Facial Expression Recognition

Cemre Zor

Submitted for the Degree of

Master of Science in Signal Processing and Machine Intelligence

from the

University of Surrey

Department of Electronic Engineering
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey, GU2 7XH, UK

August 2008

Supervised by: Dr. Terry Windeatt

ACKNOWLEDGEMENTS

I would like to offer my gratitude to my supervisor, Dr. Terry Windeatt, who has supported, helped and encouraged me throughout the whole project. He is not only a very good and knowledgeable supervisor but also a great person; and it was an honour for me to have the opportunity to work with him.

And my most sincere thanks go to all my family members; especially to my mother Semra Zor and grandfather Kazım Erkul, for all their greatest moral supports and financial help during my studies abroad.

DECLARATION OF ORIGINALITY

I confirm that the project dissertation I am submitting is entirely my own work and that any material used from other sources has been clearly identified and properly acknowledged and referenced. In submitting this final version of my report to the JISC anti-plagiarism software resource, I confirm that my work does not contravene the university regulations on plagiarism as described in the Student Handbook. In so doing I also acknowledge that I may be held to account for any particular instances of uncited work detected by the JISC anti-plagiarism software, or as may be found by the project examiner or project organiser. I also understand that if an allegation of plagiarism is upheld via an Academic Misconduct Hearing, then I may forfeit any credit for this module or a more sever penalty may be agreed.

Facial Expression Recognition

Cemre Zor

Author Signature Date: 29/06/2008

Supervisor's name: Dr. Terry Windeatt

WORD COUNT

Number of Pages:	71
Number of Words:	20027

ABSTRACT

Face expression analysis and recognition has been one of the fast developing areas due to its wide range of application areas such as emotion analysis, biometrics, image retrieval and is one of the subjects on which lots of research has been done through solving the problems occurring in recognition of the face expressions under different illuminations, orientations and numerous other variations.

Different methods, all aiming to meet different requirements, have been used in solving facial expression analysis problems. These methods consist of pre-processing and processing parts. The detection and extraction of face images from the input data together with the normalization process, which aims to align these extracted images independent of varying environmental conditions such as illumination and orientation, form up the pre-processing part. The processing part on the other hand; aims to extract specific features from the already pre-processed images, and recognize the facial action units/facial expressions depending on these features. Several methods try to find different optimised algorithms mainly for the processing part.

Thus, the project primarily concentrates on the recognition of expressions that are generated on 44 action units on face. The steps of the resulting systems, which aim to recognize the facial expression action units; mainly consist of normalization, feature extraction, feature selection and binary & multi-class classifications of the frontal face images. Experiments are carried out on various system combinations; and comparisons have been made. In the binary classification case, in which the facial expressions are recognized in “1 vs. all” manner, the results are found to be quite satisfactory in terms of accuracy. In multi-class classification case, the combinations of different expressions occurring at the same time are treated as separate classes and different methods such as Error Correcting Output Codes have been applied. Although for a large number of classes, the results are not as accurate as the ones of the binary case; the recognition speed obtained is quite pleasing. When certain conditions such as having large numbers of training samples and small number of classes are fulfilled, the results also turn out to be a lot better in terms of accuracy as well as speed.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

1 INTRODUCTION

Automatic facial expression recognition has been used in various real life applications such as security systems, interactive computer simulations/designs, computer graphics, psychology and computer vision. In this project, the aim is to implement binary and multi-class face expression analysis algorithms based primarily on 'Facial Action Coding System' by evaluating features such as Haar-like, Gabor, Haar wavelet coefficients; and making use of classifiers like Support Vector Machines and Error Correcting Output Codes (ECOC) combined with feature selection methods such as Adaboost to be used in automated systems in real-life applications based on learning from examples.

1.1 Background and Context

Since facial expression analysis is used in various applications in real-life and is one of the topics of interest in pattern recognition/classification areas, it has been taken into consideration by various researchers in various methodologies. For example, some researchers [3][4] have taken the whole face into account without dividing it into sub-regions or sub-units for processing while some others came up with sub-sections[5] for the implementations of their methods. Different parameterisation techniques, all aiming to solve the classification problem in the most efficient way possible, have been introduced and used together with the above expressed methods.

To give an example to the application areas of facial expression recognition, one might think about the computer simulations and animations that are carried out in movies/cinema and in computer games. Recognition of the expression on face can further be used in various other aspects, such as in projects where a driver's expression is examined to decide whether he is tired and therefore an alert should be displayed to fulfil the requirements of a safe drive.

1.2 Scope and Objectives

The project mainly aims to come up with a solution to the facial expression recognition problem by dividing it into sub-problems of classifications of some specific 'Action Units'. The project's scope includes not only the two class problems which tell about whether an Action Unit is on or off, but also the multi-class problems that will inform the user about multi occurrences of more than one Action Unit at the same time. For this, different methodologies and techniques for feature extraction, normalisation, selection and classification are considered. The resulting system comes up with solutions to these problems as well as taking the computational complexity and timing issues into consideration.

1.3 Achievements

In this dissertation, several algorithms for extraction of feature vectors, followed by selection and classification methods in binary and multi-class cases, have been tried on different datasets. The binary classification results for the Upper Face Action Units have been observed to be better than the current results in the literature. Achievement of a classification rate about 94.5% for Action Unit 1, and over 96.5% for Action Unit 2 are examples for the results obtained in the final system which are quite challenging.

On the other hand, the results for the multi-class classification scheme for a large number of classes, such as 12, came out to be rather less accurate. However, when the number of classes is as small as 3; and the number of patterns in the training sets are larger than 50, classification rates over 96% could have been achieved again, which is satisfactory.

One of the other main achievements is the speed obtained as a result of using ECOC in the multi-class classification part, which can hardly be achieved by making use of other techniques in the literature.

1.4 Overview of Dissertation

In Chapter 2; 'the state-of-the-art' which includes information about different methodologies and approaches for facial expression recognition in the literature has been examined. In the chapters numbered 3 to 7, the system that has been used in the implementation of the project has been introduced in detail together with its technical and theoretical parts. The chapters are mainly about: Facial Action Coding System, Image Normalization, Feature Extraction, Feature Selection and Classification. In Chapter 8, the experiments carried out on different datasets together with their results and discussions are presented. The final chapter, Conclusion, summarizes and evaluates the project, and addresses to some future work that can be carried out in order to improve the current results.

2 STATE-OF-THE-ART

2.1 Introduction

In the literature, when facial expression analysis is considered; two main different approaches, both of which include two different methodologies, exist. Dividing the face into separate action units or keeping it as a whole for further processing appears to be the first and the primary distinction between the main approaches. In both of these approaches, two different methodologies, namely the

‘Geometric-based’ and the ‘Appearance-based’ parameterisations, can be used. In the following subtitles, details of the two approaches and the two methodologies have been further discussed.

2.2 Two Main Approaches for Facial Expression Analysis on Still Images

The two main approaches can be summarised as follows:

1. Making use of the *whole* frontal face image and processing it in order to end up with the classifications of 6 universal facial expression prototypes: disgust, fear, joy, surprise, sadness and anger; outlines the first approach. Here, it is assumed that each of the above mentioned emotions have characteristic expressions on face and that’s why recognition of them is necessary and sufficient. Ekman, Friesen [1] and Izard [2] have proposed these facts in their related work and Bartlett, Littlewort et al [3] [4] have used the method for fully automatic recognition systems.

