

https://dspace.org/

A visual complexity learning algorithm for modelling human performance in visual cognitive tests

Babshet, Kanaka

2019

https://hdl.handle.net/10539/28922

Babshet, Kanaka Sudhir (2019) A visual complexity learning algorithm for modelling human performance in visual cognitive tests, University of the Witwatersrand, Johannesburg, <http://hdl.handle.net/10539/28922>

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

University of the Witwatersrand
School of Electrical and Information Engineering

Masters Dissertation

A Visual Complexity Learning Algorithm for Modelling Human Performance in Visual Cognitive Tests

Author:

Kanaka Babshet

678851

Supervisor:

Prof. Vered Aharonson

October 2019



Declaration

I, Kanaka Babshet, declare that this dissertation titled, "A Visual Complexity Learning Algorithm for Modelling Human Performance in Visual Cognitive Tests", and the work presented in it are my own.

I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this dissertation has previously been submitted for a degree, or any other qualification at this University or any other institution, has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.

Signed: _____

Date: 14 October 2019

UNIVERSITY OF THE WITWATERSRAND

Abstract

Engineering and the Built Environment
School of Electrical and Information Engineering

Master of Science

A Visual Complexity Learning Algorithm for Modelling Human Performance in Visual Cognitive Tests

by Kanaka Babshet
678851

Supervisor: Prof. Vered Aharonson

Visual complexity has been extensively studied in the mathematical, **computational** sciences. Concurrently, psychological studies have attempted to define visual complexity as perceived by **humans**. The problem lies in that the computational and psychological studies are always explored **separately**, and thus their definitions of visual complexity are disjointed. This is evident when attempting to capture human-perceived complexity through computer vision.

This research attempts to tackle this problem in the context of cognitive assessments. This context introduces a practical application to the general question of computer, and human perception of complexity: Computerized cognitive assessments regularly employ visual stimuli, and present tasks that test a subject's primal cognitive functions. The difficulty of these tasks is not objectively quantified, which reduces the efficiency of the tests' administration, and the accuracy of the results' interpretation. This study developed and examined an algorithm that could computationally predict a visual task's human-perceived complexity.

The algorithm used a database of visual tasks and subjects' performance in terms of response times. Human subjective evaluation of tasks' complexity were captured for a subset of these tasks. Two types of feature sets were extracted from the visual stimuli presented in the tasks: object-specific, and whole image features. Several classifiers were implemented, using the features and the subjects' perceived visual complexity labels. The best algorithm configuration yielded a 58 % prediction, for a three-class complexity scale.

An analysis of the performance of the algorithm, and the relative visual features' importance values, provided insights which could help bridge the gap between mathematical complexity, and human perceived complexity.

Acknowledgements

The author would like to thank Professor Vered Aharonson from the School of Electrical and Information Engineering at the University of Witwatersrand for her continuous guidance, input, and support as the supervisor of this research.

Contents

Declaration i

Abstract ii

Acknowledgements iv

Contents v

List of Figures vii

List of Tables ix

1 Introduction 1

2 Visual Tasks in Cognitive Studies 4
- 2.1 Cognitive Assessments . . . 4
- 2.2 Computerized Visual Neurocognitive Assessments . . . 6
- 2.3 Visual Tasks Using Binary Images . . . 7
- 2.4 Assessing Visual Task Complexity in Cognitive Testing . . . 8

3 Studies on Visual Perception and Complexity 10
- 3.1 Theoretical Concepts of Visual Perception and Complexity . . . 10
  - 3.1.1 Witkin: Field-Dependence and Independence . . . 11
  - 3.1.2 Attneave: Aspects of Visual Perception . . . 11
- 3.2 Mathematical Concepts of Visual Perception and Complexity . . . 12
  - 3.2.1 Gabor Filters . . . 12
  - 3.2.2 Fractal Dimensions . . . 14
- 3.3 Binary Image Processing in Machine Vision . . . 15
  - 3.3.1 Image Segmentation . . . 16
  - 3.3.2 Object Detection: Size . . . 17
  - 3.3.3 Object Detection: Position/Location . . . 17
- 3.4 Assessing Complexity Between Multiple Images . . . 18
  - 3.4.1 Modelling Image Complexity by Independent Component Analysis, with Application to Content-Based Image Retrieval . . . 18
  - 3.4.2 A Co-Saliency Model of Image Pairs . . . 19
  - 3.4.3 Direct Image Comparison . . . 21
- 3.5 Previous Research on Binary Images . . . 22

3.5.1 Individual Image Complexity Determination . . . 22
3.5.2 Aharonson's Compression by Tracing . . . 23
3.6 Computer Vision and Machine Learning . . . 23
3.7 Summary . . . 26

4 Research Specifications 28
4.1 Research Question . . . 28
4.2 Assumptions . . . 30
4.3 Constraints . . . 31

5 Visual Complexity Learning Algorithm 32
5.1 Dataset . . . 32
5.2 Algorithm Development . . . 36
5.2.1 Human Visual Complexity Level Representation . . . 37
5.2.1.1 Linear Response Time Representation . . . 37
5.2.1.2 Logarithmic Response Time Representation . . . 39
5.2.1.3 Human-Provided Complexity Levels . . . 42
5.2.2 Feature Extraction . . . 43
5.2.2.1 Object Type Definitions . . . 43
5.2.2.2 Feature Details . . . 46
5.2.2.3 Feature Summary . . . 59
5.2.3 Random Forest Learner and Classifier . . . 60
5.3 Experiments . . . 62
5.4 Summary . . . 64

6 Results 65
6.1 Response Time Segmentation Experiments . . . 68
6.2 Prediction Accuracy for Logarithmically-Segmented Response Times Labels 70
6.3 Prediction Accuracy for Human Perception Labels . . . 72
6.4 Feature Importance . . . 72
6.4.1 Object-Specific Feature Importance . . . 75
6.4.2 Whole Image Feature Importance . . . 76
6.5 Summary . . . 77

7 Discussion and Conclusion 78
7.1 Response Time Labels for Human-Perceived Visual Complexity . . . 78
7.2 Classifier Choice . . . 78
7.3 Critical Analysis of the Response Times as Labels . . . 79
7.4 Feature Importance Analysis . . . 81
7.5 More Future Considerations . . . 83
7.6 Conclusion . . . 84

List of Figures

2.1 The Mini Mental State Examination 5
2.2 Other cognitive assessments comparable to the MMSE 5

3.1 The Gabor filter bank believed to be applied by the initial stage of visual processing in the brain (V_1) [31] 13
3.2 Original image 17
3.3 Centroid of the object in the image relative to the axis drawn in 17
3.4 Relationship between the number of trees in a random forest, and the percentage prediction error 25
3.5 Random forest classifier process flow 25

5.1 An example of a visual recognition task presented to a subject in the dataset. 33
5.2 An example of a visual recall task presented to a subject in the dataset. . 33
5.3 Method process flow 36
5.4 General linear segmentation of the response times in the dataset 38
5.5 The distribution of response times segmented linearly into 3 distinct levels slightly apart from each other 39
5.6 General logarithmic segmentation of the response times in the dataset starting from the lowest response time to the largest response time 40
5.7 **Inward** logarithmic segmentation of the response times in the dataset starting from the mode and moving outwards on either side 40
5.8 The distribution of response times segmented logarithmically into 3 distinct levels slightly apart from each other 41
5.9 Two adjacent paths (outlined in red) found in an image 44
5.10 One diagonal path (outlined in green) identified in the image 44
5.11 One single black block (outlined in blue) found in the image 45
5.12 One white adjacent path and one white single block found in the image . 45
5.13 47
5.14 49
5.15 49
5.16 50
5.17 51
5.18 51
5.19 52
5.20 The object distances between the black objects drawn for an image pair. 53
5.21 Illustration of the bit-by-bit comparison 54
5.22 The different image symmetry possibilities 55
5.23 Example of an image that is almost symmetrical. 55

5.24 The Gabor filter bank implemented in this research 56
5.25 Filtered results after Gabor filter application on example image 57

6.1 Frequency plot of the response times segmented by **age** 66
6.2 Frequency plot of the response times segmented by **gender** 66
6.3 Frequency plot of the response times segmented by **computer skill** . . . 66
6.4 The distribution of the labels calculated from linear segmentation of reaction times into 8 labels (top graph) and the distribution of the predicted complexity classes (bottom graph) 69
6.5 The distribution of the labels calculated from linear segmentation of reaction times into 3 labels (top graph) and the distribution of the predicted complexity classes (bottom graph) 69
6.6 The distribution of the labels calculated from the inward log segmentation of reaction times into 3 labels (top graph) and the distribution of the predicted complexity classes (bottom graph) 70
6.7 Importance of Relative-Features 73

List of Tables

5.1 Data extract of a recognition test battery with ten tasks provided to a single subject 34
5.2 Data extract of the subject demographics provided in the table 34
5.3 The number of classes generated from varying n in the linear segmentation technique from figure 5.4 38
5.4 Number of objects for the image pair in figure 5.13 47
5.5 Object path lengths for the image pair in figure 5.13 48
5.6 List of the experiments attempted with the algorithm 63

6.1 Results of the main experiments 67
6.2 Examples from the cognitive recognition tasks database which were correctly predicted by the algorithm as complexity levels 1, 2 and 3. 74

CHAPTER 1

Introduction

Visual perception is the ability to process and interpret the surrounding visual environment. This primal activity, common to humans and animals, begins with the eyes sensing an image and transmitting it via neural paths to the brain. The subsequent processing culminates in a decision or action based on the properties of this image. This processing, however, remains a mystery, and no model has yet been accepted to describe this translation or encoding process.

Two of the most vital cognitive actions that employ visual perception are recognition and recall. Recognition refers to the ability to compare and differentiate between visual information elements, and recall is the ability to retrieve previously perceived information. On a daily basis, humans are faced with situations where they are required to recognise and distinguish between several objects when making an observation or a choice of action. They also make decisions by recalling past observations and experiences. Individuals are shaped by these preceding experiences and the daily choices they make.

Neurocognitive, psychological and geriatric tests typically employ simple recognition and recall tasks to assess cognitive performance [3]. Visual stimuli are presented to subjects, traditionally on paper, and performance in recognition and recall of these stimuli is assessed [2] [35].

In earlier paper-based testing, the complexity of these visual tasks, if considered at all, was determined by human testers. Besides being subjective and even biased, this process would be applied to a limited number of tasks, and therefore limited the number and variability of tasks that could be included in the test. The pace and fluency of the examinees' responses, which should be a measure of their performance, were interpreted

based on the tester's observations, which limited their accuracy. This procedure directly introduced discrepancies between subjective observations of different testers.

Computerized cognitive testing has become more prevalent in the last two decades and provides a capability of enhancing both the administration and the reliability of the tests. The computerized system allows a generation of a wider variety of tasks to prevent repetition in successive periodic testing. Subsequently, the interpretation of the subjects' performance can also be enhanced by the ability to capture their response latency and duration, as opposed to mere counting of correct answers.

Computerized tests offer a platform for dynamic adaptation to a user's cognitive capabilities – adjusting the complexity of the set of tasks presented based on their previous task response times. This ensures that the subject is presented with a task of an appropriate difficulty level for them, rather than something that is too easy or too difficult, which could cause frustration or boredom, and limit the usefulness of the assessment. An evaluation of performance which takes into account response times can also be refined by considering the complexity of the task as a weighting component in the test score.

The hindrance in this concept, however, is that the difficulty or complexity levels of the different tasks employed in existing tests are not quantitatively defined.

This necessitates a computerized complexity scale that can quantify the difficulty level of a given cognitive task. In order to create this scale, an algorithm to determine visual complexity is required. While there are many **mathematical** algorithms that compute the complexity of images, the relevance of these algorithmic measures to the **human** visual complexity perception is rarely assessed. Complexity is studied in both areas, however it is discussed separately. Therefore there is generally a dichotomy between the mathematically defined visual complexity, and the human perceived visual complexity studied in the fields of psychology and neurology. Consequently, the application of most mathematical complexity measures to cognitive testing are impractical due to the inability to bridge between the mathematical complexity result, and the human perceived visual complexity when taking the test. Therefore, this sort of an algorithm must be tested according to extensive human cognitive testing.

The research aims to find a computerized visual complexity scale, in the context of visual neurocognitive tests, based on mathematical tools and concepts, such as information theory and machine learning, as well as human visual perception. This scale could provide a computer vision paradigm, where an algorithm mimics human visual perception.

This dissertation begins with background on cognitive visual tasks, with focus on computerized tests and their visual stimuli. A description and short analysis on theories of image complexity computation is then presented; aspects of which were applied to the research goal. The chapters thereafter introduce the research question, and the assumptions and constraints involved. This is followed by the methods which were employed to solve the research question. Finally, the results are presented with a concluding discussion and interpretation of the results.

CHAPTER 2

Visual Tasks in Cognitive Studies

This chapter describes visual cognitive testing employed in the psychological and neurological disciplines, and narrows down to the digital versions of these tests, which were considered for this research.

2.1 Cognitive Assessments

Many tests have been developed to assess cognitive decline in the fields of psychology, neurology and geriatrics. The tests aim to capture the core cognitive domains: memory, perception, reasoning, coordination and attention [13]. The assessments vary from the quick and coarse Mini-Mental State Examination (MMSE), to longer, half day testing procedures conducted by professionals in memory clinics. Some of these tests are still paper-based, while some have been computerised.

The MMSE is a quick and simple screening tool that is most commonly used in clinical trials and in general practice to detect cognitive decline [8]. It is a 30-point questionnaire where 0 indicates severe cognitive impairment, and 30 indicates no cognitive impairment. This examination is presented in figure 2.1 [18].

The MMSE is a coarse measure that lacks sensitivity to mild cognitive impairment. Additionally, it can be seen that this examination is highly verbal, and lacks visuospatial measures with a very small and crude visual aspect. It is therefore insensitive to impairments in visual perception [29].

Instructions: Ask the questions in the order listed.
Score one point for each correct response within each question or activity.

Maximum Score	Patient's Score	Questions
5		"What is the year? Season? Date? Day of the week? Month?"
5		"Where are we now: State? County? Town/city? Hospital? Floor?"
3		The examiner names three unrelated objects clearly and slowly, then asks the patient to name all three of them. The patient's response is used for scoring. The examiner repeats them until patient learns all of them, if possible. Number of trials: _____
5		"I would like you to count backward from 100 by sevens." (93, 86, 79, 72, 65, ...) Stop after five answers. Alternative: "Spell WORLD backwards." (D-L-R-O-W)
3		"Earlier I told you the names of three things. Can you tell me what those were?"
2		Show the patient two simple objects, such as a wristwatch and a pencil, and ask the patient to name them.
1		"Repeat the phrase: 'No ifs, ands, or buts.'"
3		"Take the paper in your right hand, fold it in half, and put it on the floor." (The examiner gives the patient a piece of blank paper.)
1		"Please read this and do what it says." (Written instruction is "Close your eyes.")
1		"Make up and write a sentence about anything." (This sentence must contain a noun and a verb.)
1		"Please copy this picture." (The examiner gives the patient a blank piece of paper and asks him/her to draw the symbol below. All 10 angles must be present and two must intersect.) 
30		**TOTAL**

FIGURE 2.1: The Mini Mental State Examination

Figure 2.2 lists assessments that are also employed to detect cognitive impairment [8]. These have been analysed and compared in previous studies to assess the feasibility of the MMSE and MMSE-2 assessment. The MMSE-2, specifically, is an enhanced, more thorough version of the MMSE which was designed to improve MMSE performance.

Order	List of neuropsychological assessments
1	MMSE-2 (red form or blue form)
2	SVLT-immediate recall
3	RCFT-copy
4	Digit span-forward
5	Digit span-backward
6	Stroop Color-Word test (word reading)
7	Stroop Color-Word test (color naming)
8	SVLT-delayed recall
9	SVLT-recognition
10	SWF-animal
11	SWF-supermarket items
12	PWF-three Korean alphabets
13	K-BNT
14	MMSE

Abbreviations: MMSE-2, Mini-Mental State Examination-2; SVLT, Seoul Verbal Learning Test; RCFT, Rey Complex Figure Test; SWF, Semantic Word Fluency; PWF, Phonemic Word Fluency; K-BNT, Korean version of Boston Naming Test; MMSE, Korean version of the Mini-Mental State Examination.

doi:10.1371/journal.pone.0163792.t001

FIGURE 2.2: Other cognitive assessments comparable to the MMSE

While these alternative assessments might have certain advantages over the MMSE, they are also not focused on visual perception measurements. These assessments are not applied further as the focus of this research is on the **visual stimuli** presented during cognitive assessments.

The assessments used to detect cognitive decline are predominately paper-based. However, over time **computerised** tests have been investigated, designed and implemented as a more feasible solution for the same. Computerised testing can offer a calmer environment for a subject with minimal human intervention. Above all, it allows for a faster, more consistent and objective testing procedure in comparison to the more traditional paper-based assessments.

2.2 Computerized Visual Neurocognitive Assessments

Different computerized tests employ different stimuli, and score subject performance in unique ways. Several computerised tests combine aspects of other traditional paper-based assessments on a digital platform, and have been validated against these paper-based tests [14]. While conducted differently, they still aim to capture the core cognitive domains.

Examples of these tests are the General Cognitive Assessment Battery (CAB) from CogniFit and the Cognitive Function Test (CFT) by Food for the Brain, both of which last about forty minutes [13] [19]. There are others, such as the Cognivue test battery, that have shorter, ten-minute tests. All of these assessments track correctness as well as response times.

In the CFT, there are six different tests that require recognising and classifying well-known pictures (such as methods of transport and household objects), matching basic visual patterns in the form of geometric shapes, and distinguishing between different line drawings. It further entails matching combinations of alphabetical letters, and remembering objects and their positions in a delayed recall test [19]. The Cognifit tests, however, are interactive with moving colourful shapes, word associations and musical tones. This introduces a gamification aspect to the testing. The tests are targeted to home usage [13].

While these assessments are different, they similarly aim to capture recognition and recall skills, with comparable underlying techniques and processes between the variety of computerised tests.

The tasks in the assessments present different types of stimuli, such as visual comparisons, word associations, and musical assessments. Therefore, while these tests employ more visual stimuli than those detailed previously, they are still mixed with other stimuli types. Consequently, isolating only the visual responses for study is a challenge. Another limitation to these tests is that, being commercial products, there is no test result data that can be investigated and applied for this research.

Studying complexity in visual tasks requires tests that focus on visual perception, and have available test results on which investigations can be conducted.

2.3 Visual Tasks Using Binary Images

The NexSig computerized cognitive testing battery, developed in 2004, employs visual recall and recognition tests using small binary (black and white) images. The response correctness and response times of the hundreds of subjects to the visual tasks presented, were collected in several studies using this battery, and are available for research.

Nexsig tests were used in studies of dementia, Alzheimer's disease, and other neurological, as well as psychiatric disorders. In all these studies, controls' data was collected. An interesting study explored Cognifit cognitive training, exercising the brain to maintain or improve cognitive abilities. It investigated whether using cognitive training could provide greater benefits than those obtained by playing conventional computer games, using NexSig tests as a cognitive assessment tool [32].

The data collected in this, and several large scale studies of both, controls and cognitively declined people, contain a detailed description of the images used in the visual tasks, and subjects' corresponding response times. This data is therefore potentially valuable for a study bridging cognitive sciences, and computational visual complexity.

The images presented in the tests are all simple, black and white, four-by-four square images. They can easily be described by sixteen-bit coding, and are therefore referred to as "binary images". The justification behind the choice of the images is that the lack of colour, and the small size of the images, would allow for a more objective visual analysis of the images by the subject. The disadvantage with presenting elaborate and colourful visual stimuli, is that certain objects in the images, or the images themselves, are more likely to visually trigger past memories or associations for a subject. These triggers could distract their focus off the required objective comparison or recollection of the image(s). This could in turn skew the capturing of the neurocognitive decline extent in the subject. The presented images in NexSig's tests are therefore enhanced images that are simple, yet conceptually similar to those in other, more common cognitive assessments.

2.4 Assessing Visual Task Complexity in Cognitive Testing

With all these cognitive tests, the complexity of the different tasks has not yet been objectively defined. There are several advantages to being able to determine task complexity: If the difficulty of a task was known prior to presenting it, then the corresponding response time could be weighted and better evaluated. This would be a finer way of assessing the level of cognitive impairment in the subject. For example, it could be said that a subject that takes very long to complete a simple task is more cognitively impaired than another subject that takes a short time to complete a relatively difficult task. Additionally, the task presentation order could dynamically adapt to the subject's responses: if the subject is struggling with difficult tasks, then the system can present a simpler task, and vice versa. This necessitates a computerized visual complexity defining solution that is cognisant of visual perception.

Before proceeding further, it is important to note that any analysis conducted on these tests will require investigations into *relative* complexity. This is because the tests involve the presentation of multiple images in each test instance. This relative complexity measurement should therefore computerise the image **comparison** in a way that indicates how complex a presented task is.

The visual testing focus, the simple nature of the presented images, and the availability of the results make NexSig's test battery a desirable one to create a computerized visual complexity scale against for this research. This should be a scale that incorporates both, mathematical, as well as human perceived, visual complexity.

The following chapter will detail visual perception concepts and image analysis techniques, explored with the aim of capturing visual perception through computer vision.

