Species Recognition for Pest-Control for Cats and Possums

Thelasinghe Arachchige Stanis Achala Perera

A thesis submitted to
Auckland University of Technology
in fulfilment for the degree
of Doctor of Philosophy

2019

School of Engineering

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning, except where due acknowledgement is made in the acknowledgements.

Thelasinghe Arachchige Stanis Achala Perera

March 2019

Abstract

In New Zealand all land mammals (except bats) were introduced through human migration. These animals mostly came to the country as pets, or were introduced for economic reasons. Currently there is a huge problem with these introduced species. Currently species recognition is one of the key methods to understand and control certain animal in an area. This thesis investigates three existing techniques for recognising possum and cat. These techniques Eigenface, Fisherface and Support Vector Machine (SVM) are a novel application in animal recognition domain. When these techniques are trialled with possum and cat images, recognition rates are not acceptable for application. To improve the recognition rate a few methods are investigated. They are different colour schemes, different image resolutions and finally an error weight-based algorithm to measure the distance between average image and test image for the Eigenface technique. This developed technique produced favourable results for possums and cats detection. This new technique is compared with typical Eigenface, Fisherface and SVM techniques to investigate its performance. To further investigate, the developed algorithm is tested with dogs to check the performance with other animal species. Results show that there is an acceptable separation, with this multiclass problem. The performance (class separation) is analysed using the Receiver Operating Characteristic (ROC) method. This method provides a unique way to compare the class separation of the above three techniques.

Finally, a ROC-based feature selection method is developed to use with Principal Component Analysis (PCA), Fisherface and SVM techniques. This new technique helps to find the optimal dataset for training the above techniques. Lastly one of the main challenges to this investigation was the limited availability of training images for the algorithms. Due to the prohibitive cost of animal ethics approval, the training images were obtained from the internet. Hence all developed identification algorithms were optimised for small training sets.

Acknowledgements

In completing this research, I received help from several people whom I would now like to thank.

First, I must thank my supervisory team Dr Jeff Kilby, Dr John Collins and Professor Peter Chong for their support, guidance and advice during this investigation. Also, I would like to give special thanks to Dr Garry Tee for his help.

Secondly, I would like to thank my son Dyon, who gave me enough courage to finish this work reminding me every 30 minutes: "*Dad enough, let's go outside and play.*"

I would also like to thank my daughter Deanne and my loving wife Madu for their patience and support during this period.

Finally, thanks go to my Mum and Dad (Nimali and Titus) and my friends and family for their support during my research.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
FLDA	Fisher Linear Discriminant Analysis
HUM	Hypervolume Under the Manifold
LDA	Linear Discrimination Analysis
PCA	Principal Component Analysis
ROC	Receiver Operator Characteristics
SVD	Singular Value Decomposition
SVM	Support Vector Machine

List of Symbols

m_x	Mean
C	Covariance Matrix
λ	Eigen Values
e	Eigen Vectors
Φ	Centred Face
S_W	Within Class Scatter Matrix
S_B	Between Class Scatter Matrix
H	Hyper Plane
σ^2	Variance
e_d	l^2 or Euclidean Distance
σ	Standard Deviation
f_n	Eigenface
w	Weights Vectors

Chapter 1
Introduction

1.1 Research Motivation

The research presented in this thesis was motivated by a desire to develop a species recognition algorithm for resettable traps using species identification-techniques in New Zealand. Classical pest-control techniques are currently used to identify the pest population in New Zealand forests. The New Zealand Department of Conservation (NZ-DOC) aims to identify pests before setting the traps, and to collect pest-population information regarding certain types of pests. The main objective of this research was to design and develop a robust vision system, which can be used to identify animals in real time in different environmental conditions.

The New Zealand government invested NZ$4 million in 2010 to introduce and carry out extensive research on self-resettable trap technologies [1]. Self-resettable traps have been introduced and trial runs have been and are still carried out. However, a major disadvantage of the current technology is that it does not identify the species trapped before it activates the killing mechanism. Therefore, such traps could kill any species which goes into these traps.

At present NZ-DOC spends about NZ$20 million a year in controlling possums and ground-based pests like rats and stoats [1]. This money is mainly spent on traditional traps and maintenance. There is currently huge public opposition in New Zealand to using present pest-control practices such as sodium fluoroacetate ($NaFC_2H_2O$), known in pesticide form as 1080 drops, which is biodegradable.

The main motivation for this research is therefore to develop an intelligent alternative to existing animal-trapping technologies and other pest-control techniques. This research has developed an image-based set of algorithms, which could be used to help identify NZ-DOC recognized targeted pests such as feral cats and possums. This alternative solution presented in this thesis can recognize and classify and determine the animal as whether it is a targeted species or not.

1.2 Research Objectives

The main research objective was to develop a suitable algorithm to detect small animals such as cats and possums found in New Zealand. Currently literature has shown there is limited knowledge regarding identifying small animals with most current investigations are generally focused on large animal identification [2-5].

By developing an algorithm, the major challenge would be to make it compatible with an embedded system such as Raspberry PI or a similar device. The main performance indicator will be the execution time of the algorithm, which is dependent on the complexity of the algorithms code along with the image pre-processing tasks. By implementing the system, the total processing time of the algorithm will determine the captivation time of the animal.

Another main challenge of this research was the limited number of usable training images on targeted species. So, the aim of this research was to develop species identification algorithms that will work with smaller training sets, this was due to the prohibitive cost and time of animal ethics approval at Auckland University of Technology and so all the training images were obtained from internet.

Lastly, all the developed algorithms were benchmarked against existing algorithms to investigate performance and determine how well the targeted species classes are able to be separated.

1.3 Thesis Organisation

This chapter provides a brief background of the research problem, highlighting the need for animal identification system to identify small-animal pests in New Zealand and introduces the challenges that arise during then investigation.

Chapter 2 presents a detailed literature review which covers the developments in animal identification systems and related work in other research areas. This chapter also presents the fundamental theories used for the techniques that have been applied in the during this research.

Chapter 3 describes the pre-processing stages of the training and test images. It outlines the standard deviation-based edge detection technique and Hough transform circle

detection-based eye detection techniques that were used to correct the head orientation of small animals.

In Chapter 4 the Eigenface technique is investigated in relation to cat and possum identification. The performance of this technique is improved by the development of a novel eigenvalue-based distance formula. Identification is further improved by introducing an error weight-based classification technique. This chapter also describes the use of multivariate Bayesian decision theory based on classifiers for the Eigenface technique. The Eigenface performance improvement is demonstrated by optimising the resolution of test and training images and the colour scheme.

Chapter 5 defines the use of Fisherface technique for cat and possum identification. This chapter investigates species class separation by grayscale colour schemes and image resolutions. It also compares different colour schemes and image resolutions against operation time to find the optimal resolution and colour scheme.

Chapter 6 focuses on the use of support vector machines (SVMs) for cat and possum classification. The SVM technique's applicability to the novel application is investigated and the integration of principal component analysis (PCA) into SVM to improve the classification rate is described.

In Chapter 7 comparisons are made between the standard Eigenface technique; the standard Eigenface technique with the developed distance algorithm; the standard Eigenface technique with the developed distance algorithm and error weights; the Fisherface technique; and the SVM technique under fixed operating conditions. This chapter also explains the use of receiver operating characteristic (ROC) curves to analyse the performance of each technique under fixed operating conditions.

Chapter 8 describes the novel ROC-based face feature selection technique. This novel technique helps to identify critical face features from all the measured features. These critical face features enable typical discrimination techniques such as PCA, Fisherface and SVM to better separate animal species in high dimensional space.

Chapter 9 presents the conclusions of the research and makes recommendations for future work.

1.4 Contributions of the thesis

The main contributions of this study are listed below:

1. To correct animal head orientaton, a Hough transform-based eye detection method was developed. This technique is based on the standard devation-based edge detction technique. The standard devaion-based image is sliced into separate images depending on the grayscale value and then Hough tranform-based circule detection is used to identify the eyes. Each separate image (sliced sub-image) consists of same circular patten for the pair of eyes.
2. An improved Eigenface classification algorithm. The developed distance formula calculates the distance from the training faces to the test image with respect to the eigenvalues. Previous researchers have found that eigenvectors with large eigenvalues produce poor results and therefore have tended to remove such eigenvectors. But these eigenvectors contain some useful training face features buried under background and light information. The developed distance formula takes such eigenvectors into the distance calculation; their effect is inversely proportional to the size of the eigenvalue.
3. Introduction of error weights to the Eigenface technique with distance algorithm. In this method, the test image is reconstructed with training weights. Then the difference is measured from actual test image and the reconstructed test image. The difference vector is added as a new eigenvector with projected new eigenvalues from the existing eigenvalues. This approach improved the performance of the standard Eigenface technique significantly.
4. A ROC-based feature selection method for the PCA, Fisherface and SVM techniques. This technique improves the species seperation dramaticially compared to existing image-based techniques. This represents a addition to existing discrimination techniques.
5. Finally, all the Eigenface, Fisherface and SVM aplications used in this research are novel in the animal detection domain. To the best of the author's knowledge these techniques have not been applied in this domain before.

This research has resulted in the publication of three international journal papers and one IEEE conference paper, listed below:

[1] T. A. S. A. Perera and J. Collins, “Novel Embedded System Based Species Recognition System for Pest Control” *International Journal of Computing and Digital Systems,* vol. 5, pp. 387-393, September 2016.

[2] T. A. S. A. Perera and J. Collins, "Image Based Cats and Possums Identification for Intelligent Trapping Systems," *International Journal of Computer Applications,* vol. 159, pp. 12-18, February 2017.

[3] T. A. S. A. Perera and J. Collins, "A Novel Eigenface based Species Recognition System," *International Journal of Computer Applications,* vol. 115, pp. 19-23, April 2015.

[4] T. A. S. A. Perera and J. Collins, "Image Based Species Recognition System," in *2015 9th International Conference on Sensing Technology (ICST)*, Auckland 2015, pp. 195-199.

[5] T. A. S. A. Perera, J. Kilby, “Receiver Operator Characteristics (ROC) Based, Feature Selection Method for PCA, Fisher and SVM Techniques for Limited Number of Training Data Applications”, *IET Image Processing,* (Submitted on March 2019)

Chapter 2
Literature Review

2.1 Problem and History

New Zealand is a country that has a considerable amount of endangered animal and plant species. Before the arrival of humans, native species had evolved for millions of years without any predators. This lack of predators led to the evolution of many flightless bird species, such as the kiwi, kakapo, and takahē. Similarly, native vegetation such as fern, kowhai, rātā, kohekohe, tawa and tōtara that have soft succulent leaves without any spikes or barbs.

2.2 Effect of Possums

In 1837 the native Australian bush possums (*Trichosurus vulpecula*), were first introduced to New Zealand for developing a fur trade. Between 1837 and 1922 there were 30 different possum batches imported and released into the wild. It is estimated that possum population had expanded across 54% of the country by 1948–50, 84% by 1961–63, 90% by 1974 and 91% by 1980 [6], shown in Figure 2.1. According to a 2008/09 Landcare Research survey, there are about 30 to 48 million possums in New Zealand [7].

Figure 2.1: Possum introduction and subsequent spread [8]

Adult possums typical size is between 65 cm and 95 cm in length and weight between 1.4 to 6.4 kg [8]. Their typical appearance is a thick, bushy tail, a pointed snout and long fox-like tapering ears and generally have two colour form, grey and black, shown in Figure 2.2. Forestry is their main habitat, especially hardwood mixed forests, where possum populations are particularly high. Possums are nocturnal and mainly feed on leaves, but they also devour buds and flowers, fruits, ferns, bark, fungi, invertebrates, native birds, eggs, snails and carrion. The total New Zealand possums population eats on

average about 12,000 tonnes of vegetation per day [9] and also birds and other insects are approximately 4.5% of their diet [8, 10].

Figure 2.2: Black Possum (left) and Grey Possum (right)

Possums in Australia face natural threats such as dingoes, owls and bush fires, and deal with spiky, less palatable vegetation which keep the possum population under control. Compare this with the New Zealand environment, where the possums have no natural predators and plenty of palatable vegetation. Hence there has been excessive growth in the possum population, which has a huge impact on New Zealand ecosystems over the years.

Possums are a serious problem in New Zealand for two main reasons: (a) damage to native ecosystems; and (b) the spread of bovine tuberculosis (BTB) [11]. They also destroy pasture, crops, commercial forests, soil conservation plantings, home gardens, and contribute to the spread of waterborne diseases such as giardia and cryptosporidium [11].

Possums are destroying New Zealand's native ecosystems mainly through selective grazing, which can eliminate species such as rata, kamahi and mistletoe from the forest which reduces the main food source for native species. Possums also feed directly on the

eggs of New Zealand's native birds and on native invertebrates, and compete with kiwis for nest sites [11].

Possums contribute to BTB by acting as carriers of the disease, on average about 2% of possums are infected with BTB [12]. The occurrence of BTB among cattle increased from 1980 to 1994, but after the introduction of stringent livestock movement practices and increased possum control from 1994 to 1999, the number of affected herds decreased by 53% for cattle and 58% for deer [11-13].

The New Zealand Animal Health Board (NZ-AHB) has estimated that the potential cost of BTB, if not appropriately controlled, could be NZ$1.3 billion over 5 years. This would include loss of expected earnings and additional BTB control measure costs. From 1990 to 1997 government investment in possum control measures increased from NZ$3.5 to NZ$30 million [11, 13]. In 2006 NZ-AHB alone spent about NZ$60 million on controlling possum and an additional NZ$27 million on research on new technologies, herd testing and compensation [12]. In the same year government agencies spent about NZ$111 million in possum control [12].

In New Zealand the task of possum control is spread across three departments: NZ-AHB, NZ-DOC and regional councils. The other stakeholders are private individuals and companies. Figure 2.3 shows how the funding is divided.

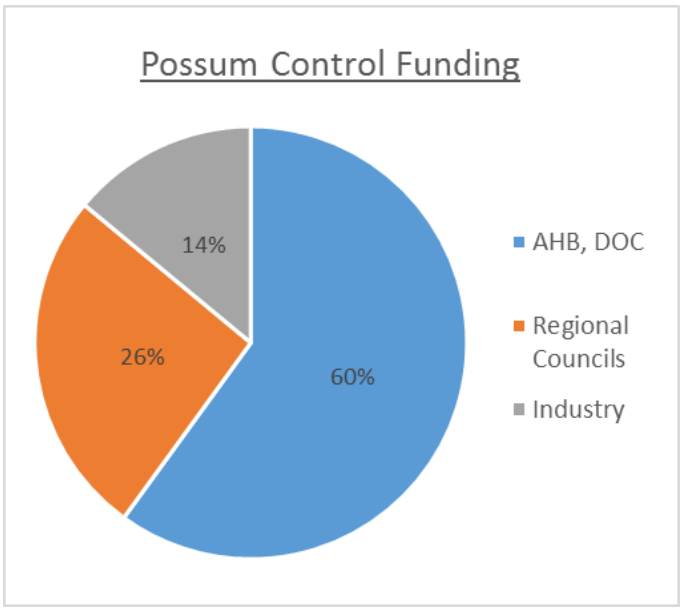


Figure 2.3: Possum control funding in New Zealand for 1998/99 [11]

Currently the most effective method of possum control is the use of toxic baits, which are the subject of huge public opposition. A range of toxic bait formulations are available, containing one of the six poisons currently registered for possum control: 1080, phosphorus, cholecalciferol, cyanide, brodifacoum, or pindone [14].

1080 is the most widely used poison (in carrot, cereal and paste baits) for situations where possum numbers need to be reduced rapidly over large areas. Carrot baits are screened to remove small pieces so as to reduce the risk of birds eating baits [14]. Cereal baits are used for both aerial and bait station control [14]. Paste baits are used extensively for ground-based follow-up maintenance control [14]. Cinnamon is sometimes added to possum baits to act as a lure and mask the taste of 1080 [14]. The toxin is an odourless, non-volatile, virtually tasteless, fine white powder which absorbs water from the atmosphere and becomes sticky. It acts by disrupting the respiration process or the energy pathway in the body, causing possums to die humanely from rapid cardiac or respiratory failure [6].

There are several advantages of using 1080 to control possum numbers, compared to other toxins. It is proven that 1080 is highly effective in gaining vigorous control over possum population. It is one of the registered toxins for aerial broadcast in mainland. 1080 is biodegradable in the environment, and insignificant levels of the toxin have been found in waterways after mass 1080 drops [13-15]. Residues in sub-lethally poisoned animals do not have prolonged persistence. 1080 is a broad-spectrum toxin, which can be effective in other rodents. Due to the extensive use of 1080 over the past decades, there is also a large amount of literature which can be used to optimise the application rates and amounts for optimal possum control [14, 15].

One of the drawbacks in using 1080, currently there is no effective antidote and there is also a high risk of secondary poisoning, especially for dogs. Also possums can develop a bait shyness if they receive a sub-lethal dose and there are also concerns about humanness of poisoning [14].

Compared to other countries, New Zealand is unique in not having native mammals apart from bats, while having a large population of introduced, highly destructive mammalian pests. While 1080 has been the choice of possum control for the past two decades [15], NZ-DOC is presently researching new techniques such as self-resettable smart traps and genetic techniques. However today aerial drops of 1080 remain the most effective technique for large scale or fast response possum control [15].

2.3 Effect of Cats

Cats (*Felis catus*) were introduced by Captain James Cook around 1769 and later by whalers and sealers [16], did not become common till the 1830s. Cats were mainly used to control the exploding rabbit population, and in the 1870s large quantities were released. As the feral cat[1] population increased, however, the native bird population disappeared in areas such as Little Barrier, Cuvier Island, Stephens Island and Herekopare Island in New Zealand [16]. To take just one example, the only known bird specimens of the Stephens Island wren were caught by a lighthouse keeper's cat in 1897 [17].

Feral cats mostly live in mountains, along the coast, bush, scrub and on farmland. Unlike possums, their main feeding source is young rabbits, rats and mice, but also native birds, lizards and larger insects. Typically 12%–15% of their diet is made up of birds, lizards and insects [16, 17]. The average adult cat weight is between 3.6 kg and 4.5 kg [18], with a head-to-body length averaging around 46 cm and tails averaging 30 cm in length. Feral cats effectively can maintain the rodent population [17].

Typically, it is difficult and very expensive to remove cats from the forest. Compared to possums and other animals, cats are more intelligent and shyer. As an example, it took 128 people, and about 400 days, to remove just 100 cats from Little Barrier Island in New Zealand [16].

As domestic pets, cats are the number one choice for households in New Zealand and undeniably offer significant benefits to their owners and society as a whole [19]. But feral cats, on the other hand, are connected with animal welfare concerns and danger to native wildlife and protected habitats [20]. The feral cat population is therefore identified as a significant problem in New Zealand [21]. Low population numbers of feral cats population can be devastating to the native wildlife [22]. One study discovered that, around New Zealand cities, domestic and feral cats are hampering the persistence of the bird population [23]. There are urban regions around Auckland that have substantial areas of forest and bushland such as the Waitakere and Hunua Ranges, where nearby urban and feral cats pose a great risk to the native bird populations. So, New Zealand policymakers

[1] A feral cat is a domestic cat that lives outdoors and has had little or no human contact.

have recently made environmental protection and conservation a priority and placed high importance on managing the impact of feral cats [24].

Similarly, local authorities are currently introducing new bylaws to limit the number of cats per house and also improve accountability of ownership by tagging and registration [25]. When human care is lacking, cats naturally become stray due to their nutritional and reproductive requirements [26]. As cats can transition to the feral state within one generation [27], the best way to limit the population to increase is by limiting the breeding population. Due to lack of lethal control, the best method to control the urban cat population is by early sterilisation (pre-pubertal) [28], where New Zealand's pet cat population has a high rate of sterilisation compared to other countries (87%–91.7%) [21] with room for improvement, especially as there is very little information about the urban feral cat population. The second most effective technique is educating owners about cat care and sterilisation [29]. Since there is limited public funding available in New Zealand to promote sterilisation and ownership obligation education. There is a need to provide a forum where owners can access important information on urban and feral cat management and its impact on New Zealand's ecosystem.

2.4 Current Control Measures and Limitations

Classical pest control techniques are currently used to identify the pest population in New Zealand forests. NZ-DOC's main objectives are to identify and collect population information of pests before setting the traps these types of pests. Existing animal detection technologies are time-consuming and costly.

Species recognition is an area in which only a limited research has been carried out. In the pest control domain basic primitive technologies are still in use today. One example of identifying species is a tracking tunnel using ink paper [30] in order to identify, understand and study the species in a given location, shown in Figure 2.4.

Once NZ-DOC has gained knowledge on the pest population for a given location, pest control traps can be placed to control the population. The major disadvantage of using this approach is that the traps do not have the ability to distinguish the targeted pest from another species. The only means of avoiding killing the wrong animal is by using the appropriate bait and lure [14].

Clearly, this method is not optimal, and a small number of New Zealand native bird species can be affected such as Kea, weka, tomtits and robins are known to be prone to eat baits and die. The approach for minimising such an outcome is to use less palatable baits and avoiding open locations to minimise any potential 'by-kill' from these species [15, 31].

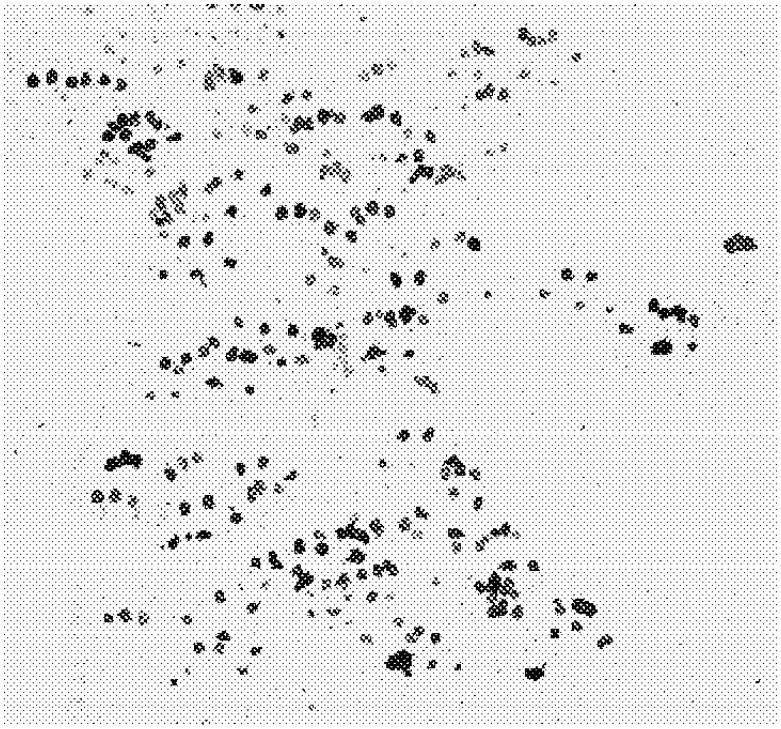


Figure 2.4: Mouse footprints [30]

Recently, self-resettable traps have been introduced and trials have been carried out by NZ-DOC. The New Zealand government has invested NZ$4 million to introduce and carry out extensive research on self-resetting trap technologies [32]. Resettable trapping technology has the potential to be a far more effective technique for trapping pests with minimal environmental effects. The main advantage of this technology is, it kills the target pest humanely within 3 minutes, as legally required by the New Zealand Animal Welfare Act 1999 [33]. Almost all the existing spring-loaded traps fail to kill within legal time period from capture. The aim of this research was to find a suitable algorithm to identify the main targeted species of cats and possums.

2.5 Current Technologies of Small Mammals Detection

One of the main methods of identifying small mammals is by their footprint, where animals firstly walk through an ink well and then walk over white paper, which records

the footprint information, shown in Figure 2.5. Numerous research studies have been carried out to correlate the footprint information to actual species [33-35].

Figure 2.5: Tracking tunnel with tracking card [35]

Typically, this type of research is carried out by scanning the tracking card and using a pattern detection algorithm to extract the actual footprint from the background [35]. To attract the animals into the tunnels, lures are used [35, 36]. Both Yuan's [36] and James's [35] studies use the technique of identifying footprint patterns by crosschecking against an existing footprint database. If any entry is matched with existing data, the animal can be identified.

Another method of identifying small mammal is by measuring the distance between toe marks [34]. In this method, scanned footprints are extracted and then the distance between each artefact is measured, which enables thee animal to be identified, shown in Figure 2.6.

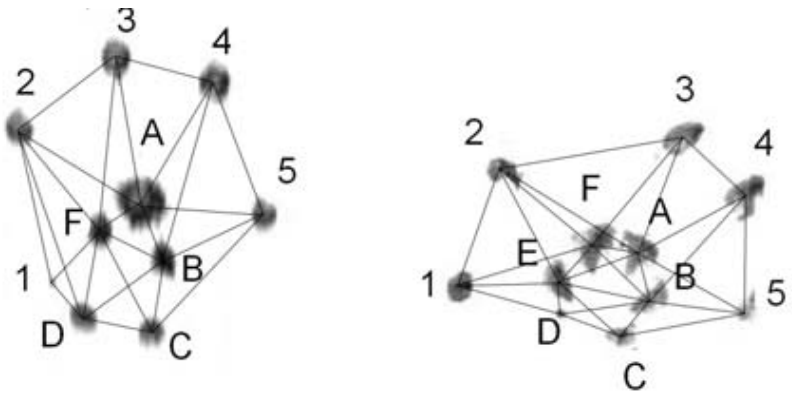


Figure 2.6: Measured footprints [34]

A limitation of footprint recognition systems is that they tend to be post-processors and as a result cannot be used with resettable trapping technology. The main drawback of this method is that the animal footprints are not always clear. Due to environmental conditions, the footprints may include other particles that may mislead the identification system and hence such a system is not suitable for real-time, remote applications.

2.6 Computer Vision for Human Face Recognition

Human face recognition is one of the fastest-growing areas in image processing. The face-recognition algorithms can be categorised into three main categories: feature-based, holistic and hybrid approach [34-36]. The feature-based method focuses on extracted features such as skin colour, distance between facial features and special patters. Holistic approach uses predefined standard face patterns. Hybrid-approach use representations of local face parts [34, 36, 37].

Since there are numerous face recognition algorithms available with their modifications, it is more meaningful to tabulate current research. Table 2.1 provides a summary for each of the following:

a) Method used
b) Explanation of the method
c) Advantages/Limitations of the method
d) Database used
e) Accuracy of the method

Table 2.1: Comparison of Face Recognition Algorithms

Method	Explanation	Advantages	Limitations	Database	Accuracy	References
Bayesian Model	Uses a probabilistic measure of similarity based primarily on a Bayesian (MAP[2]) analysis of image differences.	Error rate is minimised compared to PCA and LDA.	Computationally costly.	DARPA[3] and FERET[4]	95%	[38, 39]
Discrete wavelet transform-based feature extraction for hyperspectral face recognition	This method uses a three-dimensional discrete wavelet transform to extract features from facial hyperspectral images and decomposes hyperspectral images into a set of spatio-spectral frequency sub-bands.	Under different test scenarios revealed that accuracy of this methods is superior to alternative methods using spatio-spectral classification.	Not stated.	PolyU[5], PIE[6] and UWA[7]	PolyU 96.66% CMU 98.61% UWA 98.28%	[40, 41]

[2] Maximum A Posteriori Probability
[3] Defense Advanced Research Project Agency
[4] Face recognition technology database
[5] The Hong Kong Polytechnic University (PolyU) NIR Face Database
[6] The CMU Multi-PIE Face Database
[7] UWA Hyperspectral Face Database

Table 2.1 (continued)

Method	Explanation	Advantages	Limitations	Database	Accuracy	References
Embedded Hidden Markov Model (HMM)	This model inherits the partial size invariance of the standard HMM, and, due to its pseudo two-dimensional structure, can model two-dimensional data such as images, better than the standard HMM.	Flexible with image variations in scale, natural deformations. Fast training and recognition algorithms.	Numerically complex compared to Hidden Markov Model.	ORL[8]	98-100%	[42]
Fisherfaces	This method maximizes the ratio of between-class scatter to within-class scatter. Fisherface is one the most successful widely-used method for face recognition.	Perform well in varying lighting conditions and facial expressions.	Requires lager number of training images and more processing time in recognition compare to Eigenfaces.	Yale[9]	Not stated	[43-45]

[8] The ORL Database of Faces
[9] Yale Face Database

Table 2.1 (continued)

Method	Explanation	Advantages	Limitations	Database	Accuracy	References
Gabor filter-based feature extraction	Garbor filter is used to capture faces aligned at specific angles. Binary Particle Swarm optimisation is used to find the optimal feature subset.	Superior performance in presence of pose, illumination and expression variations conditions.	Not suitable for real time applications.	ORL, FERET, Yale and FEI[10]	ORL 76.84% FEI 81.20%	[46, 47]
Laplacianface	Laplacianface technique aims to discover local structure of the manifold.	Eliminates or reduces unwanted variations resulting from changes in lighting, facial expression, and pose.	Require frontal face images. Need large amount of training images.	Yale PIE	Yale 11.3% error rate PIE 4.6% error rate	[45, 48]

[10] FEI Face Database Brazilian face database

Table 2.1 (continued)

Method	Explanation	Advantages	Limitations	Database	Accuracy	References
Local Binary Pattern	The operator assigns a label to every pixel of an image by thresholding the 3 × 3 neighbourhood of each pixel with the centre pixel value and considering the result as a binary number. Then, the histogram of the labels can be used as a texture descriptor.	Local Binary Pattern is robust to challenges caused by lighting changes or aging of the subjects.	The performance is low compare to the other techniques. Performance can be improved by introducing CNN[11] [49].	FERET	85% 97.75% with CNN	[49, 50]

[11] CNN Convolutional Neural Network

Table 2.1 (continued)

Method	Explanation	Advantages	Limitations	Database	Accuracy	References
Low-Power Convolutional Neural Network Face Recognition	System consists of two chips: an always-on CMOS[12] image sensor-based face detector and a low-power CNN processor for face verification.	System consumes very low power level to evaluate images.	Image sensor end still has many challenges on its robustness, scalability, and applicability for more advanced detection algorithms.	LFW[13]	97%	[51, 52]

[12] CMOS Complementary Metal–Oxide–Semiconductor
[13] Labelled Faces in the Wild face database

Table 2.1 (continued)

Method	Explanation	Advantages	Limitations	Database	Accuracy	References
Principal Component Analysis /Eigenfaces	Seek a low-dimensional representation that maximise the global scatter of the training samples.	Raw intensity data are used directly and recognition without any significant low-level or mid-level processing. No knowledge of geometry and reflectance of faces is required.	Serious limitations of Eigenface representation method for face recognition under different conditions such as (varying illumination conditions and head orientations). Affected by image size changes.	Yale	Not stated	[44, 53-56]

Table 2.1 (continued)

Method	Explanation	Advantages	Limitations	Database	Accuracy	References
Scale Invariant Feature Transform with Multi-Directional Multi-Level Dual Cross Patterns	The partial face recognition using Scale Invariant Feature Transform technique is combined with multi-directional multi-level dual cross patterns technique. The robust point set matching is used to probe face image. PNN[14] and K-NN[15] are used for classification.	Operates well with presence of occlusion, random partial crop, illumination, pose and exaggerated facial expression.	Very time consuming when there are large number of subjects and number of image samples for each subject.	AR[16], Yale, LFW, Pubfig[17], EURECOM Kinect Face Database and ORL	98.33%	[57, 58]

[14] PNN Probabilistic Neural Network
[15] K-NN K-Nearest Neighbor
[16] AR Face Database computer vision centre Ohio State University
[17] Public Figures Face Database

2.7 Computer Vision for Animal Detection

Image processing-based technology is improving due to development of new cameras and smartphone-based video applications. The OpenCV source image processing toolbox [59] has played a major part in this advance, which started in the mid-1990s and it now contains more than 2500 image processing algorithms. This image processing toolbox has been widely adopted by major global companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda and Toyota, and is compatible with all major operating systems [59].

There are few systems available for animal identification, such as camera taps [60-64], image based livestock health monitoring systems [65-67] and motor vehicle manufactures to improve the driver assistance systems [68]. Mainstream vehicle manufacturers, such as Audi, BMW and Daimler are developing animal detection for driver assistance and warning systems [68], which use far infrared (FIR) cameras to detect animals, shown in Figure 2.7. They have been trained using thousands of hours of FIR videos of animals worldwide and mostly concentrates on detecting large animals like deer, horses and moose.

Figure 2.7: Examples of FIR images of deer in various poses and angles [68]

Identification of individual livestock has become a problem in recent years as farming technologies continue to be improved to find a precise objective measurement are required e.g. weight. Currently use of RFID tags are common method of identifying each individual animal, which is time-consuming for the farmer and distressing for the animal. To tackle this problem, non-invasive biometrics techniques are used to identify each individual animal by their face [69]. Image based animal health monitoring, such as lameness detection based on posture is becoming another popular area of research [65-67].

Animal identification based on based on camera-traps is becoming more desirable as the camera technology and image processing techniques continue to improve. This type of animal monitoring technique is proven to be useful in ecological, conservation and behavioural research [5, 63, 70] and has proven to be non-invasive to behaviour of the animal in their natural habitat [71] and provides a way to collect large amount of data [5, 61, 63, 64]. The main drawback of this technology is pre-processing task of selecting useful images from large number of recordings, various research studies [5, 61, 63, 64] found this process is challenging and unacceptably time-consuming, and also it was difficult to get a sufficient number of images, of targeted species [63].

Ramanan, Forsyth and Barnard [4] used pictorial structures to identify larger animals from video footage, where they divided the animal into number of pictorial representations such as rectangle structures. The configuration and orientation of these rectangle members of the animal can be used to identify it, shown in Figure 2.8. They then expanded their research to animal texture detection and developed a library of animal textures. By incorporating texture detection with pictorial representation, the accuracy of the system was improved [4, 72].

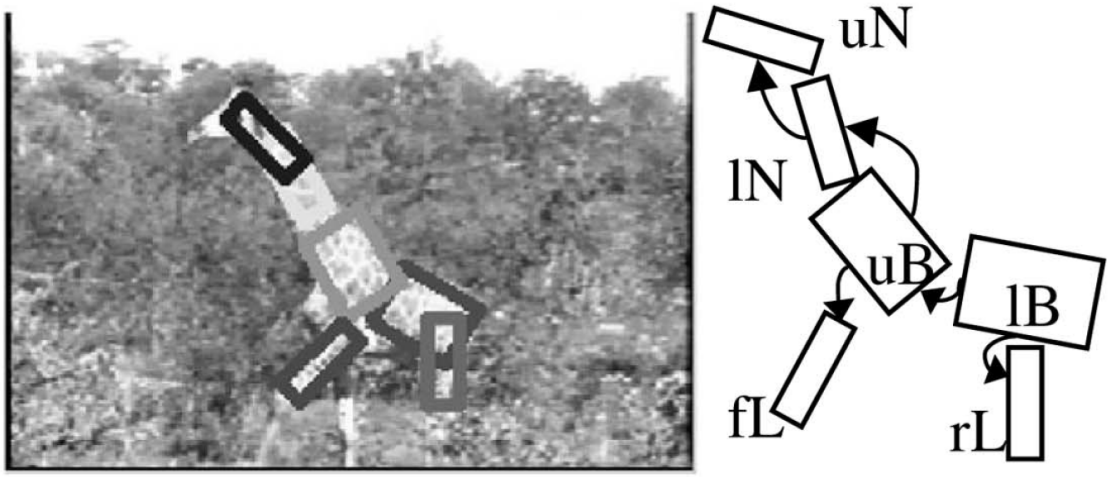


Figure 2.8: Pictorial representation of a giraffe

Burghardt and Calic performed animal face detection and animal tracking using similar techniques as those used for human face detection [60, 62]. They used a combination of several techniques to identify lion faces. Their detection strategy is based on the Viola-Jones detector technique [73]. Haar-like features [74] and AdaBoost are integrated into their system for smooth and accurate tracking [60, 62]. These human face detection techniques can be used in the animal recognition domain reasonably accurately.

According to their research, the detection rate for larger animals is high, shown in Figure 2.9.

Figure 2.9: Feature point stipulated in the centre of the detected region [62]

Loos *et al*. compared Eigenfaces, Fisherfaces, Laplacianfaces and Randomfaces approaches to identify individual great apes [75]. According to their finding with nearest neighbour classifier Laplacianface and Fisherface techniques had best accuracy 90% and 85% respectively. With SVM classifier Laplacianface, Eigenface and Randomface techniques had about 90% accuracy.

Liu and Fu used an image of a pig for weight measurement [76]. To identify the pig dimensions (measurement points), Scale-invariant Feature Transform (SIFT) [76, 77] image processing technique was used, shown in Figure 2.10. Using these measurements, a weight prediction algorithm [78] calculates the weight. For a species recognition project, animal weight could be a good indicator to differentiate animals like possums, rats or kiwis. By introducing these techniques, overall accuracy can be improved.

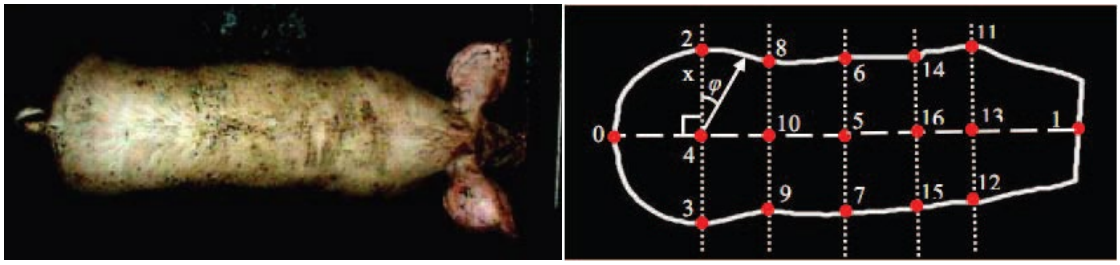


Figure 2.10: Measurement point for pigs [76]

Zhang, Sun and Tang used both texture and bodily features to detect larger animals like tigers, pandas, foxes, cats and cheetahs [2]. Their system uses the Histogram of Oriented Gradients (HOG) [79] to capture the shape and texture features on the animal's head. Once fur texture and features are captured, brute force detection and deformable detection algorithms are used to effectively exploit the shape and texture features concurrently [2], shown in Figure 2.11. The above research [2] shows promising results for animal

identification systems. The combined approach of fur texture and face detection achieves the best animal identification rate.

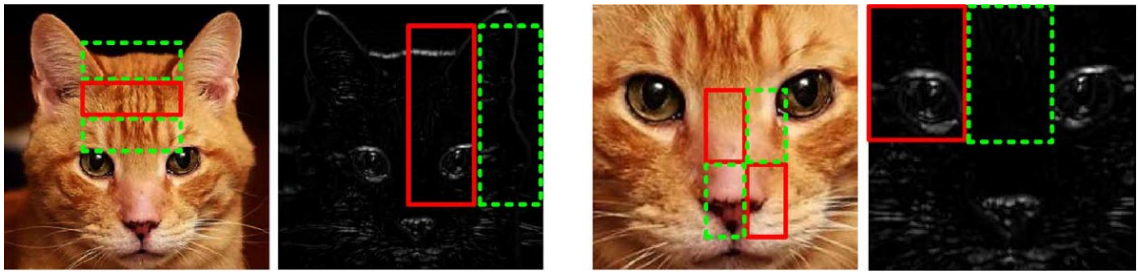


Figure 2.11: Shape (left) and texture detection (right) [2]

Deep Learning is currently one of the fastest-expanding areas in machine learning and signifies a key part of innovation in image processing. It is achieving extraordinary levels of accuracy, to the level where deep learning algorithms can beat humans at classifying images and can beat the world's best GO player [80, 81]. Deep convolutional neural networks are commonly used to classify the camera trap animal images [5, 61, 71, 82]. Tibor *et al.* [83] experimental results show that the convolutional neural networks have a positive effect on overall animal recognition performance and outperforms PCA, LDA, LBPH and SVM techniques.

2.8 Principal Component Analysis Technique

Principal Component Analysis (PCA) is widely used and applied in different research areas and can be used effectively in image recognition and feature extraction [56]. The main reason for its popularity is the balance it offers between simplicity, speed of the algorithm, and usefulness of its results.

If there are n spatially registered images each with n pixels for any given pair of coordinates (i, j), then these pixels can be arranged into a column vector, shown in Figure 2.12.

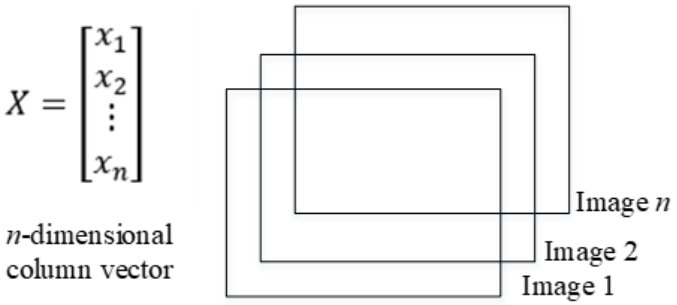


Figure 2.12: Stack of the same size images

If the images are $M \times N$, there will be M by N dimensional vectors representing all the pixels in the n images. The mean vector (or mean face) m_x given by equation (2.1) can be calculated.

$$m_x = \frac{1}{K}\sum_{K=1}^{K} X_K \tag{2.1}$$

where $K = M \times N$

The vectors covariance matrix C_x ($n \times n$) can be calculated using equation (2.2).

$$C_x = \frac{1}{K-1}\sum_{K=1}^{K} (X_K - m_x)\,(X_K - m_x)^T \tag{2.2}$$

where $K - 1$ is then used to obtain an unbiased estimation of C_x from the samples.

Let A be a matrix whose rows are the eigenvectors of C_x, ordered so that the first row is the eigenvector corresponding to the largest eigenvalue, and the last row is the eigenvector corresponding to the smallest eigenvalue. The principal components analysis can be expressed as:

$$y = A(X - m_x) \tag{2.3}$$

The rows of matrix A are the eigenvectors of C_x normalized to unit length. Hence C_x is real and symmetric, and these vectors form an orthonormal set [84; Chapter 12]), it can be shown that:

$$m_y = 0 \tag{2.4}$$

therefore:

$$C_y = AC_xA^T \tag{2.5}$$

Matrix C_y is diagonal, and it follows that the elements along its main diagonal are the eigenvectors of C_x. The main diagonal element in the i^{th} row of C_y is the variance of vector element y_i, and its off-diagonal element (j, k) is the covariance between element y_i and y_K. The off diagonals of C_y are zeros, showing that the elements of the transformed vector y are uncorrelated [84].

Since the rows of A are orthonormal, its inverse equals its transpose. Therefore, X can be recovered by performing the inverse transpose, given by equation (2.6).

$$X = A^T y + m_x \tag{2.6}$$

Instead of using all eigenvectors of C_x, matrix A_k from the k eigenvectors corresponding to the k largest eigenvalues, yielding a transformation matrix of order $k \times n$. Then y vector would then be k dimensional, and the reconstruction given in equation (2.6) no longer be exact but is an approximation [84, 85]. This method can be used for data reduction.

2.9 Singular Value Decomposition

Singular Value Decomposition (SVD) is another method used for data reduction and is a way of breaking down (decomposing) matrices into meaningful and manageable pieces. In other words, SVD takes a high-dimensional, highly-variable data set and reduces it to a lower-dimensional space that shows the main structure of the original data clearly, from most variation to least variation [86; Chapter 15, 87].

The matrix A can be factorised as in equation (2.7).

$$A = UDV^T \tag{2.7}$$

where U and V are orthogonal matrixes and matrix D is diagonal with positive real values. The main diagonal of D holds what are called the singular values. The m rows of U are called left-singular vectors and d rows of V are called right-singular vectors.

SVD is defined for all matrices, whereas the most commonly-used eigenvector decomposition requires the matrix A to be square and certain other conditions to be met to ensure orthogonality of the eigenvector [86; Chapter 15], shown in Figure 2.13.