2. Instead of using the face images as a whole, dividing them into some sub-sections for further processing forms up the main idea of the second approach for facial expression analysis. As expression is more related with subtle changes of some discrete features such as eyes, eyebrows and lip corners; these fine-grained changes are used for analysing automated recognition. This approach has been presented to be the ‘Facial Action Coding System’, which is first developed by Ekman and Friesen [5], for describing facial expressions by 44 different Action Units (AU’s) existing on face. The advantage here is that, this decomposition widens the range of applications of face expression recognition. This is due to ending up with individual features to be used in/with different processing areas/methods other than just having the 6 universal facial expression prototypes. Most of the current work done on facial expression analysis makes use of these action units.

It should be mentioned that, there are also some other methods in which neither the frontal face image as a whole nor the all of 44 action units themselves, but some other criterion such as the manually selected regions on face [6] or surface regions of facial features [7] are used for the recognition of the facial expression.

2.3 Geometric and Appearance Based Parameterizations

There are two main methods that are used in both of the above explained approaches:

1. Geometric Based Parameterization is an old way which consists of tracking and processing the motions of some spots on image sequences, firstly presented by Suwa to recognize facial expressions [8]. Cohn and Kanade later on tried geometrical modeling

and tracking of facial features by claiming that each AU is presented with a specific set of facial muscles. In general, facial motion parameters [6] [7] and the tracked spatial positioning & shapes of some special points [9] [10] on face, are used as feature vectors for the geometric based method. These feature vectors are then used for classification. The following might be regarded as the disadvantages of this method:

- The approximate locations of individual face features are detected automatically in the initial frame; but, in order to carry out template based tracking, the contours of these features and components have to be adjusted manually in this frame. (And this process should be carried out for each individual subject)
- The problems of robustness and difficulties come out in cases of pose and illumination changes while the tracking is applied on images.
- As actions & expressions tend to change both in morphological and in dynamical senses, it becomes hard to estimate general parameters for movement and displacement. Therefore, ending up with robust decisions for facial actions under these varying conditions becomes to be difficult [18].

2. Rather than tracking spatial points and using positioning and movement parameters that vary within time, colour (pixel) information of related regions of face are processed in Appearance Based Parameterizations; in order to obtain the parameters that are going to form the feature vectors. Different features such as gabor, haar wavelet coefficients, together with feature extraction and selection methods such as pca, lda, and Adaboost are used within this framework. Example research can be found in [12] [4].

The combination of the Geometric and Appearance based methods have also been used in some work in the literature. For example, Zhang [11] has tracked some fiducial points on the face images while also taking the Gabor wavelets of these points into account for the facial expression recognition.

2.3.1 More about Appearance Based Parameterizations

One of the most successful approaches for expression recognition by making use of appearance-based parameterizations consists of applying Gabor filters for feature extractions of AU's, and then using support vector machines to classify them. But, Omlin and Whitehill [12] have shown that although the recognition rates for this method are satisfactory, the approach is very inefficient in memory usage and very slow due to the high redundancy of Gabor representation. They proposed using Haar features instead of Gabor, as Haar features can be extracted quickly without a need of Fourier transform in contrast to Gabor. Secondly, for the classifier part, they made use of Adaboost, which performs feature selection at the same time with classification. As for the advantages, they have shown that Adaboost as a classifier works 300 times faster than SVM and that Haar coefficients + Adaboost yields mostly better results than Gabor wavelets (five spatial frequencies and eight orientations) + Support Vector Machines.

In the project, in order to avoid the above mentioned disadvantages of Geometric based methods, Appearance based methods have been decided to be used. Within the implementation of the binary classification scheme of each class; different from Omlin and Whitehill's suggested methods, Adaboost is just used as a feature selection method rather than a method for carrying out classification together with feature selection. The classification itself is achieved by Support Vector Machines. This scheme gives out better average classification results; although the classification time comes out to be a little bit slower than before. Apart from Haar and Gabor wavelets, 'Haar-Like' features are also used in the project as feature vectors and the difference in terms of accuracy is measured and reported.

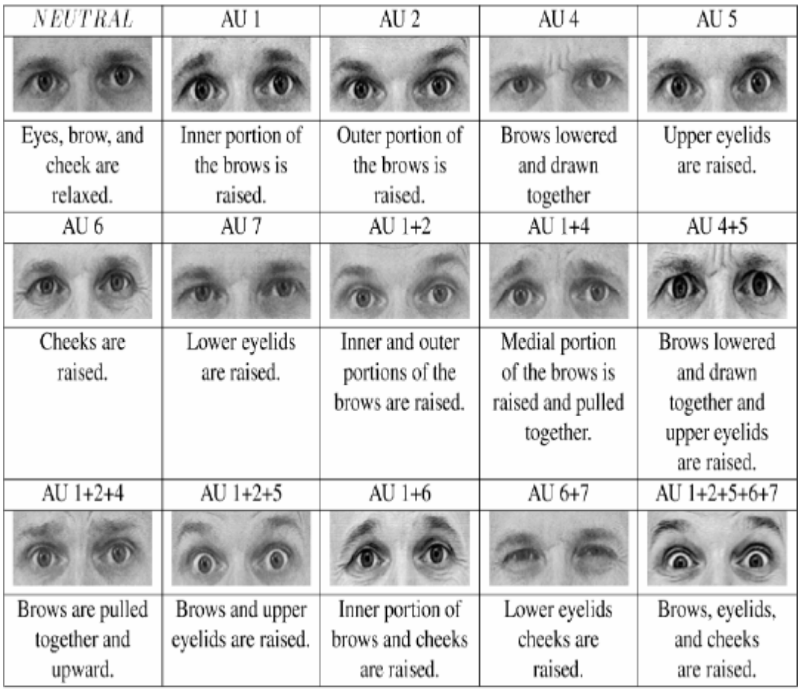
When the multi-class classification problem is taken into consideration, the algorithm implemented uses the 'Error correcting output code (ECOC)' technique which is combined with the Adaboost feature selection and Support Vector Machines classification techniques, together with an application of Bootstrapping on the training data. ECOC is examined to be giving out much smaller amount of classification time for the multiclass AU recognition problems although the classification rates comes out to be less robust.

3 FACIAL ACTION CODING SYSTEM (FACS)

This system (FACS), which was first developed by Ekman and Friesen [5] for describing facial expressions by 44 Action units, is a strong and useful method for facial expression recognition. This is because it decomposes the facial expression into individual action units by supplying flexibility, objectivity; and can be used in various applications which measure fine-grained changes in facial expressions in a comprehensive way.

