

# Image based Cats and Possums Identification for Intelligent Trapping Systems

T. A. S. Achala Perera  
School of Engineering  
Auckland University of Technology  
New Zealand

John Collins  
School of Engineering  
Auckland University of Technology  
New Zealand

## ABSTRACT

Eigenface based recognition is widely used to identify individual faces. This technique is commonly used in human recognition domain. In this research, we are investigating for implementation of a possible solution; which can be used to identify species. The eigenface technique is a novel approach in animal recognition domain [1, 2]. In this application, cats and possums are targeted as species of interest. The main reason for targeting cats and possum is, these introduced species are classified as threat to New Zealand natural environment and native wildlife. The next phase of this research project is to implement a vision based trapping system. This paper explains a novel classifier to use with eigenface techniques to separate different species.

## Keywords

Euclidean distance, face recognition, animal detection

## 1. INTRODUCTION

It was found, if the animal images were used without any padding with the eigenface technique, the recognition rate is extremely poor 55% for possums and 33% for cats. This is mostly due to the face orientation and background information. These paper discusses, about improvement of current technique by background removal, species grouping, background colour optimization and image scaling.

The one of the main contribution in this paper is, development of a new method for separation of eigenface outputs (Distance from mean). This novel technique offers a new tool, to transform multi dimensionally scattered data clusters into two dimensional graph. Which allow users to see the separation of different class clusters, and define the decision boundary. Finally, we explain the introduction of error weight vector and extra eigenvalue to improve the separation of two class problem.

## 2. EIGENFACE TECHNIQUE

The eigenface technique is normally 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. Firstly, calculates the mean of the training images, and then mean is subtracted from the training set to obtain a mean-shifted training set. This is known as normalizing the training set. From the mean-shifted training set, the scrambled covariance matrix is calculated. Then the eigenvectors and eigenvalues are obtained from scrambled covariance matrix. The eigenvectors with largest eigenvalues are kept for detection. These are known as the principle components. These principle components keep most of the facial features. Finally, the eigenface technique projects the mean-shifted images into the eigenspace, using the principal eigenvectors [1, 3-5].

The eigenface algorithm can be explained in few steps [1, 3, 5, 6].

Step 1: Obtain an animal face training set  $I_1, I_2, \dots, I_M$

Step 2: Convert each image  $I_i$  into a vector  $\Gamma_i$  (Convert the  $N \times N$  image into an  $N^2 \times 1$  vector)

Step 3: Calculate the average animal face vector  $\Psi$ :

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (1)$$

Step 4: Subtract the average face from  $\Gamma_i$  to get  $\Phi_i$ :

$$\Phi_i = \Gamma_i - \Psi \quad (2)$$

Step 5: Calculate the covariance matrix  $C$ :

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (N^2 \times N^2) \quad (3)$$

Where  $A = [\Phi_1 \Phi_2 \dots \Phi_M] \quad (N^2 \times M \text{ Matrix}) \quad (4)$

Step 6: Calculate the eigenvectors  $u_i$  of  $AA^T$  [5]

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

Once best eigenvector from training set is computed, unknown images can be feed through the system for face detection. Before the detection process, the input image need to be normalized. This processed can be split into four steps for a given unknown image  $\Gamma$  [1, 3, 4].

Step 1: Calculate  $\Phi = \Gamma - \Psi$

Step 2: Calculate  $\hat{\Phi} = \sum_{i=1}^K w_i u_i \quad (w_i = u_i^T \Phi)$

Step 3: Calculate Euclidean distance  $e_d = \|\Phi - \hat{\Phi}\|$  [5]

Step 4: If  $e_d < T_d$  then  $\Gamma$  is a face

In typical application, Euclidean distance is calculated. Then this distance is compared against a known threshold value. If the Euclidean value is less than the threshold, the input unknown image is one of the training set images. Otherwise it is not