1. The left-singular vectors of A are eigenvectors of AA^T.
2. The right-singular vectors of A are eigenvectors of A^TA.
3. The non-zero singular values of A (can be found in diagonal entries of D) are the square roots of the non-zero eigenvalues of both AA^T and A^TA.

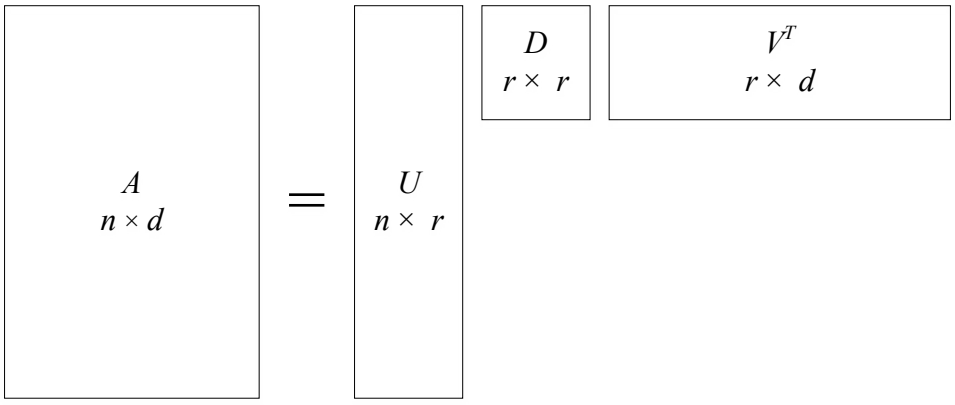


Figure 2.13: SVD of matrix A

2.10 Eigenface Technique

The Eigenface technique is commonly used in human face recognition. In a typical application, a training set is created with different human faces that need to be identified by the system. The mean of the input face image is calculated, and the mean is subtracted from the training set images to obtain a mean-shifted training set. This is known as normalising the training set. For the mean-shifted training set, the eigenvectors with the largest eigenvalues are calculated. These are known as the principal components and keep most of the facial features. Finally, the Eigenface technique projects the mean-shifted images into the eigenspace using the principal eigenvectors [45, 54, 55, 88].

In this research eigenvectors are used to distinguish between cats and possums. The Eigenface algorithm can be broken down into 6 steps [44, 54, 55, 88].

Step 1. Obtain an animal face training set $I_1, I_2, \ldots\ldots, I_M$

Step 2. Convert each image I_i into a vector Γ_i (convert the $N \times N$ image into an $N^2 \times 1$ vector)

Step 3. Calculate the average animal face vector m:

$$m = \frac{1}{M}\sum_{i=1}^{M} \Gamma_i \tag{2.8}$$

Step 4. Subtract the average face from Γ_i to get Φ_i:

$$\Phi_i = \Gamma_i - m \tag{2.9}$$

Step 5. Calculate the covariance matrix C:

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n^T \Phi_n = \mathrm{A}^T\mathrm{A} \tag{2.10}$$

Step 6. Calculate the eigenvectors e_i of $A^T A$ [54]

These are the same as M best eigenvectors from largest eigenvalues.

Once the eigenvectors with large eigenvalues are selected, unknown images can be fed through the system for face recognition. Before this, however, the input image needs to be normalised and demeaned[18]. This process can be split into four steps for a given unknown image Γ [45, 55, 88].

[18] Demeaning data is finding the mean and removing it from each value of the data, so the mean of the data will then have a zero value.

Step 1. Calculate

$$\Phi = \Gamma - m \tag{2.11}$$

Step 2. Calculate

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \ \left(w_i = u_i^T \Phi\right) \tag{2.12}$$

Step 3. Calculate Euclidean distance [54]

$$e_d = \left\| \Phi - \hat{\Phi} \right\| \tag{2.13}$$

Step 4. If $e_d < T_d$ then Γ is a face

l^2 −norm[19] is used to calculate the distance from mean image. This distance is then compared against a known threshold T_d value. If the distance value is less than the threshold, the unknown input image is belonging to one of the training.

2.11 New Developments on Principal Component Analysis/Eigenface Techniques

Principal Component Analysis technique was used originally by L. Sirovich and M. Kirby in 1987 [89] for face recognition and M. Turk and A. Pentland further develop this using the Eigenface for face classification [54, 55]. After an extensive research through the journal databases available at the Auckland University of Technology library for a 5-year period from 2013 to 2018, over 6,900 articles were found to be related to the topic of Eigenface. Table 2.2 shows some of the most recently published papers related to PCA/Eigenface techniques.

---

[19] The l^2−norm $|\boldsymbol{X}|$ is a vector norm defined for a complex vector

$$\boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ by } \quad |\boldsymbol{X}| = \sqrt{\sum_{k=1}^{n} |x_k|^2}$$

Table 2.2: PCA/Eigenface Based Face Recognition Literature

Method	Explanation	Reference
Joint Sparse Principal Component Analysis	This method relaxes the orthogonal constraint of transformation matrix, by introducing another transformation matrix and imposing joint $\ell_{2,1}$ –norms on both loss term and regularisation term. This method provides more freedom to jointly select useful features for low-dimensional representation and it is robust to outliers. Feasibility and effectiveness are benchmarked on eight different datasets.	[90]
$\ell_{2,p}$ −Norm Based PCA for Image Recognition	Wang *et al*. used $\ell_{2,p}$ −norm as the distance metric to measure reconstruction error. This method is robust for outlier and keeps PCA's properties such as rotational invariance. This technique is benchmarked with three different data bases and compared performance with PCA in papers [91-93] techniques.	[94]
Two-Dimensional Whitening of Face Images for Improved PCA Performance	This method whitens the distribution of rows and columns of an image matrix. Proposed method two-dimensional whitening transform does not enforce any alteration covariance matrix. Seghouane *et al*. showed that rows and columns of front-pose face images are approximately normally distributed. The performance is compared with paper [95] and had superior performance.	[96]

Table 2.2 (continued)

Method	Explanation	Reference
Fractional Eigenfaces	Fractional Eigenfaces is method of feature-extraction technique for high-dimensional data. This technique is an extension of paper [97], fractional PCA. The performance is compared against fractional PCA and Eigenface technique with three different data sets. It was found fractional Eigenface performs marginally better than the other two techniques.	[98]
Face recognition using Eigenface with Naive Bayes	This describe the use of Naive Bayes [99; Chapter 2] for classifying to predict the face in conjunction with Eigenface features. Proposed method improved the classification up to 70%. By introducing normalised z-score the classification rate is further improved up to 89.5%. Test and training images obtained from a digital camera and resized to 150 × 150 pixels.	[100]
Combined weighted Eigenface and BP-based networks for face recognition	In this paper test face image was divided into 9 sub blocks to reduce dimensions and computational complexity. Different weights were allocated to different parts according to the sub blocks importance at the recognition stage. As an example, on a human face key parts were enhanced. With this method recognition rate was increased to 95.62% with ORL dataset and 93.33% with Yale data set.	[101]

Table 2.2 (continued)

Method	Explanation	Reference
Sparse Approximation to the Eigensubspace for Discrimination	Lai *et al*. proposed as novel framework called sparse 2-D projections. This method interactively learns the sparse projection matrix by using elastic net regression and singular-value decomposition. Compared with existing-vector based sparse projection learning methods, proposed method saves both computation and memory costs. Proposed method achieved 97.33% recognition rate on Yale dataset.	[102]

2.12 Fisherface Technique

Similar to the Eigenface technique, the Fisherface technique is mostly used for facial recognition systems [44, 103-106], as well as fingerprint or eye iris recognition systems [107]. In recent times this technique has also become popular for commercial identification and marketing tools [108].

The Fisherface technique was developed in 1997 by Belhumeur [44] and is based on Fisher's Linear Discriminant Analysis (LDA) [109]. This technique has a much lower error rate compared to the Eigenface technique [44, 45, 110] and operates well in situations where there are different illumination conditions and different facial expressions [44, 111].

LDA seeks directions that are efficient for discrimination between the data points. It then maximises the between-class scatter matrix and minimises the within-class scatter matrix. This results in maximum separation in main direction high-dimensional space [109].

For human recognition, the Fisherface technique can be expressed mathematically as below, assuming:

1. All the training images are square images with Width = Height = *N*
2. *M* is the number of images in the database
3. *P* is the number of different persons in the database

For example, there are have two different classes with two images in each class:

$$P_{1,1} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N^2} \end{bmatrix} \quad P_{1,2} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N^2} \end{bmatrix} \quad P_{2,1} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N^2} \end{bmatrix} \quad P_{2,2} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N^2} \end{bmatrix} \tag{2.14}$$

Then compute the average of all training faces:

$$m = \frac{1}{M} \begin{bmatrix} a_1 + b_1 + c_1 + d_1 \\ a_2 + b_2 + c_2 + d_2 \\ \vdots \; \vdots \; \vdots \; \vdots \; \vdots \; \vdots \; \vdots \\ a_{N^2} + b_{N^2} + c_{N^2} + d_{N^2} \end{bmatrix} \tag{2.15}$$

Compute the average face of each class (person):

$$x = \frac{1}{2}\begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \quad \vdots \quad \vdots \\ a_{N^2} + b_{N^2} \end{bmatrix} \quad y = \frac{1}{2}\begin{bmatrix} c_1 + d_1 \\ c_2 + d_2 \\ \vdots \quad \vdots \quad \vdots \\ c_{N^2} + d_{N^2} \end{bmatrix} \tag{2.16}$$

Each class is normalised by subtracting class mean from the training images:

$$a_m = \begin{bmatrix} a_1 - x_1 \\ a_2 - x_2 \\ \vdots \quad \vdots \quad \vdots \\ a_{N^2} - x_{N^2} \end{bmatrix} \qquad b_m = \begin{bmatrix} b_1 - x_1 \\ b_2 - x_2 \\ \vdots \quad \vdots \quad \vdots \\ b_{N^2} - x_{N^2} \end{bmatrix}$$
$$c_m = \begin{bmatrix} c_1 - y_1 \\ c_2 - y_2 \\ \vdots \quad \vdots \quad \vdots \\ c_{N^2} - y_{N^2} \end{bmatrix} \qquad d_m = \begin{bmatrix} d_1 - y_1 \\ d_2 - y_2 \\ \vdots \quad \vdots \quad \vdots \\ d_{N^2} - y_{N^2} \end{bmatrix} \tag{2.17}$$

By using equation (2.17) within class scatter matrix S_W can be calculated:

$$S_W = \left(a_m \cdot {a_m}^T + b_m \cdot {b_m}^T\right) + \left(c_m \cdot {c_m}^T + c_m \cdot {c_m}^T\right) \tag{2.18}$$

The between class scatter matrix S_B can be calculated using training population mean:

$$S_B = 2(x - m)(x - m)^T + 2(y - m)(y - m)^T \tag{2.19}$$

In order to maximise the separation, the matrix, W needs to be maximised:

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|} \tag{2.20}$$

If S_W is non-singular ($M \geq N^2$), the columns of W are eigenvectors of $S_W^{-1} S_B$.

$$S_B W_i = \lambda_i S_W W_i \tag{2.21}$$

The eigenvalues are roots of:

$$|S_B - \lambda_i S_W| = 0 \tag{2.22}$$

The eigenvectors are:

$$(S_B - \lambda_i S_W) W_i = 0 \tag{2.23}$$

If S_W is singular ($M < N^2$), a data reduction method such as PCA or SVD [112] needs to be used to reduce the $N^2 \times M$ matrix to $M \times M$. Then LDA is applied as described in equations (2.15) to (2.23).

Once W is calculated, faces can be projected onto the LDA-space (Fisherface):

$$x_{LDA} = W^T x \tag{2.24}$$

$$y_{LDA} = W^T y \tag{2.25}$$

In order to classify the test images, the unknown image needs to be projected onto the LDA-space. The *k*-nearest neighbour classifier [113, 114] can be used for identification.

2.13 Support Vector Machines (SVMs)

This technique was invented by Vapnik and Chervonenkis in 1963. In 1992 Boser, Guyon and Vapnik developed a non-linear classifier by applying the kernel trick to maximise the margin hyperplanes [115]. Further to this, a soft margin [116] was developed by Cortes and Vapnik in 1993.

A SVM is a discriminative classifier defined by a separating hyperplane, which is also known as decision plane. A decision plane separates a set of objects with different class memberships [115, 117].

Typically support vectors are the data points that scatter closest to the hyperplane. Normally membership of these data points is difficult to classify. These data points have direct contribution on the optimum location of the hyperplane, shown in Figure 2.14 [115].

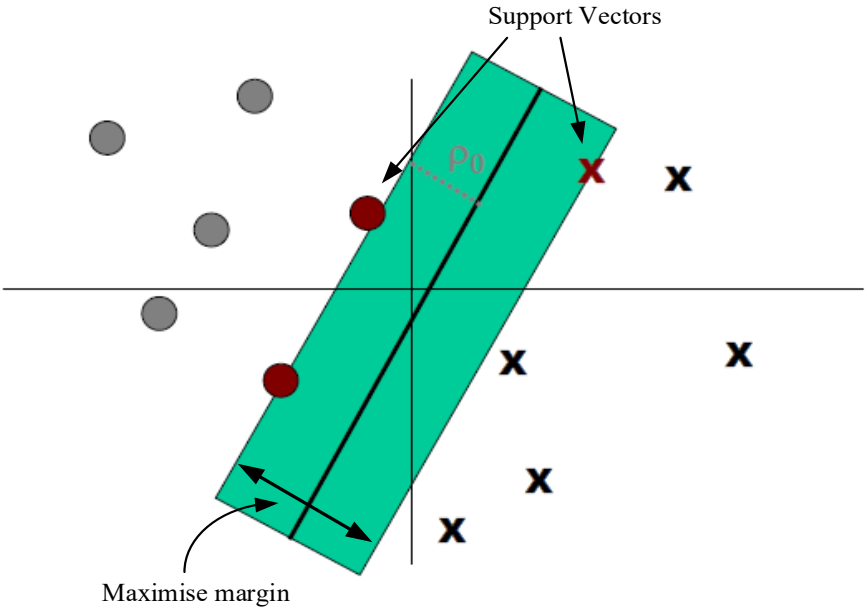


Figure 2.14: 2D support vector points

The SVM technique maximises the margin (street) around separating hyperplanes. The decision function is entirely specified by a small set of training samples, which are support vectors, shown in Figure 2.15. In a typical 2D case training data can be separated by a single line, but in higher dimensional cases data are separated by a hyperplane.

Typically, in a set of training samples, input sample features are called $x_1, x_2, \ldots x_n$ and the resulting output is called y. In higher dimensional cases, there are normally lots of input features. The output vectors of the SVM technique are called weights (feature vectors) w (or w_i). There is one weight for each feature, where a linear combination of w predicts the value of y. The main difference of the SVM technique is that it maximises the margin (street) width to reduce the number of weights that are non-zeros. These are the weights that correspond to important features, which helps to decide the optimal hyperplane. These non-zero weights correspond to the support vectors [115]. The problem of finding the support vector is an optimisation problem. Therefore, the problem can be solved analytically, using optimisation techniques such as Lagrange multipliers.

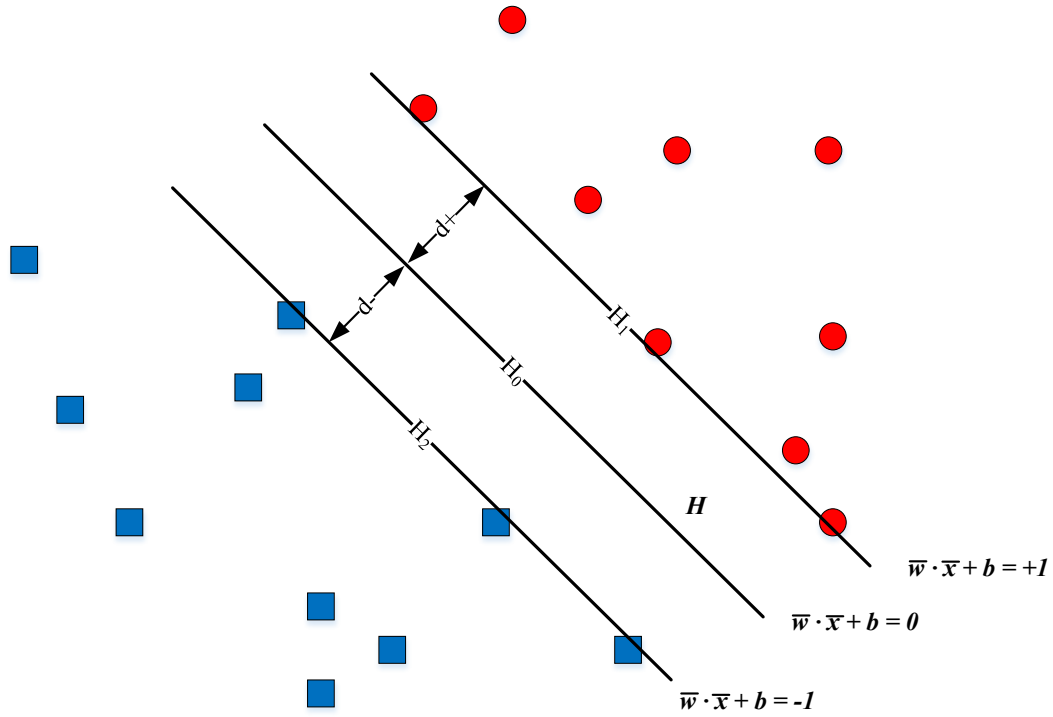


Figure 2.15: 2D SVM training data separation

The hyperplane H can be defined the equation:

$$\bar{w}.\bar{x} + b = 0 \tag{2.26}$$

where w is a weight vector, x is input vector and b is bias.

Geometrically, vector w is orthogonal to H. The class $y_i$of data x_i is defined by

$$y_i = \begin{cases} -1, & if \ \ \bar{w}.\bar{x} + b < 0 \\ 1, & if \ \ \bar{w}.\bar{x} + b \geq 0 \end{cases} \tag{2.27}$$

In order to maximise the margin, the distance between H_1 and H_2 can be formulated as follows:

The distance from a given point (x_0, y_0) to a line $(A+By+c = 0)$ is:

$$|\mathrm{Ax_0} + \mathrm{By_0} + \mathrm{c}|/\sqrt{\mathrm{A^2} + \mathrm{B^2}} \tag{2.28}$$

so distance between H_0 and H_1 is:

$$|\bar{w} \cdot \bar{x} + b| \,/\|\bar{w}\| = 1 \,/\|\bar{w}\| \tag{2.29}$$

therefore, the total distance between H_1 and H_2 is:

$$2 \,/\|\bar{w}\| \tag{2.30}$$

In order to maximize the margin, $\|\bar{w}\|$ needs to be minimise. The main condition is that there are no training data points between H_1 and H_2. To minimise $f(x)$ so that $g(x) = 0$. The equation (2.30) can be rewritten as:

$$min\, f(x) = 1/2\, \|\bar{w}\|^2 \tag{2.31}$$

Since equation (2.31) is a quadratic function, this constrained optimisation problem can be solved by the Lagrangian multiplier method. The paraboloid surface will have a single global minimum. The generalised Lagrangian multiplier can be written as:

$$L(x, a) = f(x) + \sum_i a_i g_i(x) \tag{2.32}$$

where in this case:
$f(x)$ is gradient minimum of f
$g(x)$ is constrained condition g

So in this case equations (2.31) and (2.27) can be substituted into the Lagrangian multiplier equation (2.32):

$$min\, L_P = 1/2\, \|\bar{w}\|^2 + \sum_{i=1}^{l} a_i y_i (\bar{x_i} \cdot \bar{w} + b) + \sum_{i=1}^{l} a_i \tag{2.33}$$

When the derivatives at the minimum at 0

$$\frac{\partial L_P}{\partial \overline{w}} = \overline{w} - \sum_{i=1}^{l} a_i \, y_i \overline{x_\iota} = 0 \tag{2.34}$$

$$\frac{\partial L_P}{\partial b} = \sum_{i=1}^{l} a_i \, y_i = 0 \tag{2.35}$$

$$\overline{w} = \sum_{i=1}^{l} a_i \, y_i \overline{x_\iota} \; and \; \sum_{i=1}^{l} a_i \, y_i = 0 \tag{2.36}$$

Now the answer for weight $\overline{w}$. It is a linear combination of the training input and outputs, x_i and y_i, and the values of a. Now a can be solved by differentiating the dual problem with respect to a and setting its value to zero.

In order to remove the dependence of $\overline{w}$ and b, equation (2.36) can be substituted into equation (2.33):

$$max \, L_D(a_i) = \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i=1}^{l} a_i a_j y_i \, y_j \left(\bar{x}_i \cdot \bar{x}_j \right) \tag{2.37}$$

so that

$$\sum_{i=1}^{l} a_i \, y_i = 0 \; and \; a_i \geq 0 \tag{2.38}$$

After training and calculating $\overline{w}$ by the above method, for a given unknown point $\overline{u}$ measured on feature $\bar{x}_i$, its membership can be determined by observing the sign of:

$$f(x) = \overline{w} \cdot \overline{u} + b = \left(\sum_{i=1}^{l} a_i y_i \bar{x}_i \cdot \overline{u} \right) + b \tag{2.39}$$

Most of the weights $\overline{w}_i$ will be zeros for the a's. Only the support vectors will have non-zero weights or a's, and this will help to reduce the dimensionality of the solution. The l^2 −norm is used to measure the distance from hyperplane to unknown image.

2.14 Receiver Operating Characteristic

Receiver Operating Characteristic (ROC) technique was developed by electrical engineers and radar engineers during World War II for detecting enemy objects in battlefields [118-120]. It was soon adopted for medical decision-making. This method is commonly used in signal detection domain such as radar operators or radiologist, where observer must decide whether or not a target is present or absent; or must classify given target as belonging to one category or another [119]. One of the most important property of ROC analysis is that the accuracy indices derived from this technique are not distorted by fluctuations caused by the use of arbitrarily chosen decision criteria [118].

A decision can be categorised into four outcomes [119]:

1. True negative (TN)
2. False negative (FN)
3. True positive (TP)
4. False positive (FP)

The two most commonly used are TP (sensitivity) and TN (specificity). Sensitivity relates to the ability of the observer to correctly classify the target present stimuli and specificity reflects the ability of the observer to correctly classify the target absent stimuli [119]. Sensitivity and specificity can be calculated using equations (2.40) and (2.41).

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{2.40}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.41}$$

ROC is performed by changing the decision threshold and measuring the sensitivity and false specificity, shown in Figure 2.16. These two vectors can then be plotted on a graph. From this graph two hypotheses can be obtained. The shortest distance to the curve from the maximum sensitivity value, which is (0,1) gives the best decision point (separation point) for two classes. This also indicates how good the class separation is. The shorter the distance the better the separation and the longer the distance the worse the separation. The area under the curve will indicate how well the given technique performs. The larger the area, the better the performance of the algorithm [119, 121].

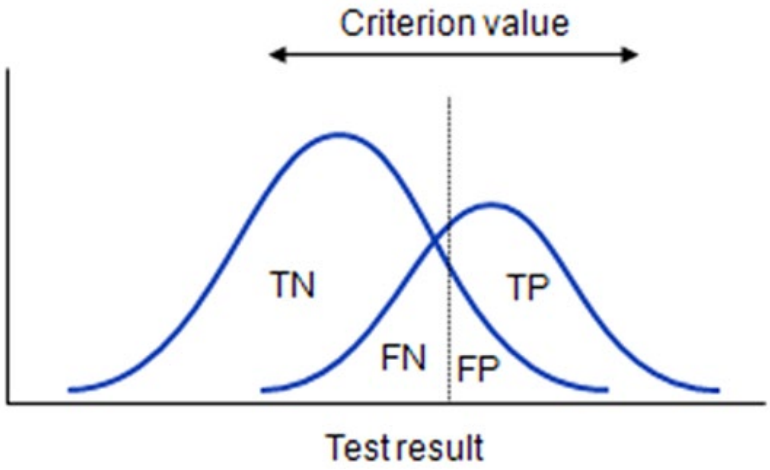


Figure 2.16: Effect on TN, FN, TP and FP fractions under changing the criterion value

Chapter 3

Face Feature Extraction and Image Correction

Image correction is an important part of creating an adequate database of different species. With different face-orientation variations, the detection rate for techniques such as Eigenface and Fisherface is somewhat hampered [55]. Without animal ethical approval, which is prohibitively expensive, it was not possible to obtain the different species images with the appropriate conditions for this research. Therefore, all the training images were obtained from the Internet. All the original images obtained had different face-orientations. In order to correct the orientation of the face, the animal's eyes need to be detected. Once the eyes are detected, the face rotation angle can be calculated and then whole face can be corrected on polar coordinate rotation. According to Shan et al. [122], tightly-controlled pose position improves the detection rate dramatically with one sample database.

This chapter outlines an image-processing method developed to detect animal eyes, it consists of two parts: (a) an edge-detection technique and (b) an eye-detection technique. The edge-detection technique used standard-deviation and image slicing-based; and for eye-detection technique the Hough transform-based was used. These two techniques are by-products of the species feature-detection process. De Souza et al. [123] suggested a similar edge-detection approach by fitting curves to the left and right regions of every adjacent pair of samples, and applying a statistical test to parameters of the fitted curves.

3.1 Eye-detection

The first step in eye-detection is to generate a standard-deviation-based image from the original image. The second step is, standard-deviation image is sliced 12 times according to threshold values for edge-detection. Hough transform-based circle detection is then used to identify circles in the sliced images. The circles' information is further processed to extract the actual pair of circles that represent the eyes.

3.1.1 Standard-Deviation-based Image-Processing

Standard-deviation is a way of quantifying how the data is spread out from the mean, and this information can be used to identify edges on an image [124]. Compared to most common edge-detection techniques, using the following Prewitt operator [85; Chapter 11], Canny edge detector [85; Chapter 11] and Sobel operator [85; Chapter 11], standard-deviation-based edge-detection is easy to implement and less mathematically-intensive [124], and hence it was used to remove uniform areas such as fur patterns.

To minimize the effect of the fur, the standard-deviation-based technique (or standard-deviation face) was developed, to bring out features like eyes and nose due to extreme pixel intensity change in these regions. Unique colour patterns such as fur will have smaller standard-deviation due to uniform distribution of numbers, shown in Figure 3.1 and Figure 3.2. This approach supresses the importance of uniform areas like background, skin colour and (particularly in this application) fur patterns.

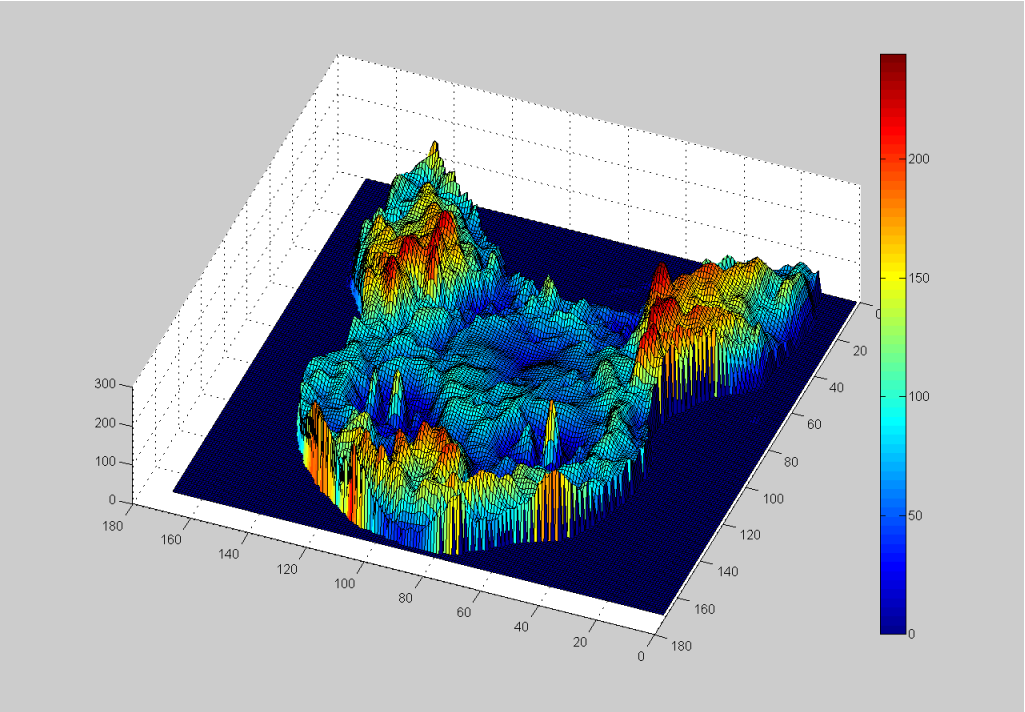


Figure 3.1: Three-dimensional surface plot of a possum face

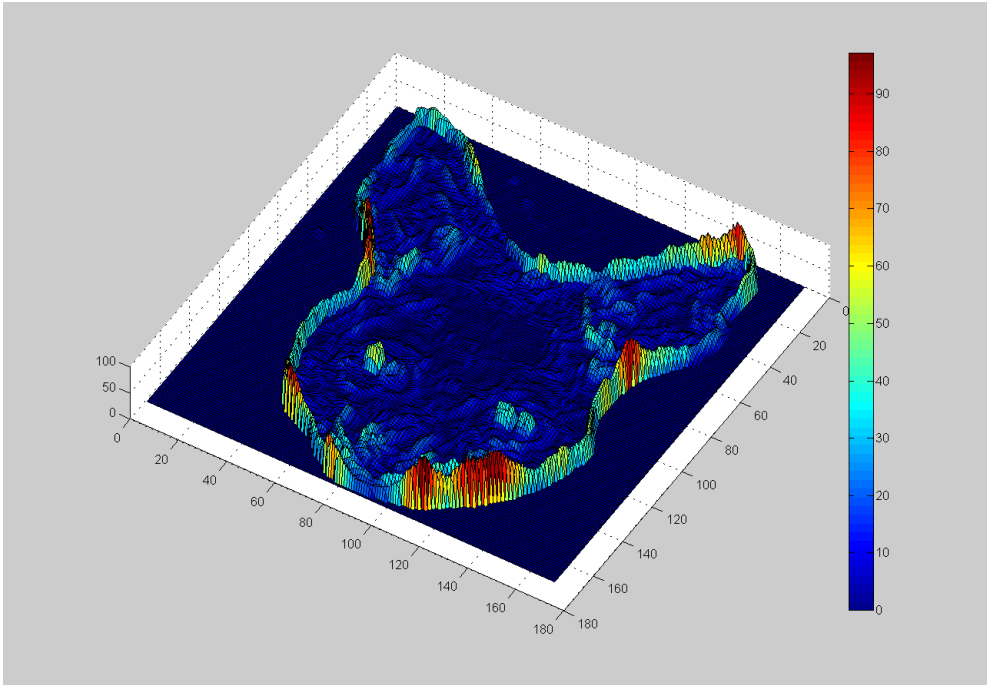


Figure 3.2: Three-dimensional surface plot of a standard-deviation image of a possum face

The standard-deviation face algorithm uses a similar principle to de Souza [123], which looked for large changes in pixels compared to its immediate neighbourhood. But in this set-up, change in standard-deviation is used to suppress the effect of changing local fur details. This process can be expressed in a few steps as follows. Initially the outer pixel

border is ignored i.e. rows 1 and 170 and columns 1 and 170. The image is divided into 3 × 3 blocks, shown in Figure 3.3. Since outer borders are removed the image is now 168 × 168, hence there are 28,244 of possible 3 × 3 blocks, shown in Figure 3.4.

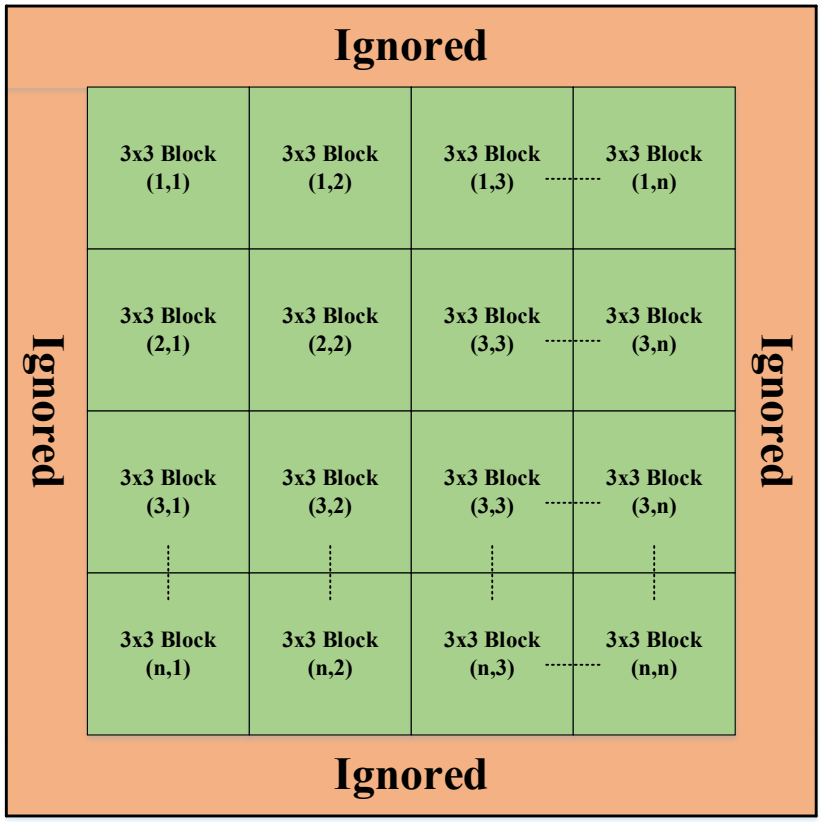


Figure 3.3: Image split into 3 × 3 block form

***p*(-1,-1)**	***p*(0,-1)**	***p*(1,-1)**
***p*(-1,0)**	***p*(0,0)**	***p*(0,1)**
***p*(-1,1)**	***p*(0,1)**	***p*(1,1)**

Figure 3.4: A one 3 × 3 block from the image

Once an image has been arranged according to the above format, the process can be described, shown in Table 3.1.

Table 3.1: Standard-deviation-based edge-detection process

Description	Process
First split the image into 3 × 3 blocks and then calculate the mean m_x for each block, shown in Figure 3.4.	$m_x = \frac{1}{N}\sum_{i=-1}^{1}\sum_{j=-1}^{1} p(i,j)$ (3.1)
Calculate the standard-deviation σ for the 3 × 3 block.	$\sigma = \sqrt{\frac{\sum_{i=-1}^{1}\sum_{j=-1}^{1}(p(i,j)-m_x)^2}{N-1}}$ (3.2)
Replace the $p(i,j)$ with σ value.	$p(i,j) = \sigma$ (3.3)
Repeat the process across the whole image *I* (for pixel *p(2,2)* to *p(n-1,n-1)*).	

Once the process is completed, the standard-deviation image can be plotted as a three-dimensional surface (or surf) plot, shown in Figure 3.2, using the original image which is an 8-bit greyscale range. Once the conversion process has been performed the maximum intensity value is changed to 100. The edge of the face has the highest standard-deviation value, which is due to the sudden colour change from background colour to fur colour. Ignoring the edge of the face and other details tends to produce a maximum standard-deviation value of 60.

3.1.2 Image Slicing

Once a standard-deviation image has been generated, the second step is to divide the image into 12 image slices. This technique of image slicing is normally used in applications such as Magnetic Resonance Image (MRI) and Computed Tomography (CT) scan technologies [125, 126]. These technologies allow users to take images at different depth level for analysis. In this research the same principle is used, but the image is sliced at 12 different depth levels.

Each step has a pixel intensity value of 5 and so the final step number 12 has an intensity value of 60. At each step, image pixel intensity $I_{i,j}$ is less or greater than the fixed constant threshold T. When $I_{i,j} < T$, pixel values are set to 0 (black) and when $I_{i,j} > T$ pixel values are set to 255 (white). Each slice ends up with black and white image, and all the edges are shown in white and the remainder of the image in black, shown in Figure 3.5.

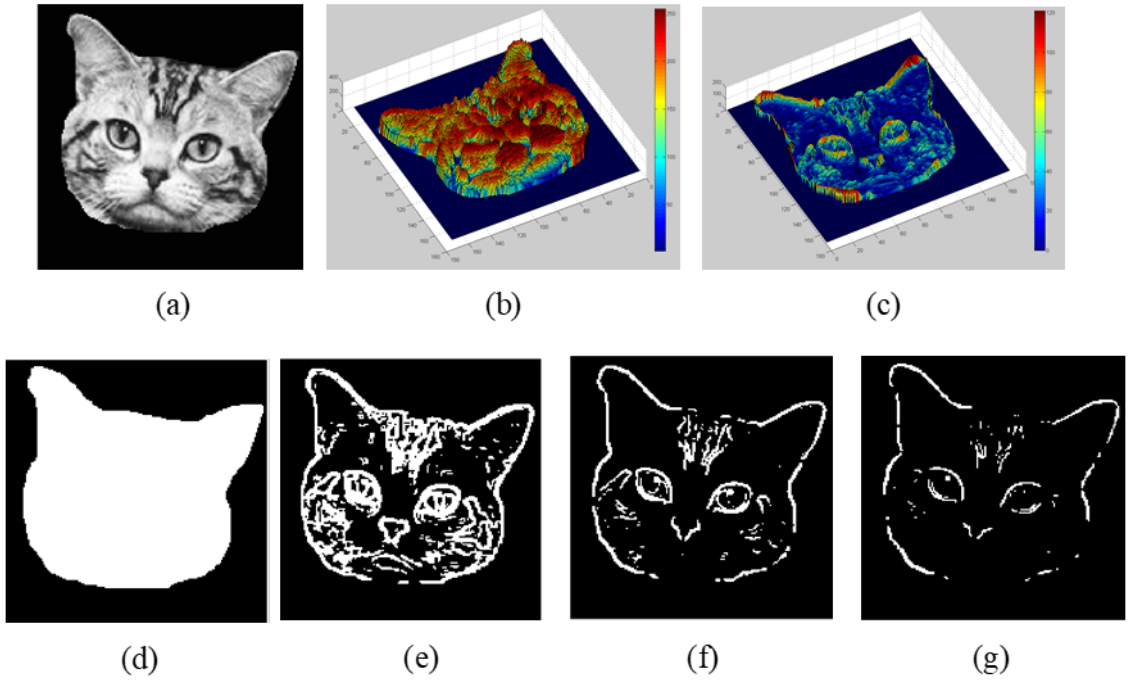


Figure 3.5: Shows (a) the Original image; (b) the Three-dimensional surface plot of the original image; (c) the Three-dimensional surface plot of the standard-deviation image; (d) Image slice 1 of (c); (e) Image slice 7 of (c); (f) Image slice10 of (c); (g) Image slice 12 of (c).

This slicing and thresholding technique removed the fur details and brings out useful information such as eyes and nose, shown in Figure 3.5. This technique was found to be a robust edge-detection tool. Figure 3.6 shows the developed edge-detection technique compared to other techniques such as Prewitt operator, Canny edge detector and Sobel operator.

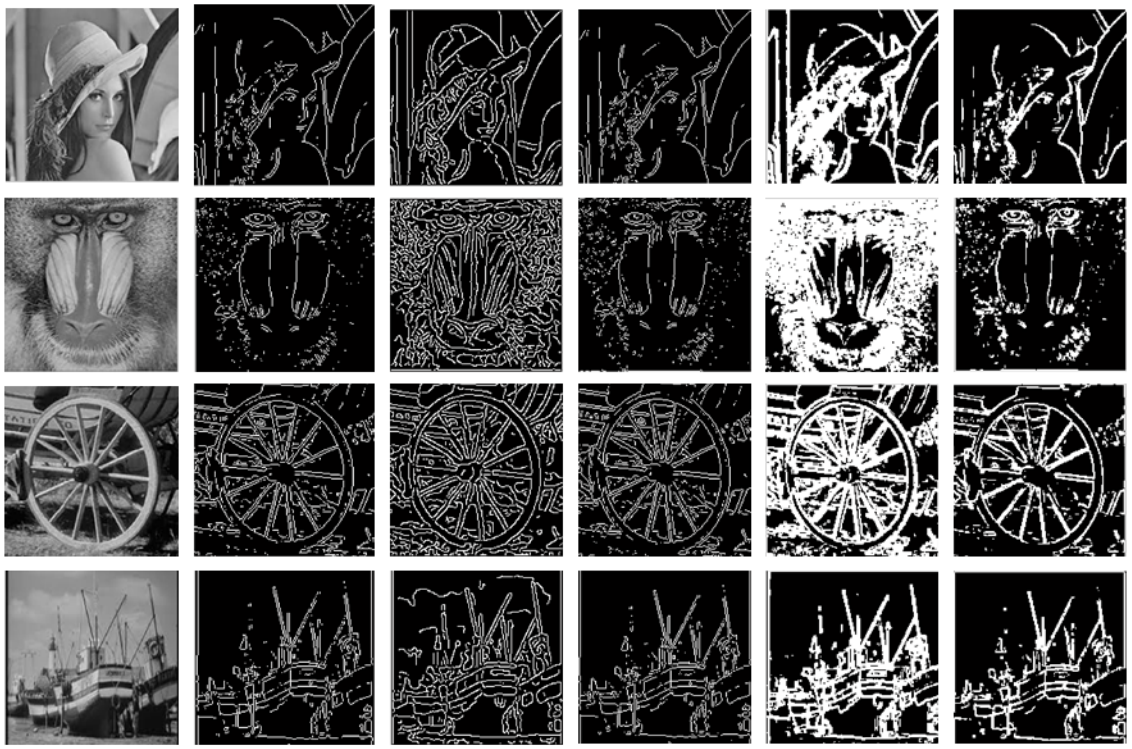


Figure 3.6: Columns from left to right. Column 1: Original images, Column 2: Prewitt (Threshold 0.1), Column 3: Canny (Threshold 0.1), Column 4: Sobel (Threshold 0.1), Column 5: Developed edge-detection technique Slice 6, and Column 6: Developed edge-detection technique Slice 7

3.1.3 Eye-Detection with Hough Transform-based Circle Detection

The next step of the process is to detect the eyes. This was achieved by using Hough transform-based circle detection [127-130]. In order to do this the MATLAB image-processing toolbox-based *'imfindcircles'* function was used. After image slicing was used to find the sharp edges on the face, the second purpose of this process is to detect symmetrical objects that have similar pixel intensity. For an example, both eyes tend to have the same pixel intensity, therefore they appear clearly on one or a few slices, shown in Figure 3.5. In order to run the Hough transform-based circle detection on each slice, shown in Figure 3.7. A radius range for the detection is 5–30 pixels (1.3 mm–7.8 mm) was used for the eyes.

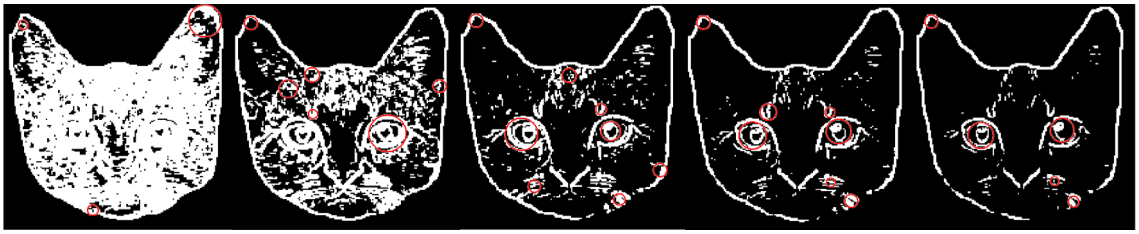


Figure 3.7: Hough transform-based circle detection on image slices 3, 5, 7, 8 and 9 (from left to right)

After completing the circle detection for the eyes, a suitable pair of circles were calculated. The first part of the selection process is to search for the best pair of the eyes by their radii referred to as facial candidates [131]. In this process, closely-matched pairs of circles are grouped by the radius. Then all the facial candidates locations (the x and y coordinates) are analysed. The system first looks for how well these circles are lined up along the y and x coordinates. If the parameters are offset by a threshold value, then those facial candidates are eliminated from the next test round. Next, the selected circle pairs are tested by how close they are to the centre of the image. The closest pair is selected as the final facial candidate for that slice and the process is repeated across all 12 slices. The most suitable circle-pair coordinates reoccur across more than one slice, shown in Figure 3.7 and the facial candidate pair with most occurrences is selected as the final pair of circles that represent the pair of eyes. These detected eyes or facial candidates can be plotted on the final image, shown in Figure 3.8 and the process is expressed as a flow chart, shown in Figure 3.9.