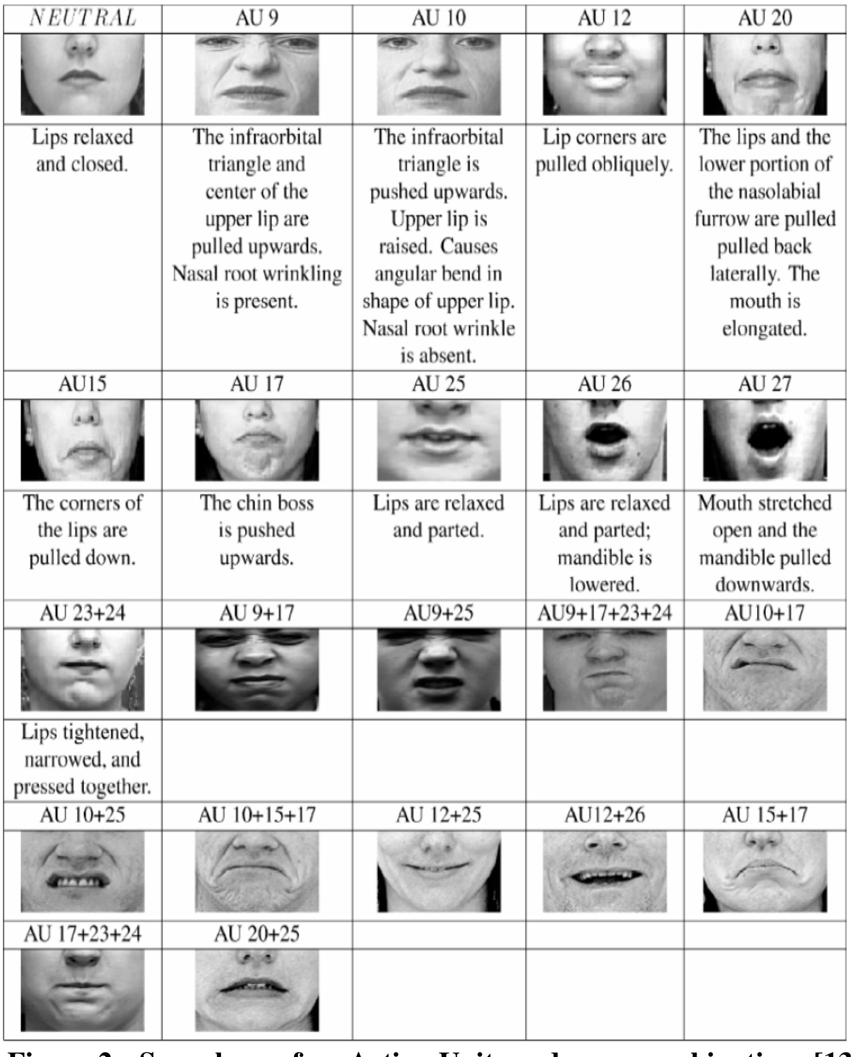
3.1 Action Units

There are 44 AU's in total and we see that 30 of them are obtained by contractions of the specific facial muscles. Out of these 30, 12 are for the upper face and 18 are for the lower [13]. Figure 1 and Figure 2 shows some upper and lower face action units together with some combinations.



NEUTRAL	AU 1	AU 2	AU 4	AU 5
Eyes, brow, and cheek are relaxed.	Inner portion of the brows is raised.	Outer portion of the brows is raised.	Brows lowered and drawn together	Upper eyelids are raised.
AU 6	AU 7	AU 1+2	AU 1+4	AU 4+5
Cheeks are raised.	Lower eyelids are raised.	Inner and outer portions of the brows are raised.	Medial portion of the brows is raised and pulled together.	Brows lowered and drawn together and upper eyelids are raised.
AU 1+2+4	AU 1+2+5	AU 1+6	AU 6+7	AU 1+2+5+6+7
Brows are pulled together and upward.	Brows and upper eyelids are raised.	Inner portion of brows and cheeks are raised.	Lower eyelids cheeks are raised.	Brows, eyelids, and cheeks are raised.

Figure 1. Upper face Action Units and some combinations [13]



NEUTRAL	AU 9	AU 10	AU 12	AU 20
Lips relaxed and closed.	The infraorbital triangle and center of the upper lip are pulled upwards. Nasal root wrinkling is present.	The infraorbital triangle is pushed upwards. Upper lip is raised. Causes angular bend in shape of upper lip. Nasal root wrinkle is absent.	Lip corners are pulled obliquely.	The lips and the lower portion of the nasolabial furrow are pulled pulled back laterally. The mouth is elongated.
AU15	AU 17	AU 25	AU 26	AU 27
The corners of the lips are pulled down.	The chin boss is pushed upwards.	Lips are relaxed and parted.	Lips are relaxed and parted; mandible is lowered.	Mouth stretched open and the mandible pulled downwards.
AU 23+24	AU 9+17	AU9+25	AU9+17+23+24	AU10+17
Lips tightened, narrowed, and pressed together.				
AU 10+25	AU 10+15+17	AU 12+25	AU12+26	AU 15+17
AU 17+23+24	AU 20+25			

Figure 2. Some lower face Action Units and some combinations [13]

It is known that AU's occur either singly or in different combinations and there are about 7000 different combinations existing. These combinations have two different kinds; *additive combinations* are the ones in which the appearance of the constituent AU's remains the same while *non-additive combinations* are the ones the appearance changes in. In Figure 3, it is examined that in the AU 1 + AU 4 combination; the brows are raised due to the action of AU 1, and this cancels out the effect of AU 4 in which brows are lowered in fact. Therefore, AU 1 + AU 4 is an example of a non-additive combination.

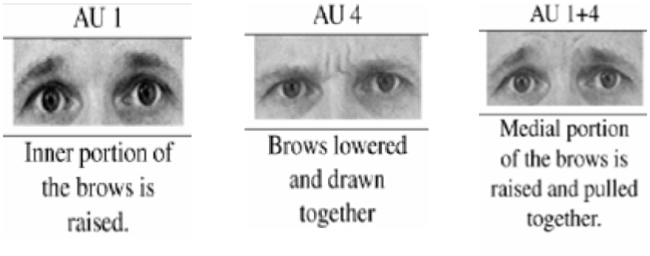

AU 1
Inner portion of the brows is raised.
AU 4
Brows lowered and drawn together
AU 1+4
Medial portion of the brows is raised and pulled together.

Figure 3. A non-additive combination example [13]

4 IMAGE NORMALIZATION

In the project implementation, the normalization process is applied on each image. This process consists of cropping the eye regions, setting the pupil locations to be in the same place, having the upper/lower faces in the same sizes and orientations, and applying histogram processing.

4.1 Automatic Pupil Localization

The normalization process has been carried out on all the images in the training and test sets before any further action such as extraction and selection of features is taken.

As the project experiments are mainly carried on upper face action units, eye centres / pupils are to be localized in the input images as the first step of the normalization process. So as to end up with accurate results in the project, localization was done manually for the experiments; however the software that has been developed for the real life application is making use of an automated way of doing this localization. This method is a combination of Haar Cascade Classifiers and the Eye Mapping technique which I have implemented for the fulfilment of my Bachelor of Science graduation project [19]; and is briefly summarized below.

4.1.1 Haar Cascade Classifiers for the Eye Region Cropping

By making use of Haar-Like features which are described in detail in the following feature extraction section 5.2, Haar Cascade Classifiers implement the idea of using the change in the contrast values between adjacent rectangular groups of pixels, instead of their individual intensity values. Also Haar-Like features' ability to get easily scaled and rotated makes them useful for the detection of the objects of various sizes.

Two trained cascade classifiers, one for each eye, have been used to detect the eye regions. The term 'cascade classifier' means that the main classifier consists of several simpler sub-classifiers each of which acts like a separate stage; and all these stages are applied one by one to the regions in the image in order to determine / detect the candidate eye region.