CHAPTER 3

Studies on Visual Perception and Complexity

This chapter presents theoretical investigations on visual perception, and computationally capturing complexity. The concepts of these were useful in providing a guideline on how a visual complexity defining algorithm for cognitive assessments had to be approached. Since the algorithm requires comparison between human complexity perception, and computational complexity classification, the model had to consider a combination of both, visual perception and information theories.

The chapter begins with theories that have previously investigated and attempted to describe visual perception. These theories motivated the subsequent choice of the mathematical techniques considered, as their implementation should mimic the visual perception process. Thereafter, binary images are analysed to see how different objects in a binary image would be able to provide image features. This is followed by image comparison techniques that are described to understand how multiple images could be compared through computer vision to capture *relative* features that define relative visual complexity. Once the technical concepts are explained, related research previously conducted is presented as preliminary work. Finally, computer vision and machine learning techniques have been analysed to provide guidance on how the preliminary work could be extended for this research.

3.1 Theoretical Concepts of Visual Perception and Complexity

These theories were explored in attempt to understand and define visual features that are vital in defining what humans perceive when observing visual stimuli, and can be captured on a computer.

3.1.1 Witkin: Field-Dependence and Independence

Herman Witkin, a psychologist who specialised in cognitive psychology, proposed a field dependence concept. He postulates that individuals can be classified as 'field dependent' or 'field independent' in their cognitive styles [9]. Field dependent individuals lack attention to detail. They initially assess their visual field as a whole, and then loosely partition the information into groups. Field independent individuals, however, immediately separate and organise the visual information into clear-cut groupings [9]. Several researchers suggest that this classification correlates to elementary cognitive processes.

Witkin's theories indicate that when attempting to interpret human visual perception, the different cognitive processes should be catered for. This can be done by capturing certain visual characteristics/features at an intricate, detailed level, and other features at a more holistic, higher level.

3.1.2 Attneave: Aspects of Visual Perception

Fred Attneave, a psychologist known for his theories in visual perception, states that there is no straight forward method by which to determine which physical measurements have greatest psychological pertinence [6]. He considers visual perception an information handling process where much of the visual information observed by a human is likely to be redundant [5] [7].

The computational consequence of this hypothesis is that any visual invariance constitutes a source of redundancy because the subject automatically begins to deduce the rest of the image, thus reducing its perceived complexity. This is with regards to the concentration of information at certain points, image symmetry, adjoining pixels of the same colours, shapes, lines, or even angles. The extent to which a human can extrapolate this information is still unknown [5] [7]. However, the occurrence of these characteristics can be captured as features that provide useful measures of human perceived complexity.

Attneave and Witkin's visual perception theories demonstrate a basic foundation on how images are observed, and how the observation process is likely to vary between humans.

These principle concepts are considered when developing the machine vision system in this research.

The algorithm must be able to mimic the visual perception process followed by humans. Therefore, through computer vision, it is important to holistically assess the images as field dependent individuals would, and also clearly segment and evaluate the images in parts as field independent individuals would. In this process, the model should capture the information held in the image, such as object-specific information. This would allow for the measurement of the level of redundancy and consequent visual extrapolation that would be conducted by a human during the visual observation. Implementing visual perception through computer vision is vital to computationally measure how simple or difficult a human is likely to find a selected test instance.

3.2 Mathematical Concepts of Visual Perception and Complexity

While there are several mathematical definitions of image complexity, few are likely to align with the human perception of visual complexity. This section describes two mathematical techniques that, based on their foundational concepts, are likely to correlate to human perception.

3.2.1 Gabor Filters

A Gabor filter is a type of bandpass filter commonly applied in image processing. It is made up by a sinusoidal plane at a certain frequency and orientation, modulated by a Gaussian envelope [22]. The filter has a real and an imaginary part (representing orthogonal directions) to collectively form a complex number [21].

The complex two-dimensional Gabor filter function is given by equation 3.1 [21]:

$$g(x,y) = e^{-\frac{(x'^2+\gamma^2 y'^2)}{2\sigma^2}} e^{i(2\pi\frac{x'}{\lambda}+\psi)} \tag{3.1}$$

Where:

$x' = xcos\theta + ysin\theta$

$y' = -xsin\theta + ycos\theta$

θ = Orientation angle

γ = Spatial aspect ratio

σ = Gaussian standard deviation

λ = Sinusoidal wavelength = $\frac{linear\ speed}{frequency}$

Many claim that these filters are synonymous with perception in the human visual system. This is because they are known to share similarity with the receptive field of simple cells in the visual cortex of mammalian brains [15] [27]. It is said that the initial stage of visual processing in the brain (V_1) applies a 'filter bank' of Gabor filters [31]. A filter bank is a collection of filters at different orientations and frequencies.

It is hypothesised that the following filter bank exists in V_1 [31]:

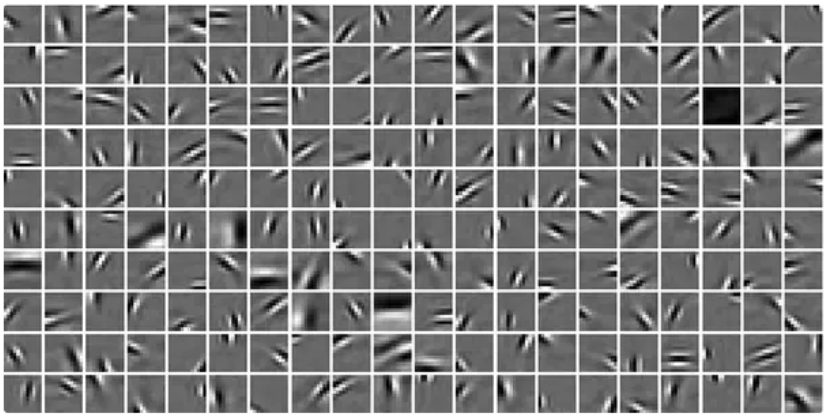


FIGURE 3.1: The Gabor filter bank believed to be applied by the initial stage of visual processing in the brain (V_1) [31]

When a filter bank is applied to a single input signal, the output is an array of single-frequency-modulated versions (at different orientations) of the input signal. Therefore, when this Gabor filter bank is applied to an image, the image is convoluted with real and imaginary vectors at different frequencies and directions to result in a feature array/vector [22].

Given the similarity to the human visual system, two-dimensional Gabor filters are frequently used for feature extraction in images with automated edge detection and texture analysis. Gabor feature based methods are also among the top performers in face detection and recognition, iris recognition and fingerprint matching [22].

This suggests that the computer application of the Gabor filter bank to the binary images in the NexSig cognitive test battery could provide information into how these images are visually processed.

3.2.2 Fractal Dimensions

A fractal pattern is an image that displays 'self-similarity'. An image displays self-similarity if it shares the same (or similar) statistical characteristics as the whole figure at different scales of magnification. I.e. each part of the image is at least approximately a smaller copy of the whole image. A very popular fractal curve is the Koch snowflake - a pattern repeatedly built off the outer two sides of several equilateral triangles [17]. The fractal dimension (D_f) is therefore the measure of the fractal properties that the figure/image displays by indicating how scaling changes an image.

The box counting method is a technique widely used to determine the fractal dimension of images. The process looks at the number of boxes, N, needed to cover the non-zero elements of an image as the size of the boxes, r, is varied. The fractal dimension is then calculated as the **slope** of the line of best fit through the log (N) (y axis) against log(r) (x axis) plot [24]. The slope/dimension tends to vary between 1 and 2. $D_f = 1$ is likely to represent a straight line, and $D_f = 2$ is likely to represent a line so out of shape that it completely fills up a two-dimensional plane. Therefore the greater/steeper the slope, the greater the fractal properties. This box-counting dimension, D_f, is also known as

the Minkowski-Bouligand dimension, or Kolmogorov capacity, or Kolmogorov dimension, and is given in equation 3.2 [24].

$$\begin{aligned} D_f &= gradient(\frac{log(N_r)}{\frac{1}{log(r)}}) \\ D_f &= -\frac{dlog(N_r)}{dlog(r)} \end{aligned} \tag{3.2}$$

An important step in the box counting technique is choosing the relevant box sizes, and increments of change in the box sizes. Naturally, the box size must never exceed the image's total size. Box sizes beyond about 50 % of the image size are likely to introduce errors. A maximum box size of 25 % of the shorter side of the image is generally optimal for standard box counting scans [20]. If an image is noisy, the largest box size should be smaller [20].

The fractal dimension signifies visual complexity by measuring the level of visual detail in a fractal at different levels of magnification [17]. A higher fractal dimension indicates more detail at smaller scales, and thus signifies greater visual complexity. Since the human eye is likely to pick up visual self-similarity as well, this measure should be synonymous, at least to a certain extent, with visual perception.

As the implementation in this study is aimed at computationally mimicking visual perception, mathematical concepts that are suggested to be synonymous with visual perception would be well suited for this endeavour. Gabor filters and fractal dimension calculations can therefore be implemented and tested against the provided cognitive tests' results to prove their relevance in human perception.

3.3 Binary Image Processing in Machine Vision

Binary images are images which have only two possible values for each pixel – black or white. Initially, machine vision largely focused on binary images (as opposed to colourful images) due to memory and computing power limitations. In terms of visual perception, people have less difficulty in distinguishing line drawings, silhouettes and other images

formed when there are only two colours being visually assessed. Binary images are therefore frequently used in several human visual applications, which makes binary vision systems continually useful [11].

The following methods are most commonly used for the analysis of binary images by humans, and consequently in machine vision systems [11]. More specifically, they are likely to be applied by field independent individuals during their visual observations as they separate and organise detail within images.

3.3.1 Image Segmentation

Humans tend to naturally identify sub-images of some sort within an image. This process is known as segmentation and is often the first step during image analysis [11]. This could be difficult to execute on a computer as the segmentation process may vary from person to person – different people focus on different areas of an image. The first step to implementing image segmentation objectively on a computer could therefore involve simply partitioning the image into smaller, equally-sized sets of pixels to be assessed independently.

Segmentation is also done on the varying colour distribution within an image. Since binary images have only two distinct variances of colour, there is a predefined level of segmentation (between the black and white pixels) within the image [11]. This is a more common and consistent level of segmentation naturally conducted by individuals, and can similarly be implemented computationally.

Another possible level of segmentation that is likely to occur is the detection of distinct objects in an image. Identifying significant objects in an image is one of the key components in image analysis [11]. Humans are likely to notice the size and location of these objects as distinctive features of the image. Note that the object calculations on the example images in 3.3.2 and 3.3.3 assume that a single cluster of black pixels in the image can be classified as an object. A formal object classification algorithm is yet to be defined.

3.3.2 Object Detection: Size

The size of an object can be given by its area [11]. In the case of binary images with large, distinct pixels, the size measure of the object can be obtained by simply summing the number of black pixels that it is made up of. Therefore, the size/area of the L-shaped object in the example image in figure 3.2, is 4.

3.3.3 Object Detection: Position/Location

The position of an object within the image can be defined using its centroid – the geometric centre of the object [11]. This theory could be applied to the objects within a binary image.

To calculate the centroid of any composite object, it must first be broken up into a series of smaller, basic shapes that have predefined local centroid locations. The formula to calculate the centroid $[\bar{x}, \bar{y}]$ of object B is given in equation 3.3 where B is made up of n basic shapes with area A. Therefore, x_i is the x coordinate, and y_i is the y coordinate of the local centroid of shape/area A_i.

$$B[\bar{x}, \bar{y}] = [\frac{\sum_{i=1}^{n} A_i.x_i}{\sum_{i=1}^{n} A_i}, \frac{\sum_{i=1}^{n} A_i.y_i}{\sum_{i=1}^{n} A_i}] \tag{3.3}$$

If the x and y axis are defined as shown in figure 3.3, the centroid for the single L-shaped object in the example image is [2.75,2.25] and is drawn in on the same figure.

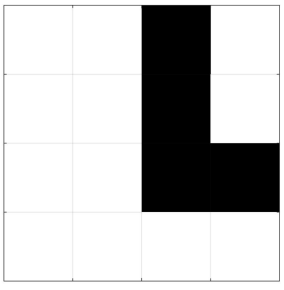


FIGURE 3.2: Original image

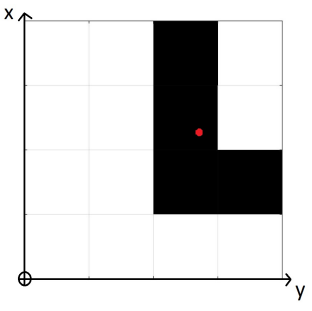


FIGURE 3.3: Centroid of the object in the image relative to the axis drawn in

Binary image processing concepts are directly applicable to the images from the chosen visual test battery where the objects in an image convey information regarding that image. Therefore, once an object is defined in an image, its size, location, shape, and even orientation can be captured as visual features. These basic characteristics can also convey other information about the image. For example, calculating the average distance between the locations of multiple objects in the image provide an indication of the object spacing within the image, or even comparing the relative positions of detected objects within two different images could be used as a measure of similarity/difference between the two images. Similarly, the other binary image information detailed in this section can be compared between multiple images to deduce comparative measures.

Further to this, additional comparative measures need to be investigated to understand how images are compared during visual perception. The next section deals with comparing images to understand relative complexity.

3.4 Assessing Complexity Between Multiple Images

The techniques explored in this section address the comparison of multiple images to gather information regarding the relative complexity of the images. These can therefore be used to provide insight into comparing multiple images through computer vision.

3.4.1 Modelling Image Complexity by Independent Component Analysis, with Application to Content-Based Image Retrieval

Authors Jukka Perkiö and Aapo Hyvärinen claim that the difficulty in defining the degree of similarity between two images stems from the fact that the similarity is largely dependent on the context of the images. Perkiö suggests that similarity between images is defined at two levels: the semantics and the syntax, where the syntax refers to the structure of the image, and the semantics are dependent on the context of the images. Simply put, different aspects or features of an image tend to be focused on in different

contexts. Consequently, the less dependent the similarity is on the context of the images, the more general the similarity measure can be with a simpler semantic/ more logical interpretation, and greater focus on the syntax of the images [33].

Perkiö and Hyvärinen's paper presents a method based on a model known as independent component analysis (ICA) that incorporates data sparsity to approximate a pair-wise similarity measure. The method estimates the entropy of images separately and combined. The similarity is then calculated from the normalised difference between the single image complexity, and the pair-wise complexity. The measured complexity is low if the components are largely dispersed [33].

The other two methods predominantly discussed in this paper, KL-divergences and NCDs, are estimated from RGB-intensity histograms, and will therefore not be studied further for binary image complexity definitions. ICA is comparable, but not similar, to other complexity measures such as normalized compression distance, and other information theoretic entropy-based methods [33].

ICA will not be directly implemented in this research, however, the process detailed in this study demonstrates how data sparsity can indicate pair-wise image similarity. Data sparsity is also an important measure in visual perception – Attneave postulated that the concentration of information at certain points in an image is one of the factors that can cause a human to extrapolate the rest of the image. The degree of extrapolation then affects an image's perceived complexity. Data sparsity could also subsequently be measured given the object information detailed in the previous subsection.

3.4.2 A Co-Saliency Model of Image Pairs

Saliency is the quality of being especially noticeable. Thus, co-saliency is the common saliency between two or more images, which is influenced by the co-occurring of particular patterns and contrast factors between the images [37].

Saliency detection models are primarily inspired by the human visual attention processes, and can be segmented into two categories: eye fixation prediction models and salient

object detection models. The objective of the former is to predict fixation points when people freely look at a visual scene. The objective of the latter is to detect and segment the full extent of salient objects that strikingly attract visual attention. Based on the hypothesis that salient regions should be distinctively noticeable from their surroundings within an image, early saliency prediction techniques involved contrast detection. More recently, deep neural networks have been used to identify informative and salient regions between comparable images [37].

This co-saliency model combines the local prominent regions within a single image with the similarity measure between an image pair. They have built the model to follow the attention search process for an image pair, which can be obtained by a calculated weighted addition of the Single Image Saliency Map (SISM) and Multi-Image Saliency Map (MISM) [23].

The SISM is generated by combining three different types of saliency maps: Itti's model saliency, frequency-tuned saliency, and spectral residual saliency where the first one mimics the human visual search process. The SISM calculation is moulded based on the specific advantages of each of the three maps [23].

The MISM is then generated by assessing the inter-image correspondence through feature matching. This is based on the premise that if there is a similar object in both the images, it would garner more visual attention. The MISM process consists of four stages: pyramid decomposition, feature extraction, SimRank optimization, and multi-image saliency computation [23].

A weighted addition of the SISM and MISM is then calculated. This method provides an effective way of predicting human fixations between multiple images and isolate co-salient areas. The authors point out that humans focus on the most valuable information from the image pair [23].

This co-saliency model demonstrates the concept of how co-occurring patterns between multiple images can affect the relative image complexity. Once again, this ties into Attneave's hypothesis where co-occurring patterns could constitute as a level of visual redundancy. This redundancy could subsequently affect visual perceived complexity.

The equations explained herein are also not applied directly to the images in this research, but they rather demonstrate how co-saliency is otherwise measured, and guide the more informal choice of measure for co-saliency. Given the binary image processing detailed in the previous subsection, the information deduced from the objects in images can be compared to find co-occurring patterns which then constitutes for a measure of co-saliency.

3.4.3 Direct Image Comparison

While the previous two studies have detailed comparing respective aspects of images, another approach can be considered where two whole images are compared directly against one another. This comparison caters for field dependent individuals (as per Witkin's postulation) who assess their visual field as a whole, and only loosely evaluate sub-images and other finer details.

A study named *Measuring Classification Complexity of Image Databases: A Novel Approach* presents a calculation of the degree of similarity between whole images from two different classes [34].

Multiple images provided in a single neurocognitive test instance can be assumed to be part of a single class. If one therefore manipulates their proposed equation to define the similarity, it results in the summation of a simple bit-by-bit AND operation. In this scenario, each respective bit is compared between the two images. Only if both of the bits are black, is the result 1 for the comparison of those two bits. Therefore, the total measure of similarity between two binary images (of the same size) with n bits each, can range between 0 and n. Here 0 denotes a complete lack of similarity, and n denotes the highest level of similarity.

Since the summation in this calculation is triggered by black squares in the same position within their respective images, this calculation assumes that black blocks are more visually striking during human observations than the white ones.

3.5 Previous Research on Binary Images

The previous sections herein have now detailed mathematical implementations that are in line with the principle visual perception theories. These can be applied in some way to this research. There are, however, other studies that have also similarly explored this research topic.

3.5.1 Individual Image Complexity Determination

A previously conducted study focused on defining the visual complexities of **single** sixteen-bit binary images. An algorithm was developed by applying Fred Attneave's primary theories on human visual perception, to classify each image in the sixteen-bit binary dataset with one of three visual complexity levels: "easy", "medium" or "hard'. The solution was tested against complexities perceived by human judges for three hundred different images. The final algorithm yielded an accuracy of 69 % [7].

Realistically, however, if a subject is simply provided with a single image and asked to define its perceived complexity, it might be tough to do so without posing the question 'with respect to what?' Even human perceived complexity is relative. Therefore, it can be said that the extent of similarity between multiple images is a better indication of a task's complexity, than the individual complexities of those same images. For instance, should two images be vastly different visually, where one is very simple, and one is very complex, being able to distinguish between them becomes effortless.

Since cognitive tests involve a subject differentiation between presented images, the degree of discrimination required is what defines the complexity level of a task. This individual complexity measure will be a foundation on which a **relative** complexity measure will be built in this research.

3.5.2 Aharonson's Compression by Tracing

This study was also conducted to define the relative visual complexity between the sixteen-bit binary square images employed in recall and recognition tasks. It is built on the postulation that the extent to which an image can be compressed is indicative of its level of complexity.

Based on this, a compression technique was applied on a sixteen-bit binary vector, derived from the square images. A pre-processing procedure used six tracing modes and converted each image into six respective binary vectors. Each vector was compressed based on the extent of change between the black and white consecutive bits. This process thus disregarded the contiguous similar bits within the sequence [1]. Aharonson then used the range of compressed vectors to propose a pattern description length (PDL) algorithm that identifies the minimum number of bits required to describe the image [1].

A relative complexity measure was then calculated using the ratio between the individual image complexities obtained from the PDL. This equation ensured that the relative complexity decreases if the difference between the two image complexities increases, or if the complexity of the simpler image decreases. The algorithm proved to provide a useful complexity scale for customary small image recognition tasks [1].

These two preliminary studies have both applied computer vision through the computational assessment of the binary images. However, further analysis needs to be conducted on how the computer vision can be advanced through this research to make better use of the available dataset, create an enhanced visual complexity scale, and ideally gain further insight into visual perception.

3.6 Computer Vision and Machine Learning

Computer vision employs image processing techniques on images to produce quantitative and qualitative information about them such as size, colour, or number of objects within the images [25]. It is aimed at deducing visual information computationally in a similar fashion to the visual reasoning process conducted through human vision. However, as

visual perception remains a mystery and is yet to be completely understood, additional machine learning methodologies, such as neural networks, can be coupled with computer vision to attempt learning and mimicking visual reasoning [25].

One such study described a computer vision system to model human behaviours and interactions in a visual surveillance task. In this investigation the system was trained on sets of perceived actions and then tested on the ability to detect eventual anomalous behaviours or potentially dangerous situations [30].

A classification model that is commonly applied in image classification studies is the random forest (RF) classifier. A random forest trains on a combination of several decision trees that make up the said forest [26].

