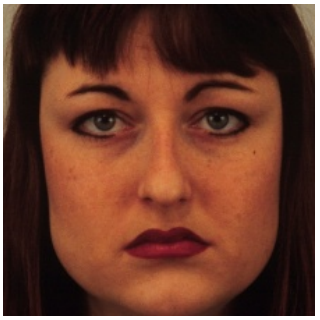
FIGURE 5.18: Groundtruth distribution of the selected sample and the distribution predicted by the TL model.

to produce similar AU distributions per emotion as a certified FACS coder would have provided. The distribution detected by the model was compared with the FACS coding for each emotion in Figure 5.5. The AU distribution detected by the model is presented in Figure 5.19. A deeper analysis of the results presented in 5.19 shows that AU4 was a dominant AU present in KDEF samples annotated with anger emotion, similar to the DISFA+ sample distribution. Possible AU combinations that depict the emotion of anger confirm the dominance of AU4 for anger. Other significant AUs include AU5 and/or AU7. As expected, the majority of samples from the neutral class are regarded as samples that do not depict any AUs by our proposed model. The set of false positives includes two samples that received AU1 annotations and one that received AU5 annotations. On further observation of the samples, we attributed this to the shape and arch of the inner and outer eyebrows for AU1 detected by the model (Figure 5.20).

The dominant AUs estimated by the proposed model for the samples annotated with

FIGURE 5.19: The FACS manual investigation guide for emotion to AU mapping and the predictions made by the TL model when given a sample of KDEF images.

(A) KDEF subject expressing the neutral emotion. The model registered AU1 in this facial expression.



(B) KDEF subject expressing the neutral emotion. The model registered AU1 in this facial expression.

FIGURE 5.20: These two neutral samples were annotated with AU1 due to the shape of the subjects' eyebrows [Lundqvist et al., 1998].

disgust were AU4, AU6, AU9, AU15, and AU25. The FACS manual uses different combinations of AU9 or AU10, AU16, AU17, and AU25 or AU26 to code the disgust emotion. AU10 and AU16 are not present in the DISFA+ dataset, which is why the model does not detect these 2 AUs. It registered only one sample as having AU17 for disgust emotion. This AU (AU17) was also in the minority for disgust in Figure 5.18. AU4 and AU6 were dominant in the model's prediction for disgust. These AUs were present in the expert annotated DISFA+ distribution. Therefore, the TL model registered these AUs in Figure 5.5. As per FACS coding, they coded the disgust emotion as AU9+AU15+AU16 (missing from the DISFA+ dataset).

The dominant AUs registered by the model for the KDEF samples annotated with fear emotion were AU1, AU2, AU4, AU5, and AU26. This was in line with the ground truth DISFA+ distribution provided in Figure 5.18 and FACS coding provided in Figure 5.5. Tested on KDEF samples annotated with happy emotion, the model registered AU1, AU4, AU5, AU9, AU12, and AU25. AU6, AU12, and AU25 were the dominant AUs in the KDEF distribution for happiness (Figure 5.18). In the FACS manual, they coded happiness with AU12 or the combination of AU6 and AU12. In the DISFA+ distribution in Figure 5.18, the dominant AUs in the happiness distribution are AU6, AU12, AU20, and AU25. Most of the subjects have their lips parted when expressing the happy emotion. Therefore, the model registered AU25 in the happiness emotion. DISFA+ distribution (Figure 5.18), AU20, AU25, and AU26 were also detected along with the AUs coded in the FACS coding. However, AU1, AU4, and AU9 were not dominant AUs in the happy distribution.

When tested on KDEF samples annotated with the sadness emotion, the model registered AU1, AU4, AU5, AU15, AU17, and AU20. The dominant AUs in this distribution were AU1, AU4, and AU15. This is in line with the FACS manual (Figure 5.5). When tested on KDEF samples annotated with the surprise emotion, the model registered AU1, AU2, AU5, AU25, and AU26. This is in line with the FACS manual, which codes the surprise emotion with different combinations of AU1, AU2, AU5, and AU26 or AU27 (missing from the DISFA+ dataset).

From the results of the mapping analysis, the model trained with the transfer learning approach has been able to generate accurate AU annotations for an unlabelled dataset. The model was able to do this without requiring the 100 hours of training needed by a FACS coder to achieve minimal competency. This shows that the knowledge gained from a model initially trained with emotion (coarse labels) can be used for the task of AU recognition and can generate labels for an unlabelled dataset.