As the algorithm may return a number of eye region candidates, some other checks have been performed by invoking some functions; such as the Canny edge detector to reject candidates with too much or too few edges, and size check functions to eliminate small regions.

The localization of the coordinates of the pupils is done by using the next technique on these already cropped eye regions.

4.1.2 Eye Mapping Technique to Locate the Pupil Centre

The EyeMap Technique presented by Rein-Lien Hsu, Anil K. Jain and Mohamed Abdel-Mottaleb [20] uses a series of morphological operators on the luma (Y – brightness information) and chroma (Cr, Cb – color information) components of the eye regions in order to emphasize bright and dark pixels. Hsu, Jain and Abdel-Mottaleb indicate that an analysis of the chrominance components around the eyes reveals high Cb and low Cr values; and analysis of the luma component reveals both dark and bright pixels in the luma component on eye [20]. Therefore, after combining some morphological operations that are applied on these two kinds of components, a final gray scale image is obtained. On this image, the eye itself is emphasized. In the project, an ellipse has been fit into this region and its centre point is taken to be the pupil centre.

Figure 4 shows the outputs of Haar Cascade Classifiers and the EyeMap technique:

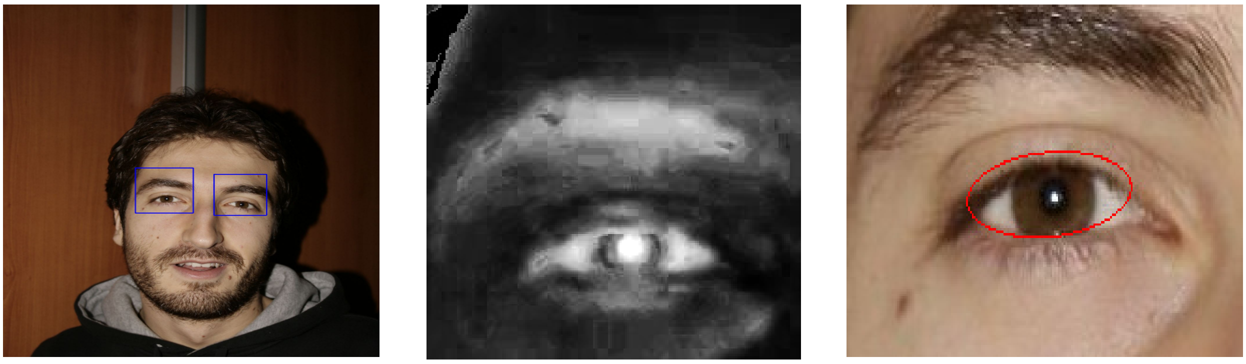


Figure 4. Haar Casade Classifiers + EyeMap technique for pupil localization

4.2 Rotation and Size Normalizations

The normalization process from this step on is carried on on the original face images (rather than the cropped region outputs of the Haar Cascade Classifiers) by making use of the extracted coordinates of the pupils. (The images are converted into gray scale).

As soon as the pupil coordinates are marked in each image, orientation normalization is carried out by having the line between the y-coordinates of the pupil centres in each image have a 0 degree slope. Therefore the images are rotated in clockwise and counter clockwise directions by going through a bilinear transform.

Secondly, in order to have the same distance (d) between the left and right eye centres of each image, distance normalization has been applied and as a result all images are bilinearly resized such that d is equal to 32. As d is selected to be 32, 32 by 32 square cropped regions around the pupils include the eyes and the eyebrows inside. Therefore 32 by 32 regions around the pupils on the so-far-normalized images have been cropped in the final stage of this part.

4.3 Histogram Processing

The already detected, cropped, resized and rotated 32 by 32 gray scale images are then made to go through a couple of histogram processing steps.

First of all contrast sketching has been applied, and afterwards low pass filtering is used in order to get rid of noise on images.

In the second stage of the histogram processing task, elimination of some pixel values has been carried out. The histograms of the images that are normalized up to this point look like shifted and scaled Gaussians. It is proved through statistics theory that about 68% of values drawn away from a normal distribution lie within one standard deviation from the mean; 95% are within two standard deviations and about 99.7% lie within three. By making use of this fact, approximations on the image histograms are made in the following way: The histogram values are sorted in descending order and the average of the indexes of the first 10 values is considered to be the index of the approximated Gaussian's mean value. From this point on, all the indexes which are within 2 standard deviations are kept and the rest are discarded. If the addition or subtraction of the 2 standard deviations cause the data to go beyond 0 or 255, then it means the original distribution is either shifted to the 0 or to the 255 side of the histogram; therefore for these kind of data, the elimination of the pixels is done for the ones having indexes less than 5 (first case) or greater than 250 (second case). This check is done in order not to end up with very dark or bright outliers. The image is histogram equalized (contrast sketched) to the newly determined minimum and the maximum pixel values.

In the next stage, the log / power-law transformation is applied in order to pull the average pixel value of each image to 128. The aim here is to get rid of the above mentioned shifts and variances of the approximated image histogram Gaussians. After this stage, contrast sketching to 0-255 is applied for the last time.

5 FEATURE EXTRACTION: HAAR WAVELET, HAAR-LIKE AND GABOR WAVELET COEFFICIENTS

In this section, detailed information about Haar and Gabor wavelets and Haar-like coefficients is given. These three kinds of coefficients form up the feature vectors that are going to be used in the classification process of the FACS action units, after going through feature selection algorithms such as Adaboost. Extraction of the images is applied on the already normalized frontal face images.

5.1 Haar Wavelet Coefficients

The advantages of making use of Haar Wavelet Coefficients instead of the most commonly used Gabor Wavelets were expressed to be the Haar Wavelets' superiority in terms of extraction times. Haar Wavelet's mother wavelet function might be considered as a kind of step function. Namely;

$$\Psi(t) = \begin{Bmatrix} 1 & 0 \le t < 1/2 \\ -1 & 1/2 \le t < 1 \\ 0 & otherwise \end{Bmatrix}$$

The wavelet decomposition of an image could therefore be defined as the combination of the resulting difference images calculated in different scales. The process of calculating Haar wavelet coefficients of an array of samples is as follows [14]: (Array length should be a power of two)

1. Find the average of each pair of samples.
2. Find the differences between the averages and the samples.
3. The first half of the array is filled with the computed averages.
4. The second half composes of the differenced calculated in Step2.
5. Recurse – The process is repeated on the first half of the array until recursion is not possible anymore.

Example Haar Wavelet decomposition of an eight-element array:

7 9 1 1 5 3 2 -10

The average of the each pair of samples is calculated:

(7+9)/2 = 8; (1+1)/2 = 1; (5+3)/2 = 4; (2-10)/2 = -4

The differences between the averages and the samples are calculated:

(7-8) = -1; (1-1) = 0; (5-4) = 1; (2-(-4)) = 6

4. First and the second halves of the array are filled with the averages and the differences.

8 1 4 -4 -1 0 1 6

The method is applied to the first half of the array recursively.