Decision trees learn to map provided data (otherwise known as "features") to expected targets. Thereafter, if the decision tree is required to predict a result given a set of sample data, it applies the feature information and structure it previously learnt. However, if this is done with only one tree, the prediction error is likely to be very high as there are too many factors to account for by only one tree. Random forests therefore eliminate this variance by combining several trees where each tree adds more knowledge to the measure [26]. A graph of the common relationship between the number of trees chosen in a random forest, and the percentage prediction error is shown in figure 3.4. The final number of trees chosen is usually around where the error percentage flat-lines so as to not make the solution too computationally expensive, while maintaining a minimal error.

An important characteristic of the random forest is that each decision tree in the forest considers a random subset of features when creating questions, and is only provided with access to a random set of the training data points [12]. The random forest uses a technique called bootstrap sampling which repeatedly selects a random sample of the training set and allocates it to a decision tree. This random sampling increases diversity in the forest and lowers the risk of overfitting a single set of training data. The final result/predicted class is then the average, or the mode (majority vote) of the predictions from the individual decision trees [12] [26]. The average is taken when the random forest

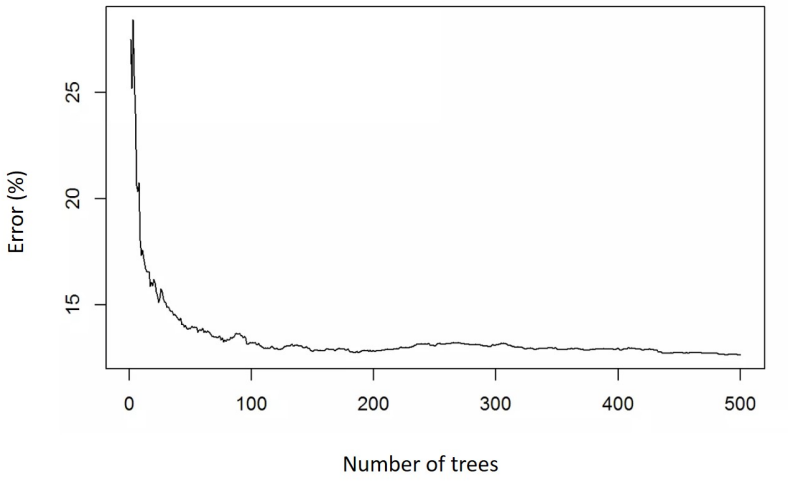


FIGURE 3.4: Relationship between the number of trees in a random forest, and the percentage prediction error

is used for regression, and the majority vote is taken when the random forest is used for classification. A high level random forest process is illustrated in figure 3.5 [26].

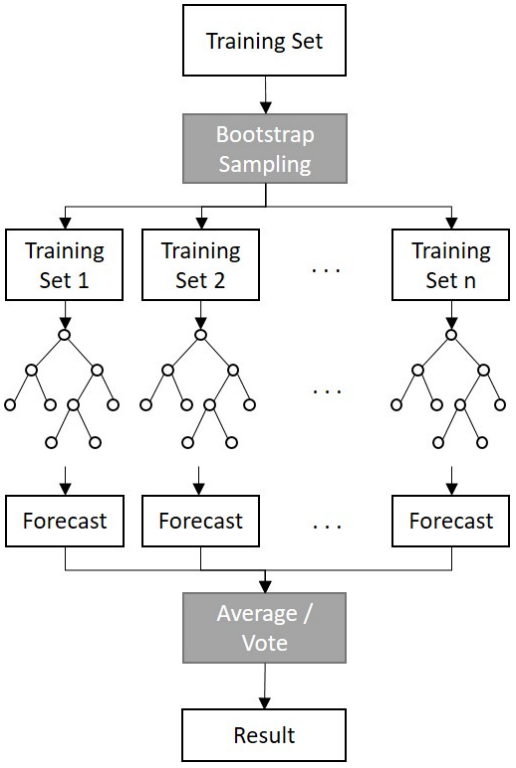


FIGURE 3.5: Random forest classifier process flow

The random forest is considered a robust model due to the minimal likelihood of overfitting the training data [12]. However, in some extreme cases this is still likely. While overfitting is checked and prevented through cross validation methods, an additional technique by which to test that a model is not overfitting the data is to obtain the out-of-bag (OOB) error. This calculation represents the prediction error on random samples from the **training** subset [28].

A significant advantage to the random forest is that it can provide feature information with the results. Since different subsets of features are used for different decision trees, the results provide better insight into which features contributed to the classifier, and hence to which lengths they had an impact on the final predication.

Other popular learning models that are employed for multi-class classification include artificial neural networks (ANN) and support vector machines (SVM). These are known to be strong learning techniques which can detect complex non-linear relationships between dependent and independent variables, and are less influenced by outliers [4]. However, these techniques are black boxes with several hidden layers between the input and the output. Consequently, the results from these techniques would not provide information on which features were used, and to which extent.

3.7 Summary

This chapter described several methods that are applicable to computationally modelling human visual perception, and that have guided, in some way, the choice of techniques implemented in this research: Attneave and Witkin's theories assist in understanding the theoretical concept of visual perception. The Gabor and fractal dimension concepts provide mathematical calculations that are likely to be synonymous with human visual perception. Binary image processing serve in identifying specific, key features observed by subjects when defining the visual complexity of these binary images. ICA, co-saliency and direct image comparison further propose solutions to define the *relative* similarity between multiple images.

The two last studies: Individual Image Complexity Determination and Aharonson's Compression by Tracing attempted to explore the correspondence of the computational complexity and human complexity perception, and have motivated this research as preliminary work. The computer vision and machine learning theories are further explored as an investigation into how these technologies have previously been used on images for a similar purpose, and can thus assist in extending the preliminary work.

To implement the computer vision system in this study, the visual perception theories guide which aspects of each mathematical, objective technique above is applied to extract notable image features. This is aimed at computationally mimicking the same process followed by a human during their visual observation. Features are extracted from the images with the techniques detailed in the binary image analysis. Object-specific and whole image comparisons are made between images using these extracted features. Co-saliency is measured by detecting the co-occurrence of different objects (as valuable focus points) within the images. Relative sparsity is measured on the objects within the images. Ideally, this process would computationally mimic visual perception by field **independent** individuals. The images can then be compared holistically as well, through the whole image comparisons, to computationally mimic visual perception by field **dependent** individuals.

Finally, the machine learning technique, random forests, is explored when building the computational predictor of visual perceived complexity by learning how the presence of the different features mentioned above affects the human perceived complexity.

CHAPTER 4

Research Specifications

4.1 Research Question

There is no reliable visual complexity scale for cognitive tests that employ visual stimuli. One of the reasons for this is the distinct disjuncture between the mathematical definitions of visual complexity, and psychological experiments on human perception of visual complexity. This study intends to bridge this gap by attempting to mathematically and computationally capture complexity in the context of human cognitive tests.

This research aims to answer the following question:

To what extent can a complexity measure, based on information theory and machine vision, model and explain human visual perception in the context of cognitive tests?

To answer this question, the following five objectives will be pursued:

1. Create a human task complexity scale based on a database of response times acquired in visual tasks. The scale requires a segmentation paradigm to map the response times of subjects into a discrete scale.

2. Extract image features that capture the visual characteristics of the images in the database tasks.

3. Develop a complexity classifier to find patterns that relate the image features in the cognitive tasks (objective 2) to the human task complexity perception scales (objective 1).

4. Assess and compare the prediction performance of the algorithm for the different human complexity perception scales.

5. Evaluate which features, or combination of features, are more relevant to the complexity prediction.

The first objective aims to quantify the human visual perception, in the context of cognitive visual tasks complexity. As visual cognitive tasks are presented to subjects, they need to perceive the images therein, and complete a task: recognition or recall. As detailed in the next section, this research assumes that the response time – the time taken to complete a task - is indicative of the difficulty of the task, which in turn is correlated to the complexity of the task. Given a suitable large database of computerized visual tasks, and their response times, a human complexity scale needs to be derived and tested.

The second objective then involves extracting features of the image sets in the tasks to capture, and specify visual characteristics that are relevant to the visual complexity of the tasks.

The third objective entails developing a machine learner to yield a computational complexity prediction, using the information theory and machine vision image features (derived in objective 2) as inputs, and using the human complexity scale (derived in objective 1) as labels.

The fourth objective evaluates the performance of the algorithm developed in objective 3. The results of this indicate how likely the algorithm is to correctly predict a human perceived visual complexity of a cognitive task. The applicability of the different labels schemes, that were derived based on human performance, is particularly examined in this stage.

Concurrently, the fifth objective deals with analysing feature importance using the information provided by the machine learning model. This will indicate which features contributed when defining the visual perceived complexity of the images.

The culmination of these five objectives aims to provide a methodology that, given visual cognitive tasks, and human perception performance, can match the two complexity scales - computational and human - and classify the complexity of the cognitive tasks.

4.2 Assumptions

The assumptions in the study were made based on the data available for it. This database - NexSig's Computerized Neuropsychological Tests - will be further detailed under the methods in the following chapter. The following assumptions are therefore made:

- The data is accurate, authentic, and reliable.
- A subject response times correlates to the perceived visual complexities of the task, and are thus a good measure for the complexity of the task. Studies in psychology have shown that reaction/response times relate to decision difficulty. When a human is presented with any sort of a task, it requires brain activity. The corresponding response time reflects the workload in the brain, and thus varies with the difficulty of the task at hand [36]. Intuitively, the more difficult a task is, the more time it takes to complete. Therefore, it was assumed that the subjects' response times relate to the complexity of the presented tasks, and could be used as a "ground truth" of the perceived complexity. This assumption was necessary as all testing was done against the response times, and the algorithm would be completely incorrect if the response times were not an indication of the perceived complexity of the task.
- The human brain is very complex, and human performance, even in very focused tasks such as the ones studied here, is affected by many processes, cognitive and other. Examples of this are emotional reaction and past experiences. It is therefore assumed that the bias or skewness caused by these factors is negligible compared to the cognitive function. This assumption can be justified by the nature of the visual stimuli, or images presented in the tasks concerned: The images are very small and abstract and were chosen after neuropsychological studies, subjective, but extensive, to mitigate this bias.

4.3 Constraints

The following constraints were identified in this algorithm design:

- As described in the assumptions section, many aspects in human visual perception, complexity included, are biased by noncognitive elements, such as emotion, culture, previous memories and more. For example, one individual might perceive a familiar object or shape within an image, whereas another does not see it. This could cause different individuals to perceive the same visual stimulus differently. Therefore, the analysis of the response times in the database may be skewed. Since these factors cannot be quantified, it was not possible to capture or account for them when analysing the data. While the possibility of this bias is significantly reduced with the lack of colour and small size of the presented images, it was still noted as a constraining factor.

- The dataset used in the study contains a large set of instances - visual tasks and response times. This dataset was collected in studies with clinical goals, and may not be optimal for the current research. The algorithm developed is therefore constrained by the range of the visual tasks in this dataset. In this way, if the dataset did not contain images of a certain complexity type, it would not be possible to train or test on them, thus limiting the capability of the algorithm's complexity labelling quality.

Given the problem specifications and the previously described related theories that could assist in solving this problem, the next chapter presents the method that was implemented in the research.

CHAPTER 5

Visual Complexity Learning Algorithm

This chapter presents the visual complexity learning algorithm that was developed to create a quantitative, computerized complexity measure. The algorithm was developed on Matlab, version 9.5, release R2018b. All code has been provided in the attached CD.

The chapter starts with a description and critical analysis of the dataset provided for this research. It then details the algorithm applied to this dataset to implement a complexity prediction.

5.1 Dataset

This computerized complexity measure made (secondary) use of the dataset from NexSig's computerized cognitive testing studies, as introduced in chapter 2. Approval to use the dataset was obtained, along with the ethics clearance (M180414) for re-use of this data. The clearance certificate is provided in Appendix B.

The primary advantage of NexSig's computerized cognitive testing studies was that the data was made available, and the simple binary images could be readily encoded and dealt with computationally. The test results could be used to deduce information regarding human visual perception in the context of visual neurocognitive tests. Consequently, this dataset could provide a benchmark against which to develop and test the methodologies that have been described in the related studies in chapter 3 to implement this complexity defining solution, and possibly better describe visual perception.

There were two visual tests employed in the studies that yielded the dataset: recognition and recall. Both tests used images of the same category as visual stimuli: small, sixteen-bit square binary images. In the recognition tasks (example shown in figure 5.1), three images were presented on the screen, and the subject was required to recognize which of the three images was different.

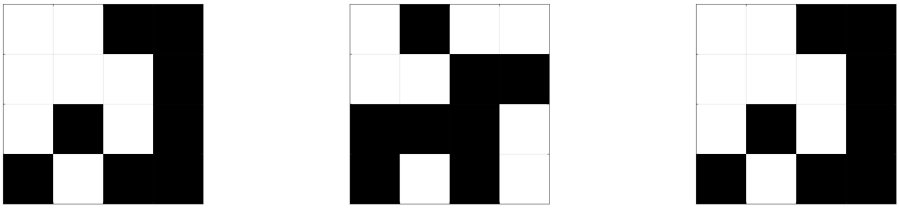


FIGURE 5.1: An example of a visual recognition task presented to a subject in the dataset.

In the recall tasks (example shown in figure 5.2), a single image was presented on the screen for a few seconds, and then erased. The subject was then presented with three images, and required to remember and select which one was previously displayed.

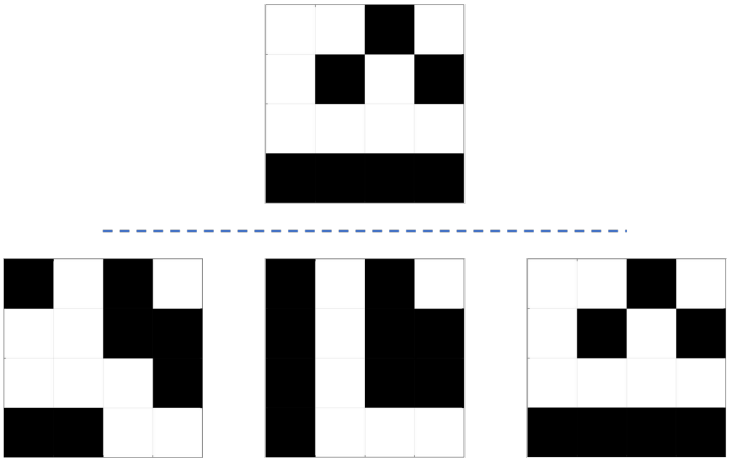


FIGURE 5.2: An example of a visual recall task presented to a subject in the dataset.

The recognition and recall test results of the subjects – correctness of answer and response times – were collected in the dataset. One table stored the recognition tasks' results, and the other stored the recall tasks' results. Each row therein represented a task instance. The first three fields in the row contained the coding of the images in the presented task, from left to right image as presented on the screen. The images were coded as sixteen-bit arrays translated from the four-by-four images starting from the top left square, going left

to right, and then downwards row-by-row. A white pixel in the image was represented with a 0 and a black pixel with a 1. For example, the first image in figure 5.1 would be coded as 0011000101011011.

After the three images, each row contained a field with what the correct (expected) answer for that task was (i.e. 1, 2 or 3), what the subject's actual answer was (1, 2 or 3), and what their corresponding task response time was in milliseconds. Results were cut off shortly after 7000 ms. A snippet of these results is shown in table 5.1.

TABLE 5.1: Data extract of a recognition test battery with ten tasks provided to a single subject

ID	Filename	Image1	Image2	Image3	Expected Response	Actual Response	Response Time
1	1106089771859.txt	1111010000110100	1111010000110100	1000101101110100	3	3	2890
2	1106089771859.txt	0011110000101110	1100000111101000	1100000111101000	1	1	2688
3	1106089771859.txt	0011000101011011	0100001111101010	0011000101011011	2	2	3594
4	1106089771859.txt	0001100110010111	0111101011001000	0111101011001000	1	1	2125
5	1106089771859.txt	1100011001010100	0110100110100110	0110100110100110	1	1	3109
6	1106089771859.txt	0101111000011010	0101111000011010	0101001110011001	3	3	4391
7	1106089771859.txt	1100010011100110	1100010011100110	1100010111000110	3	3	3640
8	1106089771859.txt	0100010011101110	0011010010111010	0100010011101110	2	2	2125
9	1106089771859.txt	0111110010001010	1000101100111010	1000101100111010	1	3	2063
10	1106089771859.txt	0011010010011101	0011010010011101	0000100111111100	3	3	1844

A separate table listing the demographics and MMSE results of the subjects that took part in the tests was also included in this dataset. This table mapped to the test results via a filename, thus maintaining subject anonymity. A snippet of this representation is shown in table 5.2.

TABLE 5.2: Data extract of the subject demographics provided in the table

Filename	Age	Gender	MMSE	Computer Skill
1106089771859.txt	50	Male	30	100
110668031945301.txt	74	Female	30	100
1107242873618.txt	44	Female	30	103

A **sub-set** of the data was selected for this study with the following motivation:

- This study did not attempt to model cognitive decline, but rather to model **normal** perception. First, normal perception needs to be defined to able be to detect cognitive decline using these tests in the future. Therefore, only control subjects -

persons with no cognitive impairment - were selected from the database. These subjects had an **MMSE value of 30**. The results would then be a sufficient indication of perceived visual complexities, uninfluenced by cognitive decline.

- The scope of this research only involved the **recognition test results**, and not the recall test results. Each presentation of the recognition test has two **unique** images (since two of the three images presented are identical), whereas the recall test has three different images. The recognition test requires a visual comparison when the subject looks for differences to select the one image that is unlike the other two. The recall test, however, involves a recollective process that requires a judgement of similarity to recall which of the presented three images was previously shown. Ergo, this research only focussed on the tasks that were primarily indicative of the visual **comparison** process, containing just two unique images.

This sub-set, containing 5 087 task instances - one row for each instance - with their respective information as detailed above, was imported into Matlab using the data importing functionality.

After extracting and importing the subset of data, an attempt was made to justify the assumption that, for this subset, the tasks' response times correlate to their human perceived complexity, and to thus confirm that the these times could be used as a fair representation of human perceived complexity.

The response times were examined according to the subjects' demographics - age, gender and computer skill - to test that none of these factors distorted the general distribution of response times.

An algorithm was then developed to predict the visual complexity in tasks that entail distinguishing between three images - with one unique image pair - with the performance results of cognitively intact subjects. An image pair was defined as the two unique images presented in the recognition tasks. The image that appeared twice between the two images in a presented recognition task was assumed to have no additional impact on the visual assessment in this research.

5.2 Algorithm Development

The algorithm was developed to achieve the objectives as ordered in the research specifications in chapter 4. A high-level process flow of the algorithm implementation is shown in figure 5.3.

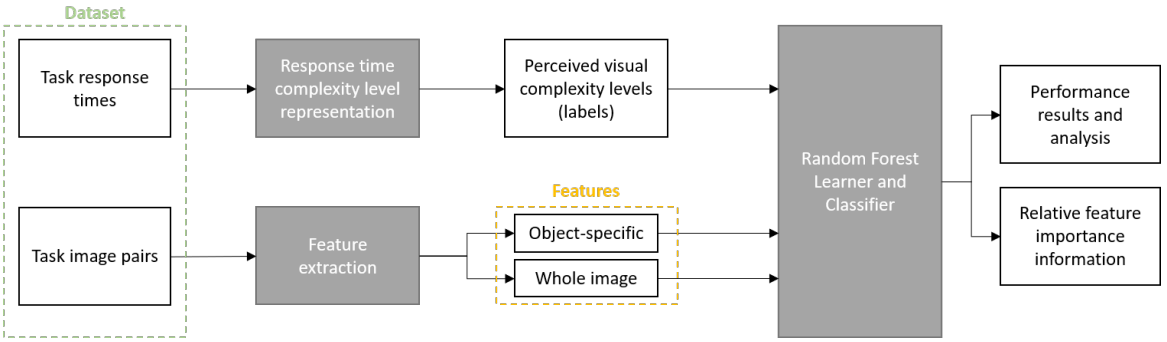


FIGURE 5.3: Method process flow

The response times in the dataset were represented with a list of human perceived complexity labels. Concurrently, the images from the tasks' image pairs were analysed to extract corresponding image information/features. Subsequently the human perceived complexity labels, and the extracted visual features were entered into a random forest learner to find a relationship between these inputs. The aim was to use this deduced relationship in a complexity classifying model to predict the perceived visual complexity of any 16-bit binary image pair. The idea was that if a relationship between the features and the corresponding perceived complexity is known, then with any given image pair, the features can be extracted, and the perceived complexity can be predicted based on this previously built relationship.

The process flow of figure 5.3 was repeated through several experiments with different values. The experiments included applying different response times scale representations, and different classification learners. Further technical details pertaining to each stage of the algorithm-defining process in figure 5.3 are detailed in the following sections.

5.2.1 Human Visual Complexity Level Representation

As described in the problem specifications, and according to the literature and cognitive testing assumption, the response times in the dataset are assumed to correlate, in some way, to the tasks' perceived complexities. Unfortunately, no prior knowledge or method was previously proposed to quantify this exact correlation, i.e. is it a linear relation or a logarithmic one? Moreover, the measured response times in the dataset are a continuous scale of values which needed a larger amount of data to classify. Therefore, a segmentation method was required to divide the scale into "complexity levels" ranging from "easy" (short response times) to "hard" (long response times). Larger response times were represented with a greater complexity level. Intuitively, the longer it took to fulfil a task, the more difficult that task was perceived to be.