## 3. EIGENFACE TECHNIQUE WITH DEVELOPED DISTANCE CLASSIFIER

The eigenface technique with developed distance algorithm can be described as follow. The size of the training image  $In$

is  $m \times 1$  matrix. In a training set (A) there are (N) number of images (refer to equation (5)). Then average face  $\mu$  is calculated refer to equation (6). Then normalize the training set by subtracting the average face as shown on equation (7). Then scramble covariance matrix B is calculated from normalized images (refer to equation (8))

$$A = [I_1, I_2, I_3, \dots, I_N] \quad (5)$$

$$\mu = \frac{\sum_n I_n}{N} \quad (6)$$

$$B = A - \mu \quad (M \times N \text{ matrix}) \quad (7)$$

$$\text{Cov } B = \frac{B^T B}{N - 1} \quad (8)$$

( $M \times N$  matrix) Use  $B^T B$  instead of  $BB^T$

Then eigenvalue  $\lambda_n$  and eigenvectors  $e_n$  are calculated.

Where:

$$B^T B e_n = \lambda_n e_n \quad (9)$$

Then:

$$BB^T (B e_n) = B \lambda_n e_n = \lambda_n (B e_n) \quad (10)$$

Therefore, normalized eigenfaces equal:

$$f_n = (B e_n) / |(B e_n)|$$

is eigenvectors of  $BB^T$  and  $\lambda_n$  is eigenvalues

Hence for full covariance matrix B:

Eigenvectors are  $f_n$

Eigenvalues are  $\frac{\lambda_n}{N-1} = \sigma_n^2$  where  $\sigma^2$  is variance

Then for unknown image J, weights and distance can be calculated by using equation (13).

$$w_n = (J - \mu) \times f_n \quad (12)$$

$$\text{distance} = \sqrt{\sum_n \frac{w_n^2}{\sigma_n^2}} = \sqrt{\sum_n \frac{w_n^2}{\lambda_n} (N - 1)} \quad (13)$$

The developed distance formula (13) measures the distance to each image from the mean. This algorithm compensates for the effect of the larger weights by dividing them by its eigenvalues (effectively the variance). Significant eigenvectors have the larger eigenvalues. Significant eigenvectors also produce significant weights. For an example if we have 5 images as a data base the covariance matrix size is 5x5. Once it convert to eigenvector and value, it can be observed significant eigenvectors have larger eigenvalues. Similarly, main eigenvectors produce larger weights. The first vector of the eigenvector matrix is not meaningful due the normalization of the training images. This also illustrates in eigenvalue matrix; the first eigenvalue's magnitude is insignificant compare to other eigenvalues.

Previous investigators found that, three eigenvectors with largest eigenvalue mostly contain the information about background lighting [6]. According to [6] research project, by removing three eigenvectors with largest eigenvalues, improves the accuracy of the classifier. But this developed distance algorithm (13), reduces the effect of the significant eigenvectors, by dividing the significant weights with its eigenvalue.

#### 4. IMPROVING DETECTION RATE BY OPTIMISING THE IMAGE COLOUR SCHEME

One of the developed technique is optimizing the image color scheme. During the initial investigations few color schemas were trials. The initial color scheme was; the image background color as grey (intensity value of 131). The internal face color was white (intensity value 255) and preserves the nose and eyes information as original colours (refer to Figure 1). The main issue of this scheme was, the area of nose and eyes was smaller percentage compare to the rest of the area of the image. The different between grey and white is significant enough to produce strong eigenvectors to extract the face outline information. Therefore, the separation of the species classes is not significant enough.

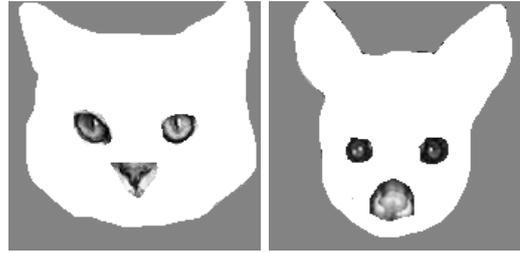


Figure 1: First trialed color scheme (11)

The second scheme was, background colour as black (intensity value of 0), face color as grey (intensity value of 160) and eyes and nose color as white (intensity value of 255) (refer to Figure 2 (a)). The main idea behind this scheme was to produce a large intensity change between background and face. Similarly eyes and nose been white (255) they have larger representation of the image and another significant intensity step change between face and features. This schemes produces much better separation between facial outline and background, compare to first scheme. Even though eyes and nose have much higher intensity value compare to face still a small area.