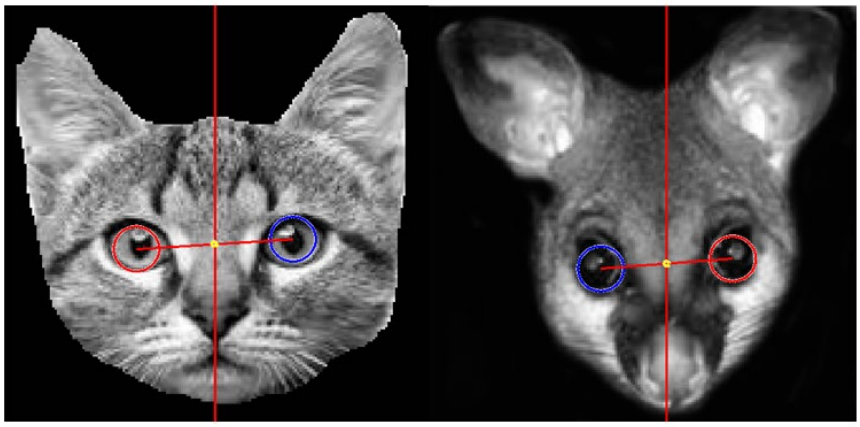


Figure 3.8: Eye-detection of cat and possum images (i) left-eye is indicated by the red circle, (ii) the right-eye is indicated by the blue circle, and (iii) the yellow dot indicates the centre of the face.

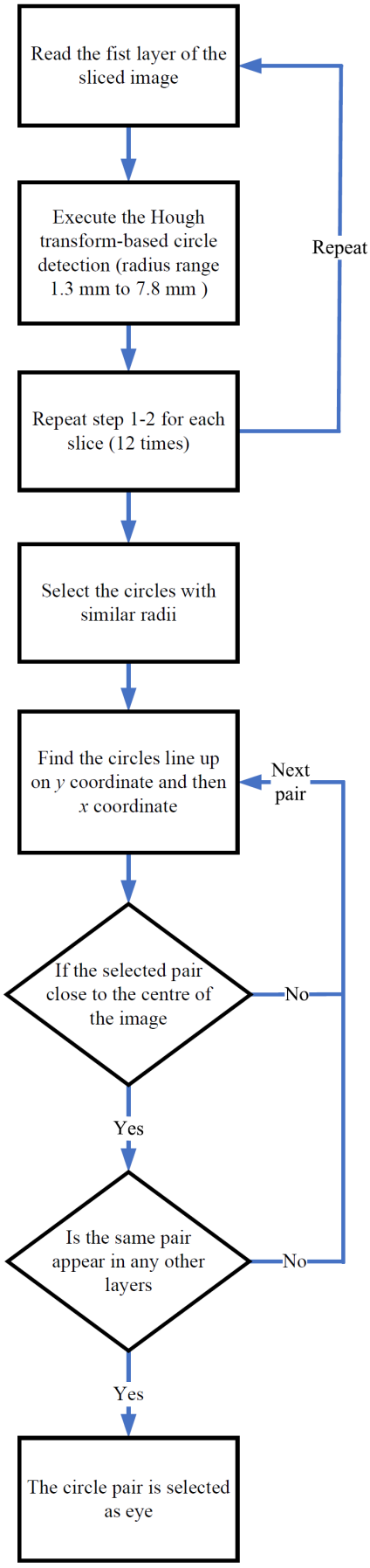


Figure 3.9: Flowchart for eye-detection process using Hough Transform-based.

In this research the eye-detection technique is based on standard-deviation, image slicing and Hough transform-based circle detection. With a visual comparison developed technique detected both eyes, when one eye was slightly covered, but MATLAB vision toolbox did not, shown in Figure 3.10. The only imperfection is that it is hard to detect the centre of the iris. The MATLAB vision toolbox uses the Haar features to detect the eyes [132, 133]. After conducting some tests, it was found that, out of 7 images, shown in Figure 3.10, MATLAB toolbox mis-identified one face. This is because, one eye was slightly covered with hair, shown in the bottom right image of Figure 3.10.

Even though the object detector was developed for species such as possums and cats, it also works well in the human eye-detection domain. A limitation of this technique is the parameter space of a Circle Hough Transform, which is a three-dimensional technique that requires a considerable amount of storage and computational power.

Toennies et al. [134] used the Hough transform to detect the irises, and this investigation, used close-up enlarged pictures of eyes. But for animal eye-detection technique, the whole face is used for the standard-deviation and image slicing processes, which improved the detection rate. Hough transform-based facial feature extraction studies by Iwasa et al. [135] and Ito et al. [136] analysed the whole face with traditional edge-detection techniques. As they used the whole image, the Hough transform produced many different candidates (greater than 50), so finding the most suitable pair of candidates was more complex. The developed process in this research reduces the number of suitable candidates to the minimum number due to standard-deviation face and image slicing.

Figure 3.10: Columns from left to right. Column 1: Original images,
Column 2: Developed sliced and circle detected image,
Column 3: Developed eye-detected image, and
Column 4: MATLAB vision toolbox eye-detection (using Haar-like features)

3.2 Image Correction

Once the eyes have been detected on the image, the central point of the face needs to be determined, which is done by calculating the horizontal distance between the eyes. The centre is the midpoint point of the horizontal distance *d*. Before attempting any amendments, the centre point of the animal's face needs to be centred at the centre of the image. If there is any angular correction, then the animal's face will be rotated about the central point *C*.

When an image is centred, the orientation of the animal's face can be calculated using the angle between the eyes. If the whole face of the animal is sitting on a two-dimensional plot; the left-eye is sitting on one quadrant and the right-eye is sitting on the diagonally opposite quadrant. Depending on the face rotation, the two eyes could be on either of the diagonally adjacent quadrants. In order to calculate the angle θ between eyes to the centre point (midpoint point of the face of animal), the vertical height *h* between two eyes and the horizontal distance *d* need to be calculated, shown in below Figure 3.11.

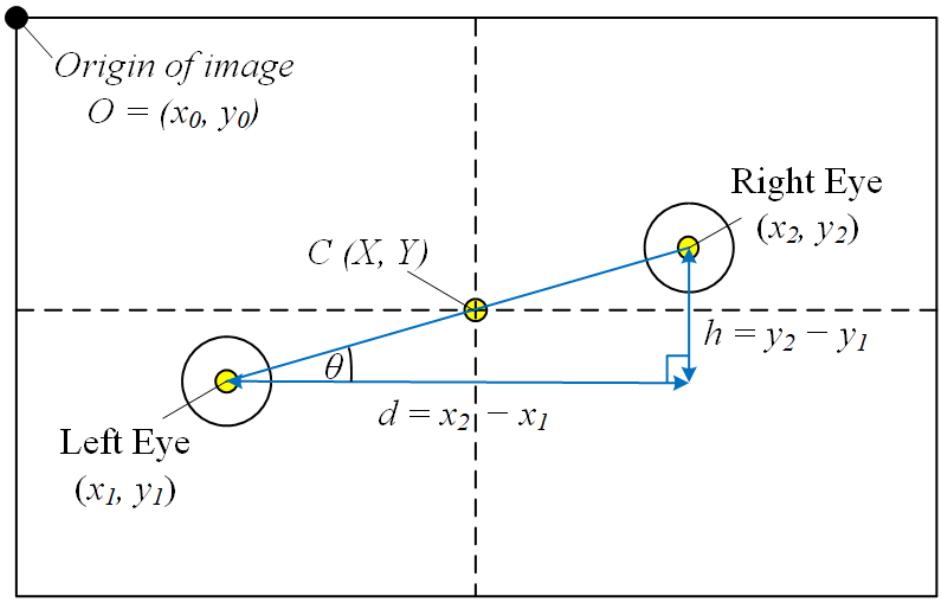


Figure 3.11: Eye location measurements.

After h and d have been derived, the angle θ in Figure 3.11 can be calculated using equations (3.4) and (3.5). Depending on the orientation of the face image, θ can have a positive or negative value.

$$tan(\theta) = \frac{h}{d} \tag{3.4}$$

therefore:

$$\theta = \tan^{-1}\left(\frac{h}{d}\right) \tag{3.5}$$

Taking the centre of rotation to C to be $x = X$ and $y = Y$, both coordinates are measured from the top left-hand corner of the image to be the origin of the image, shown in Figure 3.11.

The image is rotated by the angle θ, giving new coordinates of the left-eye referred to the origin position O *are* given by equations (3.6) and (3.7):

$$x_{(left-eye\ new\ coordinate)} = X + (x_1 - X)cos\theta - (y_1 - Y)sin\theta \tag{3.6}$$

and

$$y_{(left-eye\ new\ coordinate)} = Y + (x_1 - X)sin\theta + (y_1 - Y)cos\theta \tag{3.7}$$

The new coordinates of the right-eye reference to the origin position *O are* given by equations (3.8) and (3.9):

$$x_{(right-eye\ new\ coordinate)} = X + (x_2 - X)cos\theta - (y_2 - Y)sin\theta \quad (3.8)$$

and

$$y_{(right-eye\ new\ coordinate)} = Y + (x_2 - X)sin\theta + (y_2 - Y)cos\theta \quad (3.9)$$

For the image to be fully rotated about the central point *C* of the image, every pixel must be rotated using equations (3.10) and (3.11):

$$x_{p(i,j)} = X + \left(x_{p(i,j)} - X\right)cos\theta - \left(y_{p(i,j)} - Y\right)sin\theta \quad (3.10)$$

and

$$y_{p(i,j)} = X + \left(x_{p(i,j)} - X\right)sin\theta + \left(y_{p(i,j)} - Y\right)cos\theta \quad (3.11)$$

where $p(i,j)$, is the pixel location of the image.

There are some pixels around the edges that will not have a new location within the image canvas. In this case any pixels outside the boundary are ignored. Similarly, there will be new locations that will not have any pixel values to transfer to. In this case these values are set to zero (black). The pixels around the edges do not have any useful information apart from a black background. Therefore, these pixels can be removed and create new ones. After the image been rotated about the Y and X axis it can be used as part of the training set, shown in Figure 3.12.

Figure 3.12: Original image (left) and Corrected image (right)

Chapter 4
Eigenface for Cat and Possum Face Identification

Eigenface-based recognition is widely used in the human face-recognition domain, but less commonly applied to the animal face-recognition domain [88, 137]. The application developed in this research, is targeted at cats and possums as the species of interest. The reason for targeting cats and possums in New Zealand is to implement a vision-based trapping system. This chapter presents two image-classifiers developed for use with Eigenface techniques to separate the two different species in question.

When original images of cats and possums were used with the Eigenface technique, the recognition rate was found to be extremely poor. This was mainly due to face orientation and background information issues of the images. This chapter will describe the improvements, such as background removal, species grouping and background colour optimisation, developed to solve these problems, along with the following points:

1. The development of a new method for separation of Eigenface outputs. This technique involves transformation of multidimensionally-scattered data clusters onto a 2D graph, which allows the user to see the separation of different class clusters.
2. The inclusion of multivariate normal probability theory [138, 139] as a classifier for Eigenface technique, to find the probability of a given image belonging to each species.
3. One of the main problems in existing Eigenface technique is, the length of the eigenvectors seems to be arbitrary and they are not normalised [44, 45, 54, 55, 140]. Therefore, all the training weights have incorrect lengths in multidimensional space [141]. This causes poor separation of classes in eigenspace.
4. The improvement of the Eigenface technique by a new eigenvalue-based distance algorithm. This algorithm provides weighted distance to each training image. This technique is further improved by introduction of an error weight-based Eigenface classifier, which enables further separation of the targeted animal classes.

4.1 Image Refinements

Image refinement is a crucial part of the Eigenface technique, as previously mentioned. Without any refinements, the detection rate of this technique is unacceptable. It was found that background of the image details tends to have dominant eigenvectors with larger eigenvalues [54].

4.1.1 Image Background Modification

During the initial stages of this research animal images with different backgrounds were used. The images with random backgrounds produce strong eigenvectors primarily related to the background information, which reduced the detection rate [88].

To solve this problem, all the training set images were pre-processed by replacing the backgrounds with uniform colour, shown in Figure 4.1. After pre-processing, all the eigenvalues had very strong correlation with the actual animal face rather than the background.

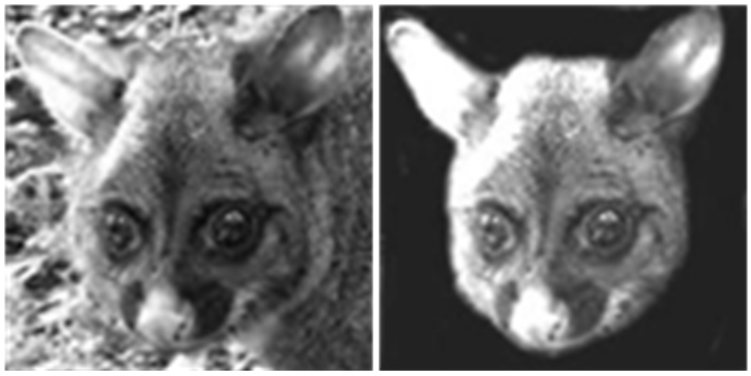


Figure 4.1: Possum image before (left) and after (right) pre-processing.

In real-world implementation pre-processing of the image would not be necessary, as the background colour can be set as a desired colour inside the trap wall and be uniform. The configuration of the trap system, would eliminate the animal's body, shown in Figure 4.2. The animal would go head first into the trap entrance and the infrared sensor detects the movement and a linear solenoid-based jaw locks the head in place against the bottom plate. The face and the background wall would be exposed to the camera, a light-emitting diode (LED) lighting turns on and the camera captures the face image. For this research no trap was implemented, so the background removal of the available images was performed manually, using a Kalman filter-based background identification [142, 143].

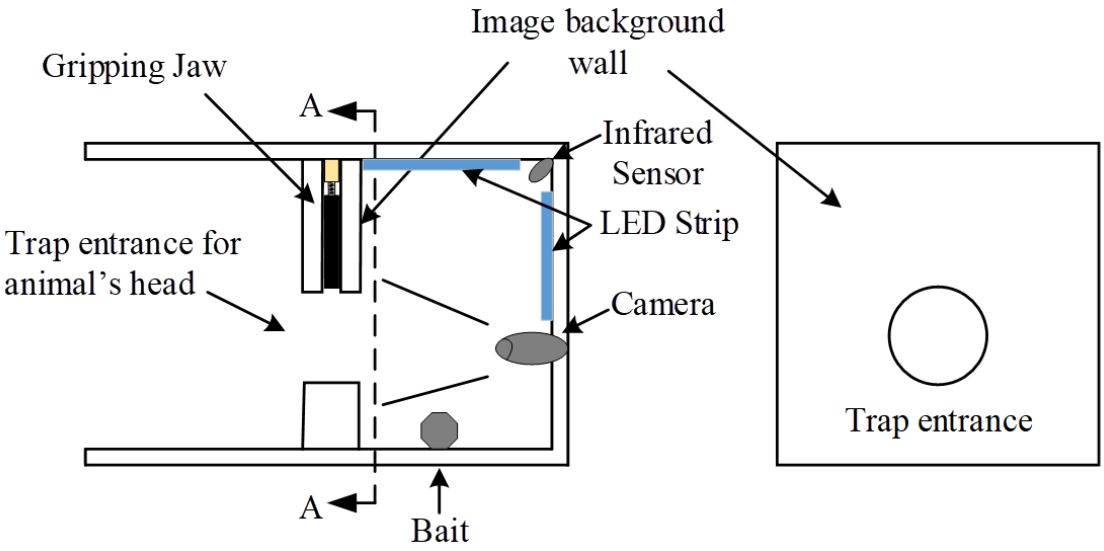


Figure 4.2: Proposed trap and camera setup

4.1.2 Species Grouping Using Sub-Training Set-Based Approach

The sub-training set-based approach uses different training sets for the same animal. These training sets are optimised according to the animal's colour and camera pose angle [88].

In the animal detection domain, an underlying problem for the Eigenface technique is that animals within the same species have different texture patterns, colour variation and face features. As mentioned in chapter 1, New Zealand possums have two different colours, namely grey and dark brown, shown in Figure 4.3. It was found that when using differently-coloured possum images accuracy of the Eigenface technique deteriorates [88].

Figure 4.3: Different-coloured possums

As most animals tend to have different poses another problem is related to finding images of the animal face at the correct orientation. In order to use the Eigenface application all the faces should to be oriented correctly i.e. looking forward. Training sets with incorrectly oriented images produce poor detection rates [44, 55, 88].

Due to the two reasons mentioned, the detection rate of the Eigenface technique for animal species is not acceptable. To overcome this problem, the sub-training set-based Eigenface technique was developed. In this technique all the likely subjects are grouped into one category. For example, all dark-coloured possums are put in one category and all light-coloured possums in another category. In order to improve the system accuracy further, these categories are broken down into further sub-groups such as light-coloured possum looking left and light-coloured possum looking right etc., shown in Figure 4.4, from which each of these categories is used as a separate training set.

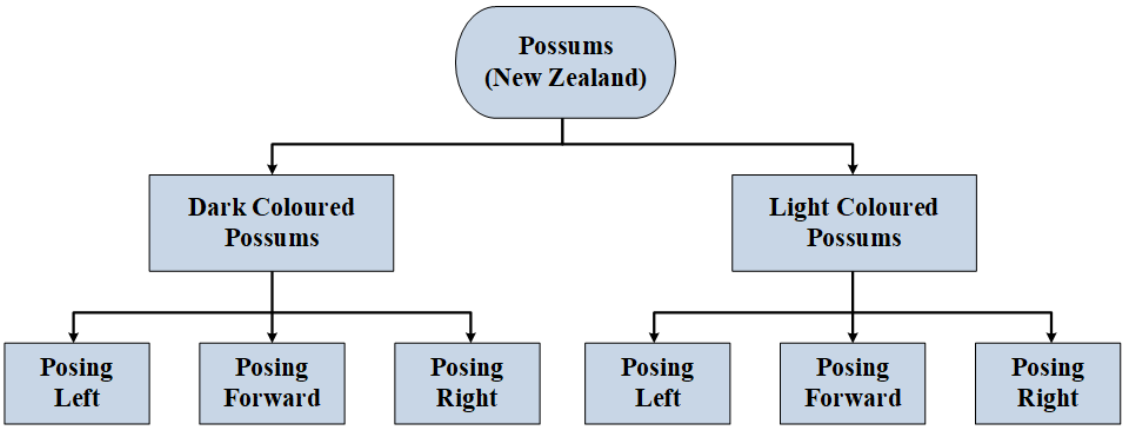


Figure 4.4: Sub-categorizing images of New Zealand possums

Animals such as cats were harder to group into colour-based groups as they have different colours and patterns on their fur. Hence for this research they were categorised by their pose. If cats were grouped by similar colour or their breed, there would be too many sub groups and hence there will not be enough training images for each sub-group for Eigenface technique to work. Cats were therefore categorised as either looking left, looking forward or looking right, shown in Figure 4.5.

Figure 4.5: Images of Cats facing in different directions i.e. left, straight ahead and right

When the images are grouped with the background removed, the sub-training set-based approach can be applied. The computational steps mostly follow those of the Eigenface detection technique, with the difference that there are multiple sets of training data sets to be formatted. The data sets are then used to generate the Eigenfaces for a given image. The process can be expressed as follows to obtain training sets *'I_i'* for:

(a) Dark coloured possums:
    (i) Posing Left (PDL)
    (ii) Posing Forward (PDF)
    (iii) Posing Right (PDR)

and

(b) Light-coloured possum:
    (i) Posing Left (PLL)
    (ii) Posing Forward (PLF)
    (iii) Posing Right (PLR)

For each sub-group the Eigenface technique was carried out and covered in section 2.10 of chapter 2. Table 4.1 shows the group represented after each iteration.

Table 4.1: Sub-Training Set-Based Eigenface steps and symbols

Explanation	Symbols
Each sub group I_i	$I_i(PDL), I_i(PDF), I_i(PDR), I_i(PLL), I_i(PLF), I_i(PLR)$
Vectorised training images Γ_i	$\Gamma_i(PDL), \Gamma_i(PDF), \Gamma_i(PDR), \Gamma_i(PLL), \Gamma_i(PLF), \Gamma_i(PLR)$
Average face each sub group m	$m(PDL), m(PDF), m(PDR), m(PLL), m(PLF), m(PLR)$
Normalised face from each training set Φ_i	$\Phi_i(PDL), \Phi_i(PDF), \Phi_i(PDR), \Phi_i(PLL), \Phi_i(PLF), \Phi_i(PLR)$
Covariance matrices C for each training set	$C(PDL), C(PDF), C(PDR), C(PLL), C(PLF), C(PLR)$
Eigenvectors e_i for each training set	$e_i(PDL), e_i(PDF), e_i(PDR), e_i(PLL), e_i(PLF), e_i(PLR)$
Eigenvectors with largest eigenvalues λ from covariance matrices C of each training set	$\lambda(PDL), \lambda(PDF), \lambda(PDR), \lambda(PLL), \lambda(PLF), \lambda(PLR)$
For each unknown image l^2-norm, e_d is calculated for each sub-group	$e_d(PDL), e_d(PDF), e_d(PDR), e_d(PLL), e_d(PLF), e_d(PLR)$

The training set with lowest l^2-norm distance is selected as the best match and it was found that this selection process is more accurate than the Eigenface technique with a large database containing all the animal images.

4.1.3 Detection Rates for Refined Images

The initial set of 4 trials was conducted using one large training set, containing possum and cat images with different poses, which were:

1. All the original images with existing background for both animals, were analysed using the Eigenface algorithm.
2. All the original images with existing original background for both animals, were analysed using the Sub-Training Set-Based Eigenface algorithm.
3. All the original images with a black ground for both animals, were analysed using the Eigenface algorithm.
4. All the original images with a black ground for both animals, were analysed using the Sub-Training Set-Based Eigenface algorithm.

The results for each trial are shown in Figure 4.6.

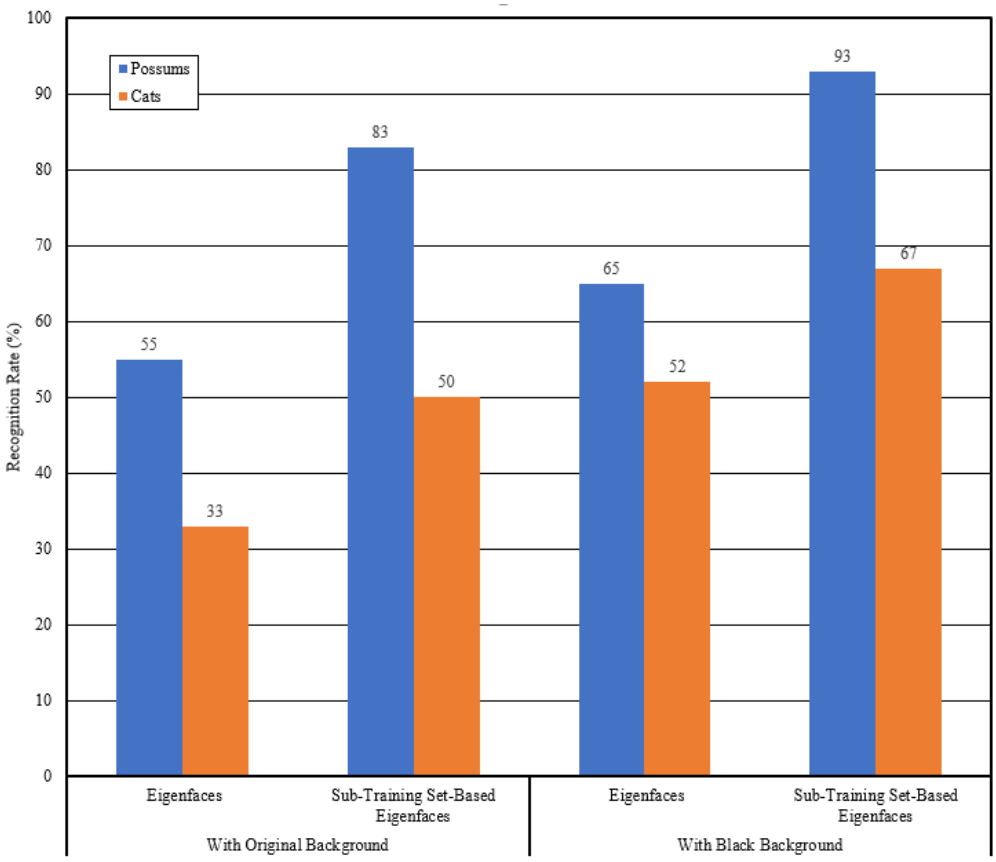


Figure 4.6: Recognition rates of possum and cat images

Using all the images with their original background, the results for both techniques used show that the possum detection has a higher detection rate than for cats. That is due to there being smaller variations between different possum images compared to the cat images.

Changing the background of the original images to black resulted in a significant improvement in detection rate for both techniques. This is because all the eigenvalues have a significant representation of the animal face rather than the background information. For with possums, the detection rate is again higher than that for the cat detection.

The overall result also shows that the Sub-Training Set-Based Eigenface technique offers a significant improvement in detection rates compared to the Eigenface technique.

4.2 Improving the Accuracy with Normalised Eigenfaces

The Eigenface technique consists of the 6 steps covered in section 2.10 of chapter 2. First the training images (I_i) are converted into column vectors (Γ_i), then from these column vectors the average face was calculated (m). All the training images are normalised by subtracting the average face, which is then used to calculate the covariance matrix (C). The covariance matrix $C = A^T A$ is used, to reduce the size and the computation time. Even though this step results in the loss of data, there is still enough data to recreate the original image [54, 55]. The eigenvalues and eigenvectors are then calculated from the covariance matrix. Since the mean is subtracted from the training images the first eigenvector (the eigenvector with smallest eigenvalue) does not contain any meaningful magnitude and the eigenvalue is extremely low in the order of $> 10^{-6}$, so the first eigenvector and eigenvalue are removed from the matrix. This step is not a part of the standard Eigenface technique, but if these vectors are not removed one of the Eigenfaces is not a meaningful representation in face space [144].

Next the eigenvectors are multiplied with the normalised training image column vectors, and after the multiplication each column vector represents an Eigenface u_i In this step all the training images are transformed from high-dimensional space to lower-dimensional space or face space. In the standard Eigenface technique, these column vectors or Eigenface column vectors are not normalised [44, 45, 54, 55, 140]. If Eigenface vectors are not normalised, the calculated weights w using these un-normalised vectors will have

arbitrary magnitudes in weight space. These weights are calculated by multiplying Eigenfaces with normalised training images given by equation (4.1):

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \tag{4.1}$$

where $w_i = u_i^T \Phi$

In order to correct the weighting of each training image in weight space, all the Eigenface vectors need to be the normalised $\hat{u}_i$, given in equation (4.2). This step is critical for separating different animal species classes in weight space.

$$\hat{u}_i = \frac{u_i}{\|u_i\|} \tag{4.2}$$

where $\|u_i\|$ the norm of u_i. Thus, $\hat{u}_i$ a unit vector.

The Eigenface classifier is based on distance e_d from face space. In order to calculate this distance for an unknown image Γ, the image needs to be normalised by subtracting the average face m given by:

$$\Phi = \Gamma - m \tag{4.3}$$

Then the image is reconstructed with calculated weights from equation (4.1). Finally, the normalised unknown image is subtracted from reconstructed image to calculate the distance e_d from face space, given by:

$$e_d = \left\| \Phi - \hat{\Phi} \right\| \tag{4.4}$$

If e_d is less than a preset threshold value, then Γ is belongs to one of the training species in the database, otherwise it is not.

4.3 Eigenface Technique for Cats and Possums Detection with Individual Minimum-Distance Classifier

This section presents an Eigenface technique for detecting specific species with a new modified classifier. In this approach one large training set was used, with 7 possum images and 7 cat images, where each training image resolution was 170 × 170 pixels, shown in Figure 4.7. The backgrounds of the all images in this database were changed to grey scale[20] with an intensity of 131, this value is the average value of the backgrounds of all the original training images. In this another manipulation technique such as face rotation was performed.

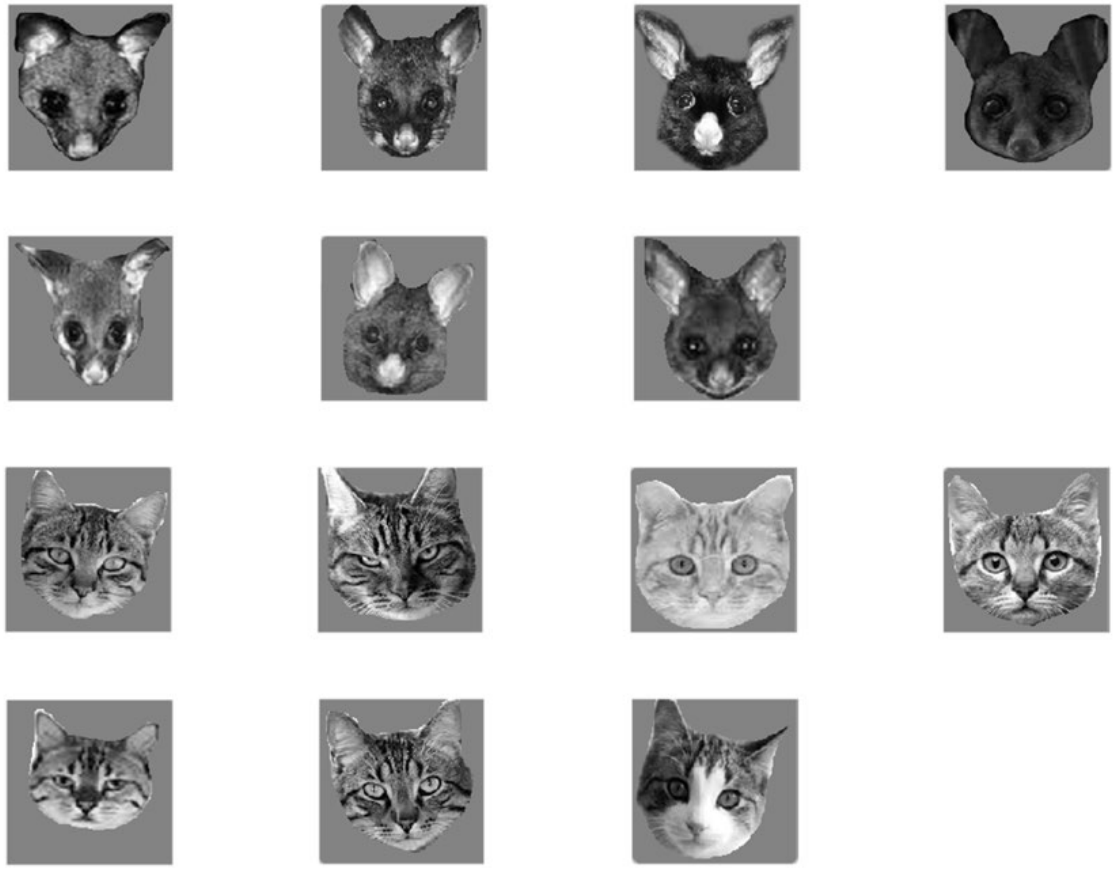


Figure 4.7: Training images of possums (top two rows) and cats (bottom two rows)

[20] A range of grey shades from white to black, as used in a monochrome display, image or printout.

The Eigenfaces are calculated, the weights for the training images as shown in Figure 4.8.

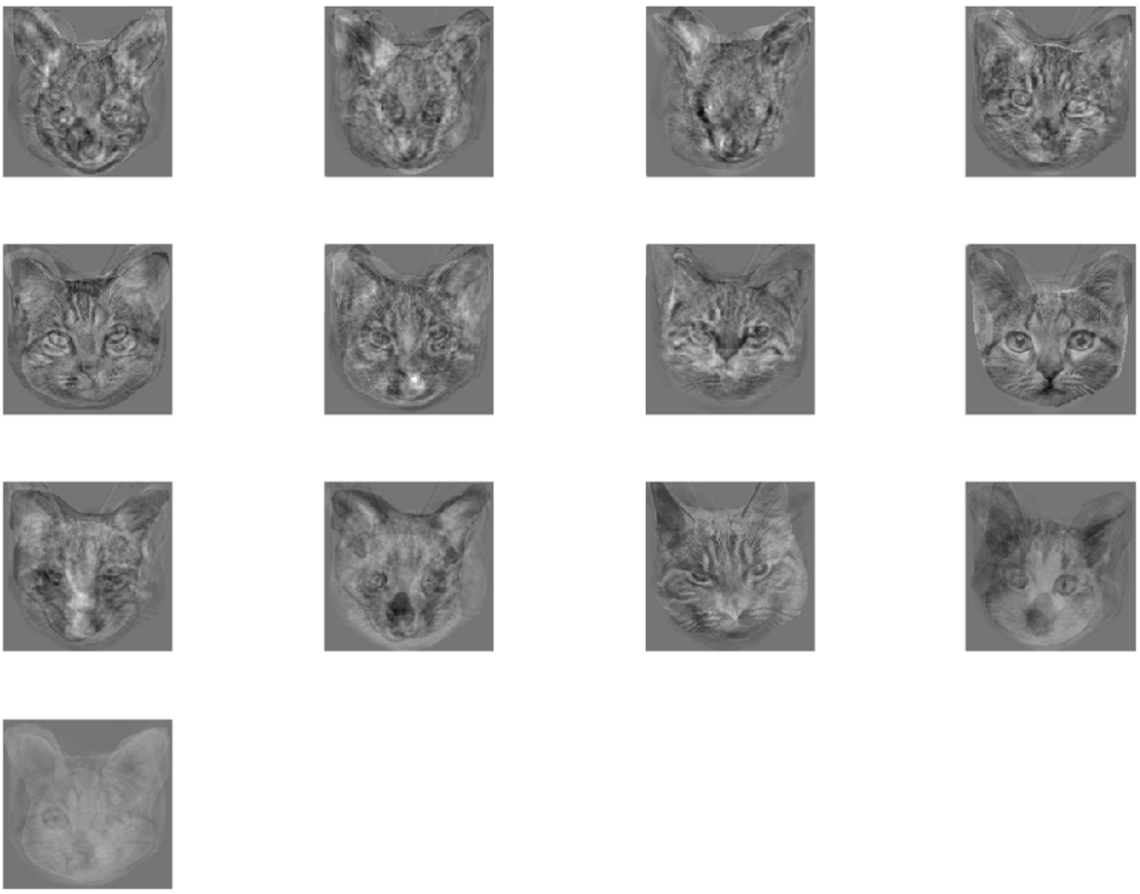


Figure 4.8: Eigenfaces of the training images

In order to calculate weights, the Eigenfaces need to be multiplied by the normalised training image values given in equation (4.1). In this case there will be 14 weight vectors representing each training image and each vector contains 13 weights. Since the average face is subtracted from the training images, one weight is lost, so there are 13 weights in each weight vector. So, for an unknown animal image Γ is normalised by subtracting the average face, and weights calculated by multiplying the normalised test image with the Eigenfaces. Since there is only one test image, it will be one weight vector with 13 individual weights, and this is the feature vector or weight which represents the unknown test image.

The distance from each training image to test image is then calculated, and the shortest distance out of all 14 different training images is selected as the closest match. If the calculated minimum distance is greater than a set threshold, the unknown image does not belong to any of the targeted species.

In order to calculate the distances e_r between the test image and training images, each training weight column vector is subtracted from the test weight vector, and take the sum of squared values of the resulting vector, which is given by:

$$e_{r_i} = \sum_i \left(\widetilde{w_i}^{Training} - \widetilde{w}^{Test} \right)^2 \tag{4.5}$$

The minimum distance is selected, which represents the closest image to the test image in the training set, which is given by:

$$e_r = \min\left(e_{r_i}\right) \tag{4.6}$$

This classifier searches for the training image closest to the test image in low-dimensional weight space. The developed classifier was tested with 16 possum images and 7 cat images. All the test images had the same background of a grey scale with an intensity value of 131, shown in Figure 4.9. The detection rate was 87% for possums and 71% for cats. Again, possums had a higher detection rate due to their fur colour being uniform across all the training and test images. Cats have more varied fur colours i.e. many cats have unique colour pattern, and therefore the algorithm mostly matches the head outline information for detection.

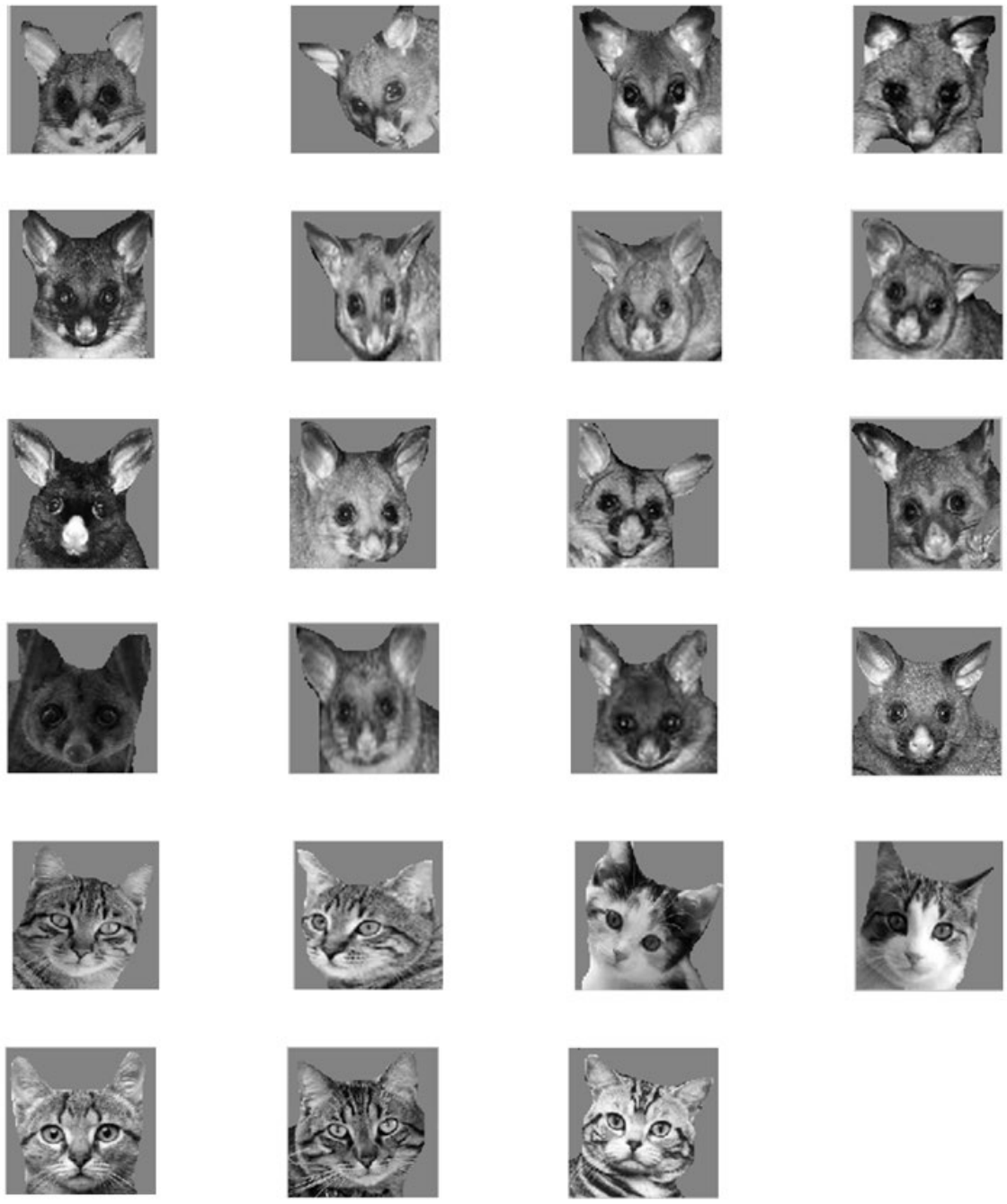


Figure 4.9: Unknown test images

Even though the detection rate is significantly higher than the standard Eigenface classifier, it is still not at an acceptable level. The main limitation of this classifier is that performance is compromised by the different fur patterns of the cat's images. Therefore, further investigation was required to detect both animals by face shape rather than by fur details.

4.4 Eigenface Technique with Developed Distance Algorithm

The Eigenface technique with the addition of a new distance algorithm can be described as follows. The size of the training image I_n is an M×1 vector and the training set Γ is given by:

$$\Gamma = [I_1, I_2, I_3, \ldots\ldots, I_n] \tag{4.7}$$

The average face m calculated by training set Γ is:

$$m = \frac{1}{M}\sum_{i=1}^{M} \Gamma_i \tag{4.8}$$

Then the training set is normalised by subtracting the average face:

$$\Phi_i = \Gamma_i - m \tag{4.9}$$

and the covariance matrix C for image A is calculated using:

$$C = \frac{A^T A}{N-1} \tag{4.10}$$

where $A = [\Phi_1 \Phi_2 \ldots \Phi_M]$ and $A^T A$ was used instead of AA^T.

The eigenvalues λ_i and eigenvectors e_i are calculated, where:

$$A^T A e_i = \lambda_i e_i \tag{4.11}$$

then:

$$AA^T(Ae_i) = A\lambda_i e_i = \lambda_i(Ae_i) \tag{4.12}$$

Therefore, the normalised Eigenfaces are equal to:

$$f_i = \frac{(Ae_i)}{\lceil (Ae_i) \rceil} \tag{4.13}$$

Hence for the full covariance matrix *C,* the eigenvectors are f_i, and the variance is:

$$\sigma^2 = \frac{\lambda_i}{N-1} \tag{4.14}$$

Then for image *J*, the weights are given by:

$$w_i = (J - m) \times f_i \tag{4.15}$$

and distance d can be calculated using:

$$d = \sqrt{\sum_i \frac{w_i^2}{\sigma_i^2}} = \sqrt{\sum_i \frac{w_i^2}{\lambda_i}(N-1)} \tag{4.16}$$

The distance formula given by equation (4.16) is a measurement of the distance of each image from the mean. This algorithm compensates for the effect of the larger weights by dividing them by their eigenvalue, effectively the variance. Significant eigenvectors have larger eigenvalues. Significant eigenvectors also produce significant weights. For example, if there are 5 images in a database, the covariance matrix size is 5 × 5. Once it is converted to eigenvectors and eigenvalues, it can be observed that significant eigenvectors have larger eigenvalues. Similarly, eigenvectors with large eigenvalues produce large weights, shown in Figure 4.10. As described in section 4.2 of this chapter, the first vector of the eigenvector matrix is not meaningful, due to the normalisation of the training images. This is shown in the eigenvalue diagonal matrix, where the first eigenvalue's magnitude is insignificant compared to other eigenvalues, shown in Figure 4.10.

$$\begin{bmatrix} 31006935 & -5599040 & -10630866 & -13034741 & -1742287 \\ -5599040 & 21913885 & -10561813 & -7322344 & 1569313 \\ -10630866 & -10561813 & 42335477 & -12297499 & -8845298 \\ -13034741 & -7322344 & -12297499 & 57234979 & -24580394 \\ -1742286 & 1569312 & -8845298 & -24580394 & 33598667 \end{bmatrix}$$

Covariance Matrix

$$\begin{bmatrix} 0.4472 & 0.0458 & -0.8255 & -0.2806 & 0.1943 \\ 0.4472 & 0.7657 & 0.3865 & -0.2372 & 0.0896 \\ 0.4472 & 0.0202 & -0.0353 & 0.8881 & 0.0985 \\ 0.4472 & -0.2352 & 0.0706 & -0.1227 & -0.8513 \\ 0.4472 & -0.5965 & 0.4037 & -0.2475 & 0.4689 \end{bmatrix}$$

Eigenvector Matrix

$$\begin{bmatrix} 1.49E-08 & 0 & 0 & 0 & 0 \\ 0 & 22326534 & 0 & 0 & 0 \\ 0 & 0 & 35140068 & 0 & 0 \\ 0 & 0 & 0 & 52680399 & 0 \\ 0 & 0 & 0 & 0 & 75942942 \end{bmatrix}$$

Eigenvalue Matrix – where the main diagonal of the matrix represents the Eigenvalues

Figure 4.10: Covariance matrix into Eigenvectors and Eigenvalues

Belhumeur et al. [44] found that the three eigenvectors with the largest eigenvalues mostly contain the information about background lighting and removing these improved the accuracy of the classifier. The new distance algorithm given in equation (4.16) reduces the effect of the significant eigenvectors, by dividing the significant weights by their eigenvalues effectively from its variance.