5.4 Model Performance Compared to State-of-the-art methods

In this section, we compare the models presented in this research work with state-of-the-art methods for AU detection using DISFA+. The metric used for comparison is the weighted average F1-score across the 12 AUs. The state-of-the-models used to compare the performance of each technique presented in this research work are: Deep Relational Metric Learning (DRML), AU R-CNN [Ma et al., 2019], and JAA-Net [Shao et al., 2021]. The results in Table 5.9 are from [Chen et al., 2020]. The WSL techniques performed well relative to these state-of-the-art models. PL-2a had superior performance to all the state-of-the-art models.

TABLE 5.9: F1-score result comparison with state-of-the-art methods on DISFA+ dataset.

Model	**DRML**	**AU R-CNN**	**JAA-Net**	**PL-1**	**PL-2a**	**PL-2b**	**LLR**	**TL**
F1-Score	0.357	0.382	0.726	0.56	0.9	0.44	0.649	0.97

5.5 Conclusion

The performance of the DISFA+ test set for each approach used in this research work was presented. The emotion-to-AU mapping analysis performed for each AU recognition approach was also presented. The AUs detected by each approach were compared with the AU mapping provided by the FACS manual. This research work set out to test whether incomplete supervision and inaccurate supervision approaches could be used to generate AU annotations for an unlabelled dataset. The pseudo-Labelling approach was used to test the incomplete supervision hypothesis. With 20% of the DISFA+ AU annotations used to train the pseudo-labelling AU recognition models, the AUs generated when tested on the unlabelled (no AU annotations) KDEF dataset were overall in line with the FACS mapping. The PL-1 model achieved a subset accuracy of 68% and an F1-score of 0.56 on the DISFA+ test. For PL-2a, the model achieved a subset accuracy of 89% and an F1-score of 0.9 on the test set. PL-2b achieved a subset accuracy of 66% and an F1-score of 0.44 on the same DISFA+ set. However, PL models struggled to detect the presence of AU12, AU15, AU17, AU20, and AU26 in the KDEF samples. AU15 and AU17 had the lowest amount of samples in the DISFA+ dataset and would hinder the ability for the model to learn the features in these two classes. Compared to the transfer learning model, which was trained without data limitations, AU15, AU17, and AU26 achieved the lowest F1-score when evaluated on the DISFA+ test set. When the TL model was tested on KDEF samples, these aforementioned AUs were also the

least detected AUs.

An AU recognition model trained with the LLR mechanism was used to test the inaccurate supervision hypothesis for unlabelled data. This model was trained with the noisy dataset generated during the pseudo-labelling version one process. When evaluated on the DISFA+ test set, the model achieved a subset accuracy of 69% and an F1-score of 0.66. The LLR mechanism also struggled to accurately learn AU15 and AU17, similar to the pseudo-labelling approach. The KDEF dataset was also used to test how the LLR mechanism would perform on unlabelled data. The AUs generated by the model when tested on the KDEF samples were also in line with the FACS manual. However, in the emotion to AU mapping performed, AU2, AU9, AU15, AU17, AU20, and AU26 were the least detected AUs in the KDEF samples.

Chapter 6

Conclusion

Facial expression recognition is a popular research area in computer vision and affective computing because of the many applications in facial behaviour analysis to which it can be applied. FER plays a significant role in application areas such as diagnosis of mental diseases, human-computer interaction, and autonomous robots [Wang and Gu, 2018; Samadiani et al., 2019]. Facial behaviour can be used to convey non-verbal communication such as determining the mental state of a person. This can be done by analysing the movements of facial muscles. The FACS manual is a standard developed to classify the different possible types of facial expressions. The FACS manual uses action units to describe the fundamental actions of individual muscles or groups of muscles to create a specific face movement.

Deep learning networks and computing hardware has advanced in recent years and has enabled computers to manage complex data representations and replicate the human performance. However, the performance of deep learning depends on the amount of data used to train the model. Deep learning researchers are now faced with the challenge of annotating large amounts of training data. The annotation process requires a significant amount of human labour and requires subject matter experts. This process is timely and cost-intensive. In FBA, a FACS coder is needed to annotate the data with FACS action units. As a subject matter expert, the FACS coder needs a minimum of 100 hours to achieve minimal competency. The process of annotating with FACS action units is also time intensive. This is why researchers have been exploring the use of creating deep learning models trained with weak supervision.