Since eyes and nose are smaller regions, it was decided to analyze the species separation without eyes and nose. In the third scheme trial image was simplified further by changing the face color to white (intensity value of 255) and background color to black (intensity value of 0) (refer to Figure 2(b)). It was found this configuration produces best result due to maximum step change between background and face shape. In this scheme, the recognition was targeting on face outline. The outline had the main intensity step change.

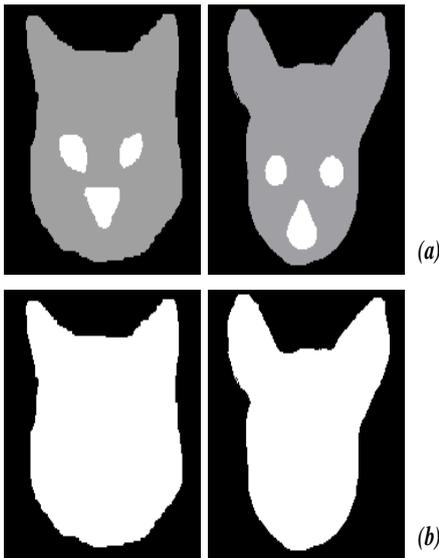


Figure 2: (a) Second colour scheme (b)Third trailed colour scheme

## 5. IMPROVING DETECTION RATE BY OPTIMISING IMAGE RESOLUTION

The image size is one of the important parameters, which can be used to optimized the separation of the species classes. Image size directly proportional to the size of the full covariance matrix. Therefore, Eigen face technique uses scrambled covariance matrix to reduce the number of data point.

The main issue of taking the scrambled covariance matrix ( $B^T B$  instead of  $BB^T$ ), large amount of the useful data will be lost. For an example, if there is a training set of 20 images and each image is 170 x 170 pixels. The actual size of the covariance matrix  $BB^T$  will be a matrix of 28900 x 28900. Since this matrix is computationally demanding, therefore scrambled covariance matrix  $B^T B$  is used. Scrambled covariance matrix is 20 x 20. Even though the most useful eigenvectors are kept, the large portion of data is lost.

In order to improve this situation, one of the technique we developed was reduce the image resolution. Since we are trying to detect animal by facial shape, fur details and nose and eyes information are not vital. Therefore, image size can be reduced to minimize the data lost due to the scrambled covariance matrix. For an example if we consider same training set of 20 images, but this time all the images are 10 x 10 resolution. The full covariance matrix  $BB^T$  will 100 x 100. If scrambled covariance matrix is use, it size will be still the same size 20 x 20. But the overall data lost lot more data than, first example.

In order to find the optimal image resolution, three different image resolution were investigated. The image resolutions were 34 x 34, 17 x 17 and 10 x 10.

To reduce the resolution, image was divide into different size blocks. For an example to obtain 17 x 17 image, original image (resolution 170 x170) was divide into 17 x 17 blocks. Each block is 10 x 10 pixels. Then take the sum of all the pixel in each block. Each sum was divided by its pixel count and replace the whole block with the calculated value (block average). Using this technique all the pixels' information is still kept (refer to Figure 3).



Figure 3: Image resolution from left to right 170 x 170, 34 x 34, 17 x 17 and 10 x 10

After this process, all the images have black background but facial edge contour colour was changed to different shade of grey values (refer to Figure 4). It was found 17 x17 is the optimum resolution, with best separation between cat and possum classes.