Removal of the significant eigenvectors may eliminate some useful information from the training set. This variance-based technique retains all the information but at the same time reduces its effect on the overall results.

4.5 Species Class Separation with New Distance Algorithm

Using equation (4.16), the distance between the mean image and a given image can be calculated. In possum and cat detection, there will be two different distances for any given test image. In order to develop a classifier to discover on which side of the decision boundary the unknown image lies, the shortest distance to the class mean could be used. But this is not an acceptable technique because when a species outside of a known class enters the trap setup, then it will not be distinguished. The solution to this problem is to set up a threshold boundary larger than a certain value, but this is not a reliable method of detecting targeted species used in this research.

It is hard to define the decision boundary between the two animal classes in multidimensional space. Therefore, it is necessary to implement a new technique which converts distance data from multidimensional space to 2D space, so that a meaningful decision boundary can be defined.

In order to transform multidimensional space data into 3D space, trigonometrical space is used. Since there are two l^2-norm distances (i) cat distance and (ii) possum distance in multidimensional space, these two distances can be plotted in trigonometrical space. It can be assumed that the cat and possum mean images are separated by a certain distance in trigonometrical space. In trigonometrical space, the minimum of the sum of both distance vectors of the cat and possum distances is used for possum and cat mean class separation. Then the calculated distances can be plotted from its mean. The angle of both distance vectors can be calculated according to the cat and possum distances where the two vectors meet in trigonometrical space, shown in Figure 4.11.

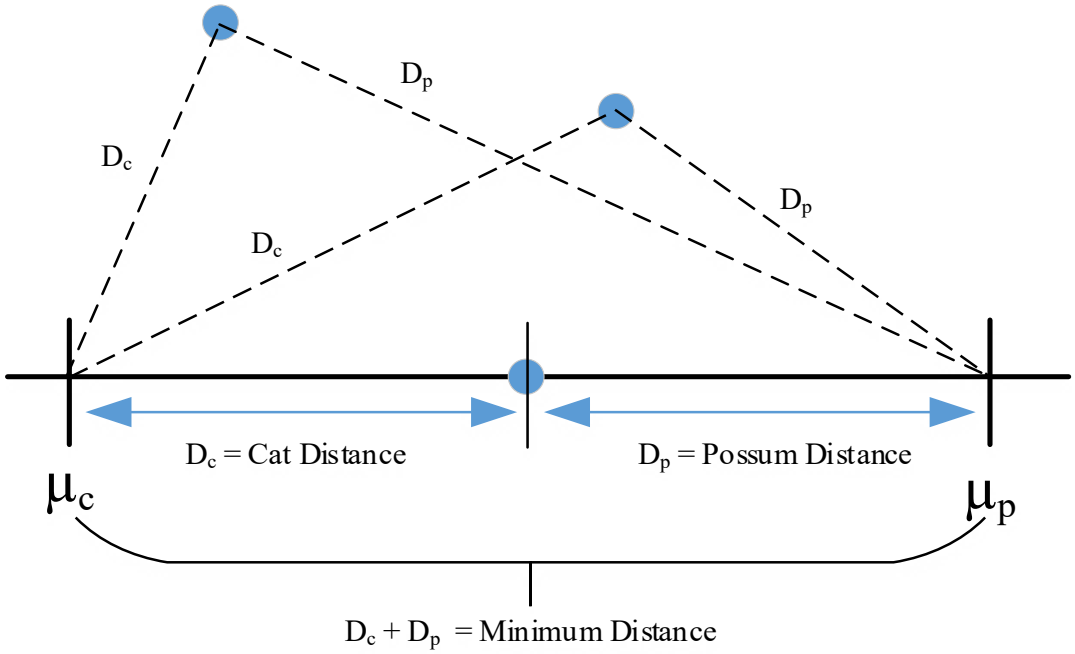


Figure 4.11: Image distances are on trigonometrical space

In order to plot these points, three scenarios needed to be taken into consideration, they are:

Case 1. The cat training set distance and the possum training set distance are similar, so all the points will scatter between the cat and possum mean values, shown in Figure 4.12.

Case 2. One of the distances is considerably larger than the other. Therefore, the point will pass one of the mean values, shown in Figure 4.13.

Case 3. One distance is much larger than the other and hence the two magnitudes will not be able to yield a point. But this application did not have case three among calculated data. The other two cases can be calculated using the developed algorithms below.

Case 1:

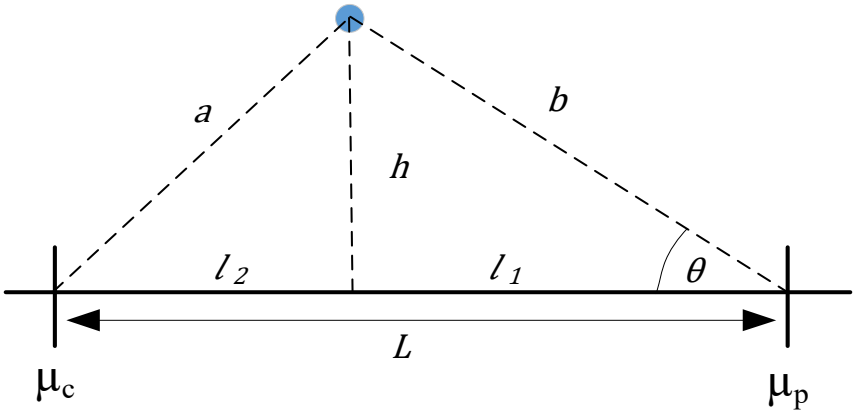


Figure 4.12: Case one

Using the cosine law a^2 can be expressed as:

$$a^2 = b^2 + L^2 - 2bL\cos(\theta) \tag{4.17}$$

and so

$$\cos(\theta) = \frac{b^2 + L^2 - a^2}{2bL} \tag{4.18}$$

and

$$L = l_1 + l_2 \tag{4.19}$$

where

$$l_1 = b\cos(\theta) = \frac{b^2 + L^2 - a^2}{2L} \tag{4.20}$$

$$l_2 = \mathrm{L} - l_1 = \frac{a^2 + L^2 - b^2}{2L} \tag{4.21}$$

and

$$h = \sqrt{a^2 - l_2^2} = \sqrt{b^2 - l_1^2} \tag{4.22}$$

Case 2:

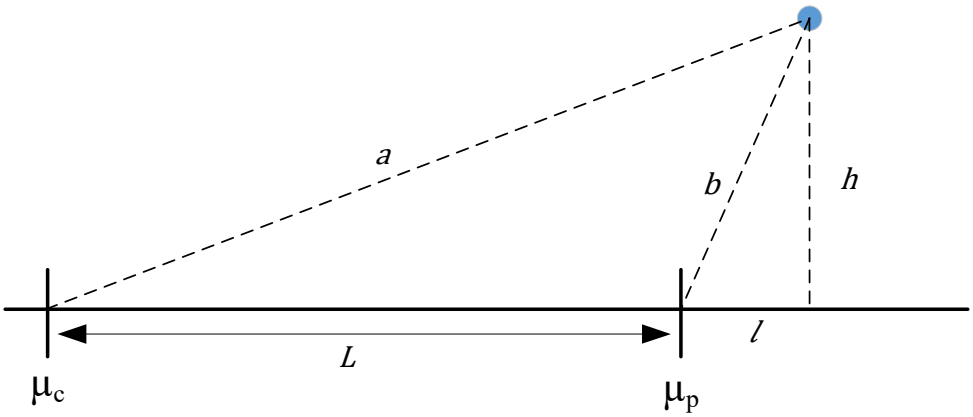


Figure 4.13: Case two

From the Pythagorean theorem, a^2 can be expressed as:

$$a^2 = h^2 + (L + l)^2 \tag{4.23}$$

where

$$b^2 = h^2 + l^2, \tag{4.24}$$

and so

$$a^2 = L^2 + 2lL + l^2 + h^2 \tag{4.25}$$

$$a^2 = L^2 + 2lL + b^2 \tag{4.26}$$

and

$$l = \frac{a^2 - L^2 - b^2}{2L} \tag{4.27}$$

$$h = \sqrt{b^2 - l^2} \tag{4.28}$$

Equations (4.21), (4.22), (4.27) and (4.28) can be used to calculate the point on the graph. If any point belongs to case three, then it will not be able to be plotted on the graph. Figure 4.14 shows an example of class separation by the new classifier. Once classes are separated, the decision boundary can be defined between two clusters.

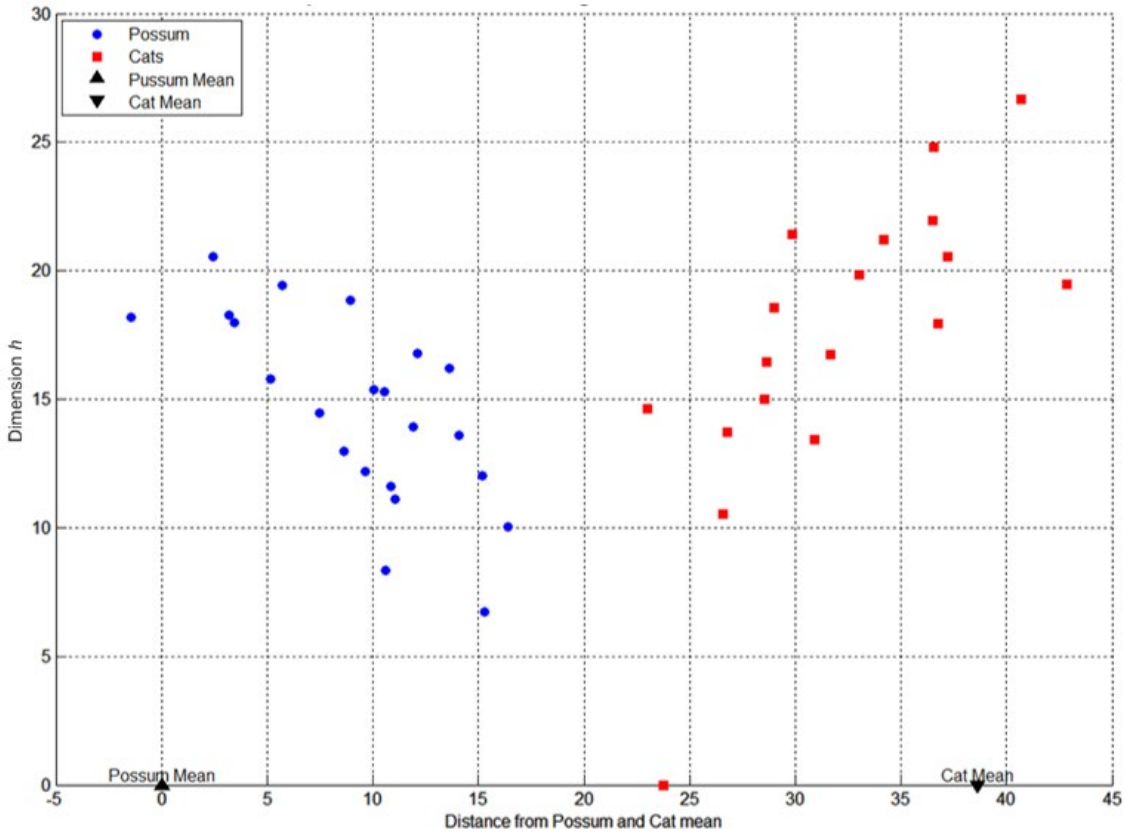


Figure 4.14: Example of the new classifier's data separation

4.6 Optimising Eigenface Technique for Detecting Cats and Possums Images by Face Outline and Features

This section describes a new approach that enhance the Eigenface technique's ability to detect the cats and possums images. The main objective was to separate the cat class and possum class as much as possible.

4.6.1 Improving Detection Rate by Optimising the Image Colour Scheme

One of the trialled approaches involved optimising the image colour scheme. During the initial investigations a few colour schemes were trialled.

The initial trialled colour scheme shown in Figure 4.15 was:

(a) The images background colour was set as grey intensity value of 131, the average value of all the backgrounds.
(b) Face colour as white intensity value of 255 and the nose and eye information were kept in the original colours.

The main issue with this colour scheme was that the area of nose and eyes was small compared to the rest of the image. The difference between grey and white was not significant enough to produce strong eigenvectors to extract the face outline information and the separation of the species classes was not significant.

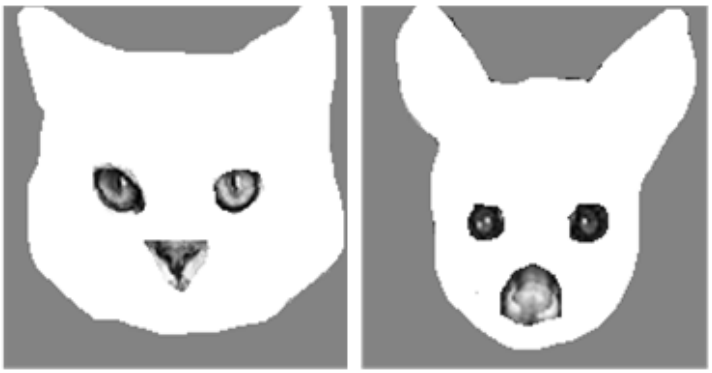


Figure 4.15: First trialled colour scheme

The second colour scheme trialled shown in Figure 4.16 was as follows:

(a) The colour of the background was set as black with an intensity value of 0.
(b) Face colour as grey with an intensity value of 160.
(c) Eye and nose colour as white with an intensity value of 255.

The main concept behind this scheme was to produce a large intensity change between the background and the face. When eyes and nose are white (255) they stand out more on the image and represent a significant intensity step change between face and features. This scheme produced much better separation between facial outline and background compared to the first scheme. Even though the eyes and nose have much higher intensity value compared to the face, they are still small in area.

Figure 4.16: Second trialled colour scheme

Since the eyes and nose were small regions, it was decided to analyse species separation without eyes and nose.

In the third scheme trialled the image was simplified further by changing the face colour to white with an intensity value of 255 and background colour to black with an intensity value of 0, shown in Figure 4.17. It was found that this configuration produced the best results due to maximum step change between background and face shape.

Figure 4.17: Third trialled colour scheme

4.6.2 Improving Detection Rate by Optimising Image Resolution

Image size is an important parameter that can be used to optimise the separation of the species classes. Image size is directly proportional to the size of the full covariance matrix, and the operation time of the algorithm can be improved with lower resolution images.

The main issue with taking a reduced-size covariance matrix, is that a large amount of useful data will be lost. For example, if there is a training set of 20 images and each image is 170 × 170 pixels, the actual covariance matrix will produce a matrix of 28,900 × 28,900. Since this matrix is computationally demanding, a reduced size covariance matrix of 20 × 20 is used. Even though the most useful eigenvectors are kept, a large portion of data is lost.

In order to improve this situation, one of the techniques developed involved reducing the image resolution. Since this technique is trying to detect animals by facial shape, their fur, nose and eye information is not vital. Therefore, the image size can be reduced to minimise the data lost. For example, consider the same training set of 20 images but this time all the images are 10 × 10 resolution. The actual covariance matrix will be 100 × 100. If the reduced size covariance matrix is used, its size will still be the same 20 × 20. But the overall data lost is a lot less than in the first example.

In order to find the optimal image resolution, three different image resolutions were investigated: 34 × 34, 17 × 17 and 10 × 10. In order to reduce the resolution, the image was divided into different size blocks. For example, to obtain 17 × 17 image, the original image (170 × 170) was divided into 17 × 17 blocks. Each block was 10 × 10 pixels. Then the sum of the pixels in each block was found. Finally, each sum was divided by pixel

count in each block and replaced the whole block with the calculated block average value, shown in Figure 4.18.

Figure 4.18: Image resolution (from left to right) 170 × 170, 34 × 34, 17 × 17 and 10 × 10

After this process all the images had black backgrounds and the facial edge contour colour was changed to different shades of grey, shown in Figure 4.19. During these trials it was discovered that a soft transitional edge produces better separation of the two classes, and it highlights different outline features with different shades.

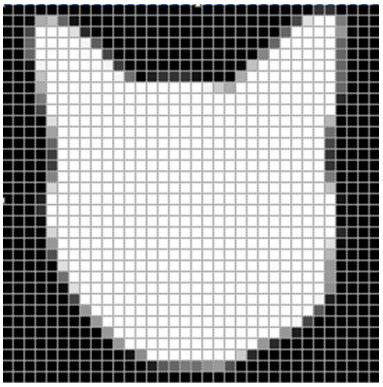


Figure 4.19: 34 × 34 image with grey facial contour

It was found that 40 × 40 was the resolution that gave the best separation between cat and possum classes.

4.6.3 Improving Detection Rate by Introducing Error Weight Vectors

The rationale behind this technique was to introduce a new weight vector to compensate for information omitted from the reduced covariance matrix *C*. As described in section 4.1.2, the cat and possum images were separated into two different training sets. Then the Eigenface technique is applied to both training sets. For this application equation (4.13) is used. At the end of the process, it ended up with two different training weights: one for the possums and one for the cats $\left(\tilde{w}^{C} \; and \; \tilde{w}^{P}\right)$.

For each image, weights w_i can be calculated using:

$$w_i = \Gamma_n f_n \tag{4.29}$$

where Γ_n is the image and f_n is its eigenface matrices.

Ideally:

$$i_m = \sum_k w_k f_k \tag{4.30}$$

$$i_m f_n = \sum_k w_k f_k f_n \tag{4.31}$$

if $k = n$ then $f_k f_n = 1$ nd if $k \neq n$ then $f_k f_n = 0$, so:

$$i_m f_n = w_n \tag{4.32}$$

In practice:

$$error \; e_m = i_m - \sum_k w_k f_k \tag{4.33}$$

Then $|e_m|$ can be used as an extra weight vector.

The equations (4.29) to (4.33) are used to calculate the weight vectors for the training images using the modified Eigenface technique described in section 4.4. The weights can be calculated using the Eigenface. Then these new weights are used to reconstruct the image i_m. Finally, the reconstructed image is subtracted from the actual image to find

the difference between the two images. The resulting difference is the extra error weight vector $|e_m|$ and this can be added to the existing weights.

Once the error weight is calculated from the image, a corresponding eigenvalue is required for the calculated vector. In order to find the next eigenvalue, existing eigenvalues can be used. By plotting an existing eigenvalue, it has an exponential decay, shown in Figure 4.20. The graph shows an exponential decay in existing eigenvalues. So this decaying phenomenon can be used to calculate the next eigenvalue.

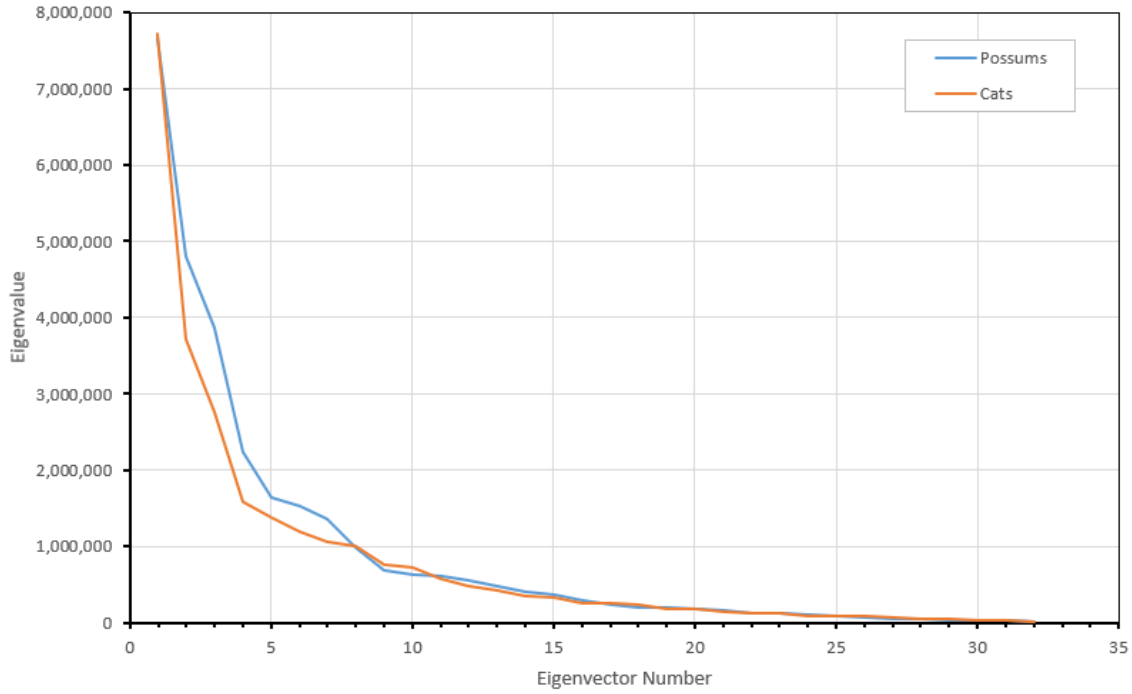


Figure 4.20: Graph of eigenvalues vs magnitude

The exponential decay formula can be used to find the next eigenvalue, given by:

$$N(x) = N_0\, e^{-\lambda x} \tag{4.34}$$

where x is an independent value and constant λ decay constant, where e^x is the exponential function and $N_0 = N(0)$ is the initial value.

The decay constant can be expressed as:

$$\lambda = \frac{-ln\left(\frac{N}{N_0}\right)}{x} \tag{4.35}$$

The equations given in (4.33) and (4.34)(4.35) can be used to calculate the extra eigenvalues for the cat and possum $\left(\tilde{\tilde{w}}^{C} \; and \; \tilde{\tilde{w}}^{P}\right)$ species classes. Adding the error weight and estimated eigenvalue makes the data classes to separate further and members within classes cluster together.

4.6.4 The Effect of Optimised Images (Colour and Resolution) and Introduced Error Weights

This section presents the results from the above colour scheme optimisation and the trials with different image resolutions. It also demonstrates the class separation improvements from the introduction of error weights and a new distance algorithm.

For these trials there were 66 training images in total: 33 possums and 33 cats. Forty test images were used. These were randomly selected and did not have any relationship to the training images. Out of 40 test images there were 18 cats and 22 possum images, shown in Figure 4.26 and Figure 4.27.

Figure 4.21 shows the effect of the various colour schemes on class. The poorest separation is produced by the original image with a grey background. The best separation is produced by a grey face, black background and white nose and mouth colour scheme. The second-best separation was produced by a black and white colour scheme. Similarly, as the resolution of the image increases, the class separation improves. But it was evident that the optimum resolution lay between 20 × 20 and 40 × 40. In order to simplify the image pre-processing tasks, it is best to use 40 × 40 resolution with black and white images. To implement this in an embedded platform, the pre-processing stage will be faster with a black and white image scheme and will result in an overall reduction in operation time for species identification.

Figure 4.23 shows the effect of the introduction of the new error weight algorithm into the Eigenface technique. The error weight separates the data classes further. When using original images to train and test, the class data separation was poor, shown in Figure 4.21,

but when using the same sets of trial with error weights, the class data begins to separate out.

When analysed the class data set with error weights, it is evident from Figure 4.21 that the best separation is with a grey face, black background and white eyes and nose colour scheme. The black and white colour scheme produced similar separation to the grey face, black background and white eyes and nose colour scheme.

With error weights both colour schemes have adequate data separation between cat class and possum class. As discussed previously, the best practical resolution is 40 × 40. These trials have determined the best colour scheme with maximum separation for cat and possum identification and established the optimal resolution for the training and test data. With a smaller resolution such as 40 × 40 the computation time of the algorithm is reduced dramatically.

When both data sets were analysed, it was evident that as the resolution of the training images increases the data tend to cluster in a vertical pattern. This is due to there being limited training images for the high dimensionality of the problem. As the resolution increases, the number of principal components stays the same. This is evident from the class separation not being much improved despite higher resolution. With lower resolutions data sets tend to spread out more. This is because there are enough training images for the dimensionality. Since there are 66 training images, 33 cats and 33 possums in total, the selection of 40 × 40 resolution gives a 1600-dimensional space. This seems to be the optimal point in terms of number of training images versus the dimensionality of the problem.

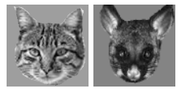
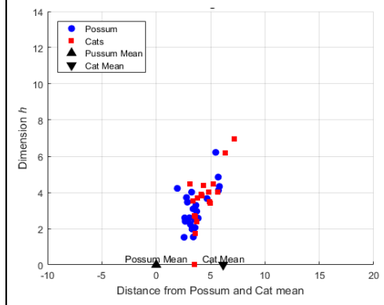
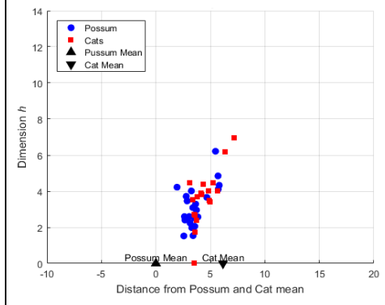
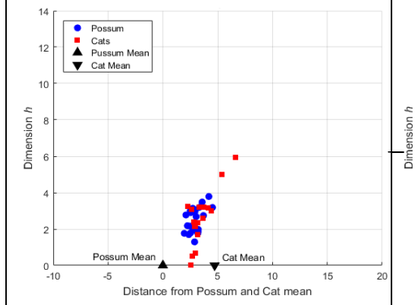
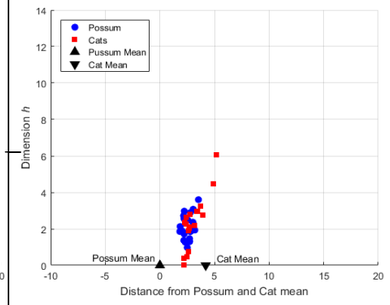
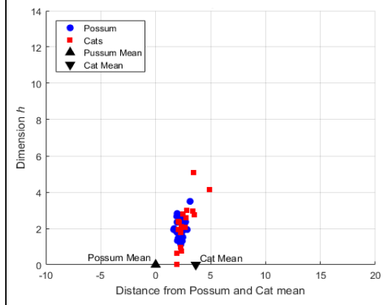
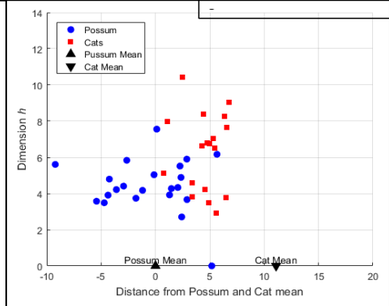
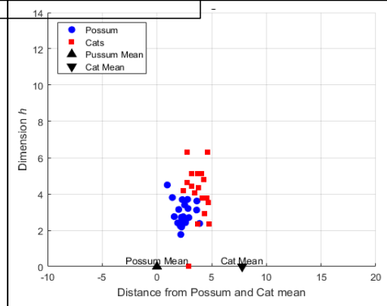
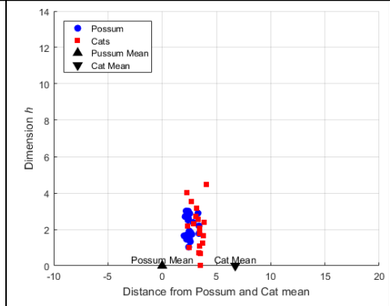
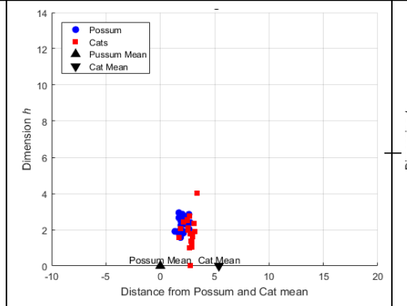
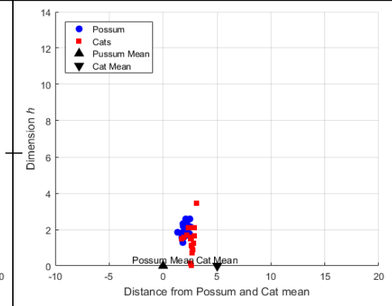
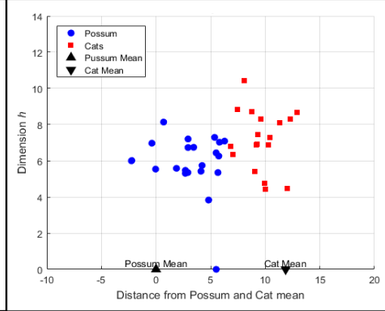
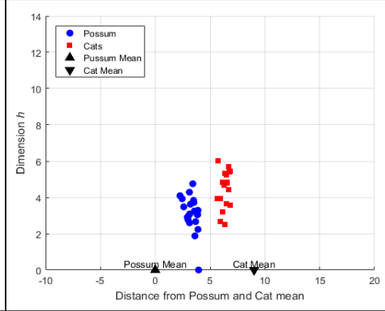
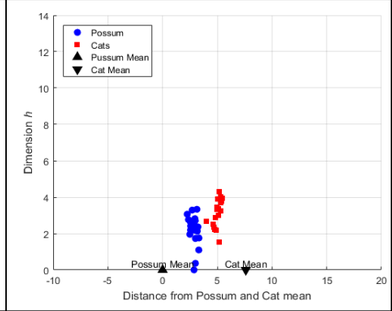
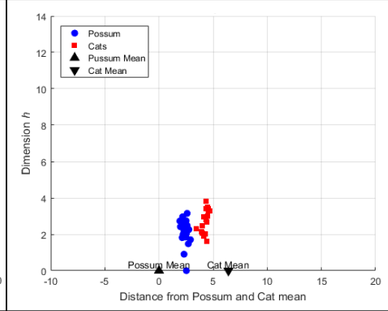
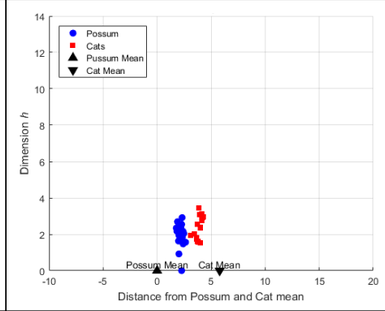
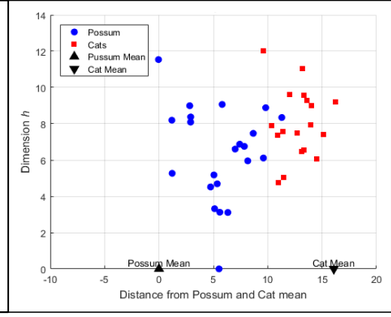
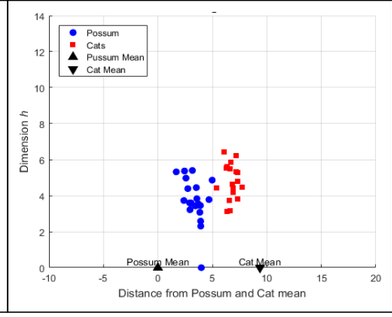
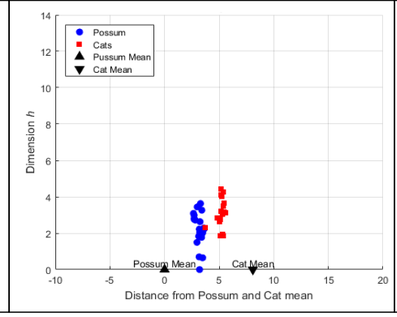
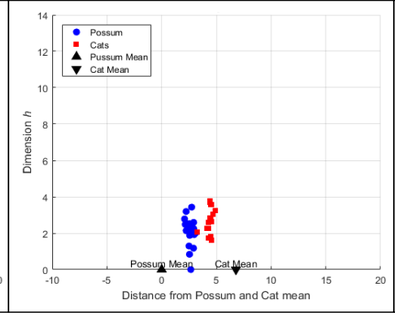
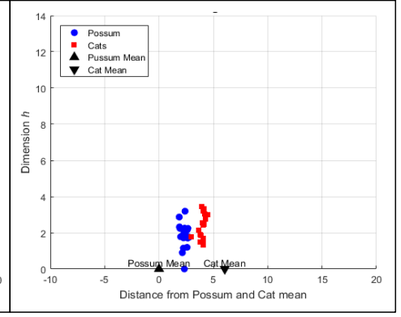
	Resolution 10x10	Resolution 20x20	Resolution 40x40	Resolution 80x80	Resolution 160x160
Original face and grey background 					
White face, Grey background and original eyes and nose 					
Grey face, black background and white eyes and nose 					
White face, black background and no eyes and nose 					

Figure 4.21: Eigenface species separation for improved colour schemes and different image resolutions using the Developed Distance Algorithm

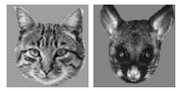
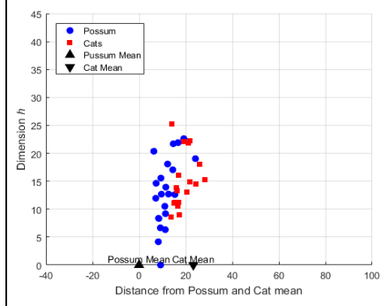
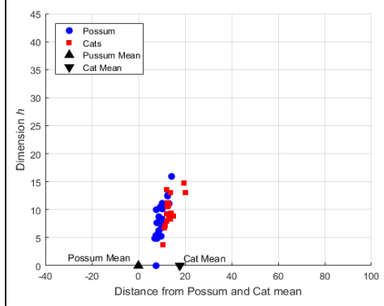
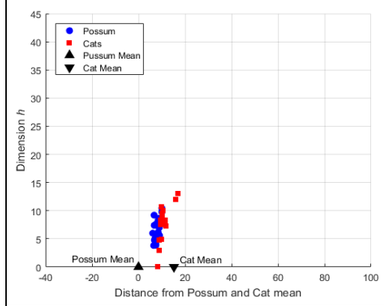
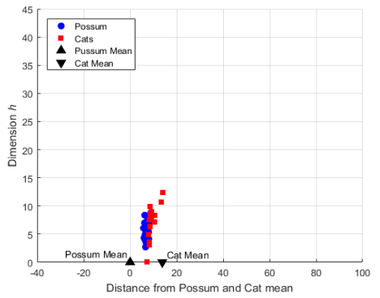
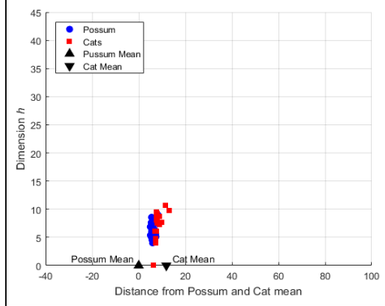
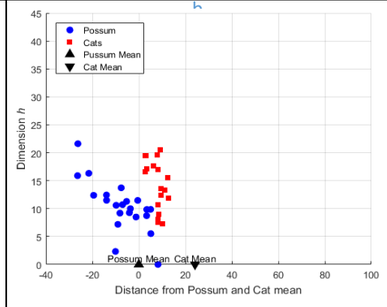
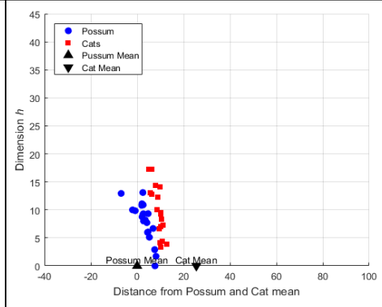
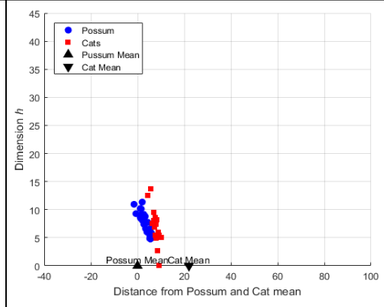
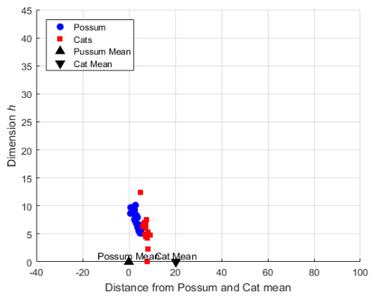
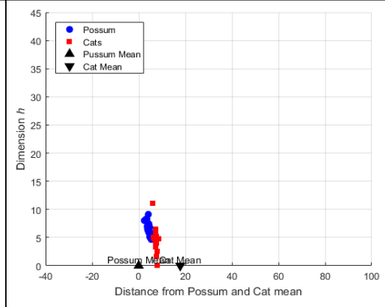
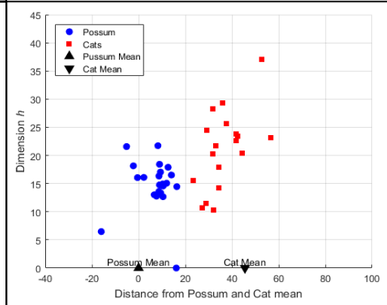
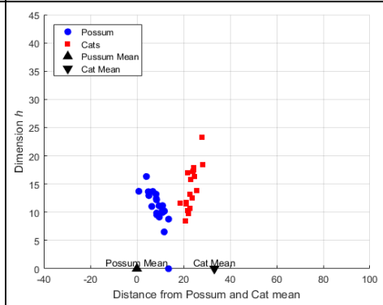
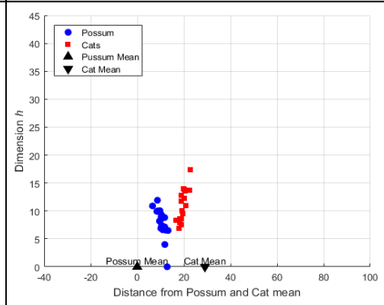
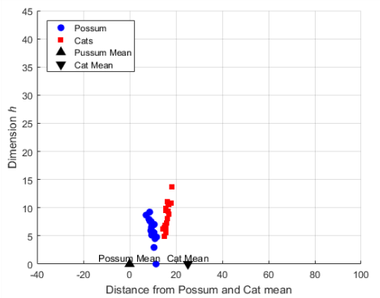
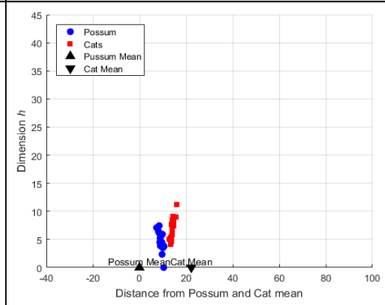
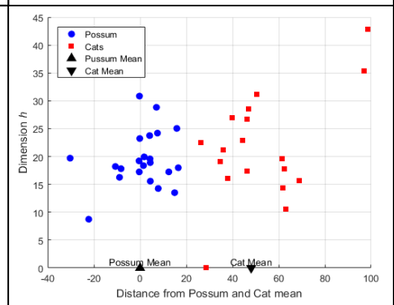
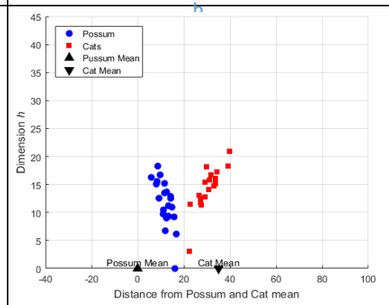
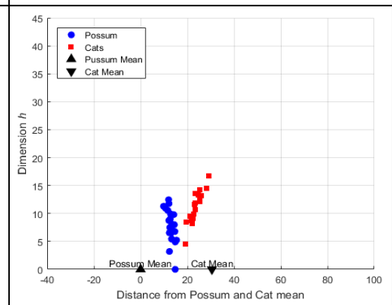
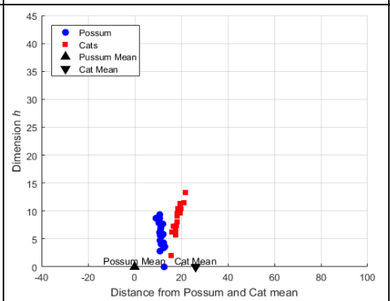
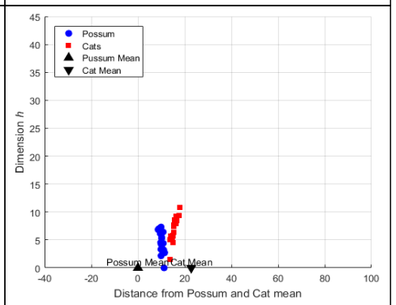
	Resolution 10x10	Resolution 20x20	Resolution 40x40	Resolution 80x80	Resolution 160x160
Original face and grey background 					
White face, Grey background and original eyes and nose 					
Grey face, black background and white eyes and nose 					
White face, black background and no eyes and nose 					

Figure 4.22: Eigenface species separation for improved colour schemes and different resolutions with introduced Error Weights and Developed Distance Algorithm

4.7 Multivariate Bayesian Decision Theory-based Classifier

Bayesian decision theory is an important statistical tool used to solve problems in pattern classification. It is considered the ideal case in which the probability structure underlying the categories is known perfectly. While this sort of situation rarely occurs in practice, it permits the Bayes optimal classifier which can be used to compare other classifiers.

The multivariate normal density [139] in terms d dimensions is given as:

$$p(x) = \frac{1}{(2\pi)^{d/2}|C|^{1/2}} e^{\left[-\frac{(x-m)^T C^{-1}(x-m)}{2}\right]} \tag{4.36}$$

where $p(x)$ is the probability mass, x is a d-component column vector, m is the mean vector, C is d × d covariance matrix, $|C|$ is its determinant and C^{-1} is its inverse.

In this application, for example, if there are 30 training images, d will become 30, this is a 30-dimensional problem. The covariance matrix from eigenvector equation (4.10) will become C, x is the weight from the test images and m is the average face image.

In this application there are two separate training images (cats and possums). Hence there are two sets of covariance matrices and two average face images from training sets. Then for an unknown image the probability mass formula given in equation (4.36) is applied twice, once for the cat data and once for the possum data. Then there will be two separate probability mass values. To detect the test image, the value with the highest probability mass is selected. The highest value has the highest probability of being close to the mean value of that class. The graph below shows the probability mass for 40 unknown images. Out of those 40 images, the first 22 belong to possums and numbers 23–40 belong to cats, shown in Figure 4.23.