By the end of this research work, we were able to answer the two research questions and compare the performance of each weakly supervised technique with that of a transfer learning model in generating AU annotations.

1. *Given a limited amount of AU-annotated data samples and a significantly larger amount of unlabelled data samples, can we use WSL to train an accurate CNN that can generate AU-annotations for the unlabelled test data?*

   An incomplete supervision technique, Pseudo-labelling, was used to answer this research question. Here we used a limited amount of AU-annotated samples combined with a larger sample with pseudo-AU labels to train a model. There were two methods used to accomplish this technique. Three models, PL-1, PL-2a, and PL-2b, were created and used to generate AUs on the KDEF dataset. We trained a modified VGG16 network with 20% of the DISFA+ samples with AU annotations. The model was used to generate pseudo-labels for an unlabelled sample which was 70% of the DISFA+ dataset. An existing VGG16 network was trained with the expert and pseudo-labelled datasets and was evaluated on the final 10% of the DISFA+ dataset. The pseudo-label model trained with the first version achieved a 68% subset accuracy for AU recognition when evaluated on the test split. The model was used to generate AUs for the KDEF dataset with no AU annotations. When compared to the emotion to AU mappings provided by the FACS manual, the AU generated by the model were in line. However, the model struggled to detect some AUs in the KDEF samples. Some of these AUs were the least represented in the DISFA+ dataset.

2. *Given an AU dataset with inaccurate annotation, can we use weak supervision to train a CNN that can generate accurate labels?*

   An inaccurate supervision technique, Large Loss Rejection, was used to answer this research question. Here the model was trained with a noisy dataset. The technique was the Large-Loss Rejection approach. In this approach, the labels that had a large loss were rejected by the model to prevent the model from learning erroneous labels. For the inaccurate supervision scenario, the LLR model achieved a subset accuracy of 66% and a weighted average F1-score of 0.56 on the DISFA+ test set. The LLR model was used to generate AU annotations for the KDEF dataset. When tested on the kDEF dataset, the AUs generated by the model were in line with the emotion to AU mapping provided in the FACS manual. Similar to the PL-model, the AUs with the lowest samples in the DISFA+ dataset were not detected in the KDEF samples.

These two approaches were compared with a transfer learning model that was initially trained with emotion-labelled data. The baseline emotion recognition model was trained with samples from the DISFA+ dataset. This emotion recognition model was

fine-tuned to transfer the knowledge gained during the emotion recognition training to AU recognition. The transfer learning model was tested on the DISFA+ dataset and reported an accuracy of 97% and a weighted average F1-score of 0.98 on the DISFA+ test set. The model was tested on the KDEF dataset which does not contain AU annotations. The AUs detected by the model using the transfer learning approach were in line with the FACS manual and the coding provided in the DISFA+ dataset.

In conclusion, this research work was able to illustrate the potential for WSL for generating AU annotations for unlabelled and inaccurate datasets. The emotion to AU mapping for each WSL approach was different to the transfer learning model's performance. Given the misclassifications discussed in Chapter 4, it is clear that both DISFA+ and KDEF datasets have potential samples with incorrect annotations. A TL model would learn these erroneous annotations and predict the same for any unseen samples from the same distributions. However, WSL models, on the other hand, were only exposed to limited labelled data and have shown some promising results. For future work, other approaches to better generate pseudo labels and have balanced rejection policies might improve the performance of the classification models trained on limited "good" data.

Bibliography

[Ahuja et al., 2018] Karishma Ahuja, Anand Krishnan K V, A. Kiran, E.M. Dalin, and Shivaji Sagar. Smart office surveillance robot using face recognition. *International Journal of Mechanical and Production Engineering Research and Development*, 8:725–734, 06 2018. doi: 10.24247/ijmperdjun201877.

[Albawi et al., 2017] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[Albawi et al., 2018] Saad Qassim Fleh Albawi, Fatih Yetkin, and Kerem Altun. Social touch gesture recognition using deep neural network. 2018.

[Almaev et al., 2015] Timur Almaev, Brais Martinez, and Michel Valstar. Learning to transfer: Transferring latent task structures and its application to person-specific facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[Atikuzzaman et al., 2019] Md Atikuzzaman, Md Asaduzzaman, and Md Zahidul Islam. Vehicle number plate detection and categorization using cnns. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–5. IEEE, 2019.