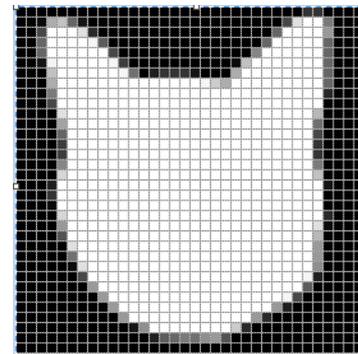


Figure 4: 34 x34 image with grey facial contour

## 6. SPECIES CLASS SEPARATION WITH DEVELOPED DISTANCE CLASSIFIER

Using equation (13) distance between mean image and given image can be calculated. In possums and cat detection problem, there will be two different distances for any given image. In order to develop a classifier to discover which cluster is close to the given image. Or in another word which side of the decision boundary the unknown image lies.

Trivial method of implementing above is, by comparison of these two distances. Shortest distance to the class mean could be selected as the likely species for a given image. Since this is multidimensional problem, it is hard to define the decision boundary between these two animal classes in multidimensional space. Therefore, it is important to implement a novel technique which convert distance data from multidimensional space to two dimensional (2D) space. In 2D space decision boundary can be defined to classify a given unknown image.

In order to transform multidimensional space data point into 2D space, trigonometrical space is used. Since there are two distances (cat distance and possum distance) in multidimensional space, these two distances can be plotted in trigonometrical space. It can be assuming in trigonometrical space, cat mean image and possum mean image is separated by a certain fixed distance. This distance can be set as a length between shorts distances vectors (cat distance and possum distances). Then the unknown image distances can be plotted from fixed mean points. The angel of both

distance vectors can be calculated according to the point, where those two vectors meet in trigonometrical space (refer to Figure 5 (a)).

In order to plot these point, there are three scenarios, which need to be taken into consideration. First case distance from cat training set and possum training set has similar weight vectors. So all the points will scatter between cat and possum mean values (refer to Figure 5 (b)). In case two, one of the distance get considerably larger than other. Therefore, the point will pass one of the mean values (refer to Figure 5 (c)). In third case one distance is extremely larger than the other, hence those two magnitudes will not be able to yield a point. The first two cases can be calculated using the developed algorithm below.

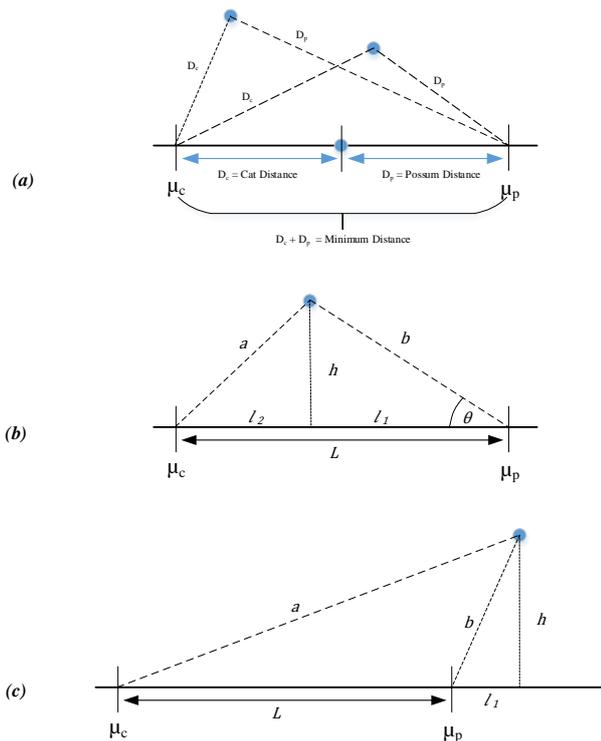


Figure 5: (a) Image distances on trigonometrical space (a) Case one (b) Case two

Case 1:

From law of cosine  $a^2$  can be expressed as [7]:

$$a^2 = b^2 + L^2 - 2bL \cos(\theta) \quad (14)$$

$$\cos(\theta) = (b^2 + L^2 - a^2)/2bL \quad (15)$$

$$L = l_1 + l_2 \quad (16)$$

$$l_1 = b \cos(\theta) \quad (17)$$

$$l_1 = \frac{b^2 + L^2 - a^2}{2L} \text{ and } l_2 = \frac{a^2 + L^2 - b^2}{2L} \quad (18)$$

$$h = \sqrt{a^2 - l_2^2} = \sqrt{b^2 - l_1^2} \quad (19)$$

Case 2:

From Pythagorean theorem  $a^2$  can be expressed as [8]:

$$a^2 = h^2 + (L + l)^2 \quad (20)$$

$$b^2 = h^2 + l^2 \quad (21)$$

$$a^2 = L^2 + 2Ll + l^2 + h^2 \quad (22)$$

$$a^2 = L^2 + 2Ll + b^2 \quad (23)$$

$$l = (a^2 - L^2 - b^2)/2L \quad (24)$$

$$h = \sqrt{b^2 - l^2} \quad (25)$$

Equations (18), (19), (24) and (25) can be used calculate the point on the graph. If any point belongs to the case three, then it will not be able to plot on the graph. Figure 9 shows the example classes separation of developed classifier. Once classes are separated as shown on Figure 9, the decision boundary can be defined.

## 7. IMPROVING DETECTION RATE BY INTRODUCING ERROR WEIGHT VECTOR

In this novel technique, new weight vector is introduced to compensate for omitted information by selecting scrambled covariance matrix  $B^T B$ .

Firstly, cats and possums are separated into two different training sets. Then eigenface technique is applied to both training sets. This time normalized eigenfaces are ( $f_n = \frac{B e_n}{|B e_n|}$ ) used. At the end of the process, we ended up with two different training weights one for possums and cats ( $\hat{w}^C$  and  $\hat{w}^P$ ). Once this process is completed additional weight vector can be calculated as shown below.

Where:

$$\text{eigenfaces} = f_n$$

$$\text{image} = i_m$$

For each image  $w_n$  weight vector can be calculated:

$$w_n = i_m \times f_n \quad (26)$$

Ideally:

$$i_m = \sum_k w_k f_k \quad (27)$$

$$i_m f_n = \sum_k w_k f_k f_n \quad f_k f_n = 1 \text{ if } \quad (28)$$

$$k = n \text{ and } f_k f_n = 0 \text{ if } k \neq n$$

$$i_m f_n = w_n \quad (29)$$

In practice  $e_m$  error vector can be calculated as below

$$\text{error } e_m = i_m - \sum_k w_k f_k$$

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

The above algorithm basically, calculate the weight vectors for the training images using modified eigenface technique. After that for a given image  $i_m$  weights can be calculated using the eigenface. Then these calculated new weight are used to reconstruct the original image  $i_m$ . Finally, reconstructed image is subtracted from actual image to find the difference between these two images. The resulting difference is used as an extra error weight vector  $|e_m|$ . This can be included into the existing weights.

Once error weight is calculated from the image,

corresponding magnitude (eigenvalue) is required for the vector. In order to find the next eigenvalue, existing eigenvalue can be used. If we plot the eigenvalue, it has an exponential decay (refer to Figure 6) [9]. Then exponential decay formula (31) can be used to find the next eigenvalue.

$$N(t) = N_0 e^{-\lambda t}$$

Where  $t$  and  $\lambda$  are known as the decay constant, and  $e^x$  is the exponential function and  $N_0 = N$  zero is the initial value [10]. The decay constant can be expressed as equation (32). The proof can be found in [10, 11].

$$\ln\left(\frac{N}{N_0}\right) = -\lambda t \quad (32)$$

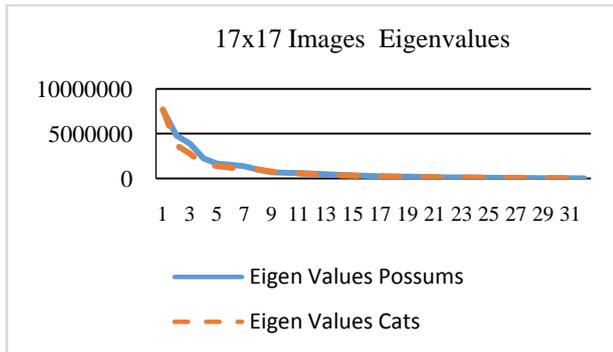


Figure 6: Graph of eigenvalue Vs magnitude

Then above algorithms (equations (31) and (32)) can be used to calculate the extra error weights and eigenvalues for possums and cats ( $\hat{w}^C$  and  $\hat{w}^P$ ) classes. By adding the error weight and estimated eigenvalue, the classes separation can be increased and data points within the class clustered together tighter.