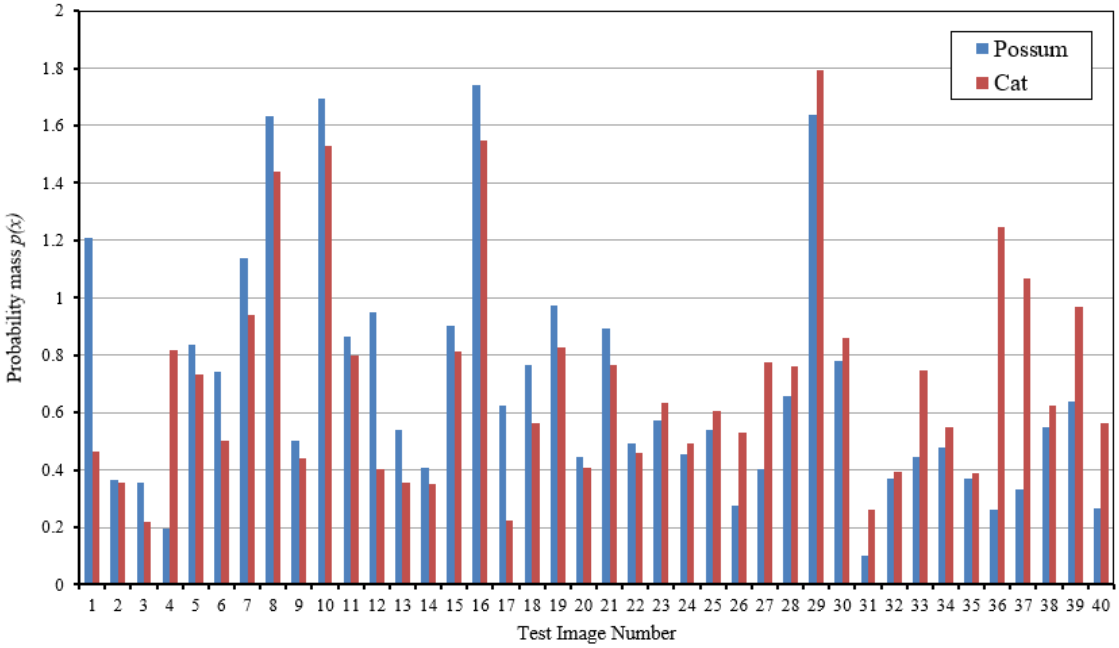


Figure 4.23: Multivariate normal density for test images

Figure 4.24 shows the separation between these two classes using the technique presented in section 4.5.

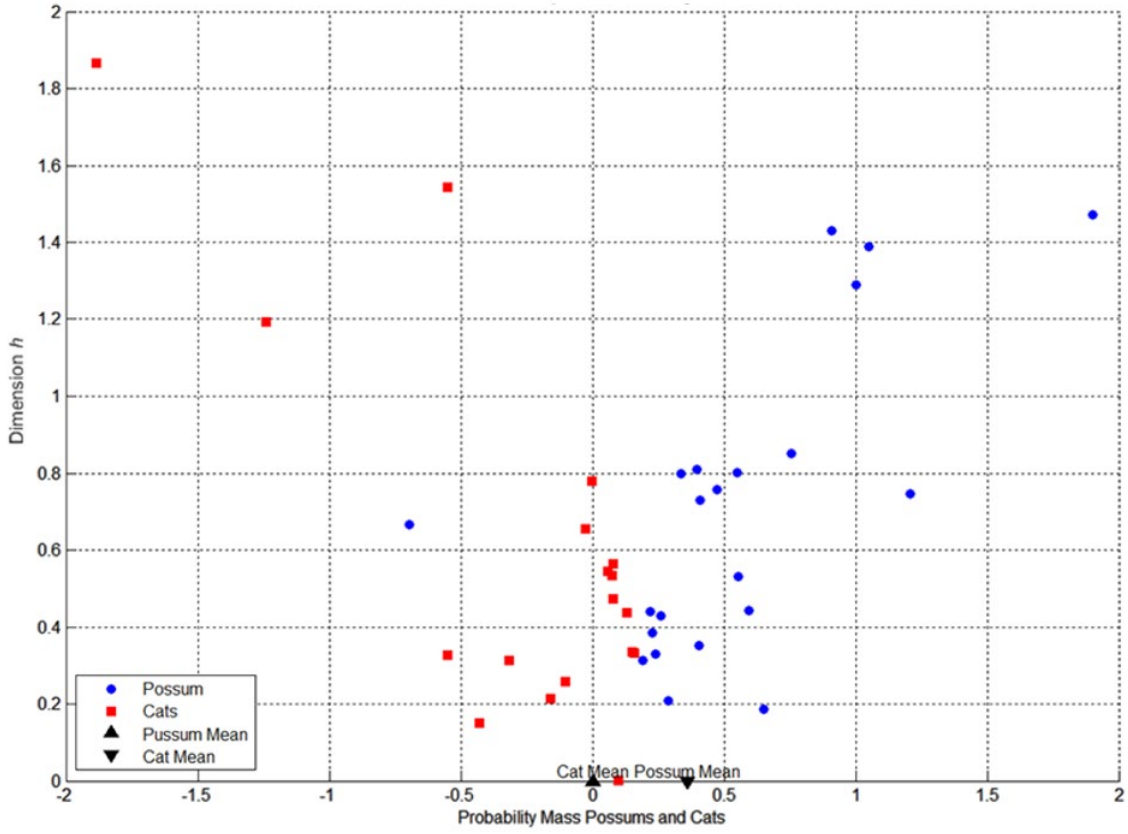


Figure 4.24: Multivariate normal density-based class separation with developed classifier

Even though possum identification rate is 95% and cat identification rate is 100%, the class separation is not satisfactory, and it is hard to define the boundary for these two classes, shown in Figure 4.24.

4.7.1 Error Function for Multivariate Normal Probability Density

Once the multivariate normal probability density is calculated, it is important to investigate the error value for the probability mass. A simple 1D case error function can be expressed as follows [145]:

$$\operatorname{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt \tag{4.37}$$

In the multidimensional case, the error function can be expressed as in equations (4.36). The full proof for these formulae can be found in [146].

Equation (4.38) shows that all $\operatorname{erf}(x)$ for even-dimensional error functions are expressible in closed form, although for sufficiently large *m* the closed form expressions become increasingly complicated. With x as the calculated probability mass:

$$erf_{2m}(x) = 1 - e^{-x^2}\left\{1 + \frac{x^2}{1!} + \frac{x^4}{2!} + \cdots + \frac{x^{2(m-1)}}{(m-1)!}\right\} \tag{4.38}$$

where $m = 0, 1, 2, \ldots$

From equation (4.38) it can be seen that:

$$1 - e^{-x^2} \equiv \operatorname{erf}_2(x) \geq \operatorname{erf}_4(x) \geq \operatorname{erf}_6(x) \geq \cdots \tag{4.39}$$

Similarly, equation (4.40) shows that all erf (x) for odd-dimensional error functions are expressible in closed form.

$$\begin{aligned}\mathrm{erf}_{2m+1}(x) = \mathrm{erf}(x) \\ -\frac{e^{-x^2}}{\sqrt{\pi}}\left\{\frac{(2x)0!}{1!}+\frac{(2x)^3 1!}{3!}+\cdots\right. \\ \left.+\frac{(2x)^{2m-1}(m-1)!}{(2m-1)!}\right\}\end{aligned} \tag{4.40}$$

where $m = 0, 1, 2, \ldots$

It can be clearly seen from equation (4.40) that:

$$\mathrm{erf}(x) \equiv \mathrm{erf}_1(x) \geq \mathrm{erf}_3(x) \geq \mathrm{erf}_5(x) \geq \cdots \tag{4.41}$$

The equations (4.38) and (4.40) were used to calculate the error and the plot is shown in Figure 4.25, the user can input the probability mass (x) and standard deviation value for training images.

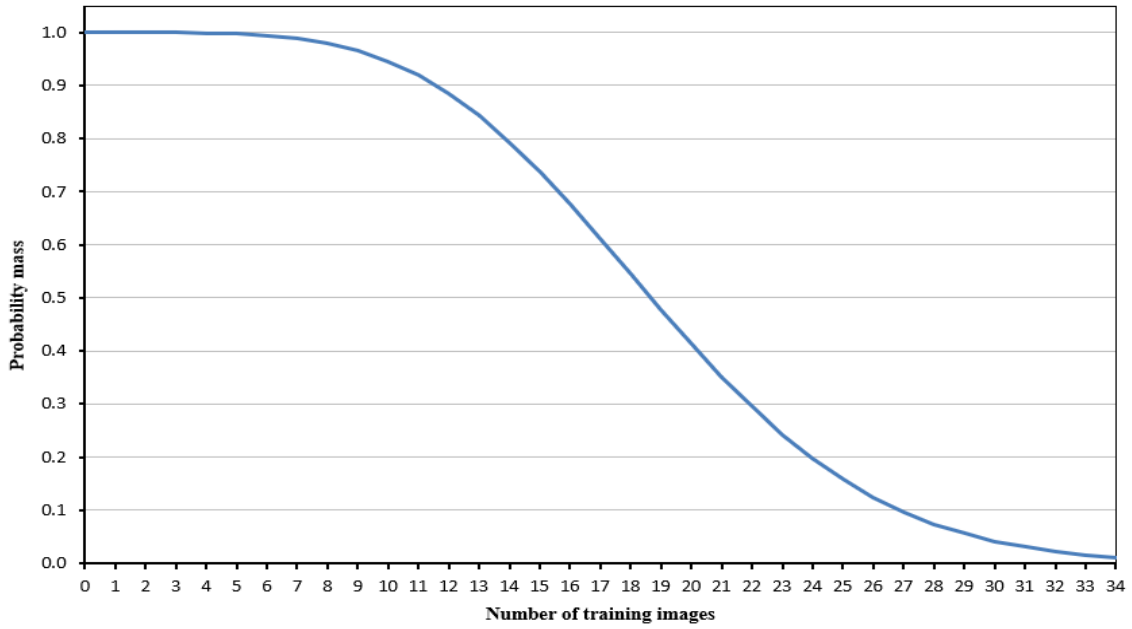


Figure 4.25: Implemented multidimensional error function

This application then calculates the error values for up to 34 dimensions. The main issue with this application is that as the dimensions get larger, the probability values get

smaller. As the number of training images increases, it is difficult to obtain a reliable error value. This technique therefore performs best with lower numbers of dimensions.

4.8 Classifier Evaluation

The main aim of this section is to evaluate the new distance algorithm (equation (4.12) with respect to the existing distance algorithm $d = \sqrt{\sum_n w_n^2}$. In this section, four different combinations were compared, shown in Table 4.2:

Option 1. The existing Eigenface technique
Option 2. The improved Eigenface technique
Option 3. The existing Eigenface technique with the new distance algorithm
Option 4. The improved Eigenface technique with the existing distance formula

Table 4.2: Equations used for Options 1 - 4

Option 1	$f_n = Ae_n$	$d = \sqrt{\sum_n w_n^2}$
Option 2	$f_n = \frac{Ae_n}{\lvert Ae_n \rvert}$	$d = \sqrt{\sum_n \frac{w_n^2}{\lambda_n}(N-1)}$
Option 3	$f_n = Ae_n$	$d = \sqrt{\sum_n \frac{w_n^2}{\lambda_n}(N-1)}$
Option 4	$f_n = \frac{Ae_n}{\lvert Ae_n \rvert}$	$d = \sqrt{\sum_n w_n^2}$
Where f_n is Eigenfaces, A is normalised images, e_n is eigenvectors, w is weights and d is distance.		

In order to test these algorithms, two training sets with 33 images (of cats and possums), were selected, shown in Figure 4.26. Then 40 unknown images were selected to test the system. Of the unknown images 22 were possums and 18 were cats, shown in Figure 4.27. These training sets and test images were padded as discussed in section 4.6.2. All

the training images and test images were 170 × 170 in resolution. In this set of trials error weights given in equation (4.33) was not used.

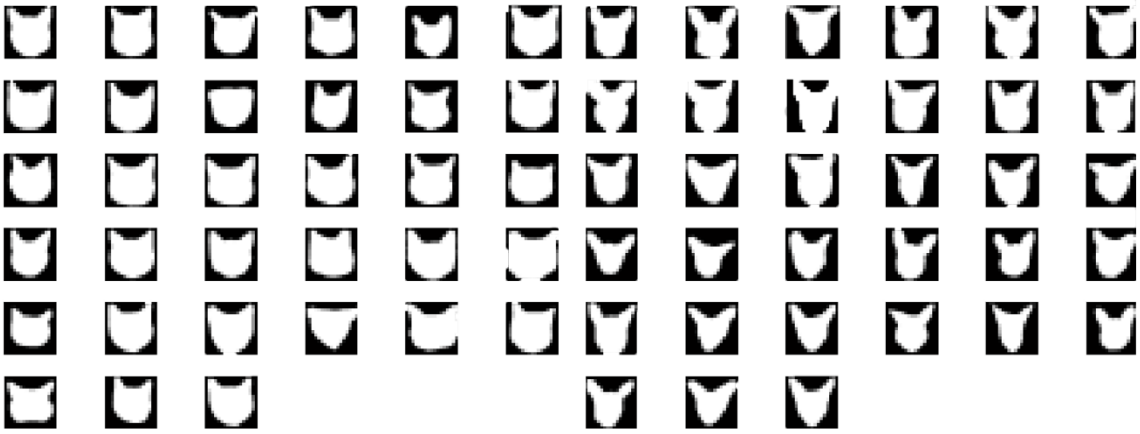


Figure 4.26: Training images (cats and possums)

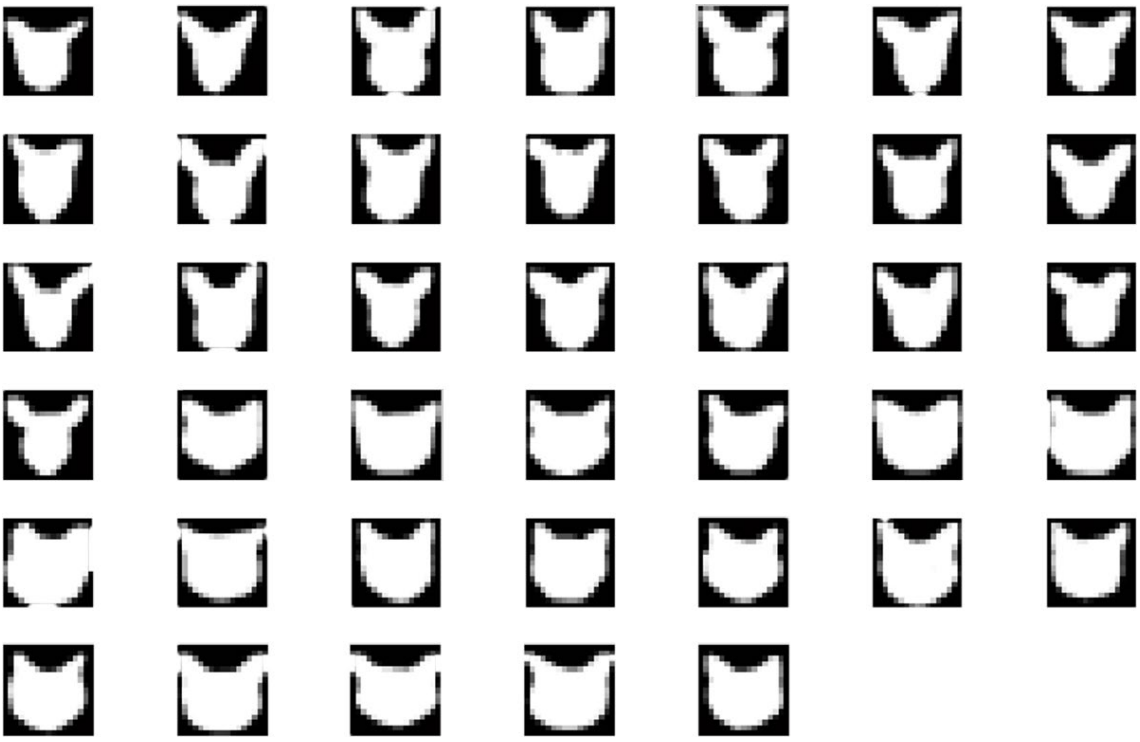


Figure 4.27: Unknown test images

The new classifier was then used to plot the distance information and analyse the separation of the species classes. If the classes are close together, it is hard to define a decision boundary between classes. Figure 4.28 to Figure 4.31 respectively show the performance of all four options.

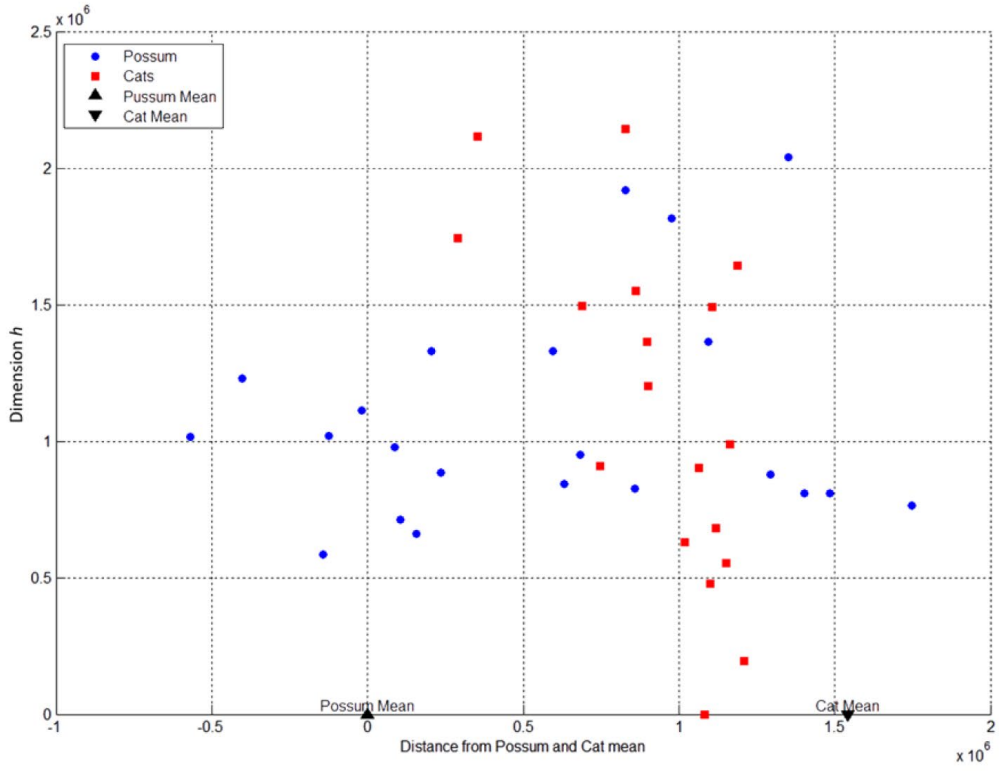


Figure 4.28: Plot for Option 1 – Distance with Standard Eigenface and Standard distance formula

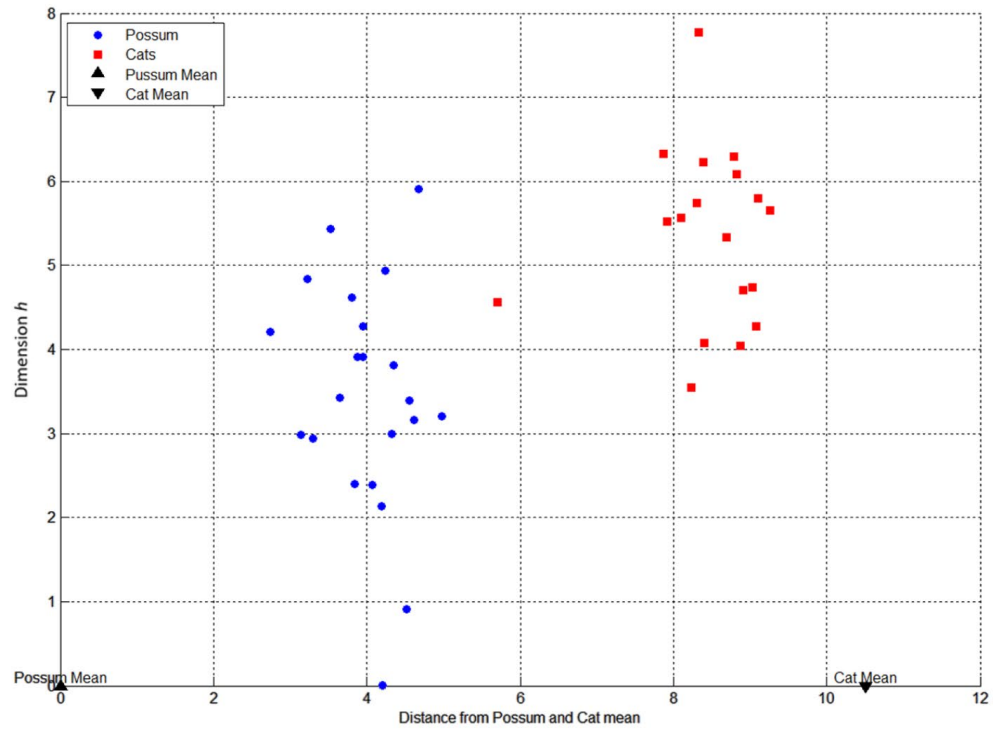


Figure 4.29: Plot for Option 2 – Distances with normalised Eigenfaces and altered distance formula

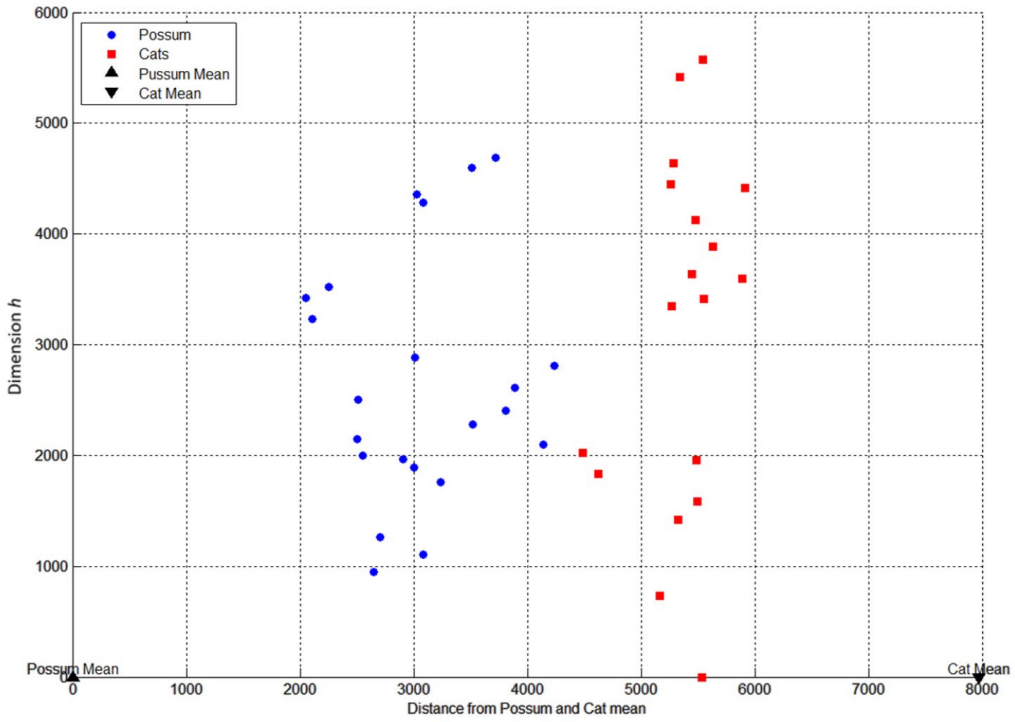


Figure 4.30: Plot for Option3 – Distance with standard Eigenfaces and altered distance formula

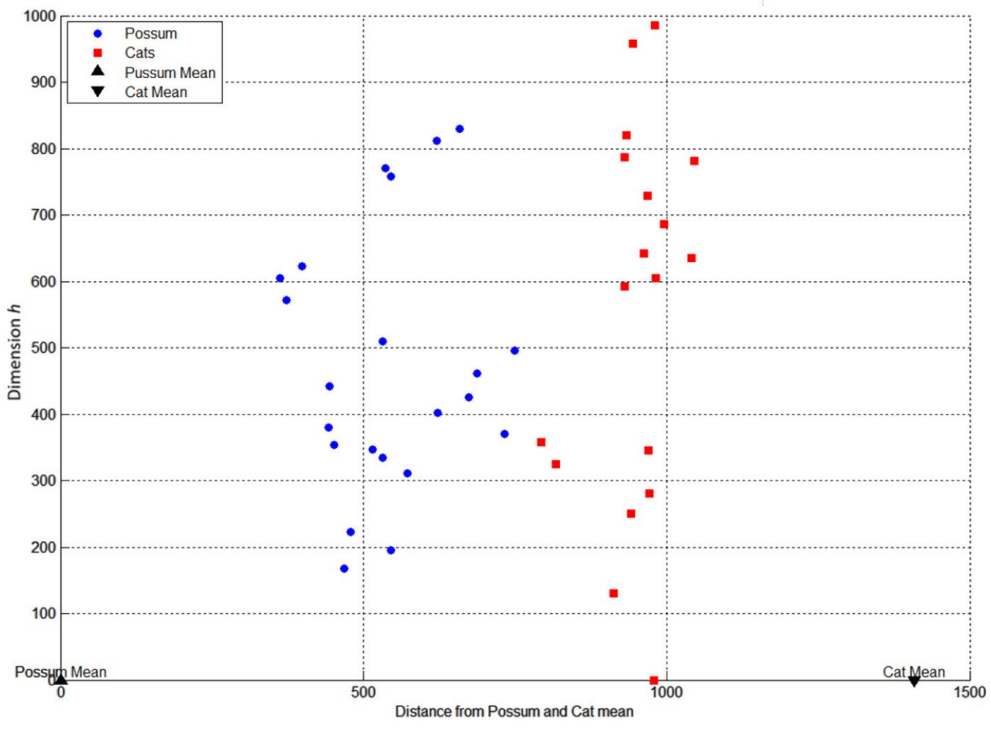


Figure 4.31: Plot for Option 4 – Distance with Normalised Eigenfaces and Standard distance formula

By analysing the results shown in Figure 4.28 to Figure 4.31 it can be clearly seen that the best separation is achieved by the improved Eigenface technique option 2, the next best are options 3 and 4. Even though these two techniques have different scales on the *x* and *y* axes, both graphs have the same separation. So, it can be concluded that the effect on the weights by normalising the Eigenface and dividing the normalised weights by their eigenvalue, which is the variance. When combined these two factors together achieve the best separation of the classes. The standard Eigenface technique had the worst performance out of all four options. It showed poor separation between the two species. This was to be expected, due to the vectors having arbitrary magnitudes in face space or low-dimensional space. Perhaps this is one of the reasons the original Eigenface technique used the threshold-based detection technique.

The work presented in this chapter therefore represents a significant addition to the existing knowledge of Eigenface techniques available.

Chapter 5
Fisherface Technique for Cat and Possum Detection

This chapter describes the use of the Fisherface technique for animal (cat and possum) image identification. That is a new application in the animal-identification domain. The Fisherface technique is widely used in the human recognition domain [147, 148]. The results for a new version are better than the standard Eigenface technique [44]. This linear discriminant technique performs well in different illumination conditions and different facial expressions. It minimises the within-class scatter matrix and maximises the between-class scatter matrix, to optimise class separation in high dimensional space. The technique seeks directions which are efficient for discrimination within the data [44, 45, 109, 111, 148].

Despite its improved performance in the human-identification domain, when unedited animal training images were used with the Fisherface technique the classes were inseparable. Therefore, at first this technique seems unsuitable to application of animal identification.

During investigation of the Fisherface technique, it was discovered that class separation can be improved by padding the training images. Two class separation optimisation approaches were used:

1. The first approach was to optimise the greyscale colour scheme.
2. The second approach was to find the optimum image resolution to optimise the class separation and computational time.

After applying the optimisation techniques, the Fisherface technique's class separation dramatically improved, and the best colour schemes and resolutions were selected for optimum processing time and class separation. Since the class separation is calculated from class mean, the probability of belonging to a certain membership can be calculated. Since there was a limited number of training images, the training data set was singular i.e. the number of images M is less than the number of column pixels times the number of row pixels N^2, as the column and row values are of equal size N. Therefore, SVD was used to reduce the dimensions of faces from $M \times N^2$ to $M \times M$ dimensional vectors.

5.1 Fisherface for Classifying Cats and Possums

This research shows that there is a further need for dimensionality-reduction methods when performing image detection, even though Principal Component Analysis (PCA) and the Eigenface technique perform well.

PCA finds a linear combination of features that maximises the total variance in data. Even though this is undoubtedly a good way to represent data, it doesn't consider classes. Therefore, most of the discriminative information may be lost when throwing less useful components away.

When considering a situation where the variance is generated by an external source, such as light, the principal components identified by a PCA do not necessarily contain any discriminative information at all, so the projected samples are smeared together, and classification becomes difficult.

To find the combination of features that best separates classes, Linear Discriminant Analysis (LDA) maximizes the ratio of between-class scatter to within-class scatter. The main idea is that the same classes should cluster tightly together, and at the same time different classes separate as much as possible from each other. Therefore, in theory the Fisherface technique should perform better than the Eigenface technique for multiclass problems [44], shown in Figure 5.1. According to Martinez and Kak [149], this is not true for all cases; in some cases PCA outperforms LDA.

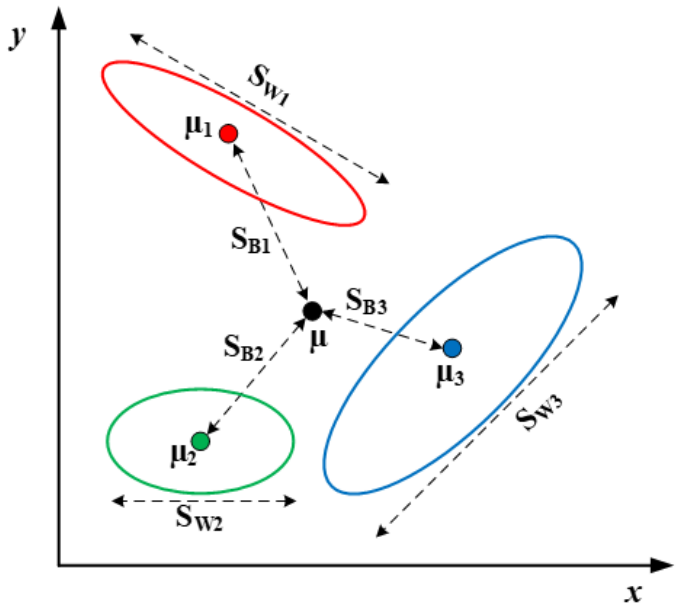


Figure 5.1: Three-class problem: Scatter matrix S_B and S_W, μ is the mean of three classes and μ_1, μ_2, μ_3 are the class means

To compute the Fisherfaces it has been assumed that all the images in cats and possum training classes are normally distributed. This can be denoted by using the multivariate distribution:

$$N_i(m_i, C_i) \tag{5.1}$$

where m_i is mean and C_i is covariance matrix, and the probability function of this can be expressed as:

$$f_i(X|m_i, C_i) \tag{5.2}$$

For the cat and possum cases, the Cat Class is C ($N^2 \times Z$) and Possum Class is P ($N^2 \times Z$), where Z is number of training images in each class, N is the height and width of images and M is the total number of images in both training sets ($2 \times Z$). X is the full training set [C, P] ($N^2 \times M$).

Calculate the average for full training set of both cat and possum classes, using:

$$m = \frac{1}{M}\sum_{i}\Gamma_i \tag{5.3}$$

Centre the data (Φ is centred data)

$$\Phi_i = \Gamma_i - m \tag{5.4}$$

Since A sample vectors are larger than the total number of samples ($M < N^2$), the within-class scatter matrix S_W is singular in this case. Therefore, SVD is performed on the centred data set A in order to reduce the dimensionality. For further details on SVD see section 2.8 of chapter 2.

SVD of A can be written as:

$$A = USV^T \tag{5.5}$$

Where:

U is left singular vectors of A are set of orthonormal eigenvectors of AA^T

S is square roots of the non-zero eigenvalues of both A^TA and AA^T

V is right singular vectors of A are eigenvectors of A^TA

The left singular vectors can be used as dimensionally reduced-vectors of A. So, U can be used to calculate the weights (Y) for normalised training images:

$$Y = U^TA \tag{5.6}$$

The eigenvectors of U are associated to non-zero eigenvalues, which are the useful vectors with the most information. Typically, in the Fisherface technique *M – number of classes* vectors have the most of information. The main reason for this is that when the original data was centred some of the information was lost. Since there are two classes in this application, there will be $M - 2 = V$ vectors from Y. The Fisher Linear Discriminant Analysis (FLDA) can now be applied to the dimensionality-reduced training set Y.

For the FLDA the mean m_Y of the training set *Y* needs to be calculated:

$$m_Y = \frac{1}{M}\sum_i Y_i \tag{5.7}$$

Then the mean for the cat and possum classes need to be computed. For the cat class, the mean m_c is given by:

$$m_c = \frac{1}{Z}\sum_{i=1}^{Z} Y_i\,, \tag{5.8}$$

and for the possum class the mean m_p is given by:

$$m_p = \frac{1}{Z}\sum_{i=Z}^{M} Y_i\,, \tag{5.9}$$

Once the cat class mean and possum class mean are calculated, each class data can be centred (normalised), C_N for the cat-centred training class and P_N for the possum-centred training class.

$$\Phi_C = Y(1{:}Z) - m_c \tag{5.10}$$

and

$$\Phi_P = Y(Z{:}M) - m_p \tag{5.11}$$

After centring the training classes, the within-class scatter matrix S_W can be calculated, using:

$$S_W = \Phi_p \Phi_p^T + \Phi_c \Phi_c^T \tag{5.12}$$

Then the between-class scatter matrix S_B can be calculated, using:

$$S_B = Z(\Phi_C - m_Y)(\Phi_C - m_Y)^T + Z(\Phi_P - m_Y)(\Phi_P - m_Y)^T \tag{5.13}$$

The next step is to find the projection *W* which maximises the class separation criterion:

$$W_{opt} = \arg max_W \frac{|W^T S_B W|}{|W^T S_W W|} \ . \tag{5.14}$$

The solution for the above optimisation problem, equation (5.14) is found by solving the general eigenvalue problem:

$$S_W B = \lambda_i S_W W_i \tag{5.15}$$

where *W* is the eigenvectors of $S_W^{-1} S_B$ and λ is the eigenvalues, so that:

$$S_W^{-1} S_B W_i = \lambda_i W_i \tag{5.16}$$

Using the *W* eigenvectors, the Fisherfaces *(F)* can be calculated:

$$F = UW \tag{5.17}$$

In order to project the cat class and the possum class into Fisherspace, the class means need to be calculated. Here *C_m* is the training cat class mean:

$$C\_m = \frac{1}{Z} \sum_i C_i \ \ (N^2 \times 1) \tag{5.18}$$

and *P_m* is the training possum class mean:

$$P\_m = \tfrac{1}{Z} \textstyle\sum_i P_i \ \ (N^2 \times 1) \ . \tag{5.19}$$

Now the average faces can be projected into Fisherspace, and sets of weights can be obtained for the cat class and the possum class using:

$$w_C = F^T C\_m \tag{5.20}$$

where w_C are the cat class weights.

$$w_P = F^T P\_m \tag{5.21}$$

where w_P are the possum class weights.

To classify an unknown image τ ($N^2 \times 1$), the weight vector w_τ needs to be calculated by projecting into Fisherspace:

$$w_\tau = F^T \tau \tag{5.22}$$

Since these weights are obtained from eigenvectors, their individual values are perpendicular to each other. So, the first weight from these weight vectors has the main separation between the two class means. The second weight will be perpendicular to the first one, shown in Figure 5.2.

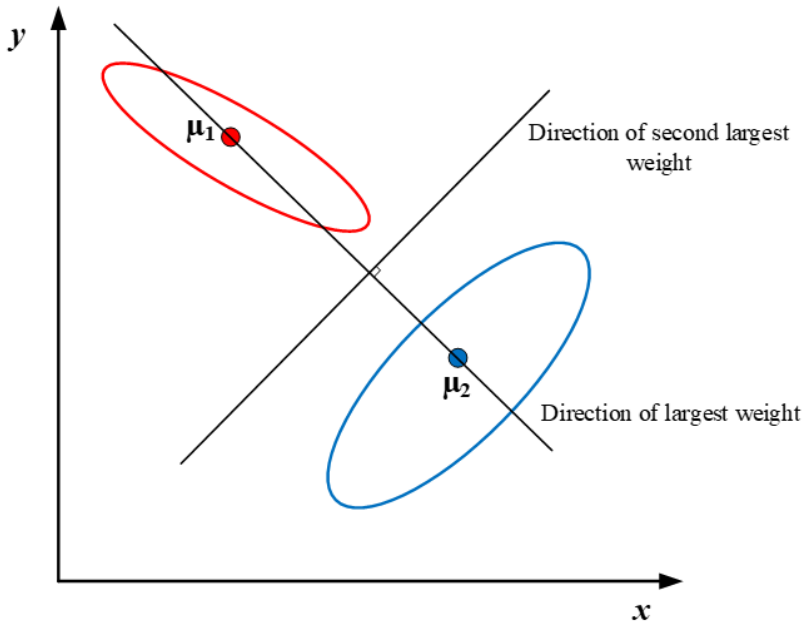


Figure 5.2: Direction of Fisher weights

Therefore, in order to calculate the distance of the cat mean and the possum mean on the main linear axis, the first largest weight vector is used, where *C_Dis* is the cat distance from the mean given by:

$$C\_Dis = W_\tau - W_C \,, \tag{5.23}$$

and *P_Dis* is the possum distance from the mean given by:

$$P\_Dis = W_\tau - W_P \,. \tag{5.24}$$

This calculated distance can be used to classify membership of the class. The l^2 -norm is used to calculate the distance between these two memberships. The distance with lowest value is the likely membership. If the test data overlaps on the linear axis the membership, the data is unclassifiable.

5.2 Training Images and Their Properties

In order to improve the detection rate, it is important to enhance the main face features of the training images. As discussed previously, by enhancing the main features the detection rate can be improved. If the original images are used, the Fisherface technique gives extremely poor performance in the animal recognition domain.

The training images were selected randomly and therefore had different head orientations and head sizes, as discussed in section 3.2 of chapter 3. With the Fisherface technique it is important to analyse the distribution of the training images. To do this training images were projected into Fisherspace and the sums of the weights were plotted. To perform this trial, equations (5.1) to (5.9) were used and the plotted weights are shown in Figure 5.3 and Figure 5.4.

In this trial 160 × 160 black and white possum and cat images were used. The identification was performed on face outline, so all the face features were turned into white pixels.

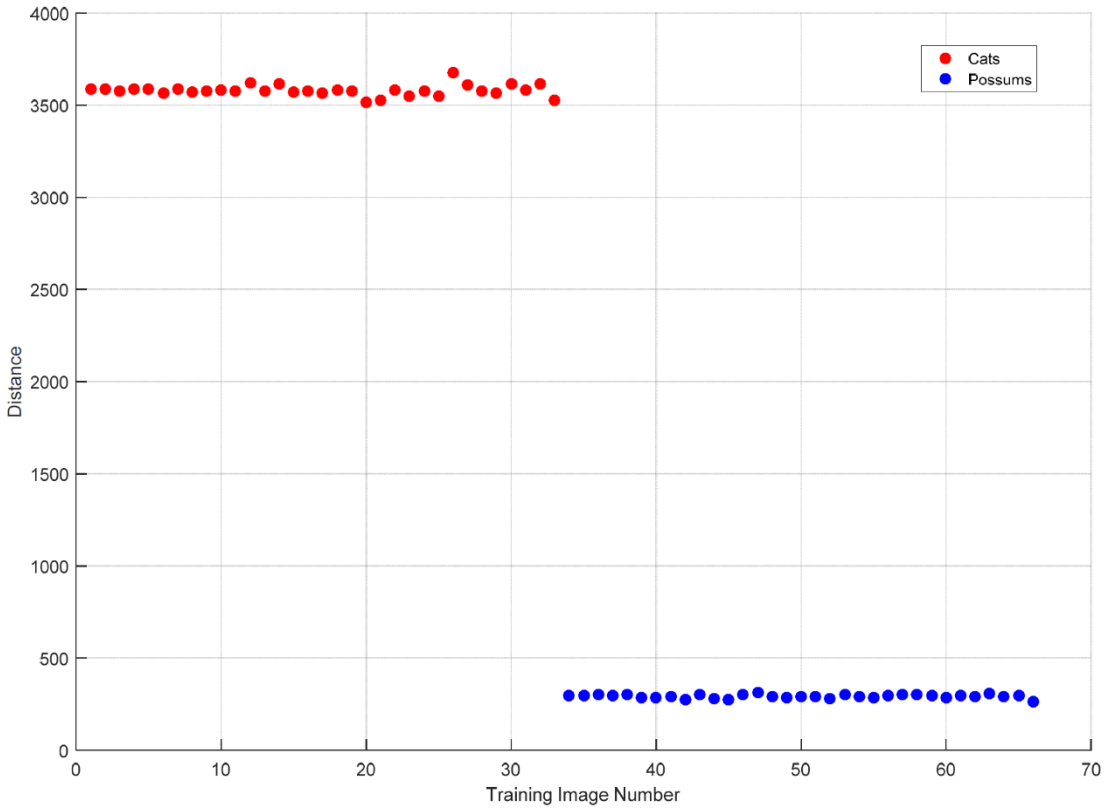


Figure 5.3: Fisherface training weights for enhanced images

As expected, all the training images in the same class had similar weight values and shown in Figure 5.3, the cat class and possum classes have good separation. There is a small fluctuation in cat weights due to different cat species having different head sizes. There is hardly any fluctuation in possum weights. This is due to both possum species have very similar head sizes. This phenomenon can be clearly seen in the above training data.

When the original images were used to calculate the weights, the above setup showed much larger variation across individual class weights. Moreover, the class separation distance was about 1.7 times less than that achieved with the face outline training images, shown in Figure 5.4.

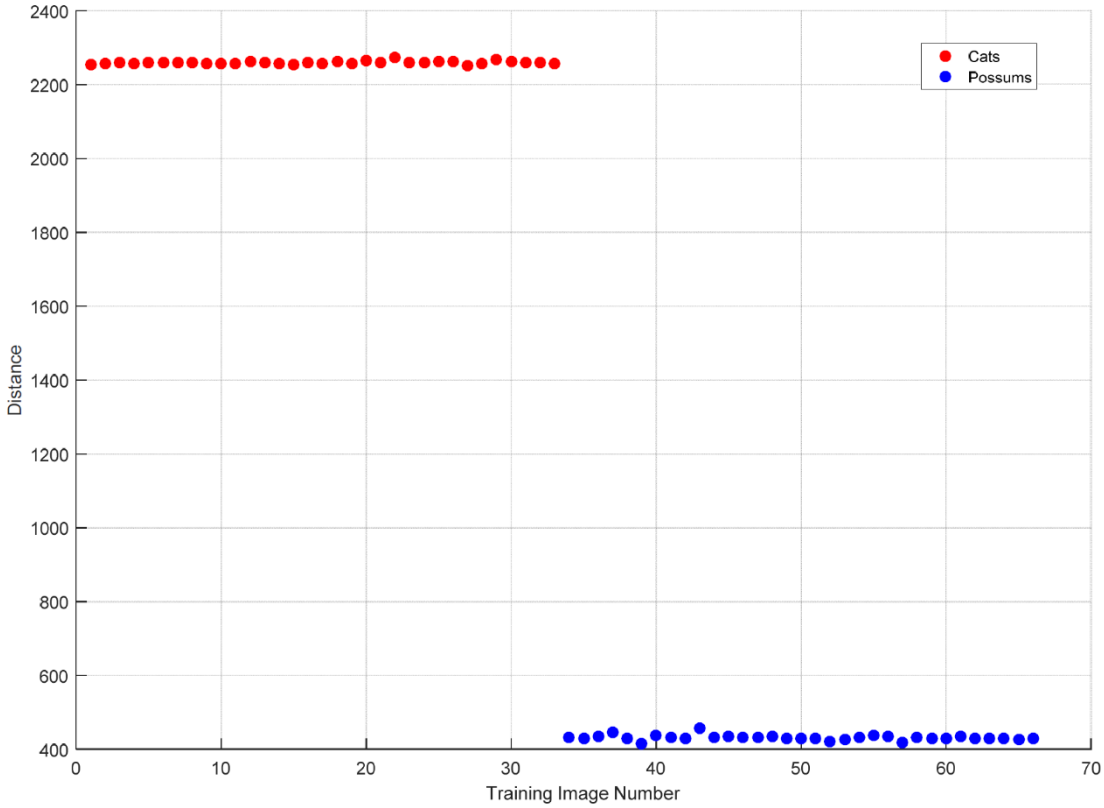


Figure 5.4: Fisherface training weights for original images

Unlike face outline detection, in this scenario the original images were used as training images. All the images had similar backgrounds and the animals' fur had similar colour spectra. Therefore, all the images in the same class looked very similar to each other, with very little variation in training weights. Similarly, there was less variation between the two separate classes. This was due to most of the animals' colours being similar to each other. This can be clearly seen in the smaller separation of cats and possums with the original images, shown in Figure 5.4.