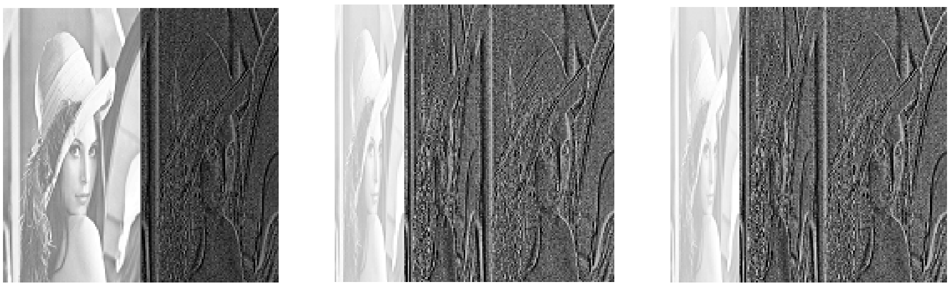


Figure 5. Haar transform through rows: First, second, 9th iterations[15]

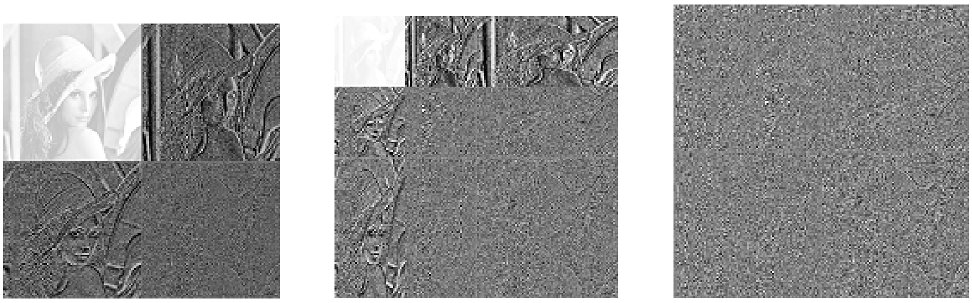


Figure 6. Haar transform through columns & rows: First, second, 9th iterations [15]

5.2 Haar-Like Coefficients

Haar-Like coefficients are features which are reminiscent of Haar Basis functions introduced by Viola and Jones [16]. They make use of three kinds of features: two-rectangle, three-rectangle and four-rectangle features. In all of these, differences between the sums of pixels between same-size pairs of rectangles are computed as features. The disadvantage of these features is that it takes a long time to extract all the Haar-like features of an input image. To be more precise, it might be given as example that, a 24*24 image has 160000 Haar-like features while a 32*32 one has more than 450000. This means that the set of features is many times over-complete. However, the one of the main advantages is that, any rectangular sum can be computed in only four array differences by making use of the "integral image" method that is expressed by Viola and Jones [16]. This, together with the fact that the features are going to proceed through a feature selection process, causes an important reduction in the complexity of the extraction of the features. Another advantage of the method can be mentioned as the features' sensitivity to edges, boundaries and other important information hidden in pixel values such as the difference between the pixel values on the regions of motion on face.

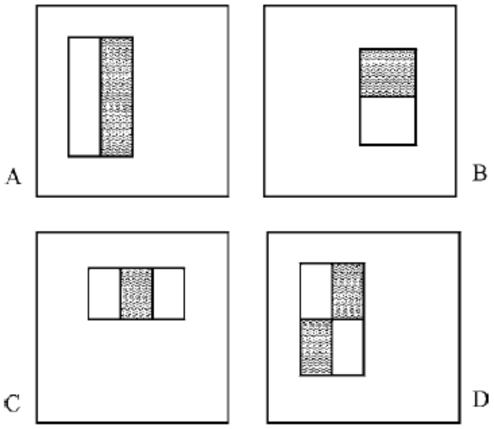


Figure 7. Two (A and B), three (C) and four (D) rectangle features. Sum of pixels in the white rectangles are subtracted from sum of pixels in the black ones [16].

5.3 Gabor Coefficients

Gabor wavelet transformation has been used in various kinds of signal and pattern processing / analysis areas both in spatial and in frequency domains and is found to give out satisfactory results in application areas such as texture segmentation [22], fingerprint recognition [23] and face recognition [21]. The characteristics of Gabor wavelets such as their ability to get easily adjusted for detailed localization in spatial and frequency domains [24] and the similarity of their frequency and orientation representations to those of the human visual system have made them popular for particular usage areas and as a result they have been found to be revealing satisfactory results in the application areas mentioned above.

Gabor wavelets are formed from the multiplication of a complex sinusoidal carrier with a Gaussian envelope. The derivations of the Gabor wavelets can be done in the following way [25]:

<u>The complex sinusoidal</u> can be defined as $s(x, y) = \exp(j(2\pi(u_0 x + v_0 y + P))$; u_0 and v_0 being the spatial frequencies, and P being the phase. It is straightforward that this sinusoidal carrier consists of two sinusoids, one cosine wave on the real, and a sine wave on the imaginary domains. In order to switch from the Cartesian coordinates to polar; F_0, the magnitude of the frequencies is defined together with w_0, the direction.

$$F_0 = \sqrt{u_0^{\ 2} + v_0^{\ 2}}$$
$$w_0 = \arctan(v_o / u_o)$$
$$s(x, y) = \exp(j(2\pi F_0 (x \cos w_0 + y \sin w_0) + P))$$

Figure 8 shows the real and imaginary part of a complex sinusoidal.

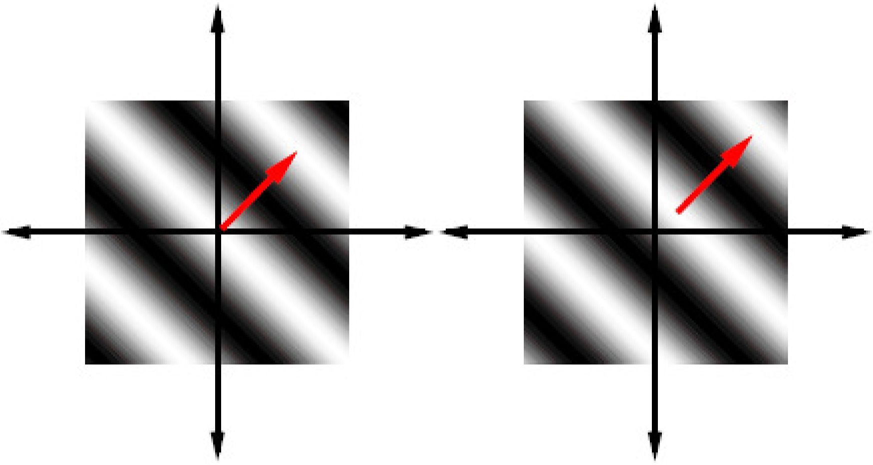


Figure 8. The real and imaginary parts of a sinusoidal carrier $u_0 = v_0 = $1/80 cycles/pixel and P =0 degrees [25]

The second part of the Gabor wavelets, namely the Gaussian envelope, can be defined as: $w_r(x,y) = K\exp(-\pi(\alpha^2(x-x_0)_r{}^2 + b^2(y-y_0)_r{}^2))$. Here, (x_0, y_0) is the peak point of the function. a and b are scaling parameters such that when they get smaller, Gaussian gets larger in spatial domain, and $(x-x_0)_r$ $(y-y_0)_r$ are the clockwise rotated versions of $(x-x_0)$ and $(y-y_0)$ and can be defined in the following way:

$$(x-x_0)_r = (x-x_0)\cos\theta + (y-y_0)\sin\theta$$
$$(y-y_0)_r = -(x-x_0)\sin\theta + (y-y_0)\cos\theta$$