### 8. CLASSIFIER EVALUATION

The main aim of this section is to evaluate the developed new distance algorithm (refer to equation (13)) with existing eigenface distance algorithm  $d = \sqrt{\sum_n w_n^2}$ . In this study, four different combinations were analysed. Firstly, existing eigenface technique (33), secondly improved eigenface technique (34), thirdly existing eigenfaces with developed distance algorithm (35), finally improved eigenfaces with existing distance formula (36).

Option one: Standard	$f_n = Be_n$	$d = \sqrt{\sum_n w_n^2}$	(33)
Option two: Proposed	$f_n = \frac{Be_n}{ Be_n }$	$d = \sqrt{\sum_n \frac{w_n^2}{\lambda_n} (N-1)}$	(34)
Option three:	$f_n = Be_n$	$d = \sqrt{\sum_n \frac{w_n^2}{\lambda_n} (N-1)}$	(35)
Option four:	$f_n = \frac{Be_n}{ Be_n }$	$d = \sqrt{\sum_n w_n^2}$	(36)

Where  $f_n$  is eigenfaces,  $B$  is mean subtracted (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 in each training set were selected (refer to Figure 7 (a)). Then 40 unknown images were used, to tests the system.

Inside the unknown images, first 22 images were possums and remaining 18 were cats (refer to Figure 7 (b)). These training sets and test images were scaled as discussed in section 5. All the training images and test images are 17x17 resolution. In this round of experiments error weights (refer to equation (30)) are not used.

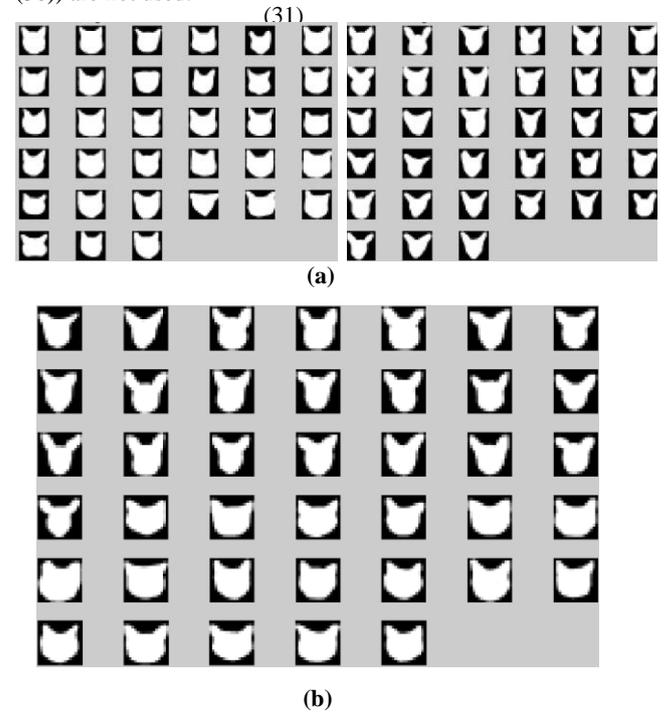


Figure 7: (a) Training images (b) Unknown test images

To plot the data developed plotting technique was used. This developed classifier helps to analyse the separation between species classes. If the classes are closer together, it is hard to define a decision boundary. The four figures below indicate the performance of all four options (refer to Figure 8, Figure 9, Figure 10 and Figure 11).