From this trial it can be clearly seen that the Fisherface technique achieves better separation on the face outlines of animals. One of the main advantages of this new method is that, when applying SVD to training images, most of the lost data is background information and middle part of the facial information. Since the majority of pixels from these areas have the same value, most of the lost dimensions will be these values. This results in greater separation of the training data in high-dimensional space.

5.3 Class Separation Optimised by Greyscale Colour Scheme

Colour scheme optimisation was performed using a two-stage process. At the first stage, four different colour schemes were tried out. The first scheme was grey background and original face; the second was white face, grey background and original nose and mouth; the third was grey face, white nose and mouth and black background; and the fourth black background and white face. In this trial all the test images were 160 × 160 pixels, shown in Figure 5.5.

Figure 5.5: Grayscale Colour schemes

From these four schemes the best performing one was then selected. Performance was compared on class separation and computational time. The class separation is measured using equations (5.20) and (5.21). The computational time is measured using MATLAB *'tic'* and *'toc'* functions to run equations (5.18) to (5.22). In order to benchmark the time, the same algorithm was looped 1000 times and the average computational time for one loop was calculated. It is difficult to measure loop time with a single computational loop. Depending on operating system task allocation, this processing time could vary considerably. The looping technique is one of the MATLAB recommended techniques to measure time, when the program execution time is very short (less than a half a second).

After the initial trial it was found the original images produced negative separation. This means that the main axis test data from the two training groups overlapped and therefore there was no clear separation. This is mainly due to the original images having similar colour details across the two classes. Hence most of the original images contained lots of data and the processing time was long. With white faces and original nose and mouth information, better separation was achieved compared to the first option. But separation was not significant enough to be useful in a real-life application. The third option showed better separation compared to previous two options and the processing time was also faster. The last option produced the best separation, but the processing time was slower than the second and third options, shown in Table 5.1. Since the third and fourth options

showed better separation and acceptable execution times, it was decided to investigate a combined colour scheme.

Table 5.1: Fisherface data separation and executing time for initial colour scheme

	Separation	**Time (secs)**
Grey background, original face	-816.6	0.341
White face, grey background, original nose and mouth	275.7	0.324
Black background, grey face, white nose and mouth	1663.7	0.318
Black background and white face	2275.1	0.337

In order to combine the third and fourth schemes, the background colour was changed to black, face colour was changed to white, and nose and mouth colour was changed to varying greyscale. In this trial the nose and mouth colour were changed from 0 to 200 greyscale, with an intensity change step of 50 and all the images size were 160 × 160 pixels, shown in Figure 5.6.

Figure 5.6: Nose and mouth with varying greyscale from 0 to 200

Table 5.2 shows the class separation and execution time for each colour scheme. The first column shows each colour scheme with ascending greyscale value. The second column shows the possum and cat class separation for each colour scheme. Third column shows the execution time.

Table 5.2: Separation and execution time and of optimised colour scheme

	Separation	Time (secs)
Black background and white face 0 nose and mouth	1636.8	0.159
Black background and white face 50 nose and mouth	2097.1	0.176
Black background and white face 100 nose and mouth	2369.6	0.173
Black background and white face 150 nose and mouth	2314.6	0.165
Black background and white face 200 nose and mouth	2318.4	0.178

It can be clearly seen that the new application has improved separation results, with best scheme being the black background and white face with 200 greyscale nose and mouth. But the actual separation difference between the black background and the white face is not considerable. In order to simplify the whole process, the black background and white face is the most optimal scheme. When the system is implemented on an embedded platform it will be important to have a trivial colour scheme, which will mean the pre-processing tasks will be more efficient.

5.4 Class Separation and Processing Time Optimised by Changing Image Resolution

As described in chapter 4, image resolution improves Fisherface class separation and processing time. After the optimum colour scheme is developed, seven different image resolutions were trialled: 10 × 10, 16 × 16, 20 × 20, 32 × 32, 40 × 40, 80 × 80 and 160 × 160 pixels , shown in Figure 5.7. In order to reduce the resolution, same technique used in section 4.6.2 of chapter 2 was applied.

Figure 5.7: Trialled resolutions: (left to right) 10 × 10; 16 × 16; 20 × 20; 32 × 32; 40 × 40; 80 × 80; and 160 × 160

SVD is a similar data reduction technique to PCA. Therefore, as discussed previously, when the dimensionality of the training images is reduced, less data is lost at low resolution. This is one of the main advantages of using lower resolution images. By reducing the resolution, however, some finer details still get lost. So, it is important to find the optimal resolution for optimal separation.

Reducing resolution will have direct effect on the execution time of the program. Therefore, it is important to optimise the execution time with class separation. As discussed above, execution time was the average of 1000 loops of same program.

The main aim of this test was to find a useable resolution with reasonable execution time. Therefore, in this trial all the colour schemes shown in Table 5.3 were trialled with all seven different resolutions.

Table 5.3: Resolution, separation and execution time data for tables (a) to (f) with varying greyscale colour from 0 to 200 and a step of 50 for animal noses and mouths

(a) Black background and white face		
Resolution	Separation	Time (secs)
10 × 10	-13.4	0.003
16 × 16	-13.5	0.004
20 × 20	84.4	0.004
32 × 32	239.4	0.006
40 × 40	352.4	0.008
80 × 80	955.7	0.031
160 × 160	2275.1	0.156

(b) Black background and white face, 0 nose and mouth		
Resolution	Separation	Time (secs)
10 × 10	-30.2	0.003
16 × 16	13.2	0.003
20 × 20	22.5	0.003
32 × 32	204.2	0.005
40 × 40	259.1	0.007
80 × 80	679.5	0.025
160 × 160	1636.8	0.160

(c) Black background and white face, 50 nose and mouth		
Resolution	Separation	Time (secs)
10 × 10	-27.2	0.003
16 × 16	15.2	0.003
20 × 20	39.1	0.004
32 × 32	253.3	0.005
40 × 40	363.7	0.006
80 × 80	913.9	0.025
160 × 160	2097.1	0.176

(d) Black background and white face, 100 nose and mouth		
Resolution	Separation	Time (secs)
10 × 10	-14.4	0.004
16 × 16	7.7	0.004
20 × 20	56.4	0.004
32 × 32	251.8	0.006
40 × 40	259.1	0.008
80 × 80	960.9	0.042
160 × 160	2369.6	0.173

Table 5.3 (continued)

(e) Black background and white face, 150 nose and mouth		
Resolution	Separation	Time (secs)
10 × 10	-26.1	0.004
16 × 16	5.5	0.004
20 × 20	73.4	0.005
32 × 32	248.8	0.006
40 × 40	363.0	0.008
80 × 80	992.5	0.027
160 × 160	2314.6	0.165

(f) Black background and white face, 200 nose and mouth		
Resolution	Separation	Time (secs)
10 × 10	-27.4	0.003
16 × 16	1.8	0.003
20 × 20	82.8	0.004
32 × 32	262.1	0.005
40 × 40	363.3	0.006
80 × 80	1003.5	0.031
160 × 160	2318.4	0.178

Figure 5.8 shows the class separation and execution time for each colour scheme. Every data point in the graph represents the image resolution, so that each coloured line has 7 different points representing resolution.

The best separation value of 2369 can be seen on the black background, white face, 100 nose and mouth graph, but the runtime of 0.173 seconds is one of the longest. The worst separation value of 1636 can be seen on the black background, white face and black eyes and nose graph. The optimal solution with one of the shortest runtimes is the black background and white face line. This line shows one of the best separations with the second-shortest runtime, shown in

Figure *5.8*. The main reason for this is most of the identification information for the animal is on the face outline, hence it produces best separation in Fisherface. Once other information such as face and nose are added, the percentage of information compared to the face outline is less, but this makes the covariance matrix more complicated to solve. Also, when applying PCA or SVD most of the lost dimensions on black background and white face will be similar across the whole training set. This lost information would be background border and internal face colour. Therefore, this colour scheme is least affected by data-reduction techniques.

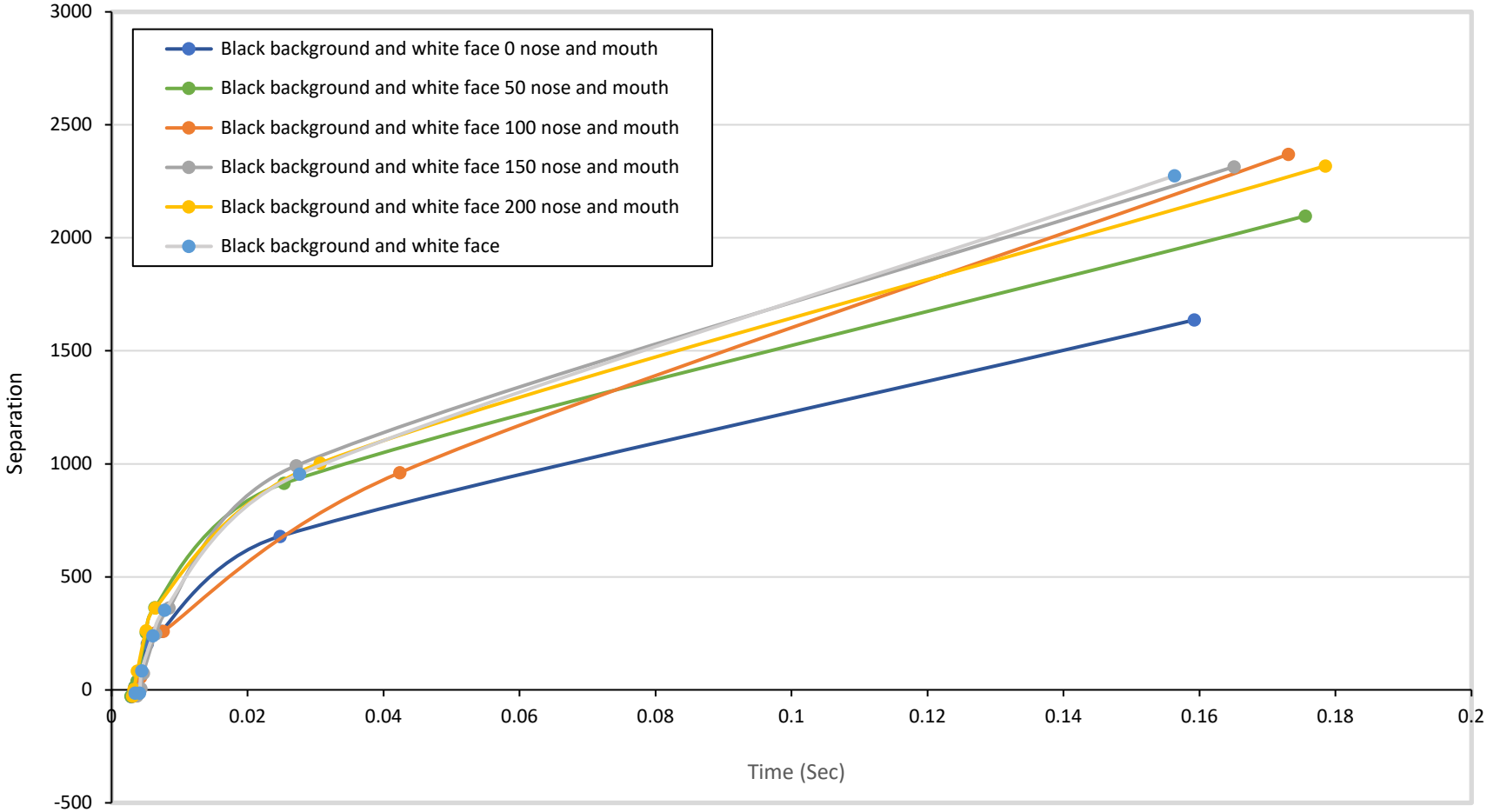


Figure 5.8: Fisherface test image separation vs time for optimised colour schemes and resolutions

From the research work presented in this chapter, it can be concluded that the best scheme to use for animal identification is facial outline with the Fisherface technique. In the animal identification domain, raw images cannot be used to obtain optimal results with the Fisherface technique. One of the advantages of this technique is that it can be extended into multiclass problems without any further modification. Therefore, the Fisherface technique is much more attractive for use in a real-world animal detection device. Unlike the Eigenface technique, it is designed for multiclass problems and it has better separation between classes when optimised training images are used.

Chapter 6
Support Vector Machine for Cat and Possum Identification

Support Vector Machines (SVMs) are used in many different research areas in order to separate two different data groups. This chapter describes the use of this technique to benchmark the performance of the techniques presented in the previous chapters. For this MATLAB's Statistics and Machine Learning Toolbox was used to implement the SVM technique for cat and possum identification.

During initial investigations it was found that SVM produced poor results with feature combination contained in original higher-resolution images. With higher-resolution images, the dimensionality of the problem increases and with a small number of training images it was difficult to produce accurate support vectors in high-dimensional space to define the decision boundary. To reduce this problem, two new techniques were developed, (a) the first technique was to improve the image by introducing custom scales. and (b) the second was a PCA-based data-reduction technique to reduce the dimensionality of the training data.

The performance of the SVMs was improved by the addition of these two developed techniques. But compared to the other two techniques Eigenface and Fisherface, SVM data separation for cat and possum detection was poor in higher resolution images. The separation was better at lower resolutions, but the error rate was similar to the Fisherface and Eigenface techniques.

6.1 Performance of SVM

MATLAB's Statistics and Machine Learning Toolbox's *'fitcsvm'* function was used to implement the SVM technique based on [150]. This function returns a SVM model for a given training data table with its data membership. The returned model class contained all the parameters of the SVM [151; Chapter 12]. Scale, bias and orthogonal vector values are the most important parameters for calculating the orthogonal distance from the optimal hyperplane that separates the data into two classes. The orthogonal distance can be obtained using the linear SVM score function.

The orthogonal distance can be obtained using following equation:

$$f(x) = (\bar{x}/s)'\overline{w} + b \tag{6.1}$$

where $\bar{x}$ is an observation the unknown image. The orthogonal vector $\overline{w}$ contains the coefficients that define an orthogonal vector to the hyperplane. For separable data, the optimal margin length is $2/\|\overline{w}\|$. The bias term is b is the and s is the scale parameter value.

This score value can be used to classify the membership of the unknown image. In order to test the data and investigate the best separation, 8 different image resolutions were trialled: 5 × 5, 10 × 10, 16 × 16, 20 × 20, 32 × 32, 40 × 40, 80 × 80 and 160 × 160. In these cases, all the different resolutions were black background and white face, shown in Figure 5.7 of chapter 5. For the training 66 images were used, of which 33 were of cats and 33 were of possums and there was a total of 40 unknown test images, of which 22 were possums and 18 were cats.

To use MATLAB's Statistics and Machine Learning Toolbox all the training images need to be categorised. For this research possums and cats are separated into two classes. Then these two classes are passed to *'fitcsvm'* function. The function returns a model, which contains values for $\overline{w}$, b and s. To classify an unknown image, the equation (6.1) can be used to calculate the training scores S. The score indicates the membership of the class of the unknown image, given in Table 6.1.

Table 6.1: SVM scores for test images

IMAGE SIZE	POSSUM SCORES					CAT SCORES				
	correct				wrong	correct				wrong
	S ≥ 1	1 > S > 0	S = 0	0 > S > -1	-1 ≥ S	-1 ≥ S	0 > S > -1	S==0	1 > S > 0	S ≥ 1
5×5	15	0	0	0	7	11	0	0	0	7
10×10	13	2	0	0	7	11	0	0	1	6
16×16	14	1	0	0	7	10	1	0	1	6
20×20	13	2	0	0	7	10	1	0	1	6
32×32	12	3	0	0	7	10	1	0	1	6
40×40	12	3	0	0	7	10	1	0	1	6
80×80	9	6	0	0	7	10	1	0	3	4
160×160	0	15	0	7	0	0	11	0	7	0

Table 6.1 has five columns for each data class membership (in this case cat and possum classes).

Column 1. The condition S ≥ 1 indicates that the unknown data is outside the gutter of the separating hyperplane.

Column 2. The condition 1 > S > 0 indicates when the data point is located between the gutter and the hyperplane, shown in Figure 2.15 of chapter 2. When the data point is in this region it is difficult to classify its membership, but there is high probability of this data point belonging to the S >= 1 class membership.

Column 3. The condition S = 0 indicates the unknown data is located on the separating hyperplane. In this situation it is impossible to classify the membership of the unknown data point.

Column 4. The condition 0 > S > -1 indicates that the data point is between the separating hyperplane and the other side (wrong side) of the gutter, shown in Figure 2.15 of chapter 2. In this situation it is still difficult to classify the membership of the data, but there is a high probability of it belonging to the other membership class.

Column 5. The condition -1 > S indicates an unknown data that belongs to the second-class membership. If this data belongs to the first-class membership, then it is mis-classified.

The above conditions are applied to the second-class columns but are mirrored. Instead of being on one side of the hyperplane, they are now on the opposite of the hyperplane.

As can be seen in Table 6.1, the performance of the SVM technique is poor compared with the Fisherface technique. Then the image resolution reduces the classification of the unknown data improves. The 5 × 5 images showed the best classifications of 15/22 for possums and 11/18 for cats, but there are still many misclassified images.

Since all the training images were black and white in a high dimensional plane, there is little change in data points in most of the axis. The main variance comes from the outline of the face. Therefore, it was decided to distort the training animal faces to a fixed size and to leave a fixed black boundary around the faces, shown in Figure 6.1. While the redundant pixels (same pixels) such as background and internal face pixels do not contribute towards the support vectors, the pixels with the most variance (face outline) have the main bearing of the support vectors.

Figure 6.1: Original black and white image (left) and Distorted and centred image (right)

Table 6.2 shows the results for the distorted test images. It can be clearly seen that there is very little improvement on, higher resolution images. At lower resolutions the cat identification rate was reduced. Therefore, this method of distorting and centring images was not satisfactory. A superior technique is required to reduce the dimensionality in higher-dimensional space.

Table 6.2: SVM scores for distorted test images

IMAGE SIZE	POSSUM SCORES					CAT SCORES				
	correct				wrong	correct				wrong
	S ≥ 1	1 > S > 0	S = 0	0 > S > -1	-1 ≥ S	-1 ≥ S	0 > S > -1	S==0	1 > S > 0	S ≥ 1
5×5	14	1	0	2	5	10	1	0	0	7
10×10	13	2	0	3	4	8	3	0	2	5
16×16	12	3	0	3	4	7	4	0	2	5
20×20	12	3	0	3	4	7	4	0	2	5
32×32	12	3	0	3	4	7	4	0	2	5
40×40	11	4	0	3	4	7	4	0	2	5
80×80	10	5	0	3	4	2	9	0	3	4
160×160	0	13	0	9	0	0	13	0	5	0

6.2 PCA + SVM for Possum and Cat Identification

As discussed in the previous section, the main problem with SVM in this research is that there are not enough training images to train the system to produce an effective separating hyperplane to divide the data into two different groups. Since there are N^2 variables as the number of rows and columns are both equal to N, the training data matrix becomes $N^2 \times M$, where M is number of training images. PCA can be used to reduce the dimensionality of this matrix.

When PCA is applied to $N^2 \times M$ matrix the pixels with the most variations are retained while all similar pixels, such as those in the internal face area and background, will not be kept. The covariance matrix extracts the pixels with most variance across the whole data set and these pixels will have a larger effect on support vectors.

Once the covariance matrix is calculated, the eigenvectors and eigenvalues are used to find the weights for the cat class and the possum class in eigenspace. For the unknown test images, set of weights can be calculated by projecting into eigenspace. Then these feature vectors can be used to train and test the system.

The mathematical steps can be expressed as follows:

Step 1. Calculate the average face for each group:

$$m = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \tag{6.2}$$

where m is the average face, Γ is the training images (cats and possums) and M is the total number of training images.

Step 2. Once the average face is calculated, the data can be centred by subtracting the average face:

$$\Phi_i = \Gamma_i - m \tag{6.3}$$

where Φ_i is the centred training data for the training images.

Step 3. Then the covariance matrix C is calculated for centred data:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n^T\Phi_n\,. \tag{6.4}$$

Step 4. Now the eigenvectors and eigenvalues can be calculated from the covariance matrix class. The centred training data is projected into the eigenspace using the calculated eigenvectors.

$$\lambda C = \lambda e \tag{6.5}$$

where λ is an eigenvalue of the covariance matrix, and e is an eigenvector of the covariance matrix.

$$f = \Phi e \tag{6.6}$$

where f is Eigenfaces for training data.

These Eigenfaces can be used to calculate the set of weights for each training class, given in equations (6.7) to (6.9).

$$w = \Phi^T f \tag{6.7}$$

$$w_C = w[1:N] \tag{6.8}$$

$$w_P = w[(N+1):M] \tag{6.9}$$

where w is all the weights for the training images, w_C is the cat training weights and w_P is the possum training weights. N is number of training images in each group. Then these weights can be used to train the SVM.

In order to compute weights for the test image Γ, it needs to be centred and projected into the eigenspace.

$$\Phi = \Gamma - m \tag{6.10}$$

$$w_T = \Phi' f \tag{6.11}$$

where Γ is the unknown test image, Φ is the centred test image and w_T is the weights for the test image.

These calculated weights from equations (6.8), (6.9) and (6.11) can be fed into the SVM as test and training data. Using this technique, the high dimensionality of the original images is reduced. The calculated full training weights are ($M \times M$) dimensions. Similarly, the test image is ($1 \times M$) dimensions.

Table 6.3 shows the separation results from SVM with PCA weights. There is significant improvement of identification at higher resolutions. The larger separation occurs at 160 × 160 resolution compared to SVM with black and white and SVM with distorted training images given in Table 6.1 and Table 6.2. This indicates that training images with higher resolution and similar features produce better separation with PCA and linear SVM. PCA with SVM is therefore an effective approach for animal detection.

Table 6.3: PCA + SVM scores for distorted test images

IMAGE SIZE	POSSUM SCORES					CAT SCORES				
	correct				wrong	correct				wrong
	S ≥ 1	1 > S > 0	S = 0	0 > S > -1	-1 ≥ S	-1 ≥ S	0 > S > -1	S==0	1 > S > 0	S ≥ 1
5×5	14	1	0	2	5	10	1	0	0	7
10×10	13	2	0	3	4	8	3	0	2	5
16×16	12	3	0	3	4	7	4	0	2	5
20×20	12	3	0	3	4	7	4	0	2	5
32×32	12	3	0	3	4	7	4	0	2	5
40×40	10	5	0	3	4	7	4	0	2	5
80×80	10	5	0	3	4	7	4	0	2	5
160×160	8	6	0	4	4	5	7	0	3	3

Chapter 7
Comparison of Identification Techniques

This chapter compares the performances of the new species-identification techniques. The main techniques were the new Eigenface technique, Eigenface and Fisherface. The Support Vector Machine (SVM) was used to compare the performance of these techniques. Finally, all three techniques were compared, to find which one produced the best result for cat and possum identification.

7.1 Different Techniques for Black and White Images with Changing Resolutions

In this section an investigation of the main techniques (a) Eigenface, (b) Fisherface and (c) SVM were compared against (d) the new Eigenface technique with distance algorithm and (e) the new Eigenface technique with error weights. All five options were applied on black background and white face training and test images. The class separation was investigated using the following image resolutions: 10 × 10, 20 × 20, 40 × 40, 80 × 80 and 160 × 160, shown in Figure 7.1.

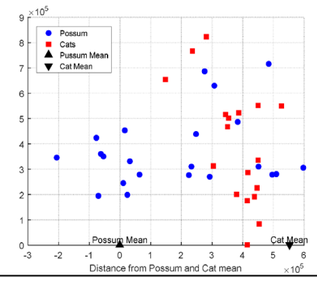
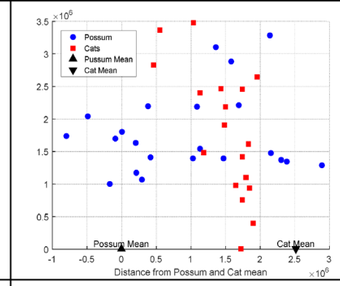
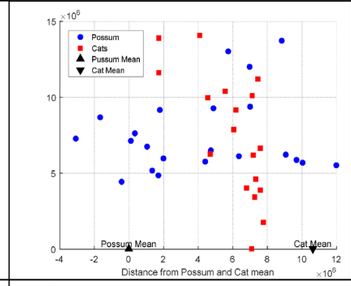
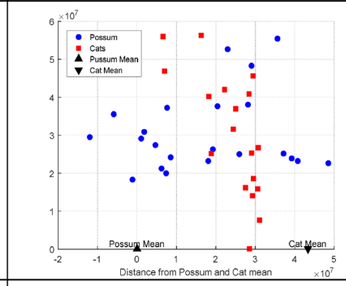
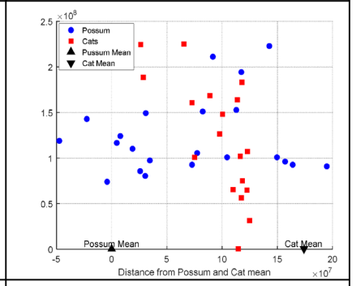
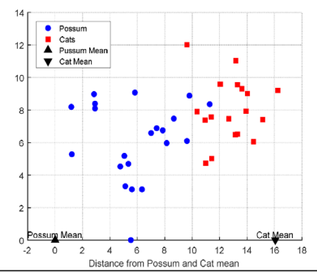
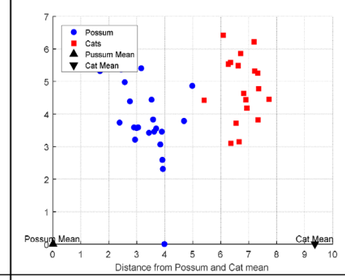
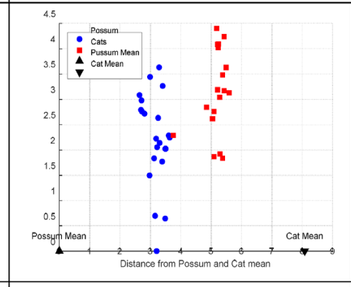
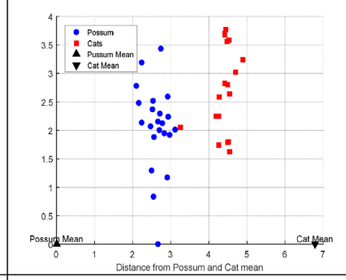
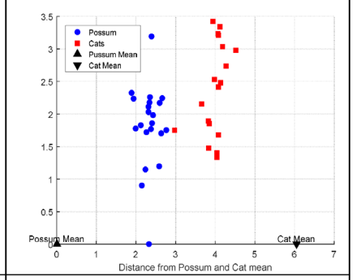
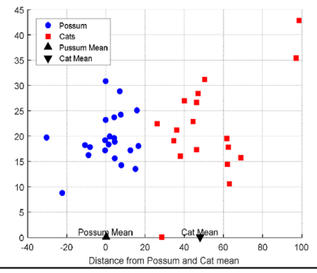
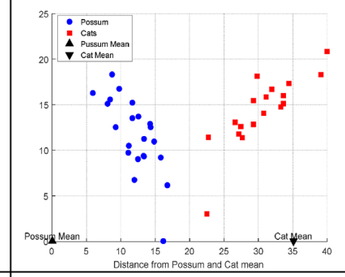
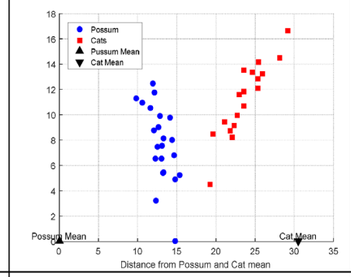
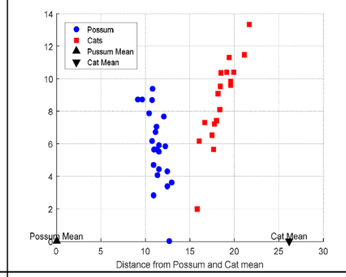
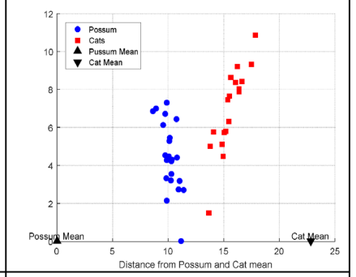
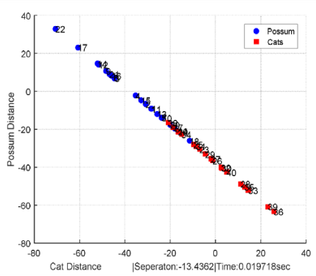
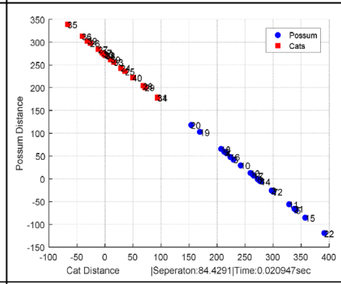
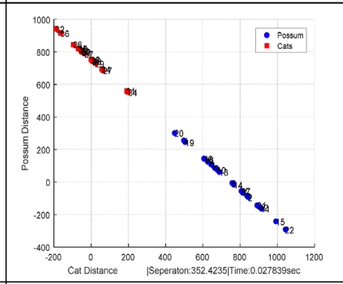
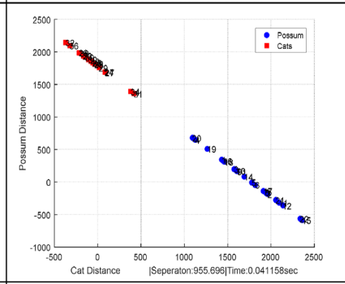
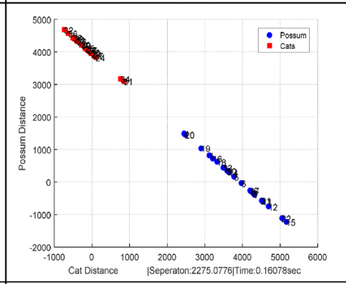
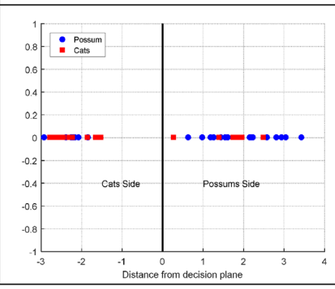
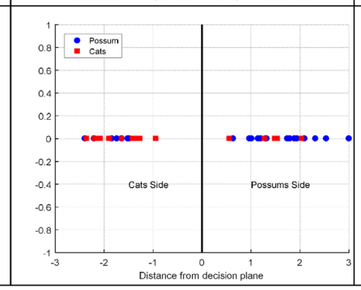
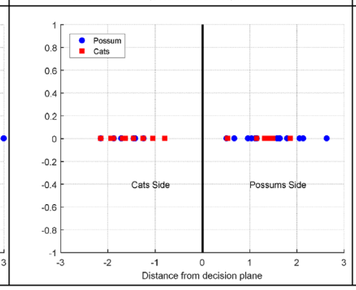
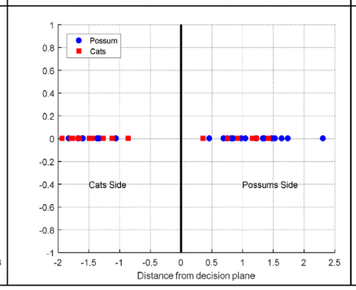
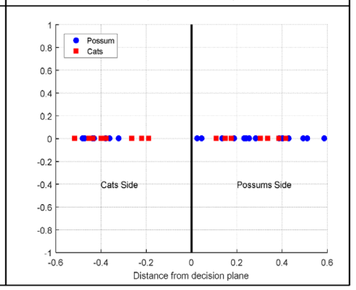
	Resolution 10x10	Resolution 20x20	Resolution 40x40	Resolution 80x80	Resolution 160x160
Original Eigenface Technique					
Proposed Eigenfaces with developed distance algorithm					
Proposed Eigenfaces with error weights					
Fisherfaces					
Support Vector Machine					

Figure 7.1: Comparison of all five techniques with varying image resolution

When comparing the results from Figure 7.1 for all five techniques, it can be clearly seen that the proposed Eigenface distance technique with error weights gave the best performance under different resolutions with black background and white face images. The second-best separation was achieved by Fisherface technique. At the lowest resolution, it could not separate two species completely. There was an overlap between two species of -13. The worst separation was obtained from the standard Eigenface technique and SVM.

As the resolution increases, the Fisherface technique's class separation improves considerably. Similarly, the separation of the new Eigenface with error weights technique improved with resolution, but the maximum separation plateaus at 40 × 40 resolution. The developed Eigenface technique with distance algorithm performed much better at higher resolutions, but there were still some overlaps and misclassifications. On the other hand, SVM's classification rate decreases as the resolution increases. This is mainly since in high dimensional space there is always a possible solution plane to separate the small training data set. When trying to classify the test data, some will be located on the wrong side of the decision plane

7.2 Receiver Operating Characteristic Identification for Black Background and White Face Images

Figure 7.1 shows that it is possible do a visual representation of the performance of each technique, but it is difficult to analyse the separation of each technique using these different sets of plots with different sets of axes scales. So Receiver Operating Characteristic (ROC) technique was used for this research, as a tool to analyse the different techniques and compare their performance, as discussed in section 2.13 of chapter 2.

In this application ROC was used to compare the performance of the five techniques. In order to measure the sensitivity and specificity of the plot shown in Figure 7.1, the decision boundary gradient was increased in 1° increments. At each increment the specificity and sensitivity were measured, shown in Figure 7.2. Then the ROC curves were plotted for each plot, shown Figure 7.3.

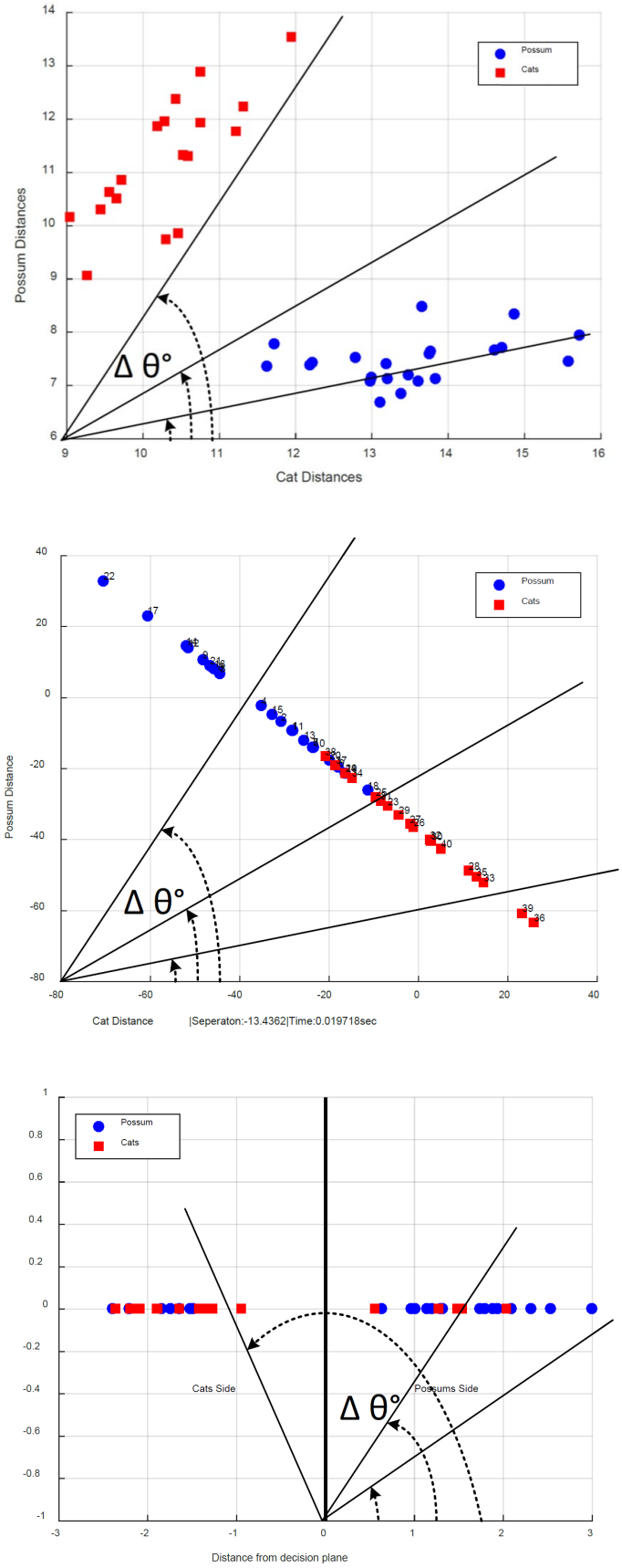


Figure 7.2: ROC measurements for (a) Eigenface (top plot); (b) Fisherface (middle plot); and (c) SVM (bottom plot)

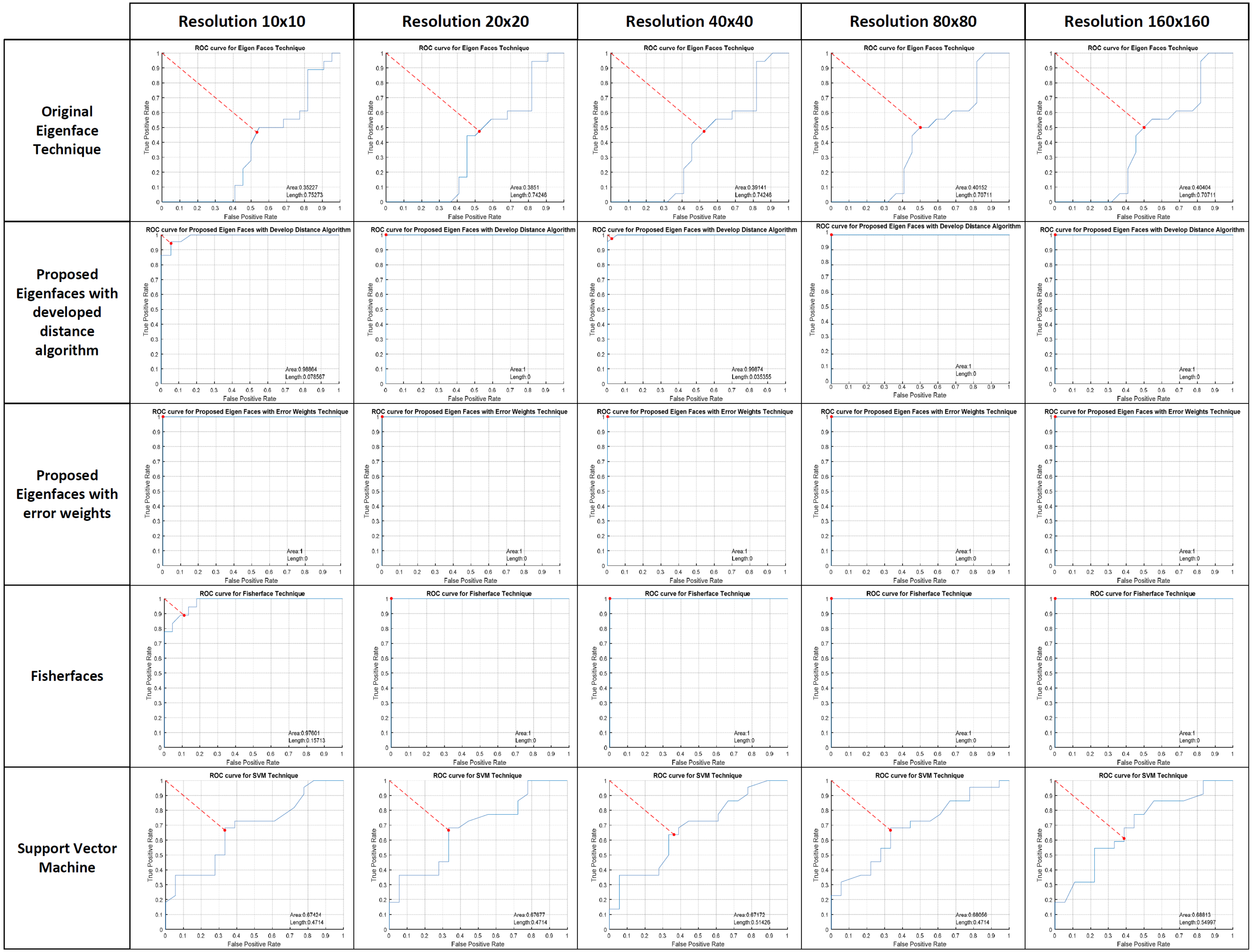


Figure 7.3: ROC curves of the different techniques for black and white images with changing resolution

Once all the ROC curves were plotted, the performance of each technique was analysed and then, comparing the areas under the ROC curves, the best performance is achieved by the Fisherface technique having the largest area under the ROC curve of 1 for the 160 × 160 resolution. At the same resolution, the second-best performer was the Eigenface technique with error weights. The SVM and the standard Eigenface performance was the worst with values 0.69 and 0.40 respectively, shown in Figure 7.3.

The 10 × 10 resolution for the standard Eigenface technique had the worst performance, with an area under the ROC curve of 0.35. The best performance was by the new Eigenface technique with error weights, with an area under the ROC curve of 1 with full separation. The second-best performer was the new Eigenface technique the distance algorithm, with an area of 0.99. The Fisherface technique had a smaller overlap and hence an area of 0.97. The SVM technique showed somewhat similar performance to the standard Eigenface technique at lower resolutions, shown in Figure 7.3.

The shortest length to the ROC curve from the top left-hand side of the sensitivity axis will give the optimal separation point for the two classes. Also, the shorter the length, the better the separation between classes. At lower resolutions the best separation was obtained by the new Eigenface technique with error weights. The second-best separation was by the new Eigenface technique with distance algorithm, and the worst separation from the standard Eigenface technique, shown in Figure 7.3.

At higher resolutions, the best separation was obtained by the Fisherface technique and the second-best by the new Eigenface technique with error weights. The worst separations were shown by the standard Eigenface technique and SVM, shown in Figure 7.3.

When analysing the performance of the data it is evident that the Fisherface technique shows the best performance at higher resolutions, but the new Eigenface technique with error weights at lower resolutions has the best performance. The new Eigenface technique with error weights managed to separate the cat and possum classes completely at all resolutions.

7.3 Performance of Each Technique with Varying Number of Training Images

An important feature is to compare the execution time for each identification technique with different image resolutions. Table 7.1 shows the execution time for each technique for varying image resolutions, where the last two columns of the table gives the ROC distance and area under the ROC curves.

In order to measure the execution time, a laptop computer was used with a 64-bit, Intel quad core, i7-2820QM CPU at 2.30GHz with 8.00 GB of system memory with Windows 10 operating system. To benchmark the time, every technique was looped for 1000 times to obtain an accurate average time. The MATLAB *'tic'* and *'toc'* functions were used to measure the time.