Linear and logarithmic paradigms were investigated to define the relationship between the response times and the perceived complexities. Each paradigm was followed by a segmentation method to a number of distinct perceived complexity levels. Accordingly, each response time in the database was represented with a corresponding complexity level/label based on which division the response time would fall under. The number of levels that the response times were divided into changed based on the segmentation technique applied. These were also adjusted based on the results observed.

Initial examination of the response times' distribution yielded a normal/Gaussian one. The following linear and logarithmic segmentation paradigms were explored:

5.2.1.1 Linear Response Time Representation

This paradigm assumed that the response times are linearly related to the perceived complexity. A general illustration of this technique is shown in figure 5.4.

The mean ($\bar{x}$) and the standard deviation (σ) of the response times, and a variable n were used to define the linear class boundaries as per the illustration in figure 5.4. The mean ($\bar{x}$) of the responses times is **3167.9 ms**, and the standard deviation (σ) is **1139.5 ms**.

Each complexity class was of width $n\sigma$. Therefore the variation of n controlled the number of classes that the response time range was classified into. As an example, the different number of classes rendered from varying n is presented in table 5.3.

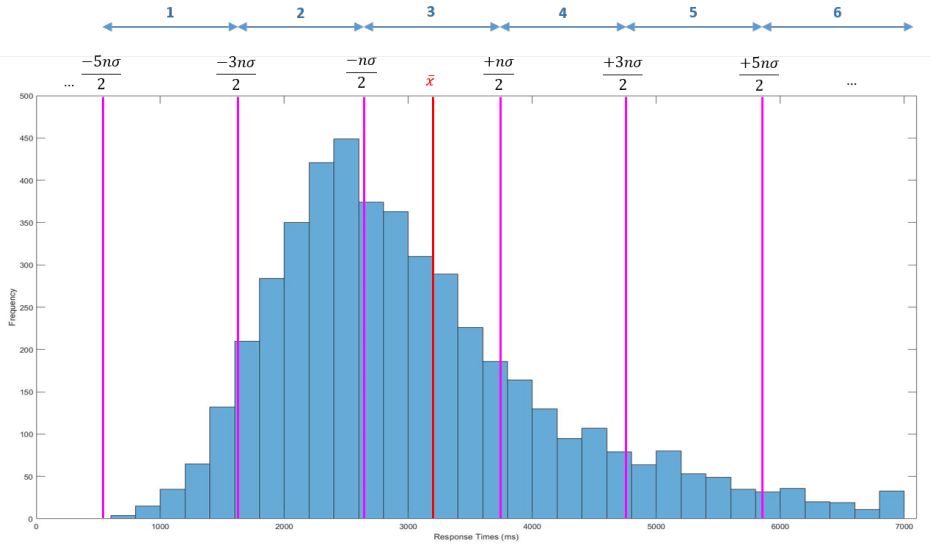


FIGURE 5.4: General linear segmentation of the response times in the dataset

TABLE 5.3: The number of classes generated from varying n in the linear segmentation technique from figure 5.4

n	Number of Classes
0.50	12
0.75	8
1.00	6

In figure 5.4, the levels are shown to be adjacent to each other. However, with such a fine scale of response times, there are responses at almost every millisecond; some of which might fall **in between** certain classes. Therefore, there also needs to be a technique that employs levels slightly separated from each other. An example of a linear response time segmentation with **three** classes slightly separated from each other is shown in figure 5.5.

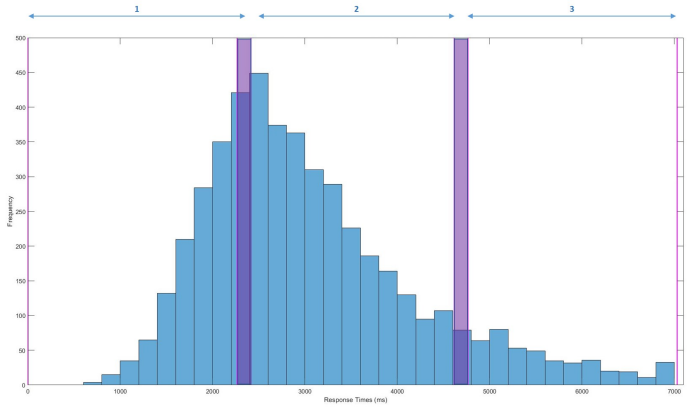


FIGURE 5.5: The distribution of response times segmented linearly into 3 distinct levels slightly apart from each other

The purple regions mark the response times that were 'mid-levels'; these instances fell in between two different classes. This segmentation was implemented as an additional experiment where the mid-levels could be tested to see where (in which class) the algorithm predicted them to fall.

5.2.1.2 Logarithmic Response Time Representation

This paradigm assumed that the response times have a logarithmic relationship with the corresponding perceived complexity. This relationship is motivated on the fact that, as per the Weber-Fechner law, human perception has a logarithmic trend where humans perceive distance, light intensity, and even sound intensity logarithmically [16].

The logarithmic class boundaries were found with the *logpsace* function on Matlab that takes a lower and upper boundary, and provides an array of logarithmically spaced values in this range. Therefore, initially, the shortest response time (625 ms), and the longest response time (7 030 ms) were taken as the two outer limits for this calculation to split the response times into a variety of logarithmically sized classes.

Implementing this straight-forward logarithmic segmentation proved to be challenging as the initial levels contained minimal response times, whereas the latter, larger levels, encapsulated all of the response times. The majority of the response times were being classified with "high" difficulty levels. An example of this technique with 4 classes is shown in figure 5.6.

Doing this, the classification learner (described later in this chapter) would have learnt that most image combinations are classified as one of the latter levels, and the predictor would have been biased towards the same level for all future predictions.

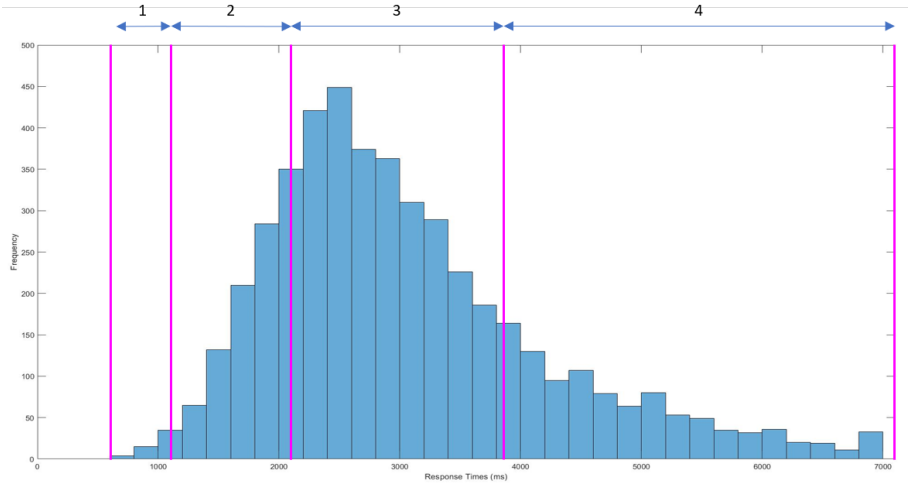


FIGURE 5.6: General logarithmic segmentation of the response times in the dataset starting from the lowest response time to the largest response time

This motivated the decision to attempt an **inward** logarithmic segmentation that began from the mode of the response times. The boundaries would then move outwards on either side. By doing this, the smaller-sized classes were in the middle, most populous area of the response times. The levels then increased in size as the volumes of response times decreased going outwards from the mode of the response times on either side. An illustration of the inward logarithmic segmentation into 4 classes is shown in figure 5.7.

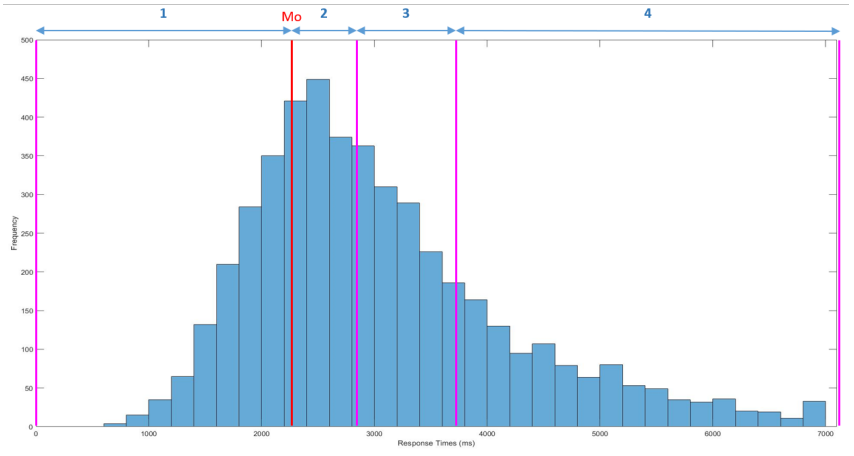


FIGURE 5.7: **Inward** logarithmic segmentation of the response times in the dataset starting from the mode and moving outwards on either side

This inward logarithmic model was therefore chosen to execute the logarithmic segmentation paradigm in the trials of the response time level representation step. Once again the number of classes was varied (for different experiments) using the *logpsace* function in Matlab to provide the boundaries based on the desired number of classes.

Additionally, the point (mentioned previously in the linear segmentation technique) around the fine response times scale applies in this segmentation paradigm as well. Consequently, an inward logarithmic segmentation module with three slightly separated levels was also attempted, and is shown in figure 5.8.

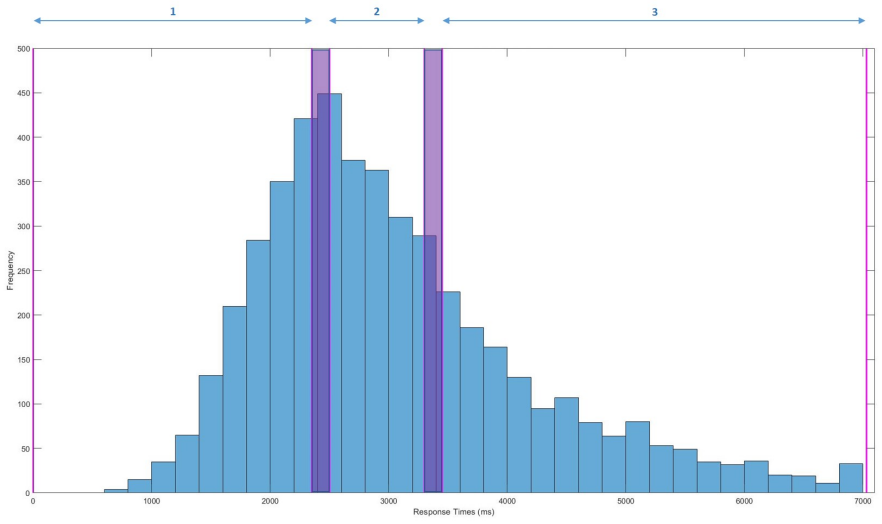


FIGURE 5.8: The distribution of response times segmented logarithmically into 3 distinct levels slightly apart from each other

Following the above paradigms, each response time (and hence cognitive task) in the provided dataset was uniquely labelled with a 'human perceived' complexity level. These were the labels used by the machine learning tool when learning and predicting the task complexities.

Each representation model was applied to the response times independently (one at a time) to produce a set of labels for each separate experiment of the algorithm. Therefore, each complexity representation model had its own set of corresponding test results. The comparison between the results from the two segmentation paradigms was done to find the optimal combination of a segmentation model, and a complexity predicting technique that makes up the algorithm.

5.2.1.3 Human-Provided Complexity Levels

This experiment has not been listed in the process flow of figure 5.3 as it was simply explored as a proof of concept to test the algorithm without the response time representation step.

While the response times have been assumed to correlate to the perceived complexity, another attempt was also made to extend a concept implemented in the preliminary work, where the human perceived complexities were provided directly by human subjects.

As referred to in chapter 3, the preliminary work focused on the visual complexity of **single** images from this same set. Although the images were from the same dataset that has been provided for this research, the same response times could not be used as a measure of perceived complexity in that study. This is because these response times are indicative of the responses to the **complete** tasks, which required comparing **multiple** images ('relative' complexity). However the scope of that study was only on defining the complexity of single images. Therefore, certain single images were directly labelled by three individuals with their perceived complexity as 1 ('easy'), 2 ('medium') or 3 ('hard').

While the preliminary work had a significantly smaller scope than this research, with only a fraction of the features extracted and images tested manually, the idea of directly labelling the tasks with perceived complexity labels prior to training was also explored in this research.

Consequently, in this attempt, 10 subjects and 120 recognition tasks from the dataset were chosen at random. From this pool, different subjects were asked to label, for different recognition tasks, how relatively difficult it was to distinguish between the three presented images when finding the image that was different (where 1 was 'easy', 2 was 'medium' and 3 was 'hard'). As a result there were a total of 120 recognition tasks directly classified with perceived complexity levels.

These were used in another small experiment to assess algorithm performance when the input human perceived levels were provided directly by human subjects.

Sections 5.2.1.1 to 5.2.1.3 have detailed various techniques by which the provided data was labelled with human perceived complexities. These labels were required for the algorithm to learn the correlation between human perceived complexities, and the various features in the image sets. These features are now described in the following section.

5.2.2 Feature Extraction

This section explains the extraction of features, chosen based on the background studies detailed in chapter 3, that describe different characteristics of the images in the recognition tasks.

5.2.2.1 Object Type Definitions

Before presenting the different features, this section details the object **types** that will be referred to in certain feature descriptions [7]. The objects are conceptual and scalable, and can apply to pixels or square black or white shapes of any size. All of the object types are defined for both, the black **and** white squares.

Given an image A, with cells $A(i, j)$ where $i, j = 1 : 4$, the various objects were detected by scanning the image row by row, left to right, from the top row to the bottom row. These object types will be referred to and incorporated in some of the different features listed later under 5.2.2.2.

- **Adjacent paths**

  An adjacent path is a line of consecutive adjoining squares of the same colour (immediately alongside each other, against one of the four **sides** of each square).

  When scanning the image, if the algorithm detects a black block, then the blocks adjacent to this will also be checked for a black block. Given that the initial black block is detected at A(i,j), then the adjacent cells will be $A(i-1, j)$, $A(i+1, j)$, $A(i, j-1)$ and $A(i, j+1)$. This process will run recurrently until there are no other black blocks found in this line/path, thus ending the black adjacent path. The same process runs to also detect white adjacent paths.

For illustration, the detected **black** adjacent paths are outlined in red in the example image (figure 5.9).

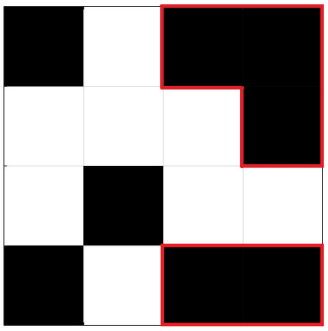


FIGURE 5.9: Two adjacent paths (outlined in red) found in an image

- **Diagonal paths**

  A diagonal path is a line of consecutive squares of the same colour diagonal to each other (immediately alongside each other, against one of the four **corners** of each square).

  The algorithm will once again scan the image in a similar manner. If a black block is found, the blocks diagonal to this will be checked for black blocks as well. Given that the initial black block is found at A(i,j), the diagonal cells will now be A(i-1,j-1), A(i-1,j+1), A(i+1,j-1) and A(i+1,j+1). This process will run recurrently until there are no other consecutive black blocks found in the diagonal path, thus ending the diagonal path instance. The same process runs to also detect white diagonal paths.

  For instance, the one black diagonal path present in the example image is outlined in green in figure 5.10.

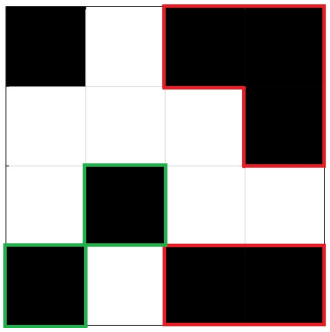


FIGURE 5.10: One diagonal path (outlined in green) identified in the image

- **Remaining single blocks**

  Once the adjacent and diagonal paths are found respectively in that order, and for each colour, any remaining, unclassified black and white squares are detected and classified as single blocks. An example black single block is shown in figure 5.11.

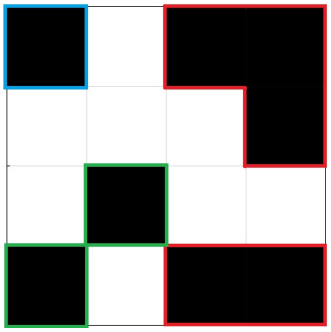


FIGURE 5.11: One single black block (outlined in blue) found in the image

By these definitions, the binary image in figures 5.9 to 5.11 also has **one white** adjacent path (outlined in orange), and **one white** single block (outlined in purple) as shown in figure 5.12.

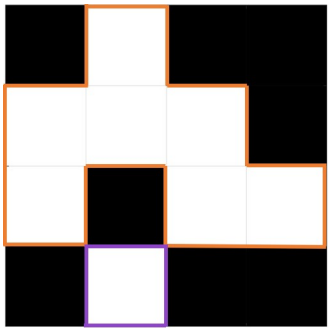


FIGURE 5.12: One white adjacent path and one white single block found in the image

There were therefore three object **types** predefined for the feature extraction: adjacent paths, diagonal paths, and single blocks.

5.2.2.2 Feature Details

The purpose of this study was to be able to computationally define the human perceived complexity when viewing **multiple** images in a cognitive task. Therefore, understanding the task complexity required understanding the *relative* visual complexity of these images, and hence required capturing *relative* characteristics between the unique image pair.

Based on this, it was initially assumed that each feature would be extracted for each image in a unique image pair from a recognition task. The absolute difference between the respective features would then be computed to represent a set of 'relative' features. These calculated differences would then be inserted into the machine learning algorithm to describe the image pair's relative features. However, while the objective is to obtain a comparative measure, image feature comparisons might not always be as simple as an absolute difference between each image's measure of that feature.

Therefore for the following extracted features described, each respective feature was then obtained and loaded into the machine learning algorithm independently. This was done with the idea that the algorithm would itself learn how the features for each image in the pair related to each other, and how they consequently impacted the relative visual complexity between the image pair. There are, however, still certain features that are comparative in nature, and describe a common or different characteristic in the pair of images. These directly measure some form of co-saliency between the images. Those particular comparative features have been marked with "(Relative measure)" next to the feature name in this section.

Two types of extracted features are detailed under this section: **object-specific** features, and **whole image** features. As per Witkin's theories, the object-specific features cater for field independent visual observations that focus on specific object-related details within the images. The whole image features cater for more holistic observations that assess a higher level detail of the image as a whole.

Object Specific Measures

The following features describe something object-specific for each image. The object types are as defined previously in 5.2.2.1.

- **Number of Objects**

  This feature is simply a count of the number of adjacent paths, number of diagonal paths, and number of single blocks within each image, for both black and white objects. There are therefore 6 measures for the number of objects in each image.

  As an example, these have been counted in table 5.4 for the example image pair presented in figure 5.13.

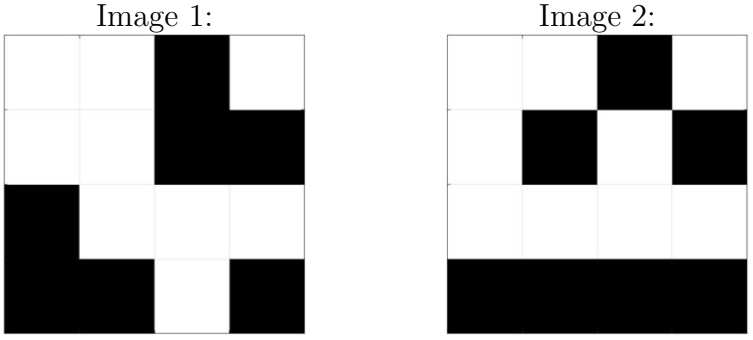


FIGURE 5.13

TABLE 5.4: Number of objects for the image pair in figure 5.13

	Image 1		Image 2	
	Black	White	Black	White
Adjacent Paths	2	1	1	1
Diagonal Paths	0	0	1	0
Single Blocks	1	1	0	1

- **Object Path Lengths**

  This measure is the total lengths of the (black and white) adjacent paths and the diagonal paths present in each image. This feature is irrelevant for single blocks. There are therefore 4 measures for the object path lengths.

  As an example, these path lengths have been calculated in table 5.5 for the same image pair in figure 5.13.

TABLE 5.5: Object path lengths for the image pair in figure 5.13

	Object Path Lengths			
	Image 1		Image 2	
	Black	White	Black	White
Adjacent Paths	6	8	4	8
Diagonal Paths	0	0	3	0

- **Object Angles (Relative Measure)**

  This is the number of (black and white) adjacent and diagonal paths that are at the **same** angle, and the number of (black and white) adjacent and diagonal paths that are at the **opposite** angle between the two images. These angles are calculated with respect to the x axis of the images. This feature is only relevant to adjacent paths and diagonal paths as all single blocks are always at the same angles in these images.