Figure 9 shows an example Gaussian envelope on a 128*128 image:

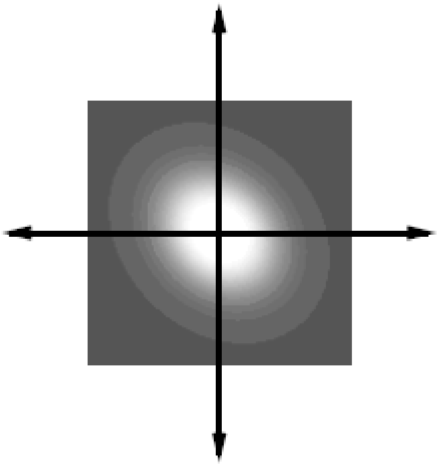


Figure 9. Example Gaussian envelope

$$x_0 = y_0 = 0, a = 1/50\,pixels, b = 1/40\,pixels, \theta = -45\deg,$$
$$F_0 = \sqrt{2}/80\,cycles/pix, w_0 = 45\deg, P = 0\deg$$

The Gabor Wavelet which is the combination of the sinusoidal carrier and the Gaussian envelope can then be defined through plugging in the above derived equations:

In spatial domain with Cartesian coordinates:

$$g(x,y)=K\exp(-\pi(a^2(x-x_0)_r{}^2+b^2(y-y_0)_r{}^2))\exp(j(2\pi(u_0x+v_0y)+P))$$

And in polar coordinates:

$$g(x,y)=K\exp(-\pi(a^2(x-x_0)_r{}^2+b^2(y-y_0)_r{}^2))\exp(j(2\pi F_0(x\cos w_0+y\sin w_0)+P))$$

The parameters that have been used so far can be summarized as follows:

K : The scale parameter of the magnitude of the Gaussian envelope.

a,b : The scale parameters of the x and y axis of the Gaussian envelope.

θ : Parameter for the rotation angle of the Gaussian envelope.

(x_0, y_0) : Peak value of the Gaussian envelope occurs at (x_0, y_0)

(u_0, v_0) : Sinusoidal carrier's Cartesian spatial frequencies. Is equal to (F_0, w_0) .

P : Phase of the sinusoidal carrier

In Figure 10, Gabor wavelet function is shown on real and imaginary parts.

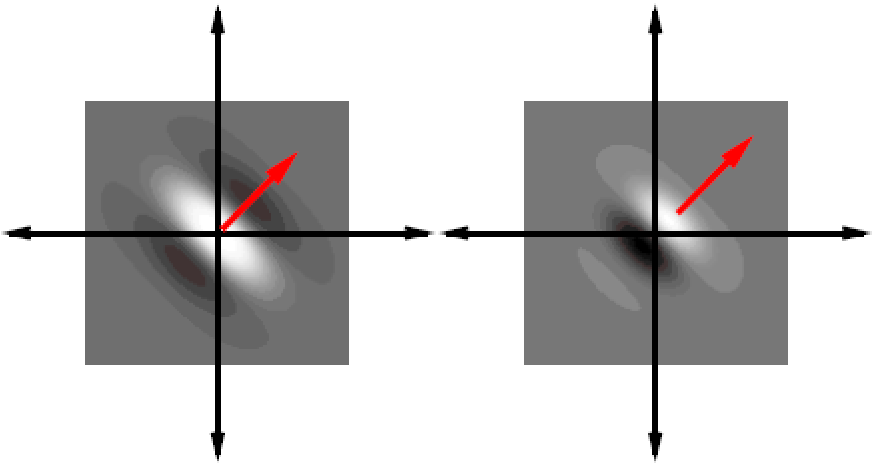


Figure 10. The real and imaginary parts of a Gabor wavelet function

$$x_0=y_0=0, a=1/50\,pixels, b=1/40\,pixels, \theta=-45\deg,$$
$$F_0=\sqrt{2}/80\,cycles/pix, w_0=45\deg, P=0\deg$$

5.3.1 Interpretation of Some Coefficients

It has been proved and can be found in the literature that the direction of the frequency of the sinusoidal carrier is roughly taken to be equivalent to the rotation angle of the Gaussian envelope. Therefore, in the experiments carried out, $w_0 = \theta$.

Also, one of the parameters of the Gabor filters, the maximum value of the sinusoidal carrier's frequency F_0 used in the implementation, should be determined according to the sampling theorem: We know that the Gabor wavelet function in polar coordinates is:

$$g(x,y) = K\exp(-\pi(a^2(x-x_0)_r{}^2 + b^2(y-y_0)_r{}^2))\exp(j(2\pi F_0(x\cos w_0 + y\sin w_0) + P)) .$$

An image in fact is a sampled version of this function, that is to say, on each integer value of x and y, one image pixel value is sampled. Therefore, $fs = 1 samples/\sec$; if you define x and y as time axis for simplicity. According to the Nyquist sampling theory: $fs > 2f\max$; and this brings about the fact that, in order to reconstruct properly and completely, a 2D image should contain the greatest F equal to 0.5 Hertz (cycles/pixel).

6 FEATURE SELECTION

Different feature selection methods have been applied on the already extracted features in this stage of the project. Apart from Adaboost which came out to be the main feature selection method for the whole facial expression recognition system, Forward Feature selection and Branch & Bound algorithm has also been used in the first stage experiments carried out on the UCI Machine Learning Repository (MLR) [26].

6.1 Forward Feature Selection Method:

Forward feature selection is a feature selection method in which the feature giving out the best classification results when combined with the previously selected ones is iteratively selected in each single run. Features might be selected based upon their classification accuracies in certain classifiers as well as their abilities to fulfil certain evaluation criterion such as minimum of squared Euclidean distances. The algorithm is known to be fast however suboptimal, as the feature combinations might not always give out the best possible classification results; sometimes even the best feature, by which the data can be separated completely, might come out to be redundant in this method.

6.2 Branch and Bound Algorithm

Branch and bound algorithm is a technique to find the optimal feature subset in the search tree of the subsets of features. By keeping the best solution found so far, it abandons the partial solutions if they do not improve the best. The solutions are checked due to some criterion function.

The high computational complexity of this algorithm comes out to be a problem; however, there are some techniques which try to increase the speed of the algorithm through applying methods like increasing the prediction mechanism so that the number of criterion evaluations decreases.

6.3 Adaboost

Boosting is the name of the general method in which the aim is to improve the performances of the learning algorithms of classifiers. Adaboost is a strong and a fast classifier, which is first introduced by Freud and Schapire [17], and which makes use of a weak binary classifier and strengthens its decisions in each iteration to end up with a final hypothesis with the lowest error rate.

The weak binary classifier is any classifier for which the weighted classification error is expected to be better than chance. In this project, one node decision tree – stump, has been used as the weak binary classifier.

There is another version of Adaboost (AdaFs) that makes use of feature selection during the classification process, and is therefore one of the powerful and fast classification and feature selection methods.

6.3.1 Decision Tree – Stump

A Decision Stump is a weak machine learning model which consists of only one level of branching (maximum depth is equal to 1). A stump therefore contains a single decision node and that node is connected to two prediction leaves. The decision node as a weak binary classifier is in fact a single rule which assigns the input vector to one of two classes by defining a threshold on a *single* feature which is selected such that the weighted error is minimized.