[Baheti, 2023] Pragati Baheti. Activation functions in neural networks [12 types amp; use cases], Feb 2023. URL https://www.v7labs.com/blog/neural-networks-activation-functions.

[Baldi and Sadowski, 2013] Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26, 2013.

[Barrett, 2006] Lisa Feldman Barrett. Solving the emotion paradox: Categorization and the experience of emotion. *Personality and social psychology review*, 10(1):20–46, 2006.

[Ben-Yosef and Weinshall, 2018] Matan Ben-Yosef and Daphna Weinshall. Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images. *arXiv preprint arXiv:1808.10356*, 2018.

[Bennett, 2015] Janet M Bennett. *The SAGE encyclopedia of intercultural competence*. Sage Publications, 2015.

[Blauch et al., 2021] Nicholas M Blauch, Marlene Behrmann, and David C Plaut. Computational insights into human perceptual expertise for familiar and unfamiliar face recognition. *Cognition*, 208:104341, 2021.

[Bottou, 2012] Léon Bottou. Stochastic gradient descent tricks. *Neural Networks: Tricks of the Trade: Second Edition*, pages 421–436, 2012.

[Bradski, 2000] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[Cao et al., 2021] Zixuan Cao, Yongmei Zhou, Aimin Yang, and Sancheng Peng. Deep transfer learning mechanism for fine-grained cross-domain sentiment classification. *Connection Science*, 33(4):911–928, 2021.

[Cascante-Bonilla et al., 2021] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6912–6920, 2021.

[Chakrabarty and Chatterjee, 2019] Navoneel Chakrabarty and Subhrasankar Chatterjee. A novel approach to age classification from hand dorsal images using computer vision. 03 2019. doi: 10.1109/ICCMC.2019.8819632.

[Chapelle and Zien, 2005] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics*, pages 57–64. PMLR, 2005.

[Chen et al., 2015] Bor-Chun Chen, Chu-Song Chen, and Winston H Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Transactions on Multimedia*, 17(6):804–815, 2015.

[Chen et al., 2020] Jing Chen, Chenhui Wang, Kejun Wang, and Meichen Liu. Computational efficient deep neural network with difference attention maps for facial action unit detection. *arXiv preprint arXiv:2011.12082*, 2020.

[Chen and Qian, 2019] Zhuang Chen and Tieyun Qian. Transfer capsule network for aspect level sentiment classification. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 547–556, 2019.

[Chollet, 2019] Francois Chollet. Keras documentation: Writing a training loop from scratch, Mar 2019. URL https://keras.io/guides/writing_a_training_loop_from_scratch/.

[Coan, 2010] James A Coan. Emergent ghosts of the emotion machine. *Emotion Review*, 2(3): 274–285, 2010.

[Cohen et al., 2003] Ira Cohen, Nicu Sebe, Fabio G Cozman, and Thomas S Huang. Semi-supervised learning for facial expression recognition. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 17–22, 2003.

[DeepAI, 2019] DeepAI. Batch normalization, May 2019. URL https://deepai.org/machine-learning-glossary-and-terms/batch-normalization.

[Deng et al., 2012] J Deng, A Berg, S Satheesh, H Su, A Khosla, and L Fei-Fei. Imagenet large scale visual recognition competition. *ilsvrc2012*, 2012.

[Dietterich et al., 1997] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

[Dong et al., 2021] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson WH Lau. Location-aware single image reflection removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5017–5026, 2021.

[Du et al., 2018] Hao Du, Yuan He, and Tian Jin. Transfer learning for human activities classification using micro-doppler spectrograms. In *2018 IEEE International Conference on Computational Electromagnetics (ICCEM)*, pages 1–3. IEEE, 2018.

[Earnshaw, 2017] Berton Earnshaw. A brief survey of tensors, Mar 2017. URL https://www.slideshare.net/BertonEarnshaw/a-brief-survey-of-tensors.

[EIA, 2022] EIA. Facial action coding system (facs), Jul 2022. URL https://www.eiagroup.com/study/facial-expressions/facial-action-coding-system-facs/.

[Ekman and Friesen, 1971] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

[Ekman and Friesen, 1978] Paul Ekman and Wallace V Friesen. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*, 1978.