  Computing the averages or sums of the angles would lack relevance because knowing an overall or average object orientation does not provide any substantial insight into the individual object details. Finding an average of 45° for two objects at angles of 0° and 90° does not add value to capturing object information. Therefore, finding where an image pair contains objects at the same angles, and where it contains objects at exactly opposite angles might be more useful. Only the extreme scenarios of the angles were considered because slight degree variations are likely to be visually insignificant in images of this size, and in scenarios where the images are presented in a time constrained environment. However, objects occurring at opposite angles could indicate some level of object symmetry, and objects at the same angle could indicate object similarity.

  The example image pair presented in figure 5.14 contains one set of **black diagonal paths** at **opposite** angles, and one set of **black adjacent paths** at **equal** angles between the two images.

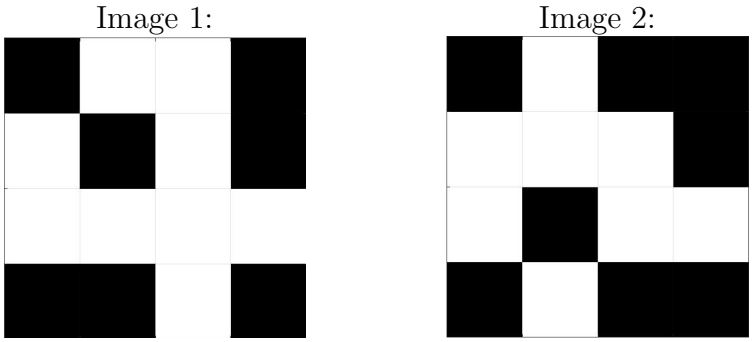


FIGURE 5.14

- **Objects with Similar Locations (Relative Measure)**

  This is a comparative measure that indicates the number of objects that have **similar** locations between the two images. Since the location is such a sensitive value, any objects with locations within approximately half a square of each other are counted as similarly located objects.

  Equation 3.3 previously presented in chapter 3, and shown below, describes the centroid formula that was used to classify the locations of any adjacent paths, diagonal paths or single blocks in the images. Given an object B made up of n basic shapes with area A, x_i is the x coordinate, and y_i is the y coordinate of the local centroid of shape/area A_i used to calculate centroid coordinate $B[\bar{x}, \bar{y}]$.

$$B[\bar{x}, \bar{y}] = [\frac{\sum_{i=1}^{n} A_i.x_i}{\sum_{i=1}^{n} A_i}, \frac{\sum_{i=1}^{n} A_i.y_i}{\sum_{i=1}^{n} A_i}] \tag{3.3}$$

  An example of the centroids found for the black objects in an image is illustrated in figure 5.15.

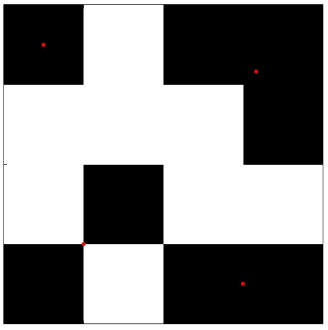


FIGURE 5.15

Only objects of the **same type and colour** are checked for similar locations between the two images. I.e. there are 6 counts, where the first 3 are the number of black adjacent paths, number of black diagonal paths and the number of black single blocks in similar locations, and the next 3 are the same counts for the white object types with similar locations.

Based on the example pair in figure 5.16, the model would detect that there is **one set of black adjacent paths, one set of white adjacent paths, and one set of white single blocks** with **similar** locations between the two images.

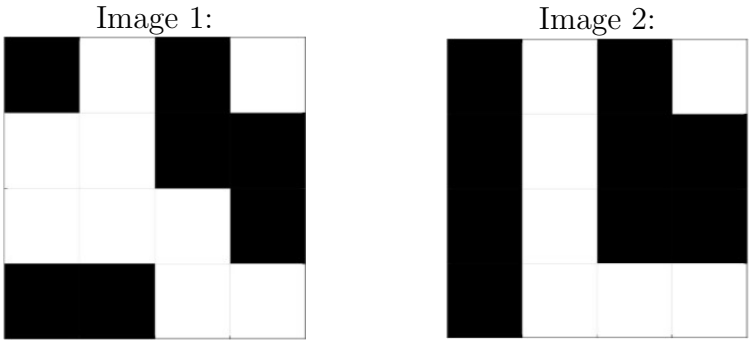


FIGURE 5.16

- **Different Object Types Present (Relative Measure)**

  This measure is a binary indicator that flags if there are any object **types** (adjacent paths, diagonal paths or single blocks) that exist in one image, but not the other.

  Since the object types are quite distinct, it was considered that perhaps the occurrence of an object type in one image, but the lack thereof in the other image could constitute for a level of visual difference between the images. The **number** of objects was not relevant in this measure.

  The indicator was calculated separately for black objects and white objects respectively. For instance in the example in figure 5.17, Image 2 has one black diagonal path while Image 1 does not have any black diagonal paths, and Image 1 has a black single block while Image 2 does not have any black single blocks. Therefore the indicator for the black different objects present would flag. However there is a white adjacent path and white single block in both images, and no white diagonal paths in either of the images, therefore the indicator for the white different objects present would not flag.

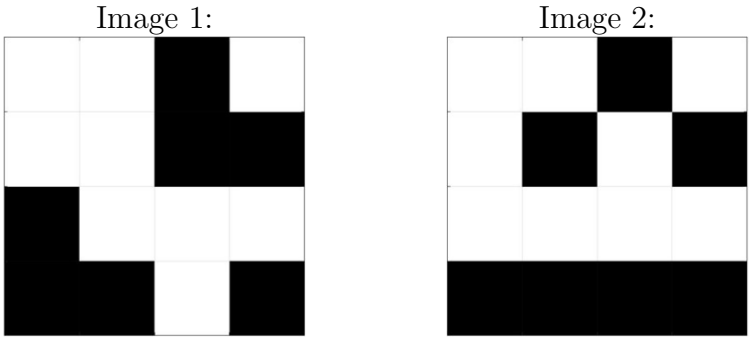


FIGURE 5.17

- **Similar Objects Within Each Image**

  This is the number of objects that are the same (shape and size) **within** an image.

  While the previous features indicate the length and quantity of the present objects, they did not provide any information on the objects' direction or shape. For instance the black objects in figure 5.18 are both single black adjacent paths with a length of 4, and even have similarly located centroids, however their shapes are different, and they could thus be perceived differently.

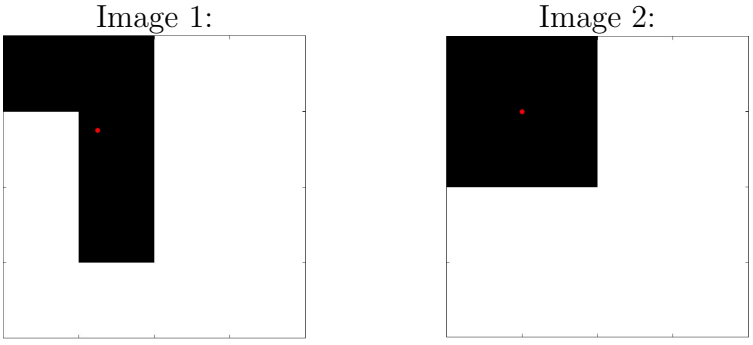


FIGURE 5.18

  Therefore, this shape measure was included to detect objects within an image that are of the same shape. This measure also included objects that are the **same shape, but at a different angle**.

  For the shape comparison, the algorithm isolated each object found, and used the *RegionProps* function on Matlab to store each object as its own temporary image. This helped separate the shape of each object. These shapes were then compared directly against one another (and also compared when rotated at 90°, 180°, 270° and 360°).

As an example, this calculation indicates that in figure 5.17 (same example image pair presented in the previous feature), Image 1 has two of the same objects, and Image 2 has none.

- **Similar Objects Between Images in Pair (Relative Measure)**
  This is the number of objects that are the same (shape and size) **between** the two images. The same shape detection method from the previous measure was applied to detect objects of the same shape occurring **between** the images. Once again, this measure also counted objects that are the same, but at a **different angle**.
  For example, the algorithm would detect one similar object between the two images in figure 5.19.

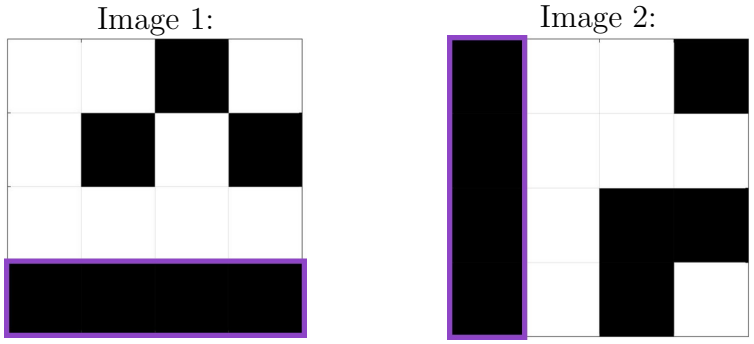


FIGURE 5.19

This concluded the seven **object-specific** features captured for the image pairs in the recognition tasks from the dataset.

Whole Image Features

The following features were extracted when assessing each image as a **whole**, without focusing on isolated object details.

- **Whole Image Object Spacing**
  The overall spacing measure was calculated as the average distance between all objects (of one colour) in the image. This assumed that the average should indicate how those objects are spread out across the image.

As mentioned previously, an object's centroid was used to define its location. Based on this, the distance formula given in equation 5.1 was applied on the object centroids to calculate the distance between all objects (of one colour) within each image.

Given object centroids $P(x_1, y_1)$ and $Q(x_2, y_2)$, the distance $d(P, Q)$ is:

$$d(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5.1}$$

An average was then taken of all possible object distances to indicate the overall object spacing in each image.

Given the example in figure 5.20: for the distances calculated between all the black objects for each image, the averages would indicate that the black objects are more spaced out in Image 2, than they are in Image 1.

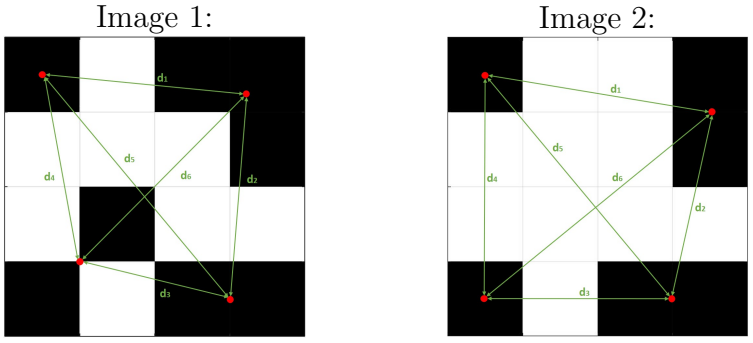


FIGURE 5.20: The object distances between the black objects drawn for an image pair.

- **Whole Image Bit Comparison (Relative Measure)**

  This whole image feature extracted was a **direct** comparison technique between the two images as detailed in chapter 3. However, that was presented as an AND calculation, whereas for this feature, the calculation was slightly modified to be an XOR one. This was still a logical operation that computed a bit-by-bit comparison between each respective bit in the images to map every single point of difference found. For the calculation, each image was taken as a 16-bit binary array (similarly to how the images were encoded to be written in the provided dataset).

The generic XOR calculation is shown in equation 5.2 below for two binary arrays B_1 and B_2:

$$XOR(B_1; B_2) = B_1\bar{B}_2 + \bar{B}_1 B_2 \tag{5.2}$$

The result from the XOR computation on the two images was also a 16-bit binary array that can be visualised back into a binary image as shown in figure 5.21. While this was not the format in which it was inserted into the machine learning algorithm, it simply illustrates how the resulting array maps the points of differences between the two images. The highlighted areas in the consequent matrix indicate the regions of visual dissimilarity.

Once again, since an array cannot be inserted into the machine learning algorithm, a summation of the resulting array was made as an indication of how many respective bits were different between the two images. This whole number formed the overall bit difference measure between the two images.

It can also be observed that the two images in figure 5.21 have similar-shaped black objects, but these would not be detected in this simple bit comparison as the similar shapes do not have the same/similar locations within their respective images. This justifies why the other object-specific features detailed previously were also required.

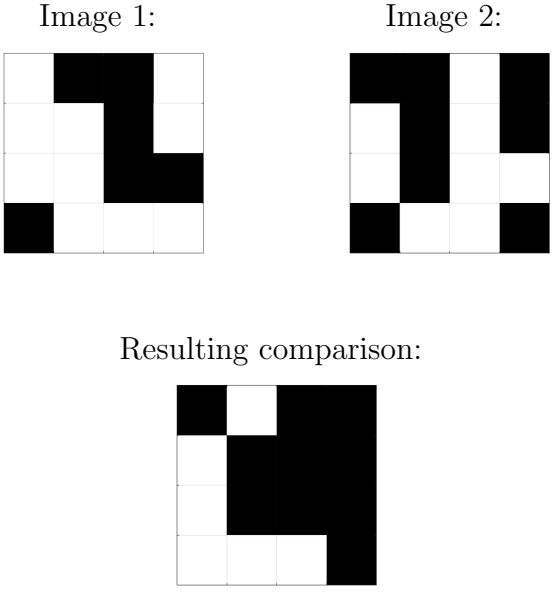


FIGURE 5.21: Illustration of the bit-by-bit comparison

- **Relaxed Image Symmetry**

  This whole image measure checked for symmetry in both of the images. To tie in with Attneave's theories, symmetry adds a level of visual redundancy which should simplify an image.

  Each image was scanned to test for symmetry in one of the following ways (examples of each shown underneath):

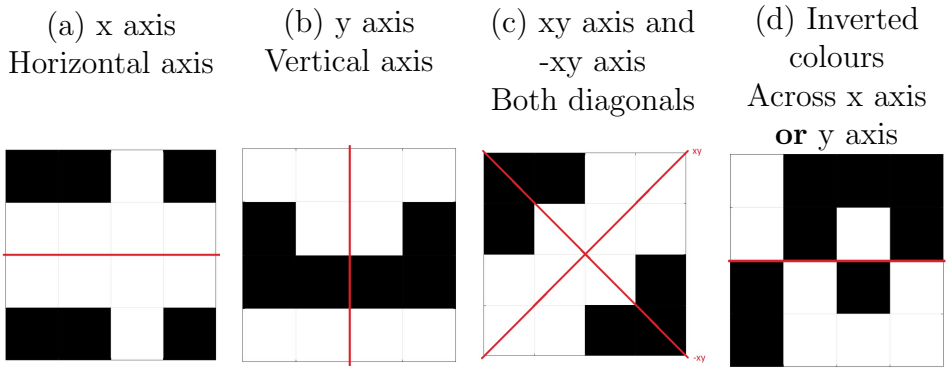


FIGURE 5.22: The different image symmetry possibilities

The algorithm looked for exact symmetry, as well as approximate ('relaxed') symmetry. Approximate symmetry is if the image is only one block (black or white) off from being exactly symmetrical. In this instance the image is still noticeably symmetrical at a visual glance. This was also done to realistically cater for additional test instances, because the probability of being presented with images that are exactly symmetrical in these recognition tasks could be relatively low.

An example of an 'approximately symmetrical' image is shown in figure 5.23. This image is approximately symmetrical across the xy, as well as -xy axis.

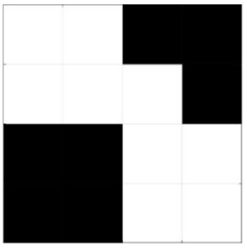


FIGURE 5.23: Example of an image that is almost symmetrical.

In the event of an image being symmetrical, or 'almost' symmetrical, across any of the above ways, a binary indicator was flagged to indicate some aspect of symmetry in this image.

- **Gabor Filter Feature Vector**

  The Gabor filter is a mathematical model. As introduced in chapter 3, it is a bandpass filter made up by a sinusoidal plane at a certain frequency and orientation, modulated by a Gaussian envelope [22]. These filters are commonly proposed as texture-segmentation models as some hypothesise that the filters share similarity with human visual perception.

  In this study, a filter bank was created of Gabor filters devised from equation 3.1 presented in chapter 3. The filter bank contained a combination of Gabor filters at 8 different orientations, and 5 different frequencies, thus resulting in 40 filters.

  The generated Gabor filters in the filter bank are as shown in figure 5.24 - the illustration separates these into their real parts, and their magnitudes, as the Gabor filter is a complex one.

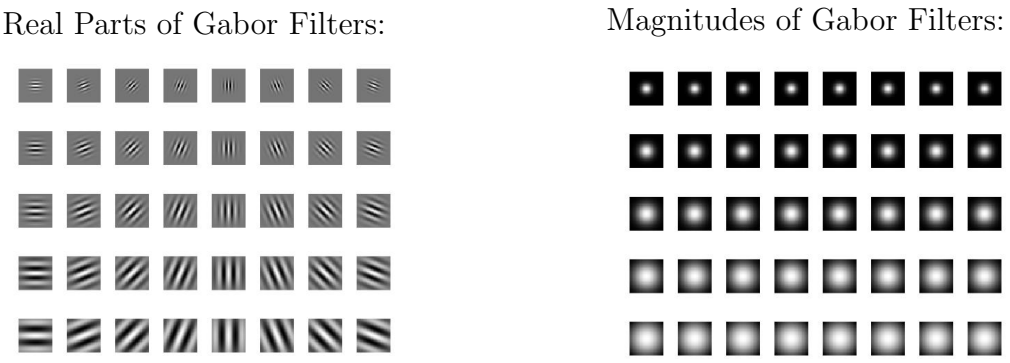


FIGURE 5.24: The Gabor filter bank implemented in this research

  The filter bank was then applied to each image in the image pair. An illustration of the filter application on one example image is shown in figure 5.25.

  The results were subsequently downscaled to obtain the final feature vector that contained the results of the application of all filters on the image.

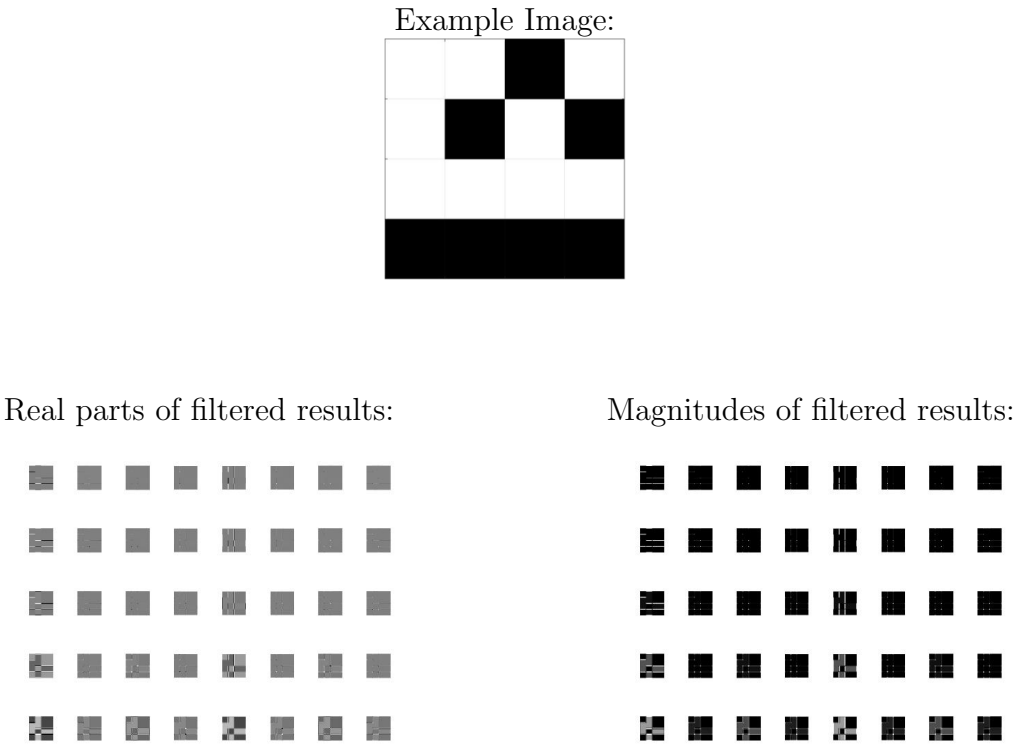


FIGURE 5.25: Filtered results after Gabor filter application on example image

While the final feature vectors from each image contained the results of the filter bank application, an entire vector could not be inserted as a feature into the machine learning algorithm. Therefore single points of information about each feature vector were calculated to yield two separate feature inputs into the learner: the sum and distribution (standard deviation) of each Gabor output feature vector. It was considered that these two features would hold some level of information about each Gabor output vector.

- **Fractal Dimensions**

  The fractal dimension (D_f) is another mathematical calculation. As initially presented in chapter 3, it is a ratio that indicates visual complexity by measuring the level of visual detail in a 'fractal' pattern at different levels of magnification [17].

  In this study, the fractal dimension was calculated to gauge if the images from the tasks display any level of self-similarity, and to consequently capture another form of visual complexity. One image with a larger fractal dimension than another image

contains more detail at smaller scales, and can be assumed to be more visually complex.

For each image, the fractal dimension was calculated using the box counting technique. As mentioned in chapter 3, this technique estimates the number of boxes required to cover (the non-zero parts of) an image at different box sizes [10]. The smaller the box size, the greater the number of boxes required to cover the image.