Table 7.1: Execution time and ROC properties for different image resolutions

Technique	Resolution	Execution time (sec)	ROC distance	ROC area
Eigenface	10×10	0.048	0.75	0.35
	20×20	0.051	0.74	0.38
	40×40	0.058	0.74	0.39
	80×80	0.071	0.71	0.40
	160×160	0.090	0.71	0.40
Proposed Eigenface algorithm	10×10	0.064	0.08	0.99
	20×20	0.065	0.00	1.00
	40×40	0.066	0.04	0.99
	80×80	0.072	0.00	1.00
	160×160	0.094	0.00	1.00
Proposed Eigenface with Error Weights	10×10	0.063	0.00	1.00
	20×20	0.065	0.00	1.00
	40×40	0.066	0.00	1.00
	80×80	0.072	0.00	1.00
	160×160	0.095	0.00	1.00
Fisherface	10×10	0.019	0.16	0.97
	20×20	0.021	0.00	1.00
	40×40	0.028	0.00	1.00
	80×80	0.041	0.00	1.00
	160×160	0.161	0.00	1.00
SVM	10×10	0.118	0.47	0.67
	20×20	0.121	0.47	0.68
	40×40	1.348	0.51	0.67
	80×80	1.764	0.47	0.68
	160×160	1.842	0.55	0.69

Figure 7.4 shows a combined bar-chart plot of all three properties from Table 7.1 for each technique. It is clear from the plot that the Fisherface technique has the best performance at higher resolutions; as the resolution increases, however, the execution time increases by 60% to 75%. At lower resolutions the Fisherface technique's performance is poorer than that of the new Eigenface technique with error weights.

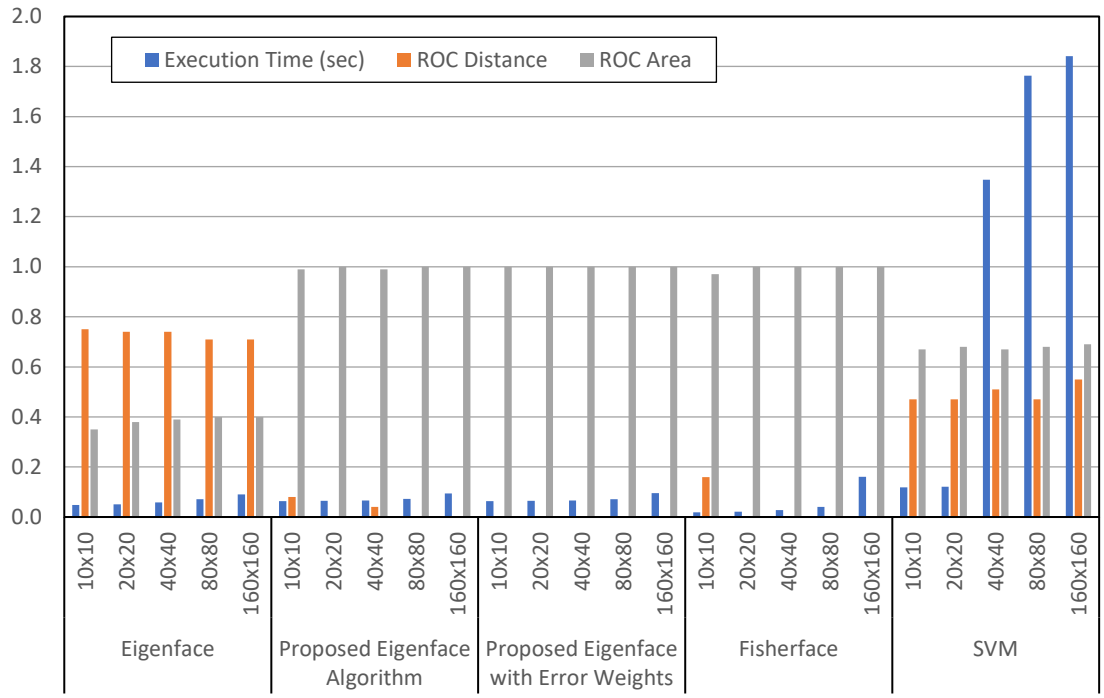


Figure 7.4: Image resolution vs runtime and ROC properties

The new Eigenface technique with error weights showed a small time-variance of 0.000172 across the whole range of resolutions but gave the best separation for all resolutions. When compared to the standard Eigenface technique and SVM, it has better separation overall with acceptable execution time, and at lower resolutions, this technique outperformed the Fisherface technique. Since these developed algorithms are to be deployed on an embedded platform, the lower-resolution training images are attractive where there are limited amounts of processing memory available.

The SVM had the longest execution time of the five techniques, and is also one of the worst-performing techniques for this limited number of training images application. To improve the performance of this technique, it will need to have a larger number of training images i.e. > 5000 for the hyperplane to be a meaningful separation between the two

classes. With lower-resolution images SVM still has a lower separation result than other techniques. The longer execution time is one of the drawbacks of this technique.

The standard Eigenface technique and the new Eigenface technique with distance algorithm showed slightly increasing execution times across the different resolutions, with the standard Eigenface technique having the worst overall performance. Therefore, it is evident that the two new Eigenface techniques are superior to the standard Eigenface technique.

7.4 Performance in Optimised Conditions

To investigate the effect of the number of training images, it was decided to plot ROC curves for varying numbers of training images. In this trial the number of training images used the following five sets of (i) 5 cats and 5 possums images, (ii) 10 cats and 10 possums images, (iii) 15 cats and 15 possums images, (iv) 20 cats and 20 possums images and (v) 33 cats and 33 possums images. All five techniques were tested with black and white images for a fixed image resolution of 40 × 40. The results for these trials are shown in Figure 7.5.

The values for the ROC distance and area for all five techniques are plotted as a combined bar chart, shown in Figure 7.6.

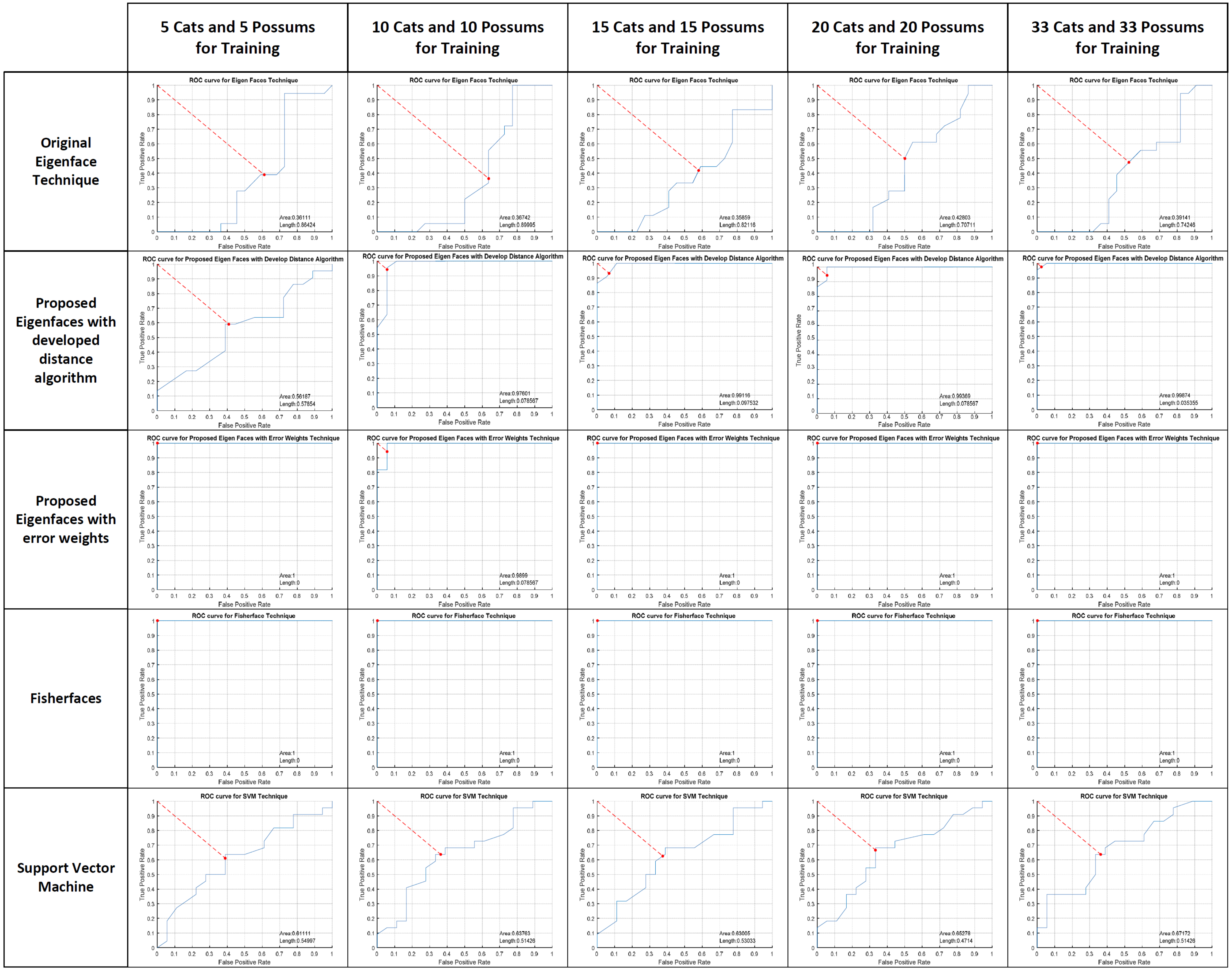


Figure 7.5: ROC properties for 40 × 40 black and white images with changing numbers of training images

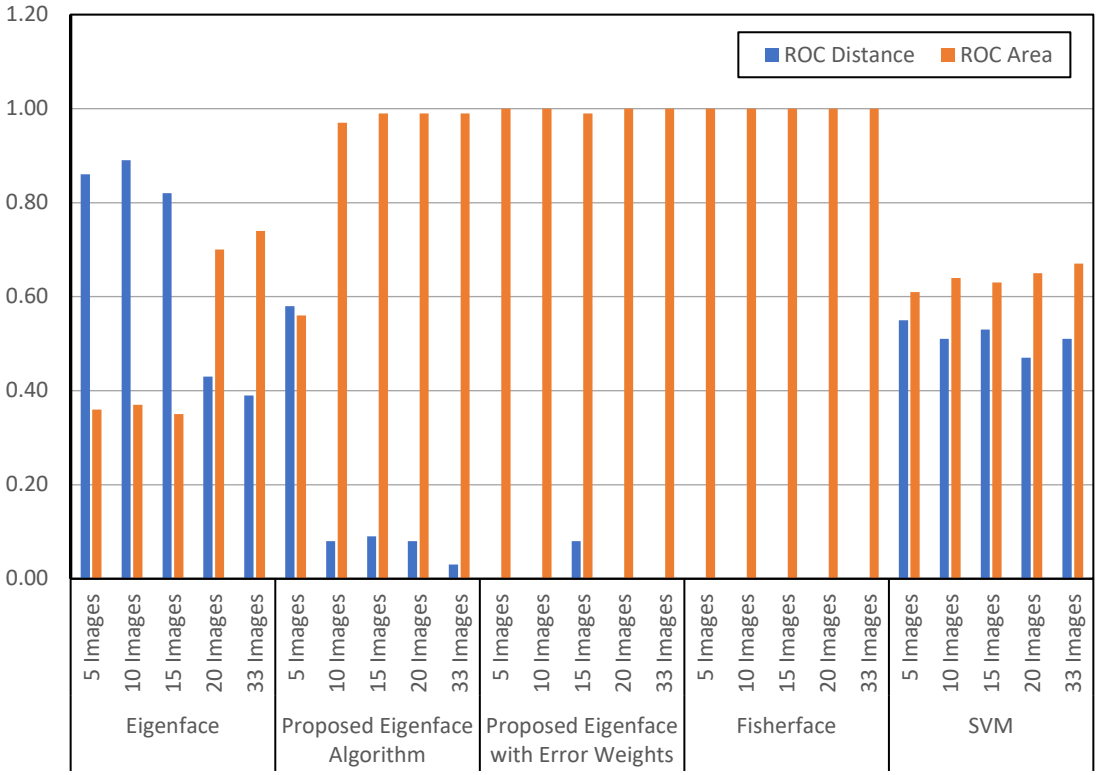


Figure 7.6: ROC properties for varying number of training images

The results in Figure 7.6 show that the Fisherface technique outperforms all the other techniques with a black background and white face and 40 × 40 resolution images. The standard Eigenface technique had the worst performance, with the largest ROC distances and ROC areas under the curve.

The proposed Eigenface with error weight technique achieved full separation for all the scenarios except the 10 training images scenario. As the training image count increased, the performance of the technique increased, i.e. the best separation is with 33 cat and 33 possum training images. This technique separates species efficiently, when the training images are similar to the test image. More similar images produce better results with this technique, hence there are lower separation values with lower numbers of training images.

As mentioned in this section the SVM tends to perform better with larger numbers of training images than with a low number of training images, but the SVM hyperplane does not produce the optimal class separation. This is seen in the plot in Figure 7.6; as the training image number increases, the class separation improves.

7.5 Comparison of recent Eigenface developments with developed Eigenface technique

After comparing the results of all five techniques used in this research, it can be concluded that the most useful techniques for application are the new Eigenface technique with error weights and the Fisherface technique.

When comparing all the characteristics, the best performance was by the Fisherface technique, with higher resolution images greater than 10 × 10 resolution. The developed Eigenface technique with error weights performs better with lower resolutions images. When comparing both techniques, shown in Figure 7.4, both produce usable results ROC areas under the curve for different resolution training and test images. The separation of targeted species classes (ROC distance) in both cases was satisfactory for this smaller training sample-based identification system.

To achieve satisfactory separation, the image colour scheme is critical. As previously discussed in chapters 4 and 5, it is important to optimise the colour scheme of both training and test images to maximise the class separation. Of all the trialled colour schemes, the most effective was a black background and white face without any fur or colour details. Identification was made by the face shape of the animal.

It was found that as the resolution increased, the runtime of Fisherface technique increased. However, the corresponding increase was lower for the new Eigenface technique with error weights.

It can be concluded from this chapter that the optimum technique for this application is Fisherface, given the fact that training and test image resolutions need to be larger than 20 × 20 pixels. If the training and test image resolution is lower than 20 × 20, the new Eigenface technique with error weights would perform better. If this research moves into implementation, both techniques can be trialled to investigate the best resolution versus power consumption and processing time.

7.6 Black and White Image Detection Rate with Different Resolutions

This section compares the performance of developed error weights based Eigenface technique with sparse Eigenface approach [152], Fractional Eigenfaces [98] and Eigenface with Naïve Bayes approach [100][21].

It is important to compare the developed technique against recent approaches to contrast the benefits. To perform this trial black and white image are used with varying resolutions (10 × 10, 20 × 20, 40 × 40, 80 × 80 and 160 × 160). 33 possum images and 33 cat images are used to train each technique. As shown in Figure 7.1 developed Eigenface technique and Fisherface technique had full separation on cats and possum test images with the above colour scheme and resolutions. The Figure 7.7 below shows the accuracy of each technique. Accuracy was calculated with 40 test images (22 possums and 18 cats). For each technique correctly classified images were calculated as a percentage.

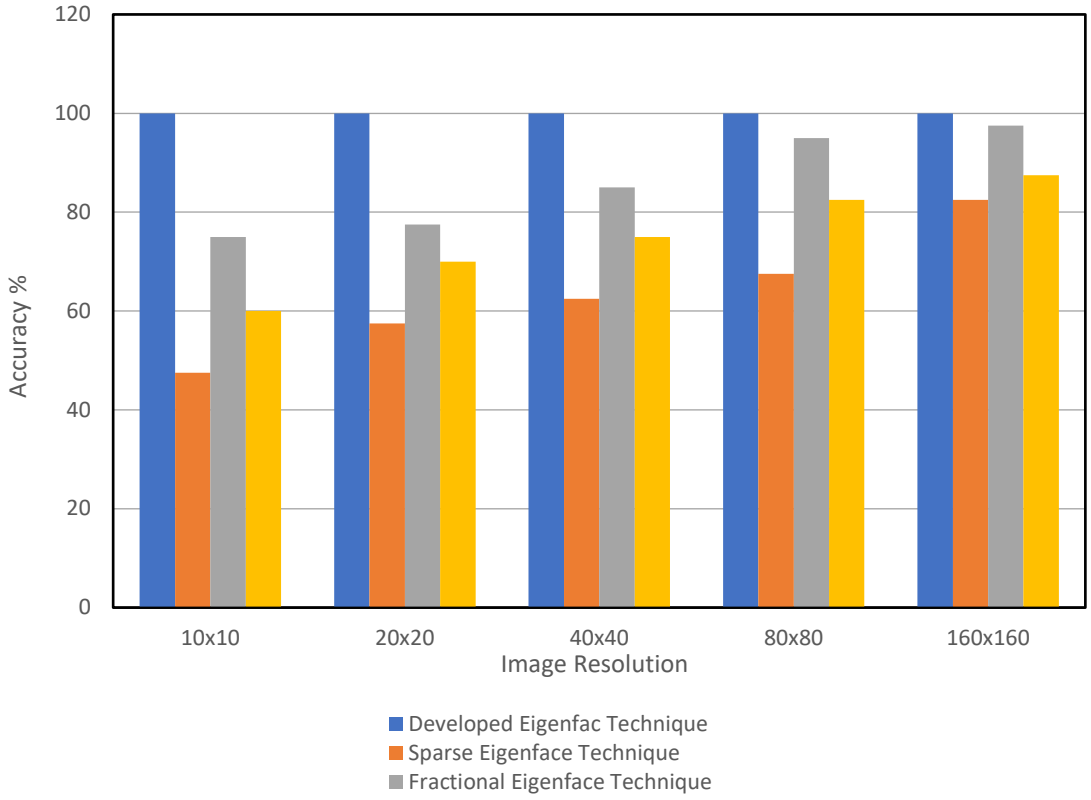


Figure 7.7: Comparison of recent Eigenface developments

[21] This is to acknowledge authors of papers [97, 99 and 151] for providing identification results with possum's and cat's datasets for this research.

When comparing the three techniques above, the Fractional Eigenface technique has a better performance across all the different image resolutions. The best separation is achieved at 160×160 with 97.5% accuracy. As the resolution increases the accuracy of the techniques improve. This can be observed with developed Eigenface technique as well (the separation margin increased with the resolution). The other two techniques (i) Sparse Eigenface and (ii) Eigenface with Naïve Bayes have a similar performance with best accuracy of 82.5% and 87.5% respectively with 160×160 resolution. The developed Eigenface technique with error weights has a better performance across different resolutions.

7.7 Performance of the developed Eigenface technique with dogs

The main aim of this trial was to investigate the performance of the developed Eigenface with error-weights technique for detection of different facial images of other animals. For this trial a different animal group was introduced, in this case images of dogs. There are three reasons for selecting dogs. Firstly, many dog images are available on internet. Secondly dogs' main facial features (ear and nose shape) are like possums and cats. Lastly dogs are curious animals, therefore they have probability of entering a trap setup. To train the system 33 dog images were used, shown in Figure 7.8. From the results shown in Figure 7.1, the 40 × 40 image resolution produced acceptable separation between possums' and cats' classes. Therefore, to conduct this trial using dog images the same resolution was used.

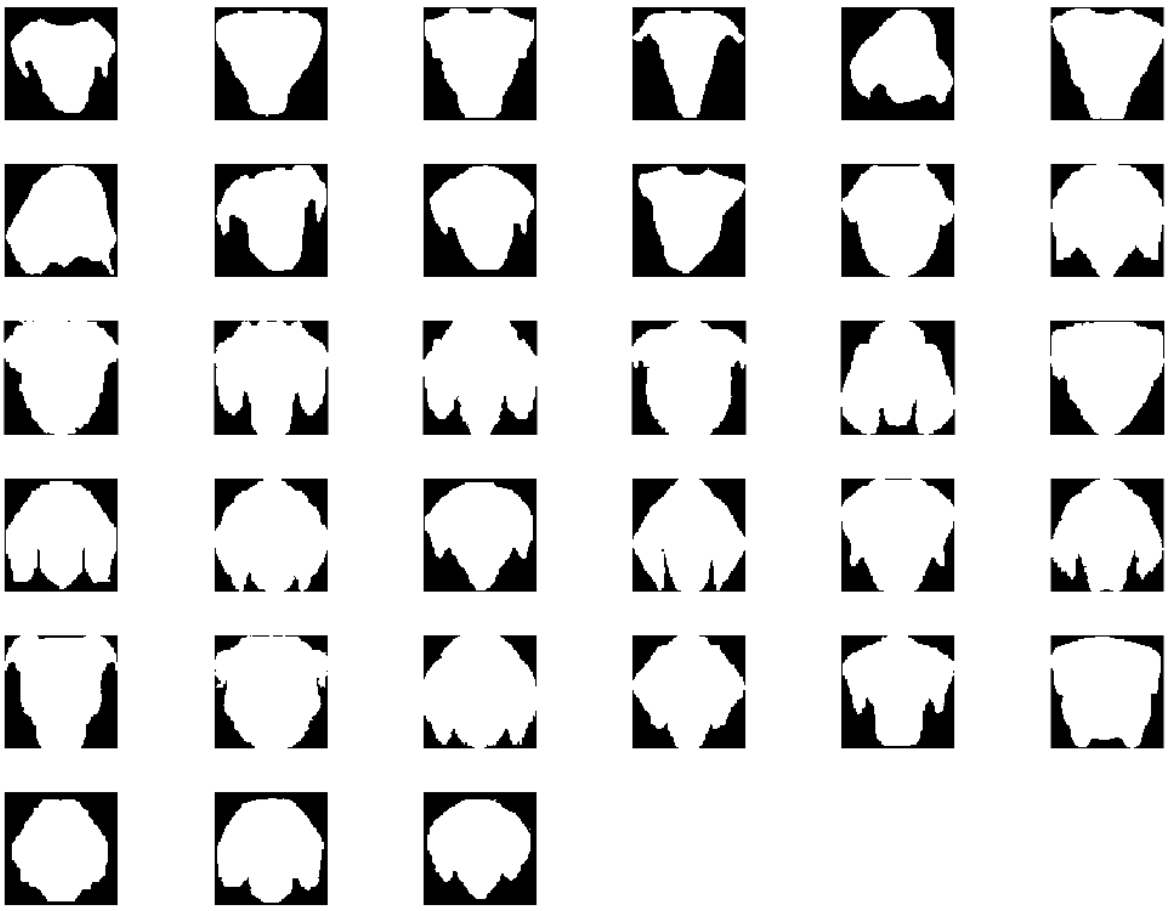


Figure 7.8: Dog training data set

To test the system 22 possums', 18 cats' and 18 dogs' images were used, these test images were not related to training images. All the test images had the same resolution as training images. The ℓ_2 norm was used measure the distance for each test image from each class average face, shown in Figure 7.9.

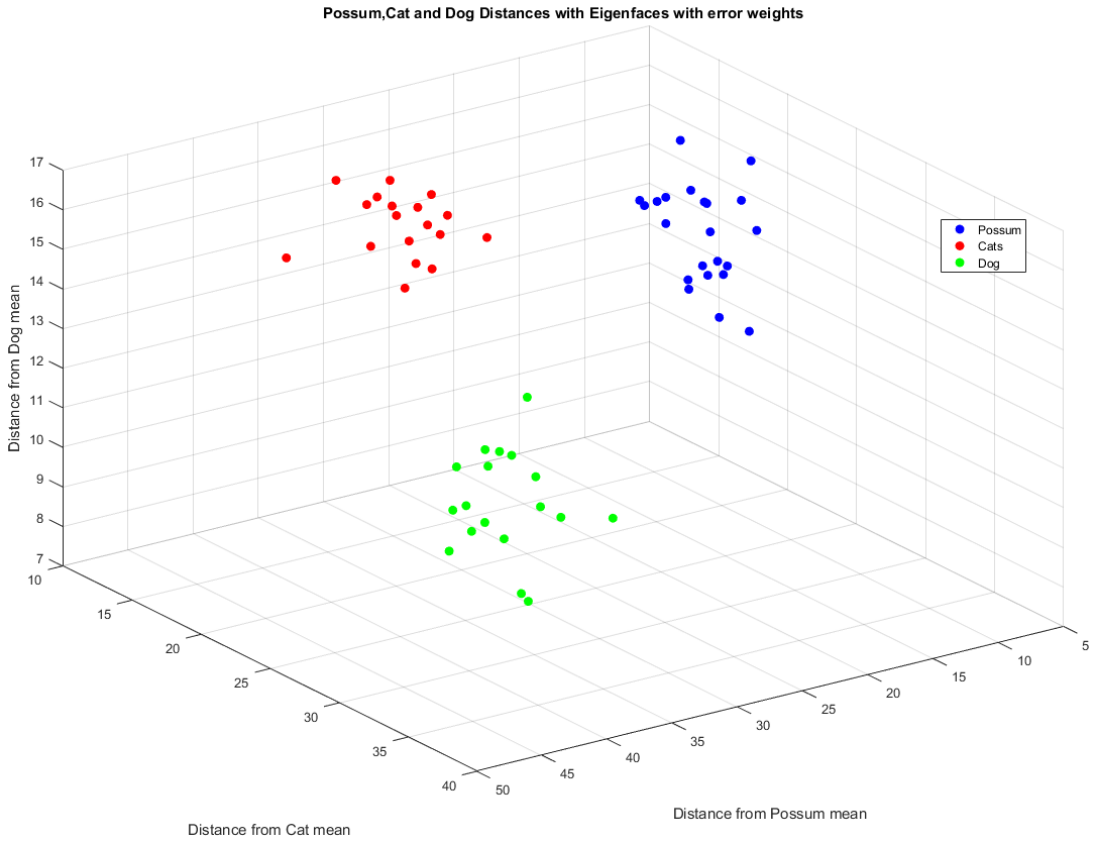


Figure 7.9: Three class (possums, cats and dog) separation with Eigenfaces with error weights

Figure 7.9 shows an acceptable separation between three all classes. To measure the separation multicategory receiver operating characteristic analysis [153] was used. Just like area under the ROC curve, the Hypervolume Under the Manifold (HUM) 3D ROC surface is used to analyse the separation of the three animal classes. The total HUM is one (1×1×1). As the separation increases HUM value gets closer to one, otherwise value gets closer to zero.

To plot the 3D ROC surface HUM calculator package[22] is used. This package plots the 3D ROC surface and calculates HUM value for a given dataset. In 3D ROC plot each axis represent Probability of Correctly Classifying (POCC) for each animal group, shown in Figure 7.10.

[22] HUM calculator package used for this research was accessed via https://public.ostfalia.de/~klawonn/HUM.htm

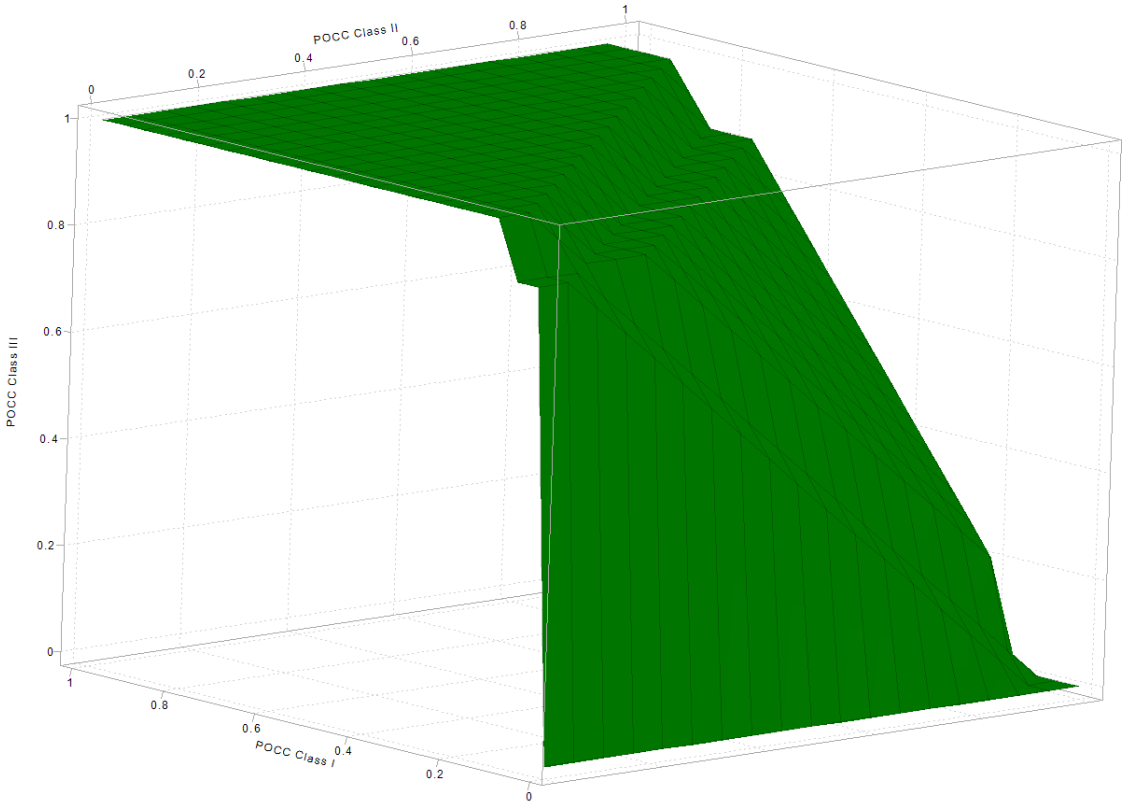


Figure 7.10: Three class (Class I: possums, Class II: cats and Class II: dog) ROC analysis

For above three classes the HUM value was calculated to have a value of 0.99 and since this value is very close to one all three classes has a full separation. Also, with multiclass problem, once distances are calculated from class average face, this data can be used with SVM for further classification.

Chapter 8

Receiver Operating Characteristic-based Feature Selection Method for the Principal Component Analysis, Fisherface and Support Vector Machine Techniques

This chapter presents a new face feature-based detection approach for the Principal Component Analysis (PCA), Fisherface and Support Vector Machine (SVM) techniques. The main aim of this approach is to reduce the dimensionality of the training data. By doing this the accuracy of the detection techniques can be improved with lower numbers of training images.

To reduce the density of the training images, the main features from each animal face were measured. These main features give meaningful representations for each training image. So, after these measurements, each training image produces a feature vector, which is used to train each discrimination technique. For each test image the same features are measured, hence each test image is represented by a feature vector.

By introducing this technique the dimensionality of training data was reduced dramatically, and the Eigenface technique is no longer required to reduce the dimensionality by scrambled covariance matrix [154; Chapter 3]. Therefore, PCA was used with the full covariance matrix, to discriminate between cats and possums in this application. Similarly, with the Fisherface technique SVD is not required, as the training data vector is smaller than the number of training images. Hence the training data set is much smaller than when using the original images, and the computational time is faster.

Finally, this approach was further improved by the Receiver Operating Characteristic (ROC)-based critical feature vector selection technique. By identifying critical features from training data sets, the separation of each technique can be improved further. Moreover, by using this method the number of features required for training the above technique is a much less, and so the computational time can be improved further by introducing this method.

8.1 Feature Extraction

In this trial the main features of the animal's face were manually measured. There were 12 main features. The first measurement is the centre point between the eyes to the 0°

point of the outer edge of the animal's face i.e. top of the head. Then other measurements are measured at every 45° interval up to 315°. Then the following measurements were taken: the distance between eyes, from the left eye to the centre of the nose, from the top of the left ear to the centre of the nose, and between ear tips, shown in Figure 8.1.

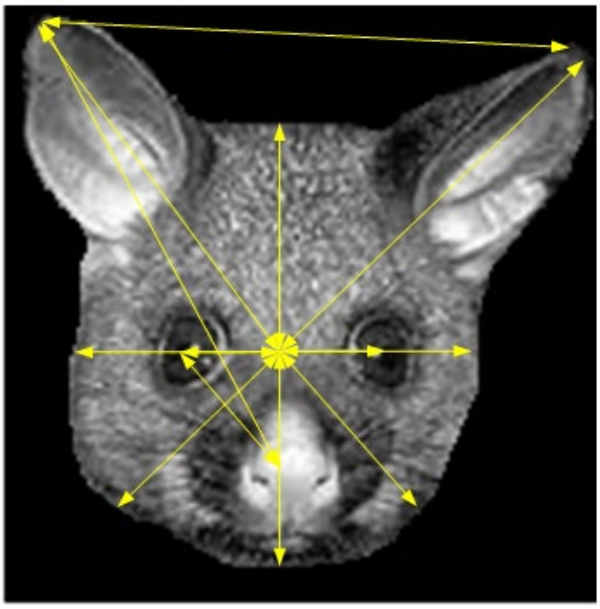


Figure 8.1: Extracted face features

Once all 12 features had been extracted, they can be plotted as shown in Figure 8.2. A few features have complete overlap between the two animal species, such as distance between ear tips and distance between left ear tip to the centre of nose. A few other data sets have smaller overlaps. So, it was important to choose the most effective data sets for species separation.

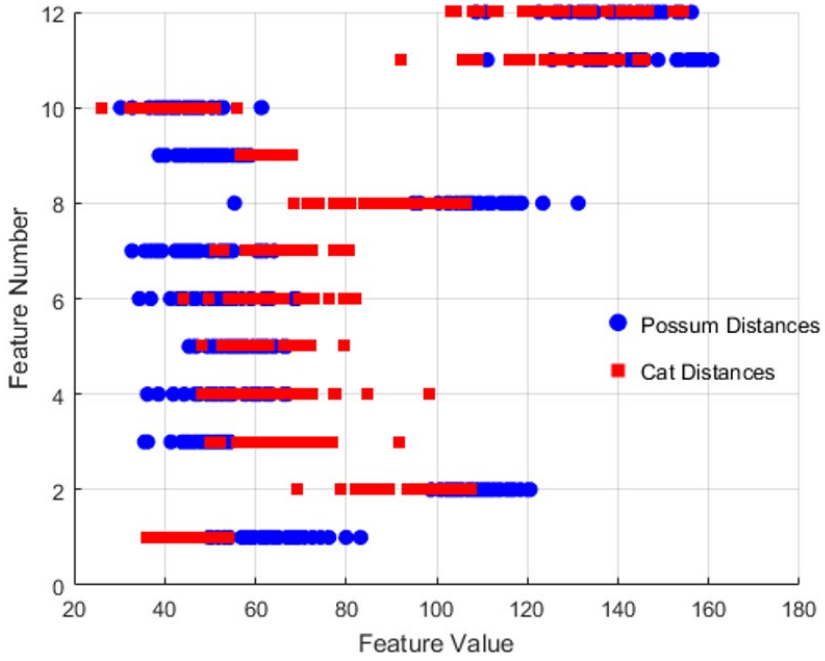


Figure 8.2: Extracted face features plot

8.2 ROC-based Critical Feature Extraction

To identify the critical datasets, ROC-based feature extraction was developed. For each measured feature a ROC curve was plotted. For each individual feature the specificity and sensitivity were measured according to Figure 7.2 (c) in chapter 7. Once all the measurements were obtained, ROC curves for each feature could be plotted, as in Figure 8.3.

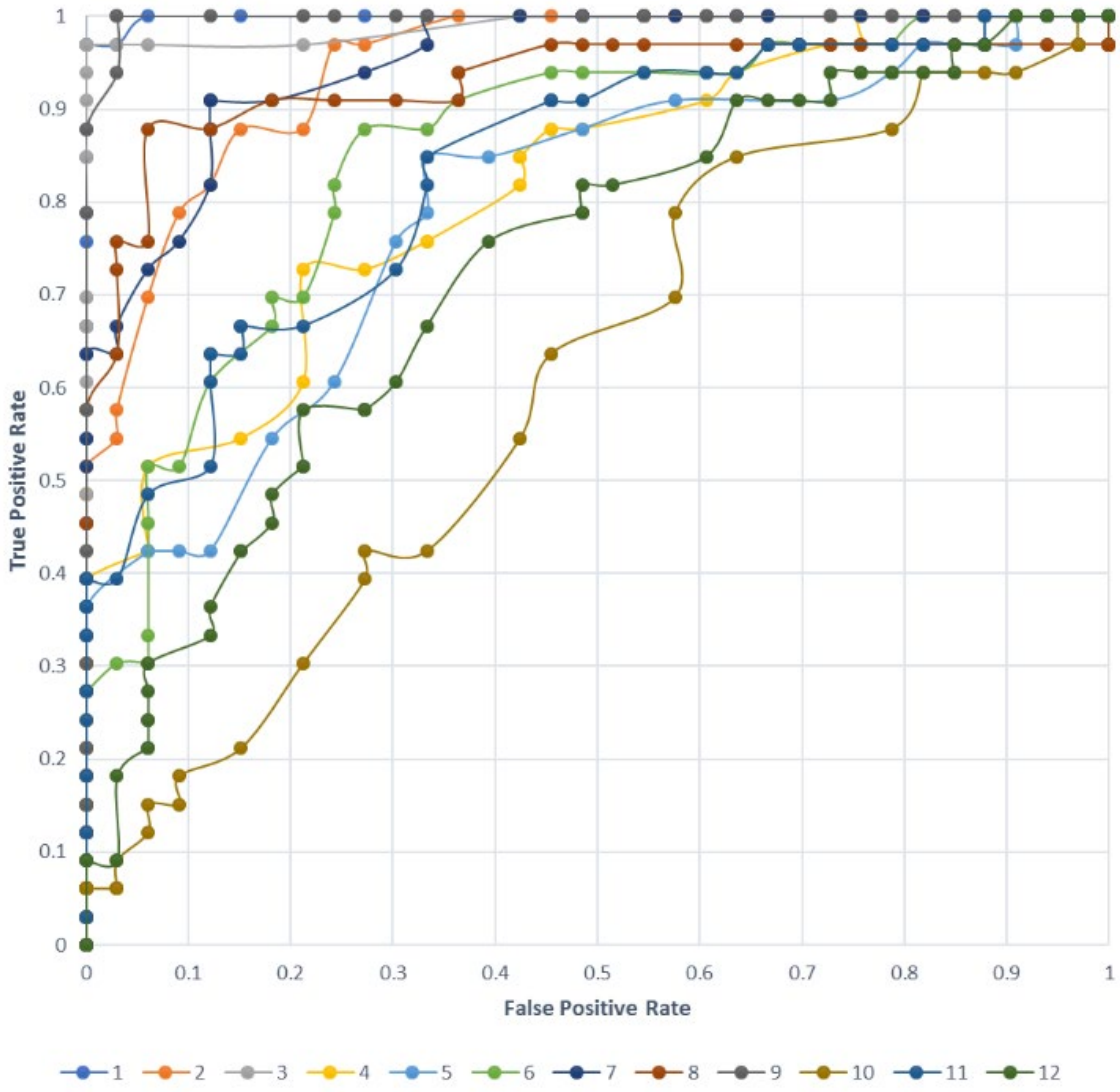


Figure 8.3: ROC curves for each individual feature

Feature vectors with higher weighting or larger eigenvalues can be selected by analysing the area under each ROC curve. The curves with the greatest areas produce feature vectors with higher weightings. All the curves that located under curves with larger areas under the ROC curves, can be ignored, this is because most of the separation data is contained by the curves with larger areas.

According to Figure 8.3, the most meaningful data can be extracted from graph 1, 3 and 9. Once these vectors are identified they can be used with the PCA, Fisherface and SVM techniques to investigate species separation. By identifying the main feature vectors, the execution time of the final system can be improved by only extracting identified vectors. Pre-processing time can therefore be reduced in the final embedded system.

8.3 Results and Comparison

To investigate the performance of ROC-based feature extraction, three selected feature vectors were used with the three main identification techniques. The trials were separated into three groups:

1. The first set of trials was conducted with original animal images, 160 × 160, 8-bit greyscale. The results from this trial are shown in Figure 8.4.
2. Second set of trials was conducted with measured 12 different features, shown in Figure 8.1. The results from this trial are shown in Figure 8.5.
3. The last trials were conducted with selected features using proposed technique above. In each scenario, performance of each technique was analysed using a ROC curve. The results from this trial are shown in Figure 8.6.

For each trial, 33 cats and 33 possums training images were used. For testing 22 possum images and 18 cat images were used. Training and test images are not related to each other; they are totally independent of each other.