There are some methods for decision trees to decide which property should be selected at each node. (In the case of decision stump we only have got one decision node; therefore only one property is to be selected.) The decisions in general are preferred to lead to simple trees and as pure immediate descendent nodes as possible [27].

Most of the popular measures are to do with information theory. For example, the entropy can be used to measure the impurity of a node N. If all patterns reaching that node have the same category label, impurity is equal to 0 whereas if categories are equally distributed it is large. Therefore the aim is to have an impurity which is as low as possible, and it follows that the query about a feature that reveals the lowest entropies is selected to be the rule on the decision node. Just similarly, information gain can be used in selecting the relevant feature that will be on the decision node together with its defined threshold. It supplies us with the knowledge about, given a specific feature's value, what the average fraction of information obtained about classification is.

As already stated, Decision Stump has been used as the weak learning algorithm in Adaboost which will be explained in detail in the following sections.

6.3.2 Adaboost Classifier without Feature Selection

The steps describing the way Adaboost works as a classifier without carrying out feature selection are as follows:

For a manually decided number of iterations (do for $t = 1, 2, ..., T$):

- Calculate the probability distribution of the data. Set the initial distributions & weights to be equal to each other if you don't have prior knowledge about them.

  Therefore calculate $p^t = \dfrac{w^t}{\sum_{i=1}^{N} w^t{}_i}$.

  Here the divisor is used for normalization purposes regarding the probability distribution.
- Call the WeakLearn algorithm to learn the N* M training data (N = number of data, M = number of features) by taking all the features into consideration and then apply the classification again on the training data. The WeakLearn is provided with the distribution p^t .
- Calculate the overall training classification error (error per pattern is equal to1 if it is misclassified, 0 otherwise).

  $$\varepsilon_t = \sum_{i=1}^{N} p_i^t \mid h_t(x_i) - y_i \mid$$
- End up with a loss function by using this overall error and use it to reset the weights of the distribution.

  Loss function here is $l_i^t = 1 - \mid h_t(x_i) - y_i \mid \;\; w_i^{t+1} = l_i^t \beta_t^{l^t{}_i} \varepsilon_t ti$

  And the weights are updated such that $w_i^{t+1} = w_i^t \beta_t^{l^t{}_i}$.

  It should be mentioned here that the parameter β_t is chosen as a function of ε_t . It is used to update the weight vector in a way that the weight and therefore the probability given to the examples which are wrongly classified (which are 'difficult' to classify) is increased and vice versa. In other words, loss l_i^t is small if the t th hypothesis/iteration suggests a bad prediction on the i th example, forcing the WeakLearn algorithm to give more importance to the classification of that [17].

- Go to the first step.

The final hypothesis is the weighted average of the all hypothesis reached by the WeakLearn algorithm in each iteration.

6.3.3 Adaboost Classifier with Feature Selection

Adaboost classifier, which is slightly modified by Viola and Jones [16] to perform feature selection at the same time with classification, is explained in this section.

In this algorithm, different from the original one, the features are taken into consideration independently within each single run. A binary classifier is created from the feature which can best discriminate among the classes, and the weights are updated accordingly.

That is to say, for a manually decided number of features times:

- Calculate the probability distribution of the data according to the weights. Set the initial distributions & weights to be equal to each other if you don't have prior knowledge about them.
- In each iteration, call the WeakLearn algorithm M times to learn the N* M data (N = number of data, M = number of features) according to a single feature each time.
- Apply each of t the trained M trained functions on the training set to calculate the individual training errors (error per pattern is equal to1 if it is misclassified, 0 otherwise).
- Out of the M functions select the one giving out the least error and calculate its corresponding loss function to be used in resetting the weights of the distribution.
- Just same as the original Adaboost the loss function, which resets the weights, gives priority to the examples which are wrongly classified (which are 'difficult' to classify) by increasing their weights; and does the vice versa for the correctly classified patterns.
- Go to the first step.

The final hypothesis is again the weighted average of the all hypothesis reached by the WeakLearn Algorithm in each iteration, which is equal to the number of selected features.

Simply, in both of the Adaboost algorithms, by making calls to the WeakLearn multiple times through altering the distribution over the feature domain each time so that the probability of "harder" samples /parts of the space is increased, the aim is to force the weak learning algorithm to generate new hypothesis that makes less mistakes on these parts [17]. Adaboost is also claimed to work more powerfully as a classifier than the boost-by-majority algorithms in which best prediction of each network is selected and then combined with a majority rule, as its final hypothesis' accuracy depends on all of the hypothesis that have been returned by WeakLearn.

In other words, the final error of Adaboost depends on the errors obtained from each WeakLearn hypotheses whereas the errors of the other boosting algorithms depend on the maximal error of the weakest hypothesis only. These algorithms therefore don't have the advantage of making use of the hypotheses whose errors are smaller [17].

7 CLASSIFICATION

Different classification methods to be used in binary and multi-class classification tasks are explained in this section.

7.1 Binary Classification

In the system, after going through the feature selection, the extracted features are classified for 1 vs. all class problems by making use of Support Vector Machines (SVMs). Detailed information about SVMs can be found in this section.

7.1.1 Support Vector Machines

Support Vector Machines, which were firstly developed from Statistical Learning Theory by Boser, Guyon and Vapnik in COLT-92, aim to perform 2-class classification via optimally separating the data by making use of an N-dimensional hyper-plane. Therefore, they belong to the family of *linear classifiers* and are examples of supervised learning. While dealing with the optimal hyper-planes that are to classify the data by minimizing the empirical classification error, Support Vector Machines take also the maximization of the margin, in other words achievement of the maximum separation into account.

Apart from the separating hyper-plane, if we consider 2 more hyper-planes which are parallel to the separating one and pass through the closest data points on each side; we end up with the "support hyper-planes" and these closest data points are named "support vectors" [28]. Margin of a linear classifier can then be defined as the width of the area between the two parallel hyper-planes. Selecting the maximum margin is one of the targets of the Support Vector Machine classifiers due to the empirical issues and the fact that the chance of having a misclassification decreases in case of a small error in the boundary's location. Also, the separating hyper-plane should be equidistant from both of the support hyper-planes in order to have chance of a misclassification in equal amounts on both sides. Figure 11 shows some of the possible boundary hyper-planes in a separable two-class problem and Figure 12 shows the corresponding margins of some possible boundaries in two different problems.