The box size (r) was increased in powers of 2 (1,2,4,..) for each increment. As mentioned previously, box sizes greater than about 50 % of the image width tend to introduce errors. The images in the dataset are of size 536 x 536. Therefore, the greatest box size was 256 (2^8), which is approximately 48 % of the image width. The corresponding number of boxes (N) was obtained for this range of box sizes using the box counting technique implemented in Matlab.

The values of $log(N)$ were plotted against the values of $log(r)$. The fractal dimension was then calculated as the slope of the **line of best fit** over the points using previously presented equation 3.2. This slope is an average measure to quantify how much the required number of boxes changes with the box size. The fractal dimension must be less than the dimension, D, of the fractal set. Therefore, for these images, it was known that the resulting fractal dimensions should have been less than 2.

During this implementation, it was observed that in many scenarios two images had similar calculated fractal dimensions, but there was more information lying in how the points varied in between the line of best fit. For certain images, the rate at which the number of boxes changed with the box size could occasionally vary considerably between consecutive pairs of points/box sizes. Therefore the overall fractal dimension calculated on a line of best fit could be misleading - this line is likely to miss certain points of variation in between [10]. Essentially, the more the slope changes at different points of the graph, the less the fractal dimension calculation accurately describes their scaling characteristics [10]. Ideally, if an image is a good fractal, the slope would not fluctuate significantly, however self-similarity is not as easily achievable with 16-bit images. Therefore, the extent to which the fractal dimension values vary with different box sizes was also measured:

Another two related measures were calculated to provide three different fractal dimension-related measures in total:

1. The fractal dimension (overall slope of the line of best fit).
2. The **range** of the fractal dimension values across the total number of measurements taken for the image: This is the largest fractal dimension value calculated between two consecutive points, minus the smallest fractal dimension value calculated between two consecutive points.
3. The distribution of the fractal dimension values across the total number of measurements taken for the image: An indication of how much the slope/fractal dimension varied between the series of points. This was calculated as the standard deviation of the different fractal dimension values obtained for each image.

All three of these fractal dimension-related features were calculated for each image in the image pair.

5.2.2.3 Feature Summary

This complete feature extraction process in the algorithm catered for visual observations focused on more intricate detail within the images, as well as visual observations focused on the holistic overview of the image pair.

The variety of features were devised based on the reviewed literature on visual perception (from chapter 3), and the relating mathematical algorithms presented herein. The Gabor filters and the fractal dimension features are state-of-the-art, previously well established and tested methods often applied in different fields of image analysis. These models were chosen based on research into which existing mathematical algorithms could mimic human visual perception computationally. The direct comparison technique is a modification of a previously well established and simple technique of comparing binary arrays. The remaining object-orientated features were novel and intuitive, motivated by Attneave and Witkin's concepts of visual perception, as well as further studies into mathematical

complexity techniques. Co-saliency was measured by detecting the co-occurrence of similar objects (as valuable focus points) within, and between, the images. The theories specified by spatial information motivated the overall object spacing features.

The above features were all extracted for the image pairs present in the recognition test results from the dataset. However, how their collective presence would affect the perceived complexity was still unknown. With the task response times in the database labelled with perceived task complexity labels, the following step was for the algorithm to learn how the presence of these features mapped to the tasks' perceived complexity.

The feature correlation was tested based on the feature importance values generated by the machine learning technique. These are explained further in the next section.

5.2.3 Random Forest Learner and Classifier

With the extracted features, and the labelled complexity scale, the following step in the algorithm necessitated a supervised learning model that could determine the relationship between the two. The random forest was chosen due to its robust characteristics as a classifier as suggested in chapter 3.

Since the random forest is a supervised learning technique, it requires an input (X), and a desired output (Y). In this study, for both training and testing, X was the range of visual features present in the visual tasks' image pairs, and Y was the corresponding perceived complexity levels.

In this technique, the training dataset (the sets of extracted features for the training tasks) was divided into random samples during the training process, each of which is called a bootstrap dataset. The algorithm then created several decision trees, one for each bootstrap dataset. Subsequently, it selected different subsets of the extracted features for splitting the nodes in each different decision tree, and trained the model against the respective complexity labels. The number of trees was selected based on the out-of-bag (OOB) error graphs plotted to gauge at which point the error saturated. Any more trees thereafter were simply taking up more computational power. This number of decision trees for this research was 250.

The decision trees were therefore trained collectively given the feature sets for the image pairs, and the complexity levels. When testing, these trained decision trees then used the inserted feature sets of the test data to each provide a complexity prediction per set of extracted features. As per the classifier, the various predictions from the trees were aggregated with a modal technique to yield the final predicted complexity label (Y_N) for the N^{th} set of extracted features (X_N).

The output of the random forest classifier when testing was therefore a set of predicted complexity labels, given the corresponding sets of extracted features. These predictions were then compared to the original, predefined complexity labels of the tasks to calculate a recognition/correctness percentage.

A 3-cross validation technique was applied for this algorithm: The dataset was divided into three subsets, where iteratively, one subset would be used for training, and the other two for testing. The final testing results for each experiment were then a combination of the test results from the 3 iterations. Doing this prevented overfitting, and ensured that all the provided data was used for both training, and testing, at some point in the model.

In addition to the complexity predictions, the random forest model provides better insight into which features were used, and to which lengths they had an impact on the complexity predications. The model can provide this information since different subsets of features were selected for different decision trees, thus allowing the model to gauge which features had a greater impact in the final complexity predictions. In Matlab, the feature importance is calculated as part of the *TreeBagger* function. As per Matlab documentation: This resulting feature importance information is "*a numeric array of size 1-by-Nvars (number of features) containing a measure of importance for each predictor variable (feature). For any variable, the measure is the increase in prediction error if the values of that variable are permuted across the out-of-bag observations. This measure is computed for every tree, then averaged over the entire ensemble and divided by the standard deviation over the entire ensemble.*"

This method yielded insight into which of the feature sets were more relevant to the complexity classifications, which achieves the fifth objective in the research question specification. The feature importance information is also vital because it could later be compared

to the visual features in human perception as suggested by psychologists, and hence contribute to a better understanding of modelling visual perception computationally.

In the event that the random forest model was not optimal in picking up patterns between the response times and visual complexities, the Matlab Classification Learner App was also employed for a quick implementation and comparison of other available classification learner techniques. This app includes several types of decision trees, support vector machines, as well as a few ensemble classifiers. The extracted features were inserted into all of the available classifiers, with the additional PCA functionality enabled to automatically remove redundant features, and get a quick assessment of the other classifiers' performance.

5.3 Experiments

A variety of experiments were conducted to test the developed algorithm with a collection of segmentation paradigms (linear and logarithmic), number of classes (that are adjacent as well as slightly distanced), and different classifiers. These experiments are listed in table 5.6.

TABLE 5.6: List of the experiments attempted with the algorithm

Experiment	**Description**
1	• Linear segmentation of RTs (adjacent) • Number of classes: 8 • Classifier: Random Forest
2	• Linear segmentation of RTs (adjacent) • Number of classes: 6 • Classifier: Random Forest
3	• Linear segmentation of RTs (adjacent) • Number of classes: 4 • Classifier: Random Forest
4	• Linear segmentation of RTs (separated) • Number of classes: 3 • Classifier: Random Forest
5	• Inward log segmentation of RTs (adjacent) • Number of classes: 4 • Classifier: Random Forest
6	• Inward log segmentation of RTs (separated) • Number of classes: 3 • Classifier: Random Forest
7	• Inward log segmentation of RTs (separated) • Number of classes: 3 • Classifier: RUS Boosted Trees
8	• Inward log segmentation of RTs (separated) • Number of classes: 3 • Classifier: Linear SVM
9	• Extra testing of experiment 7's trained model on **mid-levels** • Number of classes: 2
10	• Binary segmentation of RTs (easy and hard, from extreme sides) • Number of classes: 2 • Classifier: Random Forest
11	• Human labelling of tasks • Number of classes: 3 • Classifier: Random Forest

5.4 Summary

This chapter detailed how the complexity learning algorithm was developed given a previously collected database of certain cognitive tests' human response times. The response times were represented in a scale as an indication of the subjects' perceived complexity of the visual tests. The image features were extracted for image pairs in recognition tasks. The random forest technique was then used to try learn how the presence of the different visual features affected the visual complexity scale labels. The results and analysis from this were used to build a classifier; the prediction accuracy of which would subsequently be tested on the testing dataset. Experiments were designed to test the different components of this algorithm, and their results are presented in the following chapter.

CHAPTER 6

Results

This chapter presents the results obtained when the methods described in chapter 5 were applied to answer the initial research question: *To what extent can a complexity measure, based on information theory and machine vision, model and explain human visual perception in the context of cognitive tests?*. The results are presented in terms of the prediction accuracies, and the resulting importance levels of the features extracted from the task images.

As detailed in chapter 5, a system comprising of feature extraction, and a machine learning prediction model was developed, trained, and tested using the visual recognition tasks, to predict the human perceived complexities in these tasks. The available measure for human perceived complexity - their response times to the visual tasks - was statistically examined to check for their initial reliability as a measure of human perception of a task difficulty. Different segmentation paradigms were subsequently applied to the response times' continuous range to provide a set of labels, and then tested. The machine learner's prediction accuracy was calculated by comparing the model's predicted complexity levels to the reaction times scale.

The examination of the response times' distributions change with the dataset subjects' demographic variables are presented in figures 6.1, 6.2, and 6.3.

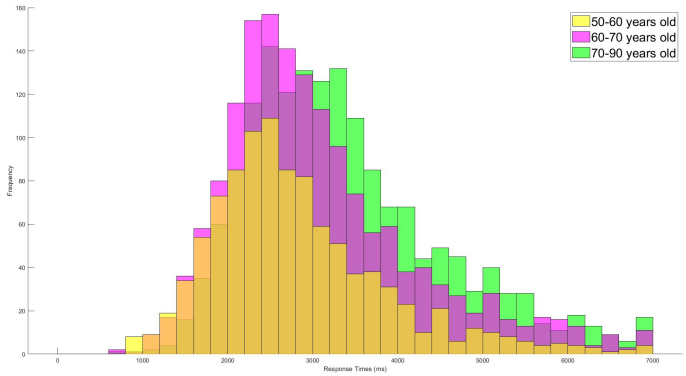


FIGURE 6.1: Frequency plot of the response times segmented by **age**

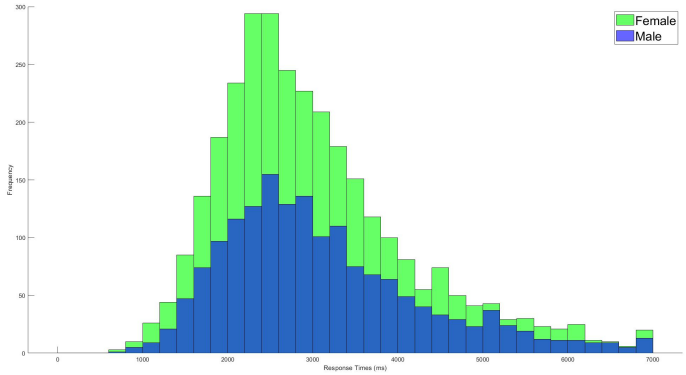


FIGURE 6.2: Frequency plot of the response times segmented by **gender**

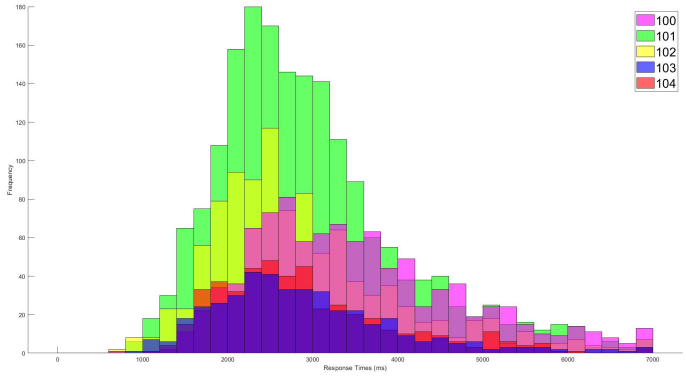


FIGURE 6.3: Frequency plot of the response times segmented by **computer skill**

The graphs demonstrate that while the response time volumes vary within the factors, the distributions are similar across the age groups, between the two genders, and across the computer skill levels. Therefore, it can be assumed that a subject's age, gender or computer skill does not noticeably offset their response time. The response times could

consequently be assumed as a valid representation of the perceived complexity of the tasks performed by the subjects.

Table 6.1 summarises the **results** from the experiments performed to test the various components of the methods as listed in table 5.6. The results are described and analysed through this chapter to explain the process and motivation behind the selection of the final set of methods based on these results. Each section in the chapter addresses one of these components of the method.

TABLE 6.1: Results of the main experiments

Experiment	**Description**	**Same Class Recognition**	**One Class Apart Recognition**
1	• Linear segmentation of RTs (adjacent) • Number of classes: 8 • Classifier: Random Forest	36%	40%
2	• Linear segmentation of RTs (adjacent) • Number of classes: 6 • Classifier: Random Forest	42%	45%
3	• Linear segmentation of RTs (adjacent) • Number of classes: 4 • Classifier: Random Forest	61%	30%
4	• Linear segmentation of RTs (separated) • Number of classes: 3 • Classifier: Random Forest	59%	41%
5	• Inward log segmentation of RTs (adjacent) • Number of classes: 4 • Classifier: Random Forest	27%	38%
6	• Inward log segmentation of RTs (separated) • Number of classes: 3 • Classifier: Random Forest	**36%**	**45%**
7	• Inward log segmentation of RTs (separated) • Number of classes: 3 • Classifier: RUS Boosted Trees	35%	41%

8	• Inward log segmentation of RTs (separated) • Number of classes: 3 • Classifier: Linear SVM	37%	38%
9	• Extra testing of experiment 7's trained model on **mid-levels** • Number of classes: 2	**70%** *	Only 2 classes; any incorrect prediction is pure misclassification
10	• Binary segmentation of RTs (easy and hard, from extreme sides) • Number of classes: 2 • Classifier: Random Forest	56%	
11	• Human labelling of tasks • Number of classes: 3 • Classifier: Random Forest	58%	40%

6.1 Response Time Segmentation Experiments

The linear and logarithmic segmentation paradigms are as described in chapter 5, and are illustrated in figures 5.4 and 5.7 respectively.

Table 6.1 illustrates that for each segmentation paradigm (linear and logarithmic), the prediction accuracies tend to increase with a fewer number of classes. This is a common finding in machine learning.

However, when experimenting with the different number of segments in the linear segmentation technique (experiments 1 - 4), it was found that the prediction accuracies were misleading: Upon closer inspection of the distribution of the prediction values, it was observed that the model repeatedly predicted the majority of the tasks to be in a narrow band of 2-3 segments. The distribution of the response times also indicated a narrow Gaussian around a mean response time value. The training tasks therefore fell within that area as well, which may have resulted in the testing tasks being biased towards the most frequent level/s, with a misleading prediction of the correct level. This bias became more significant with a decreasing number of classes, resulting in more misleading correct predictions. Figure 6.4 and figure 6.5 illustrate the predictions' distributions, for the two

extreme cases of the experiments involving linear segmentation: 8 classes and 3 classes, respectively.

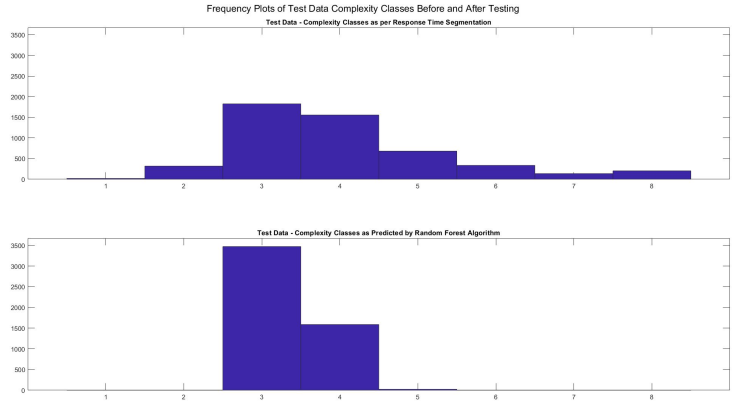


FIGURE 6.4: The distribution of the labels calculated from linear segmentation of reaction times into 8 labels (top graph) and the distribution of the predicted complexity classes (bottom graph)

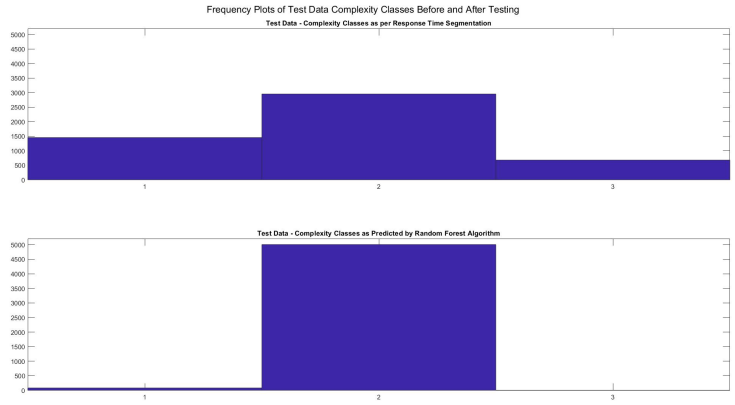


FIGURE 6.5: The distribution of the labels calculated from linear segmentation of reaction times into 3 labels (top graph) and the distribution of the predicted complexity classes (bottom graph)

The inward logarithmic model, illustrated in figure 5.7, was pursued to correct this bias by providing a more equal number of response times within each segmented level. As described in the methods, starting the logarithmic segmentation from the mode of the response times prevented a large volume of the task response times falling under a specific class/segment. The resulting predictions were more spread out across the levels, thus reducing the possibility of the classification bias. An illustration of the distribution of response times from the 3-level inward log segmentation is shown in figure 6.6

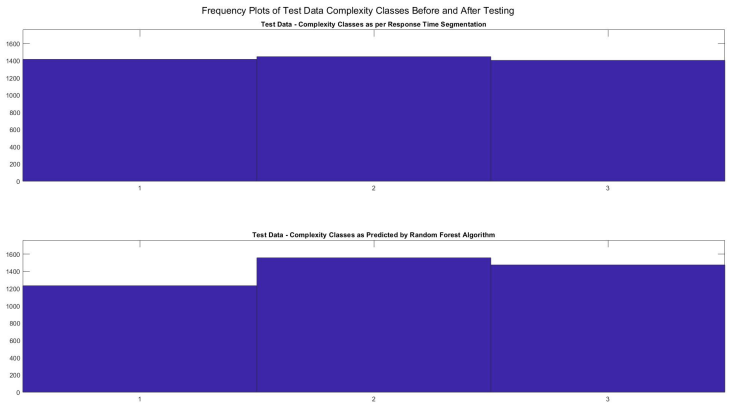


FIGURE 6.6: The distribution of the labels calculated from the inward log segmentation of reaction times into 3 labels (top graph) and the distribution of the predicted complexity classes (bottom graph)

The example logarithmic segmentation paradigm of figure 5.7 contains four levels; and was implemented as experiment 5 of table 6.1. A second logarithmic segmentation with three levels: 1 ('easy'), 2 ('medium') or 3 ('hard') where the levels were slightly separated from each other was implemented as **experiment 6** of table 6.1.

6.2 Prediction Accuracy for Logarithmically-Segmented Response Times Labels

The prediction accuracy in experiment 6 was **36 %** for the test dataset. 71 % of the misclassifications (45% of all the classifications) were "small errors" since they were incorrectly determined by only one level. Therefore, only 29 % of the misclassified tasks (19 % of all classifications) were considered "large errors" by misclassifying the test tasks by two levels. This occurred when the algorithm labelled/predicted a level 1 task as a level 3 task, or labelled/predicted a level 3 task as a level 1 task.

Experiments 7 and 8 used different classifiers and yielded prediction accuracies of 35 % and 37 % respectively. These two classifiers: the RUS Boost trees and linear SVM models, yielded the best prediction accuracies in the Matlab Classification Learner App, and were consequently listed in table 6.1. Both yielded similar results to the random forest predictions.

The results of experiments 9 and 10, which used the methodology of experiment 6: Logarithmic segmentation into 3 separated levels and a random forest classifier, conclude the response time related experiments in the table. Experiment 9 re-examined the possibility that the results from experiment 6 are misleading due to the existence of quantization effect - having too small a number of levels - or bias: The trained classifier was applied on the task instances that were "mid-level" between the three complexity levels. These mid-levels are marked by the purple regions in figure 5.8. The experiment aimed to gauge how well the classifier predicted whereabout on the response times' scale these instances fell.

As described in the methods, these tasks were labelled by "mid-level" labels, where if a task fell between level 1 and 2, it was labelled as 1.5, and if it fell between 2 and 3, it was labelled with 2.5. When the classifier was applied to these instances, 70 % of the predictions were 0.5 apart. In other words, for 70 % of the predictions, if the given task was 1.5 – a mid-level between 1 and 2 – it was predicted as either a 1 or 2, and if the task was 2.5 – a mid-level between 2 and 3 – it was predicted as either 2 or 3. The misclassifications, in an amount of 30 %, were the instances that were predicted as a 3 when the task fell between level 1 and 2, or predicted as a 1 when it fell between level 2 and 3.