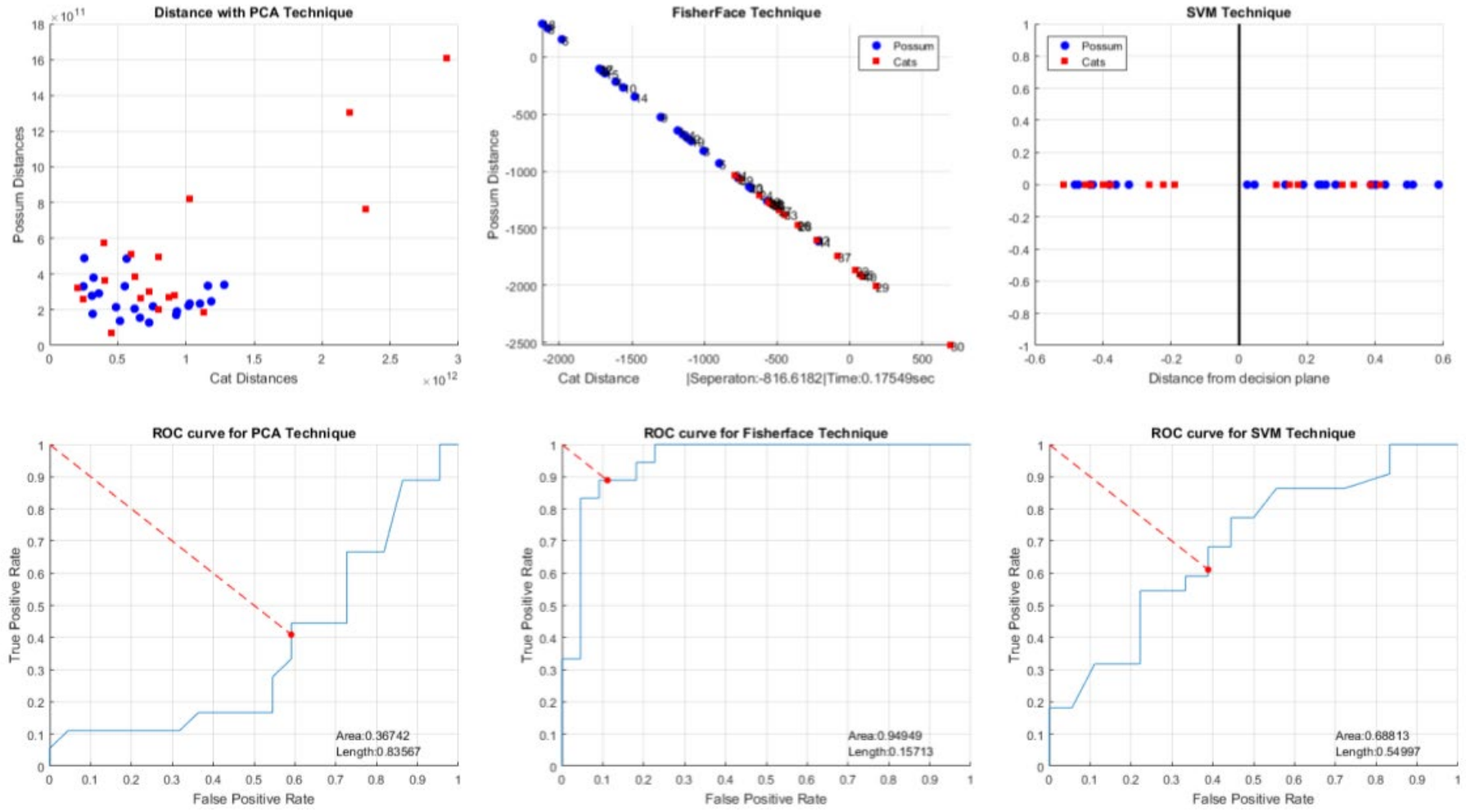


Figure 8.4: Results for the PCA, Fisherface and SVM techniques for Class Separation with Original Images

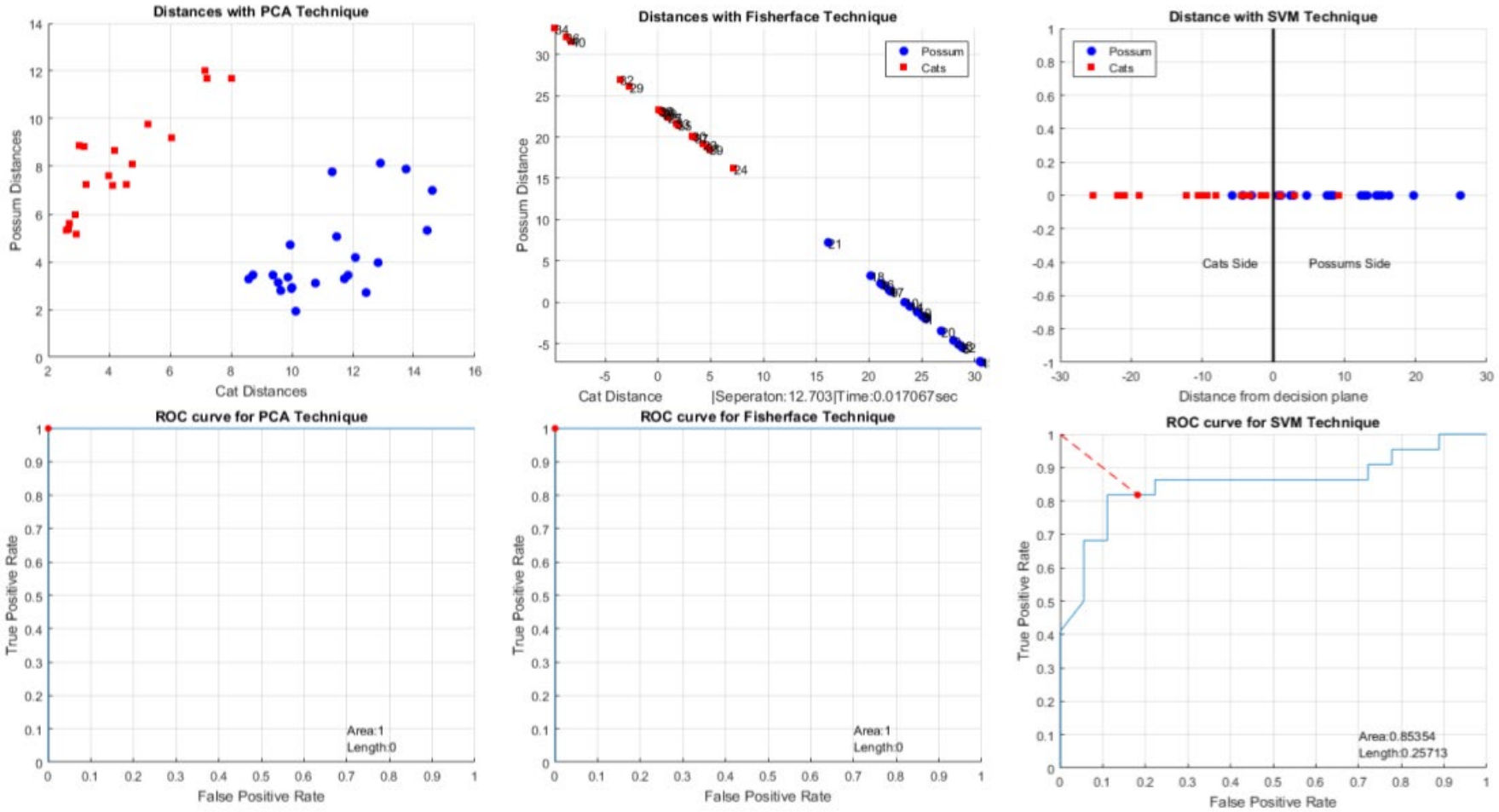


Figure 8.5: Results for the PCA, Fisherface and SVM techniques for Class Separation with Face Features

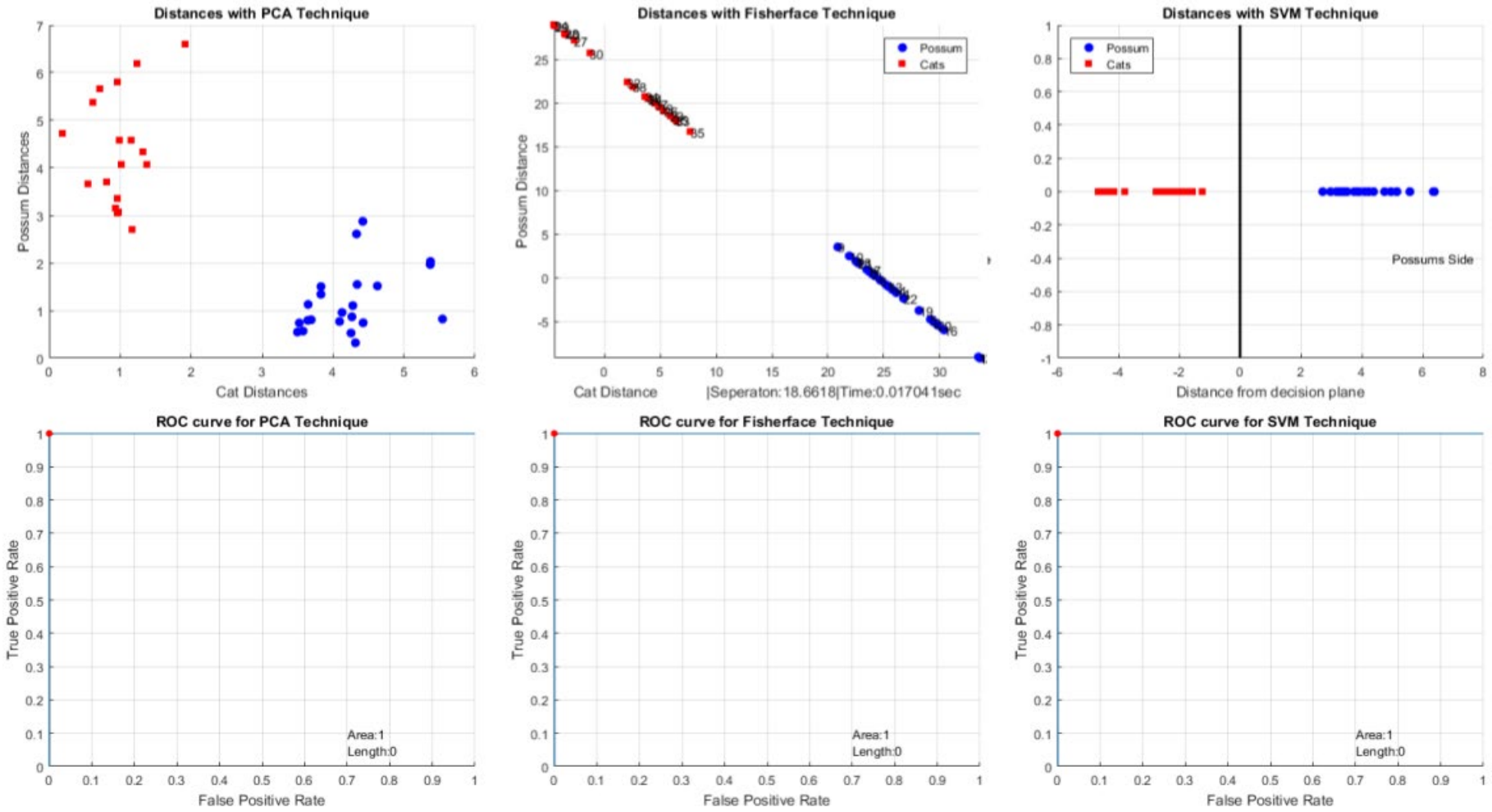


Figure 8.6: Results for the PCA, Fisherface and SVM techniques for three different trials for Class Separation with Selected Features

For this set of trials, the original cat and possum images were used with three trialled techniques. From Figure 8.4 it follows that none of the three techniques was able to separate the two species fully. The main axis test data from the two training groups overlapped, making it hard to find a clear separation. This was mainly due to the fact that images had similar colour details between two classes. Also, most of the original images contained many data points, and therefore processing time was high.

The best result was achieved by the Fisherface technique, where the other two techniques did not manage to separate the cats and possums at all. There were big overlaps between the two species groups. When the areas under the ROC curves were analysed, the PCA and SVM techniques had very small areas, but the Fisherface had an acceptable area. Since there was poor separation with the original images, a second set of trials was conducted with selected face features.

In this application from each training image, those selected features have been measured. Therefore, for each training group there were 33 × 12 feature vectors. These feature vectors were used to train the above techniques.

This developed feature-based technique produced full separation between cat and possum classes with PCA and Fisherface techniques, shown in Figure 8.5. This represents a large improvement compared to first set of trials. With the SVM technique, there is still some overlap between the two classes. But there is improved separation, compared to the original images, shown in Figure 8.5.

The ROC curves produced a maximum area of one for both the Fisherface and PCA techniques. Due to the overlap around the decision boundary, SVM's ROC curve produced a smaller area than the other two techniques.

In the last set of trials, three main ROC graph-based feature vectors, features 1, 3 and 9 were used, shown in Figure 8.6. Since there were three main features, there were 3 × 33 feature vectors for each class.

With these selected features, full separation between cat and possum classes can be achieved. All three discrimination techniques managed to separate the two species successfully. The new ROC-based feature selection method therefore helps to identify the most important feature vectors within the training data set.

With the feature-based technique the performance time is much less than when using the original images. The Fisherface technique alone with the original images had an execution time of 0.175 seconds, with the selected feature-based technique the execution time was reduced to 0.017 seconds, an 82.3% decrease, and as the number of feature vectors get less, execution time reduces dramatically.

Moreover, this new technique allows users to apply mature discrimination techniques such as PCA, Fisherface and SVM with a small amount of training data, and still produce good discrimination in high-dimensional space.

When analysing the three discrimination techniques, the Fisherface technique outperformed the others in all three scenarios. This is due to its class separation and boundary maximisation, with within-class scatter and between-class scatter matrices[23]. The new optimisation removes all the local variation within each class and at the same time maximises the class separation.

$$W_{opt} = \arg max_W \frac{|W^T S_B W|}{|W^T S_W W|} \tag{8.1}$$

The solution for the equation (8.1) is given by solving the general eigenvalue problem, where W is eigenvectors of $S_W^{-1} S_B$; S_W is the within-class scatter matrix; and S_B is the between-class scatter matrix.

The SVM technique projects data into high-dimensional space and uses Lagrangian multipliers to maximise the margin within the support vectors for a maximised hyperplane. As the training data gets scatter into the high-dimensional space as they are, there is very small variation between two species and therefore this technique produces less accurate data. With the selected feature vectors, the spread between the species is increased; hence the better separation results. When the original images were used there were not enough training images to produce a suitable decision plane. Since there is higher dimensionality with a larger number of pixel values there is poor separation with the original images.

[23] A scatter matrix is a pair-wise scatter plot of several variables presented in a matrix format.

With lower numbers of training images and higher resolutions, PCA could not create enough eigenvectors with large enough eigenvalues to recreate the original information. Therefore, a large amount of useful information will be lost in high-dimensional Eigenspace. Every time a test image is projected into eigenspace, it is hard to regenerate the species' face with a reduced number of eigenvectors. Therefore, a poor result was produced with the original images. With the new feature-based method there are fewer feature vectors than actual images, therefore this problem can be solved within Eigenspace.

Chapter 9
Conclusion and Future Work

9.1 Conclusions

The main aim of the research presented in this thesis was to develop a new cat and possum identification system for smart trapping. Very little research has been conducted on small animal detection using a camera system. Most of the animal detection research has been carried out on large animals. Most existing small animal detection systems are based on animal footprints, and therefore this research represents a new approach to small animal management systems.

Initially, human identification systems were investigated in general before three main techniques were explored further was the (i) Eigenface technique, (ii) Fisherface technique and (iii) Support Vector Machine (SVM) technique. All three techniques have rarely been used for detecting small animals such as cat and possum.

During the research, Eigenface technique performance was enhanced by development of weight-based classification algorithm and was improved further by introducing test image error weights. This is a major improvement for the existing Eigenface technique. The combined approach of both improvements proven to be much better classifier for small animal detection.

The Fisherface technique showed improved performance compared to the standard Eigenface technique. However, when the new enhancements described in this thesis were applied, the new Eigenface technique with error weights proved to give improved results, compared to the Fisherface technique with lower resolution training and test images. The Fisherface technique still outperformed the new Eigenface technique with high-resolution images and lower numbers of training images. From the results, it is clear that both techniques are practical solutions for implementation in smart trap applications.

The SVM technique proved to be less usable with the feature set which contains in the training images. To form a useable decision surface, a larger number of training images would be required (> 5000). With smaller training sets and larger numbers of dimensions there are always a few decision surfaces, which is not optimal. To overcome this situation lower resolution training and test images were used. Even though it improved the

classification, the species separation was still not acceptable, compared to the other two techniques.

To optimise the species separation, the head orientation in the training and test images was corrected. To achieve this a new standard deviation-based edge-detection technique was developed. The eyes of the animal were detected using Hough transform-based circle detection. With respect to orientation of the eyes, the whole face was corrected on the yaw axis. The main aim of this approach was to optimise the face orientation of the animals so that when training and classifying all animals in the same class have similar appearance. From the trials it was evident that this approach improved the class separation further.

The colour scheme was also optimised to improve the separation. During the test stage it was found that the image colour scheme had a great effect on species separation. Different colour schemes were investigated, and it was found that a black background and white face with no features performed best. With this scheme, the techniques were trained and classified on face shape. A black and white image provides the maximum contrast between face shape and background.

Receiver Operating Characteristic (ROC) curves were used to compare the performance the ROC area and class separation the ROC distance of each technique. From the analysis of the results, Fisherface technique emerged as the most suitable for this application, given the image resolution is greater than 10×10. When the resolution is low, the Eigenface with error weights technique is the most suitable and proved to be much more effective than the existing Eigenface technique.

Finally, a newly developed ROC-based feature selection method was able to produce greater separation for Principal Component Analysis (PCA), Fisherface and SVM techniques. Additionally, this technique assisted the other techniques with selecting critical features. Because of the low dimensionality of the training data, the training time was much less than using traditional image-based approaches.

9.2 Suggestions for Future Work

The next step for this research would be to implement these identification applications onto an embedded system such as Raspberry PI. The latest Raspberry PI Zero, which is a single-board computer module, provides a low-cost powerful embedded module with minimal hardware options. Since there is minimal hardware overhead, the total system power consumption is less than a Raspberry PI 3 model. This module can be intergraded with Raspberry PI camera module V2. This camera can be used to take high-definition video, as well as stills photographs. Since Raspberry PI Zero is a minimal board, there are few I/O pins, which can be configured to control the main trap's sensors, lighting and actuators as shown in Figure 4.2 of chapter 4. The only limiting factor for this approach is the idle power consumption of the Raspberry PI module, which would have an idle power consumption around 2 Watts.

Raspberry PI is a Linux based system, so OpenCV libraries can be used to implement the developed algorithms. Then the system could be deployed to collect more training images and determine the optimum number of training images for best separation between the targeted animal species.

The main goal for this research project was to implement the developed algorithms in a low cost 32-bit microcontroller such as Microchip Atmel SAM series microcontrollers. The idle power consumption is around 25uA [155] with floating-point capabilities. This is an option that could be implemented as a commercial product.

Even though this research and development is to solve a specific New Zealand problem, the developed system could be used to detect other similar species. Such future projects could be to install a similar system to detect squirrels, where Defence infrastructures in remote areas are damaged by squirrels chewing power cables etc. So, this developed application may have international use with different animal species to solve similar pest control problems.

Future work could be to determine the optimum resolution from a set of low-resolution greyscale cameras, as lower resolution and a low number of training images help to reduce the operation time. The faster the operation time, the less stress experienced by the trapped animal and so it is very important factor to optimise this identification time.

The proposed system could be easily used to control pest populations in different geographical areas by smart trapping, or to study the wild animal population. Another factor of the system it is that it will remove the requirement of a specialised trained expert to detect animals either by their footprint pattern or fur samples.

A similar approach could be implemented in smart animal door systems such as cat doors and farm gates, to let animal species into certain areas. Since Radio Frequency IDentification (RFID) based animal entry methods are heavily patented [156, 157], the vision-based system will become more popular among product developers.

With further work and research this type of system could also be introduced to a vehicular vision system to detect small animal species and reduce the number of endangered species killed on roads. Currently there are numerous studies being carried out to detect larger animals with built-in vision systems; the algorithms developed by this research could be a useful addition to vehicular detection systems.

Appendix A
Test and Training Images

Attached CD contains all the images used for the trials. Under each folder there are two subfolders separating the test and training images. Inside those folders cats and possums are separated into two groups. Inside the cat and possum groups there are seven folders with seven different resolutions.

1. The original 170x170 images:
   \Images\170x170

2. The black back ground and white face images:
   \Images\BlackandWhite

3. The black back ground, grey face and white nose and mouse:
   \Images\BlkBG_GryFC_WhtNM

4. Grey background and original face:
   \Images\GryBG_OrgFC

5. Optimised black background and white face images by changing eyes and nose colour with different grey scales:
   \Images\Poti

6. Black background and white face images, with the size of the face is scaled to same size:
   \Images\Scaled

7. Images with white face grey background and original nose and mouth:
   \Images\WhtFC_GryBG_OrgNM

Appendix B

MATLAB Programs CD

The attached CD contains all the MATLAB scripts and Microsoft Excel files used for these trials.

The files are:

- Eigenface technique
- Eigenface with proposed distance formula
- Eigenface with proposed distance formula and error weights
- Error function for Eigenfaces
- Feature Points detection
- Fisherface technique
- Possum nose detection
- Standard deviation-based edge detection and Hough transformed based eye detection
- SVM + PCA identification
- SVM technique

For the Face Feature-based method (Face Feature Folder):

- PCA _FD
- FisherFaces_FD
- SVM_FD
- Feature_Selection
- PCA_Selected_FD
- FisherFaces_Selected_FD
- SVM_Selected_FD

References

[1] K. Wilkinson, "$4 m invested in self-resetting trap pilot project," ed. New Zealand: Office of the Minster of Conversation, 2010.

[2] Z. Weiwei, S. Jian, and T. Xiaoou, "From Tiger to Panda: Animal Head Detection," *Image Processing, IEEE Transactions on,* vol. 20, no. 6, pp. 1696-1708, 2011.

[3] A. R. T. Palma and R. Gurgel-Gonçalves, "Morphometric identification of small mammal footprints from ink tracking tunnels in the Brazilian Cerrado," *Revista Brasileira de Zoologia,* vol. 24, no. 2, pp. 333 - 343, 2007.

[4] Ramanan D., Forsyth D. A., and Barnard K., "Building models of animals from video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, no. 8, pp. 1319-1334, 2006.

[5] A. Gomez Villa, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks," *Ecological Informatics,* vol. 41, pp. 24-32, 2017.

[6] "A Pest of Plague Proportions," ed. Department of Conservation, New Zealand, 2004.

[7] B. Warburton, P. Cowan, and J. Shepherd, "How Many Possums Are Now in New Zealand Following Control and How Many Would There Be Without It?," ed. Landcare Research Ltd, Lincoln, New Zealand, 2009.

[8] P. E. Cowan, *The Handbook of New Zealand Mammals*, 2 ed. Melbourne: Oxford University Press, 2005.

[9] G. Nugent, "Possums... Their Effects on Native Vegetation," ed. Landcare Research Ltd, Lincoln, New Zealand, 2002.

[10] C. G. Rickard, "Introduced Small Mammals and Invertebrate Conservation in a Lowland Podocarp Froest, South Westland, New Zealand," Master of Forestry Science University of Canterbury Canterbury, 1996.

[11] P. D. Rachelle, "Possum control," ed. NZ Parliamentary Library, 2000.

[12] H. Gerard, "Possums - TB and possum control," ed. Te Ara - the Encyclopedia of New Zealand: TEARA, 2012.

[13] "Possum Control by the New Zealand Department of Conservation: Background, Issues, and Results from 1993 to 1995," ed. Department of Conservation, New Zealand, 1997.

[14] L. Booth, "Possums the Pros and Cons of Different Poisons," ed. Landcare Research, Lincoln, New Zeland 2010.

[15] "Pests and threats: 1080 poison for pest control," ed: Department of Conservation, New Zealand.

[16] B. Bob, "Introduced animal pests - Stoats and cats," ed. Te Ara - the Encyclopedia of New Zealand: TEARA, 2012.

[17] B. M. Fjtzgerald and B. J. Karl, "Foods of feral house cats (Felis Catus L.) in forest of the Orongorongo Valley, Wellington," *New Zealand Journal of Zoology,* vol. 6, pp. 1007-126, 1979.

[18] B. Warburton and N. Poutu, "Effectiveness of three trapping systems for killing feral cats," ed. Department of Conservation, New Zealand, 2002.

[19] S. Staats, H. Wallace, and T. Anderson, "Reasons for Companion Animal Guardianship (Pet Ownership) from Two Populations," *Society and Animals,* vol. 16, pp. 279-291, 2008.

[20] J. Levinthal, "The Community Context of animal and human Maltreatment: Is there a relationship between animal Maltreatment and human Maltreatment: Does Neighborhood Context Matter?," PhD Publicly accessible Penn Dissertations, University of Pennsylvania, 2010.

[21] M. J. Farnworth., J. Campbell, and N. J. Adams, "What's in a Name? Perceptions of Stray and Feral Cat Welfare and Control in Aotearoa, New Zealand," *Journal of Applied Animal Welfare Science,* vol. 14, no. 1, pp. 59-74, 2011.

[22] C. King, *Immigrant Killers: Introduced Predators and The Conservation Oof Birds In New Zealand*. Auckland, New Zealand: Oxford University Press, 1984.

[23] Y. van Heezik, A. Smyth, A. Adams, and J. Gordon, "Do domestic cats impose an unsustainable harvest on urban bird populations?," *Biological Conservation,* vol. 143, no. 1, pp. 121-130, 2010.

[24] M. J. Farnworth, N. G. Dye, and N. Keown, "The Legal Status of Cats in New Zealand: A Perspective on the Welfare of Companion, Stray, and Feral Domestic Cats (Felis catus)," *Journal of Applied Animal Welfare Science,* vol. 13, no. 2, pp. 180-188, 2010.

[25] M. Forbes and J. Weekes, "Wellington bylaw would cap cat numbers, ban roosters and limit pigeon feeding," in *Stuff Environment*, ed, 2016.

[26] L. Say and D. Pontier, "Spacing pattern in a social group of stray cats: effects on male reproductive success," *Animal Behaviour,* vol. 68, no. 1, pp. 175-180, 2004.

[27] J. W. S. Bradshaw, G. F. Horsfield, J. A. Allen, and I. H. Robinson, "Feral cats: their role in the population dynamics of Felis catus," *Applied Animal Behaviour Science,* vol. 65, pp. 273-283, 1999.

[28] M. C. Calver, J. Grayson, M. Lilith, and C. R. Dickman, "Applying the precautionary principle to the issue of impacts by pet cats on urban wildlife," *Biological Conservation,* vol. 144, pp. 1895-1901, 2011.

[29] M. R. Slater, A. Di Nardo, O. Pediconi, P. D. Villa, L. Candeloro, and B. Alessandrini, "Free-roaming dogs and cats in central Italy: public perceptions of the problem," *Preventive Veterinary Medicine,* vol. 84, pp. 27-47, 2008.

[30] D. W. Craig Gillies, "A short guide for identifying footprints on tracking tunnel papers," ed. Department of Conservation, New Zealand (unpublished report), 2002.

[31] G. A. Morriss, G. Nugent, and J. Whitford, "Dead birds found after aerial poisoning operations targeting small mammal pests in New Zealand 2003–14," *New Zealand Journal of Ecology,* vol. 40, no. 3, pp. 361-370, 2016.

[32] ONENews, "Self-resetting pest traps hope to save money," ed. New Zealand, 2010.

[33] "Animal Welfare Act 1999," ed. Ministry for Primary Industries, New Zealand, 1999.

[34] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," *ACM Comput. Surv.,* vol. 35, no. 4, pp. 399-458, 2003.

[35] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, no. 10, pp. 1042-1052, 1993.

[36] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, 2 ed. New York: Springer, 2011.

[37] N. Prakash and Y. Singh, "Support Vector Machines for Face Recognition," *International Research Journal of Engineering and Technology (IRJET)* vol. 2, no. 8, pp. 1517-1529, 2015.

[38] B. Moghaddam, T. Jebara, and A. Pentland, *Bayesian Modeling of Facial Similarity* (Advances in Neural Information Processing System). MIT Press, 1999.

[39] A. Dutta, R. Veldhuis, and L. Spreeuwers, "A Bayesian model for predicting face recognition performance using image quality," in *IEEE International Joint Conference on Biometrics*, 2014, pp. 1-8.

[40] A. Ghasemzadeh and H. Demirel, "3D discrete wavelet transform-based feature extraction for hyperspectral face recognition," *IET Biometrics,* vol. 7, no. 1, pp. 49-55, 2018.

[41] A. Ghasemzadeh and H. Demirel, "Hyperspectral face recognition using 3D discrete wavelet transform," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2016, pp. 1-4.

[42] A. V. Nefian and M. H. Hayes, "Maximum likelihood training of the embedded HMM for face detection and recognition," in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, 2000, vol. 1, pp. 33-36 vol.1.

[43] A. G. Joshi and A. S. Deshpande, "Review of Face Recognition Techniques," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 5, no. 1, pp. 71-75, 2015.

[44] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, no. 7, pp. 711-720, 1997.

[45] M. Smiatacz, "Eigenfaces, Fisherfaces, Laplacianfaces, Marginfaces – How to Face the Face Verification Task," in *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, vol. 226, R. Burduk, K. Jackowski, M. Kurzynski, M. Wozniak, and A. Zolnierek, Eds. (Advances in Intelligent Systems and Computing: Springer International Publishing, 2013, pp. 187-196.

[46] T. M. Abhishree, J. Latha, K. Manikantan, and S. Ramachandran, "Face Recognition Using Gabor Filter Based Feature Extraction with Anisotropic Diffusion as a Pre-processing Technique," *Procedia Computer Science,* vol. 45, pp. 312-321, 2015.

[47] B. S. Oh, K. A. Toh, A. B. J. Teoh, and Z. Lin, "An Analytic Gabor Feedforward Network for Single-Sample and Pose-Invariant Face Recognition," *IEEE Transactions on Image Processing,* vol. 27, no. 6, pp. 2791-2805, 2018.

[48] H. Xiaofei, Y. Shuicheng, H. Yuxiao, P. Niyogi, and Z. Hong-Jiang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 3, pp. 328-340, 2005.

[49] H. Zhang, Z. Qu, L. Yuan, and G. Li, "A face recognition method based on LBP feature for CNN," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017, pp. 544-547.

[50] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 2037-2041, 2006.

[51] K. Bong, S. Choi, C. Kim, D. Han, and H. J. Yoo, "A Low-Power Convolutional Neural Network Face Recognition Processor and a CIS Integrated With Always-on Face Detector," *IEEE Journal of Solid-State Circuits,* vol. 53, no. 1, pp. 115-123, 2018.

[52] K. Bong, S. Choi, C. Kim, and H. J. Yoo, "Low-Power Convolutional Neural Network Processor for a Face-Recognition System," *IEEE Micro,* vol. 37, no. 6, pp. 30-38, 2017.

[53] I. T. Jolliffe, *Principal Component Analysis* (Springer Series in Statistics). New York: Springer-Verlag, 2002, p. 488.

[54] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[55] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, 1991, pp. 586-591.

[56] S. Nedevschi, I. R. Peter, A. Mandru, and x, "PCA type algorithm applied in face recognition," in *Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference on*, 2012, pp. 167-171.

[57] W. S. L. Jebarani and T. Kamalaharidharini, "Robust face recognition and classification system based on SIFT and DCP techniques in image processing," in *2017 International Conference on Intelligent Computing and Control (I2C2)*, 2017, pp. 1-8.

[58] Q. Zhou, R. S. u, Y. Zhou, X. Wei, L. Wang, and B. Zheng, "Face Recognition Using Dense SIFT Feature Alignment," *Chinese Journal of Electronics,* vol. 25, no. 6, pp. 1034-1039, 2016.

[59] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725-1730.

[60] T. Burghardt and J. Calic, "Analysing animal behaviour in wildlife videos using face detection and tracking," *Vision, Image and Signal Processing, IEE Proceedings -,* vol. 153, no. 3, pp. 305-312, 2006.

[61] H. Nguyen *et al.*, "Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2017, pp. 40-49.

[62] T. Burghardt and J. Calic, "Real-time Face Detection and Tracking of Animals," in *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*, 2006, pp. 27-32.

[63] K. R. R. Swinnen, J. Reijniers, M. Breno, and H. Leirs, "A Novel Method to Reduce Time Investment When Processing Videos from Camera Trap Studies," *PLOS ONE,* vol. 9, no. 6, p. e98881, 2014.

[64] X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang, and T. Huang, "Automated identification of animal species in camera trap images," *EURASIP Journal on Image and Video Processing,* journal article vol. 2013, no. 1, p. 52, September 04 2013.

[65] K. Abdul Jabbar, M. F. Hansen, M. L. Smith, and L. N. Smith, "Early and non-intrusive lameness detection in dairy cows using 3-dimensional video," *Biosystems Engineering,* vol. 153, pp. 63-69, 2017/01/01/ 2017.

[66] A. Poursaberi, C. Bahr, A. Pluk, A. Van Nuffel, and D. Berckmans, "Real-time automatic lameness detection based on back posture extraction in dairy cattle: Shape analysis of cow with image processing techniques," *Computers and Electronics in Agriculture,* vol. 74, no. 1, pp. 110-119, 2010/10/01/ 2010.

[67] T. Van Hertem *et al.*, "Automatic lameness detection based on consecutive 3D-video recordings," *Biosystems Engineering,* vol. 119, pp. 108-116, 2014/03/01/ 2014.

[68] D. Forslund, J. Bj, x00E, and rkefur, "Night Vision Animal Detection," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 737-742.

[69] M. F. Hansen *et al.*, "Towards on-farm pig face recognition using convolutional neural networks," *Computers in Industry,* vol. 98, pp. 145-152, 2018.

[70] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna," *Scientific Data,* Data Descriptor vol. 2, p. 150, 2015.

[71] A. Gomez, G. Diez, A. Salazar, and A. Diaz, "Animal Identification in Low Quality Camera-Trap Images Using Very Deep Convolutional Neural Networks and Confidence Thresholds," in *Advances in Visual Computing*, Cham, 2016, pp. 747-756: Springer International Publishing.

[72] K. J. Dana, S. K. Nayar, B. van Ginneken, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 151-157.

[73] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *Int. J. Comput. Vision,* vol. 57, no. 2, pp. 137-154, 2004.

[74] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, vol. 1, pp. I-900-I-903 vol.1.

[75] A. Loos, M. Pfitzer, and L. Aporius, "Identification of great apes using face recognition," in *2011 19th European Signal Processing Conference*, 2011, pp. 922-926.

[76] L. Tonghai, T. Guanghui, and F. Weisen, "Research and development of pig weight estimation system based on image," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pp. 2774-2777.

[77] Z. Liang and L. Jian-hui, "Obtain the depth of field based on two-view," presented at the Proceedings of the 2nd international Asia conference on Informatics in control, automation and robotics - Volume 1, Wuhan, China, 2010.

[78] Y. Wang, W. Yang, P. Winter, and L. T. Walker, "Non-Contact Sensing of Hog Weights by Machine Vision," *Applied Engineering In Agriculture* vol. 22, no. 4, pp. 577-582, 2006.

[79] P. Viola and M. Jones, "Robust real-time face detection," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2001, vol. 2, pp. 747-747.

[80] MathWorks, *Introducing Deep Learning with MATLAB*. Humusoft, 2017, p. 15.

[81] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature,* Article vol. 550, p. 354, 2017.

[82] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 858-862.

[83] T. Tibor, K. Patrik, O. Richard, B. Miroslav, and S. Peter, "Animal Recognition System Based on Convolutional Neural Network," *Digital Image Processing and Computer Graphics,* vol. 15, no. 3, pp. 517-525, 2017.

[84] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3 ed. Upper Saddle River NJ.: Prentice Hall, 2008.

[85] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*, 2 ed. United States: Gatemark Publishing, 2009.

[86] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999, p. 680.

[87] Y. Jiang, I. Hayashi, and S. Wang, "Knowledge Acquisition Method Based on Singular Value Decomposition for Human Motion Analysis," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 12, pp. 3038-3050, 2014.

[88] T. A. S. A. Perera and J. Collins, "A Novel Eigenface based Species Recognition System," *International Journal of Computer Applications,* vol. 115, no. 20, pp. 19-23, April 2015.

[89] L. Sirovich and M. Kirby, *Low-Dimensional Procedure for the Characterization of Human Faces*. 1987, pp. 519-24.

[90] S. Yi, Z. Lai, Z. He, Y. Cheung, and Y. Liu, "Joint sparse principal component analysis," *Pattern Recognition,* vol. 61, pp. 524-536, 2017.

[91] N. Kwak, "Principal Component Analysis Based on L1-Norm Maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, no. 9, pp. 1672-1680, 2008.

[92] D. Zhang and Z.-H. Zhou, "(2D)2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing,* vol. 69, no. 1, pp. 224-231, 2005.

[93] R. He, B. G. Hu, W. S. Zheng, and X. W. Kong, "Robust Principal Component Analysis Based on Maximum Correntropy Criterion," *IEEE Transactions on Image Processing,* vol. 20, no. 6, pp. 1485-1494, 2011.

[94] Q. Wang, Q. Gao, X. Gao, and F. Nie, "ℓ2,p -Norm Based PCA for Image Recognition," *IEEE Transactions on Image Processing,* vol. 27, no. 3, pp. 1336-1346, 2018.

[95] X. Shi, Z. Guo, F. Nie, L. Yang, J. You, and D. Tao, "Two-Dimensional Whitening Reconstruction for Enhancing Robustness of Principal Component Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 38, no. 10, pp. 2130-2136, 2016.

[96] A. K. Seghouane and N. Shokouhi, "Two-Dimensional Whitening of Face Images for Improved PCA Performance," *IEEE Signal Processing Letters,* vol. 25, no. 4, pp. 521-525, 2018.

[97] C. Gao, J. Zhou, and Q. Pu, *Theory of fractional covariance matrix and its applications in PCA and 2D-PCA*. 2013, pp. 5395–5401.

[98] T. B. A. de-Carvalho, M. A. A. Sibaldo, I. R. Tsang, G. D. C. Cavalcanti, I. J. Tsang, and J. Sijbers, "Fractional Eigenfaces," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 258-262.

[99] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[100] E. B. Putranto, P. A. Situmorang, and A. S. Girsang, "Face recognition using eigenface with naive Bayes," in *2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, 2016, pp. 1-4.

[101] Z. Jiyin, Z. Ruirui, Z. Lulu, and D. Kun, "Combined weighted eigenface and BP-based networks for face recognition," in *2008 5th International Conference on Visual Information Engineering (VIE 2008)*, 2008, pp. 298-302.

[102] Z. Lai, W. K. Wong, Z. Jin, J. Yang, and Y. Xu, "Sparse Approximation to the Eigensubspace for Discrimination," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 23, no. 12, pp. 1948-1960, 2012.

[103] L. Chengjun and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Transactions on Image Processing,* vol. 11, no. 4, pp. 467-476, 2002.

[104] Z. Wei-Shi, L. Jian-Huang, and P. C. Yuen, "GA-fisher: a new LDA-based face recognition algorithm with selection of principal components," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 35, no. 5, pp. 1065-1078, 2005.

[105] Z. Wangmeng, D. Zhang, Y. Jian, and W. Kuanquan, "BDPCA plus LDA: a novel fast feature extraction technique for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 36, no. 4, pp. 946-953, 2006.

[106] Y. C. Feng, P. C. Yuen, and A. K. Jain, "A Hybrid Approach for Generating Secure and Discriminating Face Template," *IEEE Transactions on Information Forensics and Security,* vol. 5, no. 1, pp. 103-117, 2010.

[107] C. Dinakardas, S. P. Sankar, and N. George, "A multimodal performance evaluation on two different models based on face, fingerprint and iris templates," in *Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 2013 International Conference on*, 2013, pp. 1-6.

[108] D. Salvatore. (February 2016) Facial Recognition: Who's Tracking You in Public? *Consumer Reports Magazine*

[109] R. A. Fisher, "The Use of Multiple Measures in Taxonomic Problems.," *Annals of Eugenics,* vol. 7, pp. 179-188, 1963.

[110] Y. Ming-Hsuan, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face recognition using kernel methods," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, 2002, pp. 215-220.

[111] L. Kuang-Chih, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 5, pp. 684-698, 2005.

[112] G. Bergqvist and E. G. Larsson, "The Higher-Order Singular Value Decomposition: Theory and an Application [Lecture Notes]," *IEEE Signal Processing Magazine,* vol. 27, no. 3, pp. 151-154, 2010.

[113] H. M. Hussain, K. Benkrid, and H. Seker, "An adaptive implementation of a dynamically reconfigurable K-nearest neighbour classifier on FPGA," in *Adaptive Hardware and Systems (AHS), 2012 NASA/ESA Conference on*, 2012, pp. 205-212.

[114] L. I. Kuncheva and M. Galar, "Theoretical and Empirical Criteria for the Edited Nearest Neighbour Classifier," in *Data Mining (ICDM), 2015 IEEE International Conference on*, 2015, pp. 817-822.

[115] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks,* vol. 10, no. 5, pp. 988-999, 1999.

[116] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik, "Boosting and Other Ensemble Methods," *Neural Computation,* vol. 6, no. 6, pp. 1289-1301, 1994.

[117] "Introduction to Support Vector Machines," ed: OpenCV, 2014.

[118] J. A. Swets, "Indices of discrimination or diagnostic accuracy: their ROCs and implied models," *Psychological Bulletin,* vol. 99, no. 1, p. 7, 1986.

[119] E. A. Krupinski, "Receiver Operating Characteristic (ROC) Analysis," *Frontline Learning Research - Special issue "Visual Expertise",* vol. 5, no. 3, pp. 31-42, 14/07/2017 2017.

[120] K. Hajian-Tilaki, "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation," *Caspian Journal of Internal Medicine,* vol. 4, no. 2, pp. 627-635, 2013.

[121] T. Fawcett, *ROC Graphs: Notes and Practical Considerations for Researchers*. Netherlands: Kluwer Academic Publishers, 2004.

[122] S. Ting, B. C. Lovell, and C. Shaokang, "Face Recognition Robust to Head Pose from One Sample Image," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006, vol. 1, pp. 515-518.

[123] P. de Souza, "Edge detection using sliding statistical tests," *Computer Vision, Graphics, and Image Processing,* vol. 23, no. 1, pp. 1-14, 1983.

[124] F. A. Jassim, "Semi-Optimal Edge Detector based on Simple Standard Deviation with Adjusted Thresholding," *International Journal of Computer Applications,* vol. 68, pp. 43-48, 2013.

[125] R. Bhattacharjee and M. Chakraborty, "Brain tumor detection from MR images: Image processing, slicing and PCA based reconstruction," in *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*, 2012, pp. 97-101.

[126] P. Kamencay, M. Zachariasova, R. Hudec, M. Benco, and R. Radil, "3D image reconstruction from 2D CT slices," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2014, pp. 1-4.

[127] C. Xing, L. Ling, and G. Yang, "A new concentric circle detection method based on Hough transform," in *Computer Science & Education (ICCSE), 2012 7th International Conference on*, 2012, pp. 753-758.

[128] T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," *Image and Vision Computing,* vol. 17, no. 11, pp. 795-803, 1999.

[129] H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of Hough transform methods for circle finding," *Image and Vision Computing,* vol. 8, no. 11, pp. 71-77, 1990.

[130] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, 3rd ed. Morgan Kauffman Publishers, 2005.

[131] C. Kotropoulos and I. Pitas, "Rule-based face detection in frontal views," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, vol. 4, pp. 2537-2540 vol.4.

[132] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. 511-518

[133] MathWorks, "vision.CascadeObjectDetector System object," in *Detect objects using the Viola-Jones algorithm*, ed, 2015.

[134] K. Toennies, F. Behrens, and M. Aurnhammer, "Feasibility of Hough-transform-based iris localisation for real-time-application," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 2002, pp. 1053-1056 vol.2.

[135] H. Iwasa, W. Ohyama, T. Wakabayashi, and F. Kimura, "Facial features extraction by accelerated implementation of circular hough transform and appearance evaluation," in *Frontiers of Computer Vision (FCV), 2015 21st Korea-Japan Joint Workshop on*, 2015, pp. 1-6.

[136] Y. Ito, W. Ohyama, T. Wakabayashi, and F. Kimura, "Detection of eyes by circular Hough transform and histogram of gradient," in *Pattern Recognition, 2012 21st International Conference on*, 2012, pp. 1795-1798.

[137] T. A. S. A. Perera and J. Collins, "Imaged Based Species Recognition System," in *2015 9th International Conference on Sensing Technology (ICST)*, Auckland 2015, pp. 195-199: IEE Instrumentation & Measurement Society

[138] A. Genz and F. Bretz, "Comparison of Methods for the Computation of Multivariate *t* Probabilities," *Journal of Computational and Graphical Statistics,* vol. 11, no. 4, pp. 950-971, 2002.

[139] A. Genz and F. Bretz, *Computation of Multivariate Normal and t Probabilities*, 1 ed. (Lecture Notes in Statistics, no. 195). Springer-Verlag Berlin Heidelberg, 2009, pp. VIII, 126.

[140] J. Wang, G. Yuantao, K. N. Plataniotis, and A. N. Venetsanopoulos, "Select eigenfaces for face recognition with one training sample per subject," in *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, 2004, vol. 1, pp. 391-396 Vol. 1.

[141] P. J. Groenen and I. Borg, "Past, present, and future of multidimensional scaling," *Visualization and verbalization of data,* pp. 95-117, 2014.

[142] J. Scott, M. A. Pusateri, and D. Cornish, "Kalman filter based video background estimation," in *Applied Imagery Pattern Recognition Workshop (AIPRW), 2009 IEEE*, 2009, pp. 1-7.

[143] S. Kumar and J. S. Yadav, "Background subtraction method for object detection and tracking," in *Advances in Intelligent Systems and Computing*, 2017, vol. 479, pp. 1057-1063.

[144] T. Valentine, M. B. Lewis, and P. J. Hills, "Face-Space: A Unifying Concept in Face Recognition Research," *Quarterly Journal of Experimental Psychology,* vol. 69, no. 10, pp. 1996-2019, 2016.

[145] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th ed. New York: Dover, 1972.

[146] M. Brown, "A Generalized Error Function In n Dimension," 1 ed. Point Mugu, California: U. S. Naval Missile Center, 1963, pp. 1-14.

[147] B. W. Yohanes, R. D. Airlangga, and I. Setyawan, "Real Time Face Recognition Comparison Using Fisherfaces and Local Binary Pattern," in *2018 4th International Conference on Science and Technology (ICST)*, 2018, pp. 1-5.

[148] N. Hegde, S. Preetha, and S. Bhagwat, "Facial Expression Classifier Using Better Technique: FisherFace Algorithm," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 604-610.

[149] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-233, 2001.

[150] N. Christianini and J. C. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge UK: Cambridge University Press, 2000.

[151] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 12 ed. New York: Springer, 2017.

[152] H. Zhang, W. Liu, L. Dong, and Y. Wang, "Sparse Eigenfaces Analysis For Recognition," in *2014 12th International Conference on Signal Processing (ICSP)*, 2014, pp. 887-890.

[153] N. Novoselova, C. Della Beffa, J. Wang, J. Li, F. Pessler, and F. Klawonn, *HUM Calculator and HUM package for R: easy-to-use software tools for multicategory receiver operating characteristic analysis*. 2014.

[154] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[155] *AT11491: Peripheral Power Consumption in Standby Mode for SAM D Devices*, 2015.

[156] N. P. R. Hill, "RFID Pet Door," United States of America Patent US8240085B2, 2006.

[157] A. D. Bank and N. P. R. Hill, "Pet Door with RFID Detection," European Union Patent EP2756146B1, 2011.