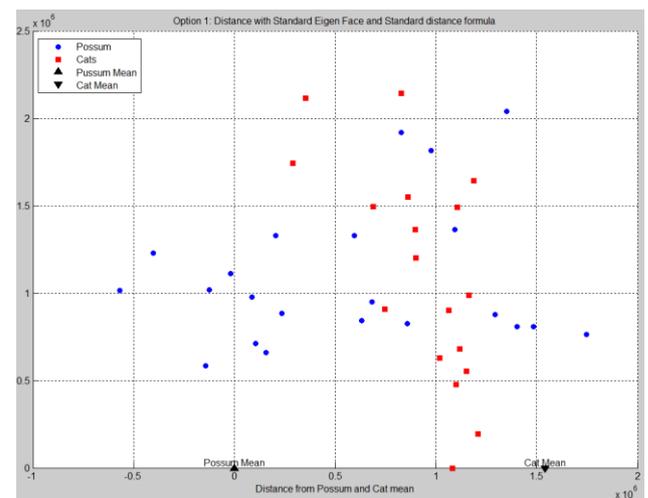


Figure 8: Graph for option one

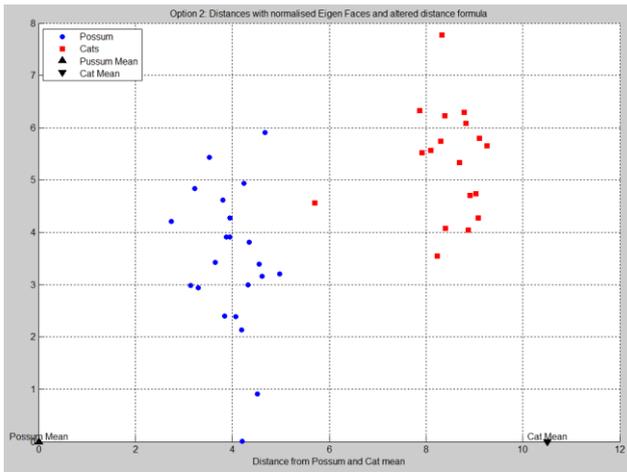


Figure 9: Graph for option two

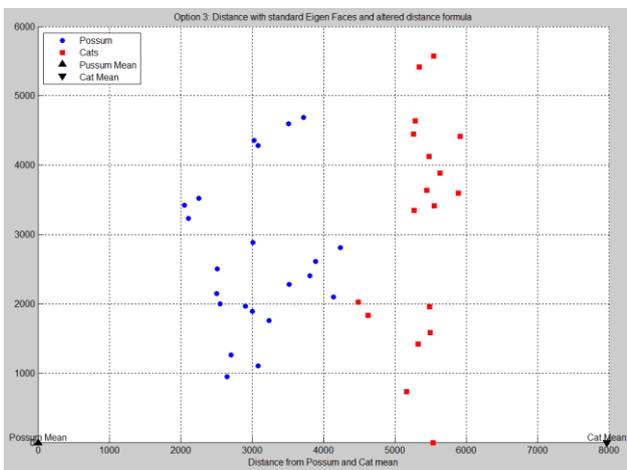


Figure 10: Graph for option three

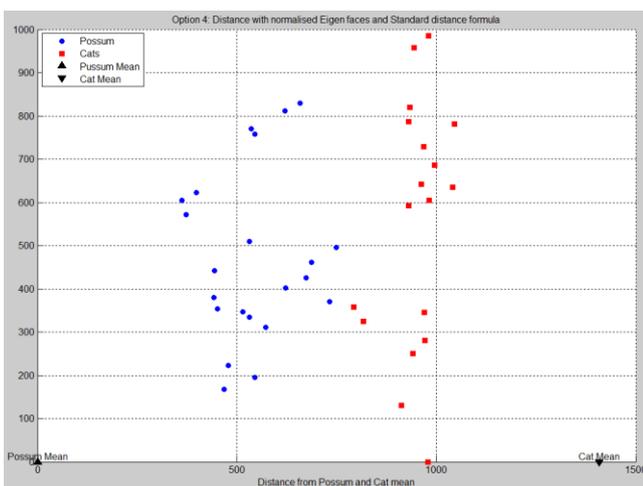


Figure 11: Graph for option four

If we analyse these four graphs; it can be clearly seen the best separation is achieved by improved eigenface technique (refer to equation (34)). The second best options are option 3 and 4. Even though, these two techniques have different scales on X and Y axis, both graph have exactly the same separation. So it can be concluded the effect on the weights, is similar, either for normalised eigenface or dividing the normalized weights by their eigenvalue. When we combine these two factors