[Ekman et al., 2002] Paul Ekman, Wallace V Friesen, and Joseph C Hager. *Facial Action Coding System: Facial action coding system: the manual: on CD-ROM*. Research Nexus, 2002.

[Frénay and Verleysen, 2013] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[Grandini et al., 2020] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

[Granger et al., 2021] Eric Granger, Patrick Cardinal, et al. Weakly supervised learning for facial behavior analysis: A review. *arXiv preprint arXiv:2101.09858*, 2021.

[Griffin et al., 2007] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[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.

[Kasinski and Schmidt, 2010] Andrzej Kasinski and Adam Schmidt. The architecture and performance of the face and eyes detection system based on the haar cascade classifiers. *Pattern Analysis and Applications*, 13:197–211, 2010.

[Khalifa and Frigui, 2016] Amine Ben Khalifa and Hichem Frigui. Multiple instance fuzzy inference neural networks. *arXiv preprint arXiv:1610.04973*, 2016.

[Kim et al., 2022] Youngwook Kim, Jae Myung Kim, Zeynep Akata, and Jungwoo Lee. Large loss matters in weakly supervised multi-label classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14156–14165, 2022.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kodzoman, 2017] Vinko Kodzoman. Pseudo-labeling a simple semi-supervised learning method, 2017. URL https://datawhatnow.com/pseudo-labeling-semi-supervised-learning/1. Last accessed 16 September 2017.

[Krizhevsky et al., 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Krizhevsky et al., 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[Kulkarni et al., 2020] Ajay Kulkarni, Deri Chong, and Feras A Batarseh. Foundations of data imbalance and solutions for a data democracy. In *data democracy*, pages 83–106. Elsevier, 2020.

[LeCun et al., 1998] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. Mnist handwritten digit database, yann lecun, corinna cortes and chris burges, 1998. URL http://yann.lecun.com/exdb/mnist/.

[Lee et al., 2013] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[Lewinski et al., 2014] Peter Lewinski, Tim M Den Uyl, and Crystal Butler. Automated facial coding: Validation of basic emotions and facs aus in facereader. *Journal of Neuroscience, Psychology, and Economics*, 7(4):227, 2014.

[Li and Deng, 2020] Shan Li and Weihong Deng. Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, 2020.

[li Tian et al., 2004] Ying li Tian, Takeo Kanade, and Jeffrey F. Cohn. Chapter 11 . facial expression analysis. 2004.

[Lundqvist et al., 1998] Daniel Lundqvist, Anders Flykt, and Arne Öhman. Karolinska directed emotional faces. *Cognition and Emotion*, 1998.

[Lydia and Francis, 2019] Agnes Lydia and Sagayaraj Francis. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci*, 6(5):566–568, 2019.

[Ma et al., 2019] Chen Ma, Li Chen, and Junhai Yong. Au r-cnn: Encoding expert prior knowledge into r-cnn for action unit detection. *neurocomputing*, 355:35–47, 2019.

[Mahesh, 2020] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.

[Mavadati et al., 2016] Mohammad Mavadati, Peyten Sanger, and Mohammad H Mahoor. Extended disfa dataset: Investigating posed and spontaneous facial expressions. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–8, 2016.

[Mehrabian, 2017] Albert Mehrabian. *Nonverbal communication*. Routledge, 2017.

[Mollahosseini et al., 2016a] Ali Mollahosseini, David Chan, and Mohammad H Mahoor. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter conference on applications of computer vision (WACV)*, pages 1–10. IEEE, 2016a.

[Mollahosseini et al., 2016b] Ali Mollahosseini, Behzad Hasani, Michelle J Salvador, Hojjat Abdollahi, David Chan, and Mohammad H Mahoor. Facial expression recognition from world wild web. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 58–65, 2016b.

[Nugroho and Yuniarti, 2022] Budi Nugroho and Anny Yuniarti. Performance of root-mean-square propagation and adaptive gradient optimization algorithms on covid-19 pneumonia classification. In *2022 IEEE 8th Information Technology International Seminar (ITIS)*, pages 333–338. IEEE, 2022.

[Phung and Rhee, 2018] Van Hiep Phung and Eun Joo Rhee. A deep learning approach for classification of cloud image patches on small datasets. *Journal of information and communication convergence engineering*, 16(3):173–178, 2018.