The asterisk for this experiment in table 6.1 accentuates the fact that this experiment is different to the rest. For this experiment, the classifier will always predict a 1, 2 or 3 for each instance as these were the classes that it was trained with. These mid-level task instances, however, were labelled with the mid-level labels as 1.5 or 2.5. Therefore the percentage of "same class prediction" in table 6.1 for this scenario refers to test mid-levels where the predicted labels - 1, 2, or 3 - were only 0.5 apart from the labels applied to the mid-level tasks before testing - 1.5 or 2.5. The other 30 % of the classifications were those that were predicted 1.5 apart from the mid-level labels, and are not listed under the "one class apart recognition" column since they are considered pure misclassifications.

Experiment 10 was designed to test the performance of the algorithm as a binary classifier using the two extreme labels - easy (1) and hard (3). The correct class/level prediction rate was 56 %. In this case, all other incorrect predictions (44 %) were pure misclassifications.

6.3 Prediction Accuracy for Human Perception Labels

Due to the small prediction accuracy of experiments 1-10, which used complexity labels based on the dataset response times, experiment 11 attempted a different label source. The method with a small number of subjects as described under section 5.2.1.3 was implemented. These volunteers were presented with a sample of recognition tasks from the original dataset. They subjectively estimated the complexity on a 3-level scale: 1 ('easy'), 2 ('medium') and 3 ('hard').

The 10 subjects collectively completed 120 recognition tasks. A random forest classification model, once again executed with 3-fold validation, was applied to this very small dataset.

The predictions in this experiment yielded an accuracy of **58 %**. Moreover, another 40 % of the classifications yielded misclassifications of only one class. Therefore only 2 % of the classifications were misclassified by two levels.

6.4 Feature Importance

Feature importance was calculated as part of the *TreeBagger* function of the random forest implementation, using the configuration of experiment 6.

The feature importance results were grouped according to feature types and are illustrated in figure 6.7. The first seven subplots in the figure represent the object-specific features. The next five subplots then represent the whole image features. The importance values are shown using the same scale such that they are easily comparable.

The more positive the feature importance value, the more an impact it is predicted to have in the classification of visual tasks complexity according to the labels provided. The negative/very low values in the feature importance graphs suggest that those features have negligible, if any, contribution to the classification.

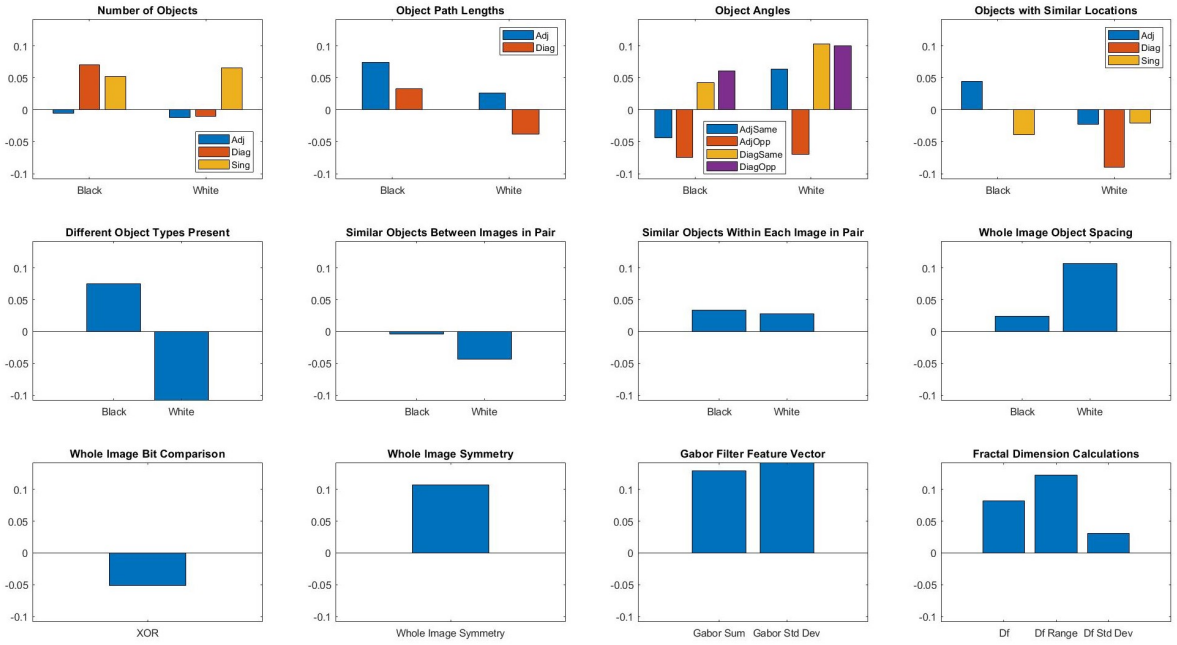


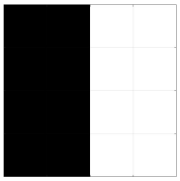
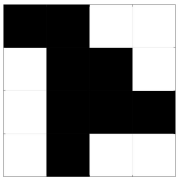
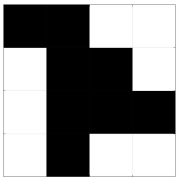
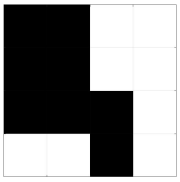
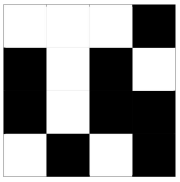
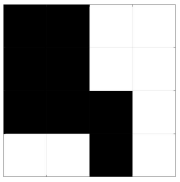
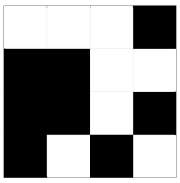
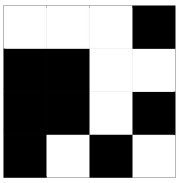
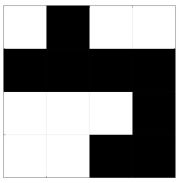
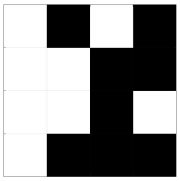
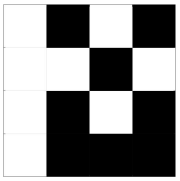
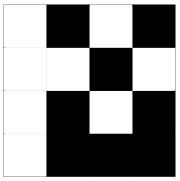
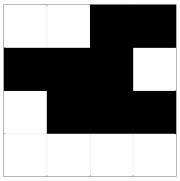
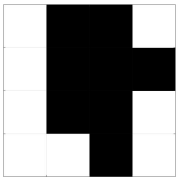
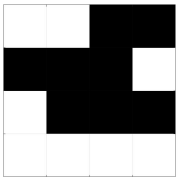
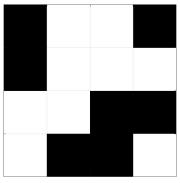
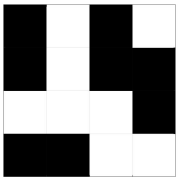
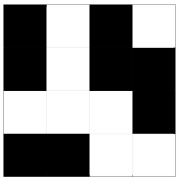
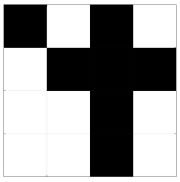
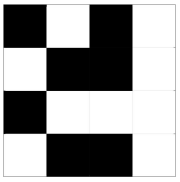
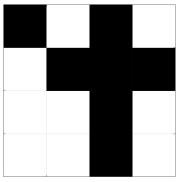
FIGURE 6.7: Importance of Relative-Features

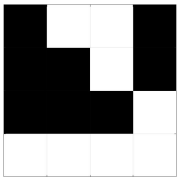
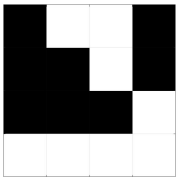
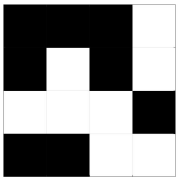
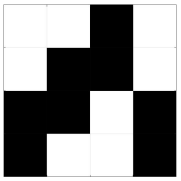
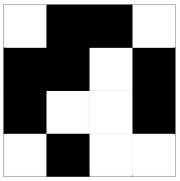
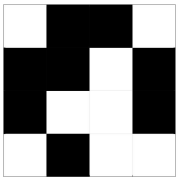
The feature importance values are further assessed below using a visual examination of nine examples - three from each complexity class. The examples were randomly selected from the test subset to illustrate and interpret the feature importance results of figure 6.7. The perceived complexities of all these tasks were correctly predicted by the model.

The nine examples are shown in table 6.2, where the tasks, their human perceived and algorithmically predicted complexity level are listed. While the algorithm was run on unique image pairs, the tasks in table 6.2 are shown with three images, the way they were presented to the subjects in the recognition task.

It is important to note that none of these features are independent. I.e. the presence, or lack of a single feature cannot independently predict the human perceived complexity level. However, the examples demonstrate trends in these features, that could explain why a feature was selected by the classifier as a strong or a weak predicting factor.

TABLE 6.2: Examples from the cognitive recognition tasks database which were correctly predicted by the algorithm as complexity levels 1, 2 and 3.

#	**Image 1**	**Image 2**	**Image 3**	**Complexity Level**
1				1
2				1
3				1
4				2
5				2
6				2
7				3

8				3
9				3

6.4.1 Object-Specific Feature Importance

In #2 and #3 of the "easy" (level 1) image pairs in table 6.2, only one of the images has **one** long black adjacent path. This makes it quite easily distinguishable from the other unique image in the task. This demonstrates the importance of the black adjacent path/s **length/s** shown by the feature importance graph (figure 6.7). The **number** of adjacent paths, however, is quite consistent in all tasks across the three complexity levels, with no specific trend. This could indicate why the **number** of (black and white) adjacent paths did not play a role in deducing and predicting the complexity level. This can be also related to the positive "Different (Black) Objects Present" feature: When one image in a pair had only one long black adjacent path and the other image had more black features, it was immediately predicted as easy (level 1); examples of this are the image pairs in #2 and #3. The"Different (White) Objects" feature, however, did not display the same characteristic. There appeared to be different white objects present within several image pairs across the different complexity classes, once again with no specific trend. This might explain the negative importance level of the "Different (White) Objects Present" feature in figure 6.7.

The similar object location features' importance values in figure 6.7 suggested that the objects locations did not play a significant role in the image observations. The only object type whose location yielded slight importance was the **black adjacent path**. In tasks

#5, #6 and #9 of table 6.2, the presence of a similarly located black adjacent path in both images could have made the distinguishing between the images in the task harder, thus increasing the perceived complexity in some way.

It appears that while similarly located objects held visual significance, the similar **shapes**, especially those occurring between the images of the task, held small visual significance.

Finally, in terms of the object-specific features, the black spacing feature did not hold considerable importance in comparison to the white spacing feature. In #2 in table 6.2, the average distance between the **white** objects for image 1 (and image 3) is noticeably greater than that for image 2. This is because there is only one long black adjacent path in the middle of image 1, whereas there are smaller black objects dispersed around image 2. This noticeable difference in the white spacing could have also justified the 'easy' (level 1) classification. Except for the average white spacing feature, most **white** object-specific features have relatively weak significance as illustrated in figure 6.7.

6.4.2 Whole Image Feature Importance

The feature of whole image, direct, bit-by-bit comparison had a negative importance in the complexity predictions (figure 6.7). This feature had values randomly ranging from 4 to 12 for the examples in table 6.2, and had no correlation to the human task complexity labels.

The symmetry features demonstrated significance in task complexity prediction (figure 6.7). As an example, this is illustrated in the first easy (level 1) image pair in table 6.2. Image 1 is very clearly symmetrical across the x axis, and is also symmetrical across the y axis, in inverted colours - as described in the various symmetry checks in chapter 5. The presence of this symmetry can explain the image as perceptually easier to discriminate in the recognition task.

The last two subplots on the feature importance graph (figure 6.7) illustrate the importance of the mathematical features. These features are the sum of the output Gabor feature vector, the standard deviation of the output Gabor feature vector, the calculated

fractal dimension, the range of the different fractal dimension values observed through the box-counting implementation, and the standard deviation of the same range of fractal dimensions values. These features were calculated for each image from the unique pair in the recognition task, and then a difference was computed. An observation of the values of these calculated differences for the examples in table 6.2 yielded that, on average, all five of these measures are **reduced** with increasing complexity. This trend can explain the positive feature importance levels for all these features in the last two subplots of figure 6.7.

The values for all features for the example tasks in table 6.2 are presented in Appendix A.

6.5 Summary

Several sets of results were obtained through different trials based on several response times' segmentation paradigms, and several machine learning techniques. The first 10 experiments were based on the initial and central assumption of this research: that the response times correlate, or represent, the human perceived complexity of visual recognition tasks.

The best methods configuration of experiment 6 yielded a prediction accuracy of 36% for a 3 classes' machine learner. Experiment 11, which used the same configuration with direct human labels of visual complexity, showed a significantly higher prediction accuracy of 58 %. The feature importance results were presented and assessed using both machine learning selection and manual examination of randomly selected examples from the test dataset.

These findings will be discussed in further detail in the following chapter.

CHAPTER 7

Discussion and Conclusion

This chapter summarises the key findings of the research, and suggests future improvements based on the collective results from this study.

7.1 Response Time Labels for Human-Perceived Visual Complexity

The final methodology chosen in this study was the one of experiment 6 in table 6.1, which used labels based on logarithmic segmentation of the response times. While the linear segmentation paradigms yielded higher prediction accuracies, the logarithmic segmentation models yielded results that were more reliable, and less biased. This choice is also in line with the fact that human perception in all senses is inherently logarithmic in nature [16].

Although the prediction accuracy from the final experiment of choice was relatively low, the analysis of the misclassifications, and the tests on the mid-levels in experiment 9 suggest that the algorithm is heading in the right direction. The results could possibly be improved in future studies by examining factors that overlap across classes computed from human response times.

7.2 Classifier Choice

Several machine learning models were explored to verify that the low rate of correct predictions was not due to the classifier choice - the random forest. The best performer

from the tested classifiers - the linear SVM's prediction accuracy was just slightly greater (< 1 %) than the initial random forest model implemented in this research.

The differences between the prediction accuracies of the various classifiers were not significant. If anything, the best classifiers tested in experiments 9 and 10 had a percentage of "large errors" which was slightly greater than the random forest. These results imply that the random forest was a valid choice for this classification. However, for future work, a **non-linear** (SVM) model could be attempted, since it may be more suited to this problem as the process deals with logarithmic labels.

7.3 Critical Analysis of the Response Times as Labels

The accuracy of a classifier's predictions will always be limited by the training data provided to it. In this research the training data included both features and labels.

The results detailed in the previous chapter imply a critical deficiency in the use of response times as labels. This research was based on an assumed correlation between response times and perceived complexity. The results suggest that other factors could have impacted the speed of response, aside from perceived difficulty. Although age, gender and computer skill were examined and ruled out as having no significant effect on the response time distributions, the response times could still be affected by factors such as stress, distraction, boredom or fatigue of subjects taking the cognitive assessments.

Another practical example in which response times may not be a suitable measure of perceived difficulty, is a scenario where a tester happens to provide cognitive tasks in incrementally increasing complexities. In this context, it is possible that a subject's response times could be similar, or not drastically different, as he/she gets used to the tasks. Based on the initial assumption in this research, which shaped the methodology developed, visual tasks which have similar response times would be segmented under the same complexity class. In the slowly incrementing complexity scenario, however, the complexities of the tasks did vary, and similar response times indicate the subject simply getting accustomed to the tasks provided.

The results could also be interpreted as a correlation still existing between subjects response times and their perceived visual tasks complexities, but that the segmentation models implemented in this research were not able to correctly capture this correlation. However, there is no way of independently testing the correctness of these segmentation paradigms.

This motivates further future research into accurately establishing if, and how, response times correlate to perceived complexity to create a more definitive, less volatile model of human responses to cognitive tasks.

Another possible future solution is to employ the method developed here with **direct** human perception labels. This was preliminarily addressed in this work, and examined in experiment 11. The predictions based on subjects' complexity labels estimation were far more accurate than those involving the response time labels.

The goal of this research was to develop an algorithm that can gauge visual complexity as closely as possible to the way humans intuitively would. Therefore, for this algorithm to learn human perception trends, other labels providing the human perceived complexity levels should be explored. A transformation from task response times into task perceived complexities may be very complex and need new experiments and studies. Different labels that capture human perceived complexity may also eliminate the segmentation needed to convert a continuous variable, like response time, into discrete labels.

The use of different labels improved the prediction accuracy in this study and suggests that the methodology has potential that can be explored more thoroughly in future studies. It is important to note that the data size in the new labels experiment was very small, and therefore is not conclusive and can only suggest a trend. Additionally, the new human perception labels are also subjective. This experiment could therefore only serve as a feasibility proof, and provide some insight into the plausibility of pursuing this concept of new labels more thoroughly in the future.

Another issue to be mindful of is that when presenting tests designed to diagnose cognitive decline on individuals with non-impaired cognition, it is likely that all images of this type are easily distinguishable for these individuals, making all the recognition tasks "easy".

Consequently, there may be a bias in these individuals subjectively labelling tasks as hard (level 3).

7.4 Feature Importance Analysis

The extracted features that were detailed in chapter 5 were based on the previously reviewed literature on computational complexity of images. The resulting importance numbers, as illustrated in figure 6.7, provided valuable insight into their significance in human perception tasks. The positive feature importance in the context of machine learning, and in visual observations in the study, implies that the presence of these features had a relation to the human perceived complexity.

The finding of importance of the number of black adjacent paths is in line with Attneave's theories on visual redundancy, where long strings of adjoining pixels of the same colour could constitute for a level of visual redundancy, thus simplifying the visual field's perceived complexity [6].

The result that the black spacing feature had no substantial importance may be due to the fact that there are many instances in the dataset with only one (large) black object, making the overall/average black object spacing in the dataset a zero. This hinders the ability to find a trend in the image pairs' black object spacing calculations. Nevertheless, the average spacing of the **white** objects was found to be important, implying that there are more black objects present in the image. This was corroborated by the observation that in the examples of table 6.2, the feature increased the perceived complexity of the image pairs. This justifies the classifier's computed importance of the white object spacing feature. Even with the presence of only one black adjacent path in an image, the black path tends to separate the white space, thus providing an average white object distance measure, and another indication of the black objects' placement.

If the white object distance measure truly indicates black objects' placement, then this, along with the fact that most of the other white object-specific features were weak, could imply that during visual observations, the white objects are commonly considered as

background space. This suggests that individuals do not typically notice specific white-object details.

Most whole image features had greater significance than the object-specific features. The only whole image feature which had a negative importance value was the bit-by-bit whole image comparison technique. This result could suggest that this "typical computer" way of scanning an image and comparing images is not a human perception model.

The significant importance of image symmetry in the complexity predictions, is once again in line with Attneave's premise on image symmetry representing another form of visual redundancy, thus reducing the visual observation's perceived complexity [6]. It indicates that when subjects observe an image that has symmetry, the mirroring makes it simpler to distinguish between other, non-symmetrical images.

Finally, the positive importance values of the Gabor and fractal dimension features imply their relation to visual perception as described in the background studies of chapter 3 [15] [17] [27] [31]. These feature importance values, and the trends showed that the smaller the **difference** in the values of these features for each of the two images in the pair, the greater the perceived complexity. This is also in line with the assumption of this research that the smaller the difference between the images in the task, the harder is the recognition of the odd one.

The feature importance trends found in the study suggest that it is possible to find a correlation between human visual perception, and the complexity calculated from mathematical models. The mathematical calculations are also likely to be more consistent and objective, and may be applicable to a wider diversity of subjects, whereas features based on **intuitive** observations, may be more likely to fluctuate between individuals.

Whole image features were found to be, in general, more significant for the complexity prediction than the object-specific features. Based on Witkin's theories, this could imply that subjects might lean slightly more towards field dependency by assessing their visual fields as a whole, and making only loose partitions of the presented visual field [9]. This tendency could also be due to the small size and simplicity of the images in the dataset, where individuals may not need to look for details at a finer level.

7.5 More Future Considerations

After obtaining the results, the dataset was further inspected to understand the range of tasks previously provided during these assessments. It was discovered that the tasks provided in the dataset were limited in terms of their variety and range of possible square binary images presented therein. All images from the tasks had either 7, or 8 **black** squares out of the possible 16 squares. This suggests that these are the type of images that are more likely to be employed in cognitive tasks. Testers might not have previously presented other, more extreme, images as these could be considered as 'too easy'.

Since the algorithm has been trained and tested on only this range of images, it becomes more difficult to learn/predict the trends with more 'extreme' images (such as mostly white or mostly black images). Additionally, the features chosen in this study aimed to capture a variety of visual characteristics. The possibility of finding that same visual variety in this dataset might be limited if there are only images with 7 or 8 black squares. Therefore, the features that were weak in terms of importance during the predictions in this study may be more significant in a greater assortment of images.

Another point to take into consideration is that some features should have also been 'relaxed', as was done with the image symmetry feature, so that they could possibly apply to more of the image pairs in this dataset, and more human observations. This could potentially be relevant to the detection of similar objects (within an image **and** between the two images) as well, because the algorithm does not currently pick up if objects are 'almost' the same. This could also explain why all the similar object related features had low/negative importance levels in the feature importance graphs - the prevalence of exactly the same shapes occurring was not high enough in the dataset to be able to gauge its importance.

7.6 Conclusion

The results in this research show that computational features have the potential to provide a visual complexity measure to human performance in visual recognition tasks. The features extracted in the study provided an insight into visual factors that affect human visual perception of small abstract images.