together, we can archive the maximum separation between classes. Standard eigenface technique has the worst performance out of all four options. It shows ill separation between two animal species. Some what this is expectable result due to vectors with arbitrary magnitudes in face-space (low-dimensional space). Perhaps this could be one of the reasons; the original eigenface technique used threshold based detection technique.

The introduced error weight vector and eigenvalue, helps to tight up the species clusters closer together. Also improves the separation of the two clusters. The Figure 12 demonstrates the improved separation between two classes. Figure 12 and Figure 9 were plotted with same training and test data set. The difference is Figure 12 was plotted with introduced error vectors but Figure 9 was not.

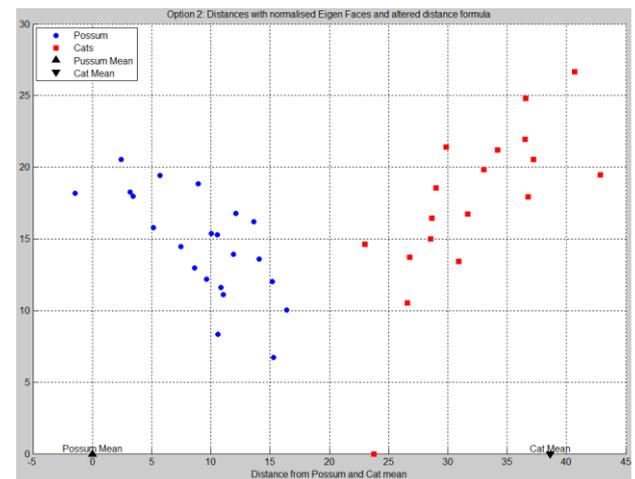


Figure 12: Graph for option two with introduced error weights and eigenvalue

## 9. CONCLUSION AND FUTURE WORKS

This developed novel techniques, improve the detection rate of the typical eigenface technique. The optimized algorithm can be used to solve any two class problem. In this application it was used to detect cats and possums.

The developed data visualising technique transform multidimensional data, into two dimensional plot. Which allows user to identify the targeted clusters and define the decision boundary around the clusters. The data cluster separation is optimized by altering the image color schema and scale. By introducing error weights, made the data clusters tighter and the same time increase the separation.

The next phase of the project is to implement the developed algorithm in Raspberry PI 2 embedded module. The main advantage of the eigenface technique is, it allows most of the processor intensive portion of the algorithm to compute in a powerful desktop computer and store the calculated weights and eigenface information in the embedded module. The only processing task left is, to detect the unknown animal image from the Raspberry PI camera module.

## **10. REFERENCES**

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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. Zolnierok, Eds., ed: Springer International Publishing, 2013, pp. 187-196.
- [5] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Cognitive Neuroscience, Journal of*, vol. 3, pp. 71-86, 1991.
- [6] 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, pp. 711-720, 1997.
- [7] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th ed. New York: Dover, 1972.
- [8] W. W. R. Ball and H. S. M. Coxeter, *Mathematical Recreations and Essays*. New York: Dover, 1987.
- [9] S. Alakkari and J. J. Collins, "Eigenfaces for Face Detection: A Novel Study," in *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, 2013, pp. 309-314.
- [10] E. W. Weisstein. "Exponential Decay." From MathWorld--A Wolfram Web Resource. Available: <http://mathworld.wolfram.com/ExponentialDecay.html>
- [11] J. Stewart, *Calculus Concepts And Contexts* Gary W. Ostedt, 2001.