[Programmersought, 2021] Programmersought. Study notes: three directions of weak supervision - programmer sought, 2021. URL https://www.programmersought.com/article/87324536709/.

[Rawat and Wang, 2017] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29:1–98, 06 2017. doi: 10.1162/NECO_a_00990.

[Rifai et al., 2011] Salah Rifai, Yann N Dauphin, Pascal Vincent, Yoshua Bengio, and Xavier Muller. The manifold tangent classifier. *Advances in neural information processing systems*, 24, 2011.

[Romera-Paredes et al., 2013] Bernardino Romera-Paredes, Min SH Aung, Massimiliano Pontil, Nadia Bianchi-Berthouze, Amanda C de C Williams, and Paul Watson. Transfer learning to account for idiosyncrasy in face and body expressions. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE, 2013.

[Ruiz et al., 2015] Adria Ruiz, Joost Van de Weijer, and Xavier Binefa. From emotions to action units with hidden and semi-hidden-task learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3703–3711, 2015.

[Russakovsky et al., 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[Samadiani et al., 2019] Najmeh Samadiani, Guangyan Huang, Borui Cai, Wei Luo, Chi-Hung Chi, Yong Xiang, and Jing He. A review on automatic facial expression recognition systems assisted by multimodal sensor data. *Sensors*, 19(8):1863, 2019.

[Schreiner et al., 2006] Christopher Schreiner, Kari Torkkola, Mike Gardner, and Keshu Zhang. Using machine learning techniques to reduce data annotation time. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 2438–2442. SAGE Publications Sage CA: Los Angeles, CA, 2006.

[Sebe et al., 2005] N Sebe, Ira Cohen, Ashutosh Garg, and Thomas S Huang. Application: Facial expression recognition. *Machine Learning in Computer Vision*, pages 187–209, 2005.

[Settles, 2009] Burr Settles. Active learning literature survey. 2009.

[Shao et al., 2021] Zhiwen Shao, Zhilei Liu, Jianfei Cai, and Lizhuang Ma. Jaa-net: joint facial action unit detection and face alignment via adaptive attention. *International Journal of Computer Vision*, 129:321–340, 2021.

[Shetty et al., 2021] Anirudha B Shetty, Jeevan Rebeiro, et al. Facial recognition using haar cascade and lbp classifiers. *Global Transitions Proceedings*, 2(2):330–335, 2021.

[Sikka et al., 2014] Karan Sikka, Abhinav Dhall, and Marian Stewart Bartlett. Classification and weakly supervised pain localization using multiple segment representation. *Image and vision computing*, 32(10):659–670, 2014.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Soo, 2014] Sander Soo. Object detection using haar-cascade classifier. *Institute of Computer Science, University of Tartu*, 2(3):1–12, 2014.

[Szegedy et al., 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[Torrey and Shavlik, 2010] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[Vadapalli, 2011] Hima Bindu Vadapalli. Recognition of facial action units from video streams with recurrent neural networks: a new paradigm for facial expression recognition. 2011.

[Viola and Jones, 2001] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[Wang and Gu, 2018] Huan-Huan Wang and Jing-Wei Gu. The applications of facial expression recognition in human-computer interaction. In *2018 IEEE international conference on advanced manufacturing (ICAM)*, pages 288–291. IEEE, 2018.

[Wang et al., 2016] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[Weiss et al., 2016] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[Werner et al., 2020] Philipp Werner, Frerk Saxen, and Ayoub Al-Hamadi. Facial action unit recognition in the wild with multi-task cnn self-training for the emotionet challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 410–411, 2020.

[Weston et al., 2008] Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th international conference on Machine learning*, pages 1168–1175, 2008.

[Zeiler, 2012] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[Zeng et al., 2018] Jiabei Zeng, Shiguang Shan, and Xilin Chen. Facial expression recognition with inconsistently annotated datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[Zheng et al., 2021] Wenzhao Zheng, Borui Zhang, Jiwen Lu, and Jie Zhou. Deep relational metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12065–12074, 2021.

[Zhou et al., 2017] Yuqian Zhou, Jimin Pi, and Bertram E Shi. Pose-independent facial action unit intensity regression based on multi-task deep transfer learning. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 872–877. IEEE, 2017.

[Zhou, 2018] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

[Zhu and Goldberg, 2009] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[Zou et al., 2019] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.