While the prediction accuracy of the final algorithm was low, the error analysis and feature analysis demonstrate that this model has the potential to be further developed into a computational complexity measurement paradigm, which can be related to the human way of perceiving visual complexity. This study could therefore serve as a proof of concept, and provide a preliminary methodology and set of features that could be applied to other visual tasks involving similar images.

In future implementations, the human perceived complexity labels will require further attention. It is important to note that these results will never be 100 % accurate because perceived complexity can be further affected by factors such as semantic memory, psychological thoughts, or cultural background, which are harder to quantify. However the results can still be improved in the future by experimenting with different methods of defining human-perceived complexity labels.

As per the research question, a complexity calculation method combining information theory, machine vision and human perception measures was successfully designed and developed. The results indicate correlations that could help bridge the gap between human perceived complexity, and mathematically defined complexity. The resulting feature importance values have assisted in explaining human visual perception in the context of cognitive tests. Although the prediction accuracy of the final method was low, the analysis of the results indicate that this complexity measure has the potential to evaluate visual complexity in the context of cognitive tests.

References

[1] Vered Aharonson. Complexity by compression. *Perception and Psychophysics*, June 2016.

[2] Vered Aharonson, Ilan Halperin, and Amos D Korczyn. Computerized diagnosis of mild cognitive impairment. *Alzheimer's & Dementia*, 3(1):23–27, 2007.

[3] Vered Aharonson and Amos D Korczyn. Human–computer interaction in the administration and analysis of neuropsychological tests. *Computer methods and programs in biomedicine*, 73(1):43–53, 2004.

[4] AS Ahmad, MY Hassan, MP Abdullah, HA Rahman, F Hussin, H Abdullah, and R Saidur. A review on applications of ann and svm for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews*, 33:102–109, 2014.

[5] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.

[6] Fred Attneave and Malcolm D Arnoult. The quantitative study of shape and pattern perception. *Pattern Recognition edited by L. Uhr, John Wiley*, 1956.

[7] Kanaka Babshet, Catherine Honegger, Ashley Gritzman, and Vered Aharonson. Visual pattern complexity determination for enhanced usability in cognitive testing. In *International Conference on Applied Human Factors and Ergonomics*, pages 195–203. Springer, 2017.

[8] Min Jae Baek, Karyeong Kim, Young Ho Park, and SangYun Kim. The validity and reliability of the mini-mental state examination-2 for detecting mild cognitive impairment and alzheimer's disease in a korean population. *PloS one*, 11(9):e0163792, 2016.

[9] Marios Belk, Christos Fidas, Panagiotis Germanakos, and George Samaras. The interplay between humans, technology and user authentication: A cognitive processing perspective. *Computers in Human Behavior*, 76:184–200, 2017.

[10] Martin Bouda, Joshua S Caplan, and James E Saiers. Box-counting dimension revisited: presenting an efficient method of minimizing quantization error and an assessment of the self-similarity of structural root systems. *Frontiers in plant science*, 7:149, 2016.

[11] Alan C Bovik. *Handbook of image and video processing*. Academic press, 2010.

[12] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[13] Cognitive. General cognitive assessment battery (cab), June 2016. `https://www.cognifit.com/cognitive-assessment/cognitive-test`.

[14] Cognivue. How cognivue works, June 2018. `https://www.cognivuesystems.com/product-tour/how-cognivue-works/`.

[15] John G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169, 1985.

[16] Stanislas Dehaene. The neural basis of the weber–fechner law: a logarithmic mental number line. *Trends in cognitive sciences*, 7(4):145–147, 2003.

[17] Yuval Fisher. *Fractal image compression: theory and application*. Springer Science & Business Media, 2012.

[18] Marshal F Folstein, Susan E Folstein, and Paul R McHugh. "mini-mental state": a practical method for grading the cognitive state of patients for the clinician. *Journal of psychiatric research*, 12(3):189–198, 1975.

[19] Food for the Brain. Food for the brain cognitive function test, January 2015. `https://cft.foodforthebrain.org/`.

[20] Kayhan Foroutan-pour, Pierre Dutilleul, and Donald L Smith. Advances in the implementation of the box-counting method of fractal dimension estimation. *Applied mathematics and computation*, 105(2-3):195–210, 1999.

[21] Jesper Juul Henriksen. 3d surface tracking and approximation using gabor filters. *South Denmark University*, 28, 2007.

[22] Joni-Kristian Kamarainen. Gabor features in image analysis. In *2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 13–14. IEEE, 2012.

[23] H. Li and K. N. Ngan. A co-saliency model of image pairs. *IEEE Transactions on Image Processing*, 20(12):3365–3375, Dec 2011.

[24] S. Liu. An improved differential box-counting approach to compute fractal dimension of gray-level image. In *2008 International Symposium on Information Science and Engineering*, volume 1, pages 303–306, Dec 2008.

[25] R Mahendran, GC Jayashree, and K Alagusundaram. Application of computer vision technique on sorting and grading of fruits and vegetables. *J. Food Process. Technol*, 10:2157–7110, 2012.

[26] W. Man, Y. Ji, and Z. Zhang. Image classification based on improved random forest algorithm. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 346–350, April 2018.

[27] Stjepan Marčelja. Mathematical description of the response of simple cortical cells. *Journal of the Optical Society of America*, 70:1297–300, 12 1980.

[28] Matthew W Mitchell. Bias of the random forest out-of-bag (oob) error for certain input parameters. *Open Journal of Statistics*, 1(03):205, 2011.

[29] Ruth E Nieuwenhuis-Mark. The death knoll for the mmse: has it outlived its purpose? *Journal of geriatric psychiatry and neurology*, 23(3):151–157, 2010.

[30] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, Aug 2000.

[31] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[32] Chava Peretz, Amos D Korczyn, Evelyn Shatil, Vered Aharonson, Smadar Birnboim, and Nir Giladi. Computer-based, personalized cognitive training versus classical computer games: a randomized double-blind prospective trial of cognitive stimulation. *Neuroepidemiology*, 36(2):91–99, 2011.

[33] Jukka Perkiö and Aapo Hyvärinen. Modelling image complexity by independent component analysis, with application to content-based image retrieval. *Artificial Neural Networks–ICANN 2009*, pages 704–714, 2009.

[34] A. F. R. Rahman and M. C. Fairhurst. Measuring classification complexity of image databases: a novel approach. In *Proceedings 10th International Conference on Image Analysis and Processing*, pages 893–897, 1999.

[35] Christina Shen, Florin C Popescu, Eric Hahn, Tam TM Ta, Michael Dettling, and Andres H Neuhaus. Neurocognitive pattern analysis reveals classificatory hierarchy of attention deficits in schizophrenia. *Schizophrenia bulletin*, 40(4):878–885, 2013.

[36] Y. Yokota, Y. Igarashi, M. Okada, and Y. Naruse. Estimation of the reaction times in tasks of varying difficulty from the phase coherence of the auditory steady-state response using the least absolute shrinkage and selection operator analysis. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6662–6666, Aug 2015.

[37] Dingwen Zhang, Huazhu Fu, Junwei Han, Ali Borji, and Xuelong Li. A review of co-saliency detection algorithms: fundamentals, applications, and challenges. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(4):38, 2018.

Appendix A

This appendix presents all the calculated features for the nine example tasks listed in table 6.2 of dissertation titled: *A visual complexity learning algorithm for modelling human performance in visual cognitive tests.*

Each subsequent page tabulates the measured features for **one** of the tasks from table 6.2.

Task #1 in table 6.2 ‖ Complexity level = 1			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	1	1	1
Number of black diagonal paths	0	0	0
Number of black single blocks	0	0	0
Number of white adjacent paths	1	3	3
Number of white diagonal paths	0	0	0
Number of white single blocks	0	0	0
Lengths of black adjacent paths	8	8	8
Lengths of black diagonal paths	0	0	0
Lengths of white adjacent paths	8	8	8
Lengths of white diagonal paths	0	0	0
Number of black adjacent paths at same angle	1		
Number of black adjacent paths at opp. angle	1		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	1		
Number of white adjacent paths at opp. angle	0		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	0		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	0		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	0		
White different object types present	0		
Number of similar black objects between images in pair	0		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	0	0
Number of similar white objects within image	0	0	0
Whole image black object spacing	0	0	0
Whole image white object spacing	0	378.77	378.77
Whole image bit-by-bit comparison	6		
Whole image symmetry	1	0	0
Gabor feature vector sum	2030.3	2542.7	2542.7
Gabor feature vector standard deviation	0.0088	0.0099	0.0099
Fractal dimension (overall)	1.8499	1.7700	1.7700
Fractal dimension range	0.6781	0.6943	0.6943
Fractal dimension standard deviation	0.2480	0.2820	0.2820

Task #2 in table 6.2 ‖ Complexity level = 1			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	1	2	1
Number of black diagonal paths	0	0	0
Number of black single blocks	0	2	0
Number of white adjacent paths	2	1	2
Number of white diagonal paths	0	0	0
Number of white single blocks	0	3	0
Lengths of black adjacent paths	8	6	8
Lengths of black diagonal paths	0	0	0
Lengths of white adjacent paths	8	5	8
Lengths of white diagonal paths	0	0	0
Number of black adjacent paths at same angle	0		
Number of black adjacent paths at opp. angle	0		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	0		
Number of white adjacent paths at opp. angle	0		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	0		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	0		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	1		
White different object types present	1		
Number of similar black objects between images in pair	0		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	1	0
Number of similar white objects within image	0	3	0
Whole image black object spacing	0	337.79	0
Whole image white object spacing	379.08	335.63	379.08
Whole image bit-by-bit comparison	10		
Whole image symmetry	0	0	0
Gabor feature vector sum	2320.1	2786.9	2320.1
Gabor feature vector standard deviation	0.0095	0.0104	0.0095
Fractal dimension (overall)	1.7971	1.7860	1.7971
Fractal dimension range	0.7388	0.5913	0.7388
Fractal dimension standard deviation	0.2809	0.2427	0.2809

Task #3 in table 6.2 ‖ Complexity level = 1			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	1	1	1
Number of black diagonal paths	1	1	0
Number of black single blocks	1	1	0
Number of white adjacent paths	1	1	2
Number of white diagonal paths	0	0	0
Number of white single blocks	2	2	1
Lengths of black adjacent paths	5	5	8
Lengths of black diagonal paths	2	2	0
Lengths of white adjacent paths	6	6	7
Lengths of white diagonal paths	0	0	0
Number of black adjacent paths at same angle	1		
Number of black adjacent paths at opp. angle	1		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	0		
Number of white adjacent paths at opp. angle	0		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	0		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	0		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	1		
White different object types present	0		
Number of similar black objects between images in pair	0		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	0	0
Number of similar white objects within image	2	2	0
Whole image black object spacing	352.94	352.94	0
Whole image white object spacing	316.72	316.72	354.94
Whole image bit-by-bit comparison	8		
Whole image symmetry	0	0	0
Gabor feature vector sum	2653.4	2653.4	2525.3
Gabor feature vector standard deviation	0.0101	0.0101	0.0099
Fractal dimension (overall)	1.7800	1.7800	1.7999
Fractal dimension range	0.8317	0.8317	0.6026
Fractal dimension standard deviation	0.3017	0.3017	0.2240

Task #4 in table 6.2 ‖ Complexity level = 2			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	1	1	1
Number of black diagonal paths	0	1	1
Number of black single blocks	1	0	0
Number of white adjacent paths	1	1	1
Number of white diagonal paths	0	1	1
Number of white single blocks	2	0	0
Lengths of black adjacent paths	7	5	5
Lengths of black diagonal paths	0	3	3
Lengths of white adjacent paths	6	5	5
Lengths of white diagonal paths	0	3	3
Number of black adjacent paths at same angle	1		
Number of black adjacent paths at opp. angle	1		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	1		
Number of white adjacent paths at opp. angle	1		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	0		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	1		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	1		
White different object types present	1		
Number of similar black objects between images in pair	0		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	0	0
Number of similar white objects within image	1	0	0
Whole image black object spacing	301.60	302.53	302.53
Whole image white object spacing	320.32	289.84	289.84
Whole image bit-by-bit comparison	4		
Whole image symmetry	0	0	0
Gabor feature vector sum	2572.9	2797.6	2797.6
Gabor feature vector standard deviation	0.0100	0.0104	0.0104
Fractal dimension (overall)	1.8194	1.8031	1.8031
Fractal dimension range	0.5519	0.4955	0.4955
Fractal dimension standard deviation	0.1953	0.2067	0.2067

Task #5 in table 6.2 ‖ Complexity level = 2			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	1	1	1
Number of black diagonal paths	0	0	0
Number of black single blocks	0	0	0
Number of white adjacent paths	2	2	2
Number of white diagonal paths	0	0	0
Number of white single blocks	1	0	1
Lengths of black adjacent paths	8	8	8
Lengths of black diagonal paths	0	0	0
Lengths of white adjacent paths	7	7	7
Lengths of white diagonal paths	0	0	0
Number of black adjacent paths at same angle	1		
Number of black adjacent paths at opp. angle	1		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	0		
Number of white adjacent paths at opp. angle	0		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	1		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	0		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	1		
Black different object types present	0		
White different object types present	0		
Number of similar black objects between images in pair	0		
Number of similar white objects between images in pair	1		
Number of similar black objects within image	0	0	0
Number of similar white objects within image	0	0	0
Whole image black object spacing	0	0	0
Whole image white object spacing	361.68	388.15	361.68
Whole image bit-by-bit comparison	6		
Whole image symmetry	0	0	0
Gabor feature vector sum	2545.9	2504.9	2545.9
Gabor feature vector standard deviation	0.0100	0.0099	0.0100
Fractal dimension (overall)	1.7702	1.7694	1.7702
Fractal dimension range	0.6943	0.6998	0.6943
Fractal dimension standard deviation	0.2825	0.2827	0.2825

Task #6 in table ‖ Complexity level = 2			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	2	3	3
Number of black diagonal paths	0	0	0
Number of black single blocks	1	0	0
Number of white adjacent paths	1	1	1
Number of white diagonal paths	0	0	0
Number of white single blocks	1	1	1
Lengths of black adjacent paths	6	8	8
Lengths of black diagonal paths	0	0	0
Lengths of white adjacent paths	8	7	7
Lengths of white diagonal paths	0	0	0
Number of black adjacent paths at same angle	1		
Number of black adjacent paths at opp. angle	0		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	1		
Number of white adjacent paths at opp. angle	1		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	1		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	0		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	1		
White different object types present	0		
Number of similar black objects between images in pair	3		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	2	2
Number of similar white objects within image	0	0	0
Whole image black object spacing	381.40	352.72	352.72
Whole image white object spacing	330.56	324.84	324.84
Whole image bit-by-bit comparison	7		
Whole image symmetry	1	0	0
Gabor feature vector sum	2748.2	2571.6	2571.6
Gabor feature vector standard deviation	0.0102	0.0100	0.0100
Fractal dimension (overall)	1.8000	1.7919	1.7919
Fractal dimension range	0.7631	0.6645	0.6645
Fractal dimension standard deviation	0.2851	0.2628	0.2628

Task #7 in table 6.2 ‖ Complexity level = 3			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	1	2	1
Number of black diagonal paths	0	0	0
Number of black single blocks	1	2	1
Number of white adjacent paths	2	1	2
Number of white diagonal paths	0	1	0
Number of white single blocks	2	1	2
Lengths of black adjacent paths	6	5	6
Lengths of black diagonal paths	0	0	0
Lengths of white adjacent paths	7	6	7
Lengths of white diagonal paths	0	2	0
Number of black adjacent paths at same angle	0		
Number of black adjacent paths at opp. angle	0		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	0		
Number of white adjacent paths at opp. angle	0		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	0		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	1		
Number of white adjacent paths with similar locations	0		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	0		
White different object types present	1		
Number of similar black objects between images in pair	0		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	2	0
Number of similar white objects within image	2	0	2
Whole image black object spacing	321.09	298.87	321.09
Whole image white object spacing	356.44	342.65	356.44
Whole image bit-by-bit comparison	4		
Whole image symmetry	0	0	0
Gabor feature vector sum	2869.0	3004.8	2869.0
Gabor feature vector standard deviation	0.0105	0.0107	0.0105
Fractal dimension (overall)	1.7845	1.7881	1.7845
Fractal dimension range	0.5934	0.6282	0.5934
Fractal dimension standard deviation	0.2509	0.2491	0.2509

Task #8 in table 6.2 ‖ Complexity level = 3			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	2	2	2
Number of black diagonal paths	0	0	0
Number of black single blocks	0	0	1
Number of white adjacent paths	2	2	2
Number of white diagonal paths	0	0	0
Number of white single blocks	0	0	0
Lengths of black adjacent paths	8	8	7
Lengths of black diagonal paths	0	0	0
Lengths of white adjacent paths	8	8	8
Lengths of white diagonal paths	0	0	0
Number of black adjacent paths at same angle	0		
Number of black adjacent paths at opp. angle	0		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	0		
Number of white adjacent paths at opp. angle	0		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	0		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	0		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	1		
White different object types present	0		
Number of similar black objects between images in pair	1		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	0	0
Number of similar white objects within image	0	0	0
Whole image black object spacing	330.98	330.98	351.63
Whole image white object spacing	329.74	329.74	299.40
Whole image bit-by-bit comparison	12		
Whole image symmetry	0	0	0
Gabor feature vector sum	2486.4	2486.4	2558.8
Gabor feature vector standard deviation	0.0098	0.0098	0.0099
Fractal dimension (overall)	1.7966	1.7966	1.7968
Fractal dimension range	0.7355	0.7355	0.7371
Fractal dimension standard deviation	0.2742	0.2742	0.2734

Task #9 in table 6.2 ‖ Complexity level = 3			
Features	**Image 1**	**Image 2**	**Image 3**
Number of black adjacent paths	2	2	2
Number of black diagonal paths	0	0	0
Number of black single blocks	0	1	1
Number of white adjacent paths	3	1	1
Number of white diagonal paths	0	0	0
Number of white single blocks	0	3	3
Lengths of black adjacent paths	8	7	7
Lengths of black diagonal paths	0	0	0
Lengths of white adjacent paths	8	5	5
Lengths of white diagonal paths	0	0	0
Number of black adjacent paths at same angle	1		
Number of black adjacent paths at opp. angle	0		
Number of black diagonal paths at same angle	0		
Number of black diagonal paths at opp. angle	0		
Number of white adjacent paths at same angle	0		
Number of white adjacent paths at opp. angle	0		
Number of white diagonal paths at same angle	0		
Number of white diagonal paths at opp. angle	0		
Number of black adjacent paths with similar locations	0		
Number of black diagonal paths with similar locations	0		
Number of black single blocks with similar locations	0		
Number of white adjacent paths with similar locations	1		
Number of white diagonal paths with similar locations	0		
Number of white single blocks with similar locations	0		
Black different object types present	1		
White different object types present	1		
Number of similar black objects between images in pair	1		
Number of similar white objects between images in pair	0		
Number of similar black objects within image	0	0	0
Number of similar white objects within image	2	3	3
Whole image black object spacing	298.50	312.42	312.42
Whole image white object spacing	351.82	395.94	395.94
Whole image bit-by-bit comparison	8		
Whole image symmetry	0	0	0
Gabor feature vector sum	2638.4	2736.0	2736.0
Gabor feature vector standard deviation	0.0101	0.0103	0.0103
Fractal dimension (overall)	1.7866	1.7696	1.7696
Fractal dimension range	0.5900	0.6929	0.6929
Fractal dimension standard deviation	0.2499	0.2743	0.2743

Appendix B

The following ethics clearance certificate was obtained for re-use of the dataset from NexSig's computerized cognitive testing studies in this research:

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

R14/49 Miss Kanaka Babshet

HUMAN RESEARCH ETHICS COMMITTEE (MEDICAL)

CLEARANCE CERTIFICATE NO. M180414

NAME: (Principal Investigator)	Miss Kanaka Babshet
DEPARTMENT:	Electrical and Information Engineering
PROJECT TITLE:	Complexity computation of patterns in neurophysiological tests
DATE CONSIDERED:	04/05/2018
DECISION:	Approved unconditionally
CONDITIONS:	
SUPERVISOR:	Prof Vered Aharonson

APPROVED BY: Prof C Penny, Chairperson, HREC (Medical)

DATE OF APPROVAL: 09/05/2018

This clearance certificate is valid for 5 years from date of approval. Extension may be applied for.

DECLARATION OF INVESTIGATORS

To be completed in duplicate and **ONE COPY** returned to the Research Office Secretary in Room 301, Third floor, Faculty of Health Sciences, Phillip Tobias Building, 29 Princess of Wales Terrace, Parktown, 2193, University of the Witwatersrand. I/we fully understand the conditions under which I am/we are authorized to carry out the above-mentioned research and I/we undertake to ensure compliance with these conditions. Should any departure be contemplated, from the research protocol as approved, I/we undertake to resubmit the application to the Committee. **I agree to submit a yearly progress report**. The date for annual re-certification will be one year after the date of convened meeting where the study was initially reviewed. In this case, the study was initially reviewed in April and will therefore be due in the month of April each year. Unreported changes to the application may invalidate the clearance given by the HREC (Medical).

Principal Investigator Signature

10/05/2018
Date

PLEASE QUOTE THE PROTOCOL NUMBER IN ALL ENQUIRIES