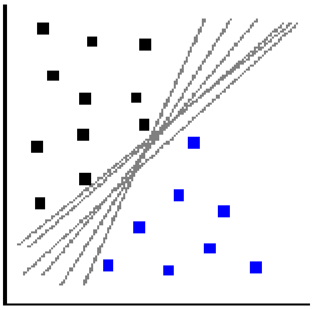


Figure 11. Possible boundary hyper-planes on a separable 2 class problem

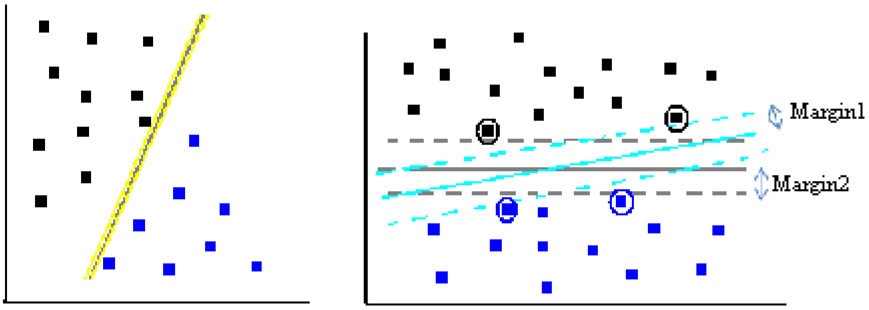


Figure 12. Margins of some boundaries

Support Vector Machines are studied for two different kinds of problems: The separable and the non-separable data problems:

7.1.1.1 The Separable Case

Label the data in the following way: $\{x_i, y_i\}$ for $i = 1,.....,l$, $y_i \in \{-1,1\}, x_i \in R^d$. x_i here are the input vectors and the y_i are the labels. In the case of a linear separation, a straight line of the form $f(x) = w.x + b$ can be considered such that the data points having labels $y_i = 1$ fall on the side where $f(x_i) > 0$ and the ones having labels $y_i = -1$ fall on the other which has the equation of $f(x_i) < 0$. Therefore, the equations for the hyper-planes come out to be $f(x) = w.x + b + \Delta$ and $f(x) = w.x + b - \Delta$. For convenience, Δ is taken to be equal to 1. The functions can be examined on Figure 13.

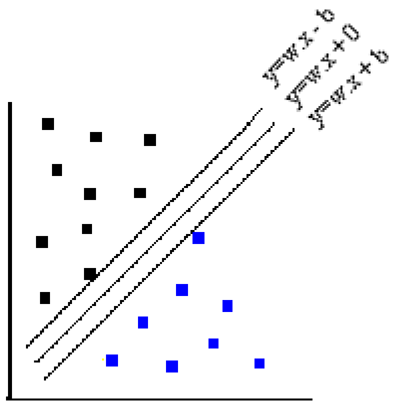


Figure 13. **Support hyper-plane equations for a separable 2 class problem**

The classification task now consists of taking the following two procedures into account:

1. Classification: Classify as $y_i = +1$ if $w.x + b \geq 1$

as $y_i = -1$ if $w.x + b \leq -1$

all of which can be combined as $y_i(x.w + b) - 1 \geq 0, \forall i$

2. Maximization of the margin: Define x^- as a point of the $y = w.x - b$ plane; x^+ as a point of the $y = w.x + b$ plane and M as the margin. It immediately follows that $M = | x^+ - x^- |$

As the hyper-planes are all parallel to each other, we can write $x^+ = x^- + a.w$. We plug this in $w.x^+ + b = +1$ and end up with

$$w.(x^- + aw) + b = 1$$
$$w.x^- + b + aw.w = 1$$
$$-1 + aw.w = 1$$
$$a = 2 / w.w$$

We know that, $M = | x^+ - x^- | = | aw | = a | w | = \dfrac{2 | w |}{w.w} = \dfrac{2\sqrt{w.w}}{w.w} = \dfrac{2}{\sqrt{w.w}}$ [39]

Maximizing the margin therefore requires the minimization of $\sqrt{w.w}$. In order to end up with/use a quadratic programming optimization, we can change the problem of the minimization of $\sqrt{w.w}$ with the one of $| w |^2$.

Now that we have the quadratic optimization problem of the minimization of $| w |^2$ according to the constraint $y_i(x.w+b)-1\geq 0,\forall i$; we can make use of the Lagrangian multipliers together with Karush-Kuhn-Tucker conditions.

Lagrange Multipliers and Karush-Kuhn-Tucker conditions: Given a set of necessary conditions (equality constraints) to identify optimal points in optimization problems, Lagrange multipliers are used. With the help of Lagrange multipliers, the constrained problem is converted into an unconstrained one by utilizing the Lagrange multiplier parameters.

As a summary, the problem of minimizing a particular problem $f(x_1,x_2,....,x_n)$ subject to some equality constraints $h_i(x)=0, i=1,..,m$ can be converted into the problem of the primal Lagrange, which is: $\min x \max \alpha L(x,\alpha)=f(x)-\alpha h(x)$ where $h(x)=(h_1,h_{2,},....,h_m)$.

In case of the constraints' having inequality rather than equality forms, there are some conditions that have to be satisfied in order for the problem to be solved again through the minimization of the primal Lagrange. These constraints are called Karush-Kuhn-Tucker conditions and are to be used in the Support Vector Machines problem as we are experiencing inequalities.

Back to the separable data classification problem:

The problem of minimizing $|w|^2$ subject to $y_i(x.w+b)-1\geq 0,\forall i$ can now be examined under the Lagrangian formulation of $\min w \max \alpha L_p(w,b,\alpha)=\frac{1}{2}|w|^2-\sum_{i=1}^{N}\alpha_i[y_i(w.x_i-b)-1]$.

It should be indicated that as the original problem is convex, the minimization can be interchanged with maximization by taking the dual of the primal Lagrangian. Therefore, $\min w \max \alpha L_p(w,b,\alpha)=\frac{1}{2}|w|^2-\sum_{i=1}^{N}\alpha_i[y_i(w.x_i-b)-1]$ becomes to be equal to the dual form: $\max w \max \alpha L_D(w,b,\alpha)=-\frac{1}{2}|w|^2-\sum_{i=1}^{N}\alpha_i[y_i(w.x_i-b)-1]$.

Here, the Kuhn-Tucker conditions (KKT) should be stated [29]:

1) $\partial wLp=0\ ->w-\sum_i \alpha_i y_i x_i=0$

2) $\partial bLp=0\ ->\sum_i \alpha_i y_i=0$

3) $y_i(w.x-b)-1\geq 0$

4) $\alpha_i\geq 0$

5) $\alpha_i[y_i(w.x_i-b)-1]=0$

In the solution of the dual Lagrange, this KKT conditions have to be fulfilled. First two rules indicate that derivatives of the primal Lagrange, Lp, with respect to w and b should be equal to 0. After computing the derivative wrt w, what we end up with is the equation: $w-\sum_{i=1}^{N}\alpha_i y_i x_i = 0$ from which it follows that $w=\sum_{i=1}^{N}\alpha_i y_i x_i$; and the application of the derivation on b brings about the equation of $\sum_{i=1}^{N}\alpha_i y_i = 0$. After these values are plugged in the dual formulation, we end up with the following final problem:

Maximize $L_D(w,b,\alpha) = -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_i y_i y_j x_i.x_j + \sum_{i=1}^{N}\alpha_i$ subject to $\sum_{i=1}^{N}\alpha_i y_i = 0$ and $\alpha_i \geq 0, \forall_i$.

It can easily be observed here that the dual formulation of the problem again comes out to be a quadratic program; however now the problem only depends on x through the inner products $x_i.x_j$. Here it is also shown that there is a Lagrange multiplier for each of the training samples and the points for which $\alpha_i > 0$ are called the support vectors.

It should be noted down that the fifth constraint, called the “complementary slackness”, can be used in order to find a value for b.

7.1.1.2 The Non-Separable Case

In this case, the difference is the training data’s being non-separable optimally. Apart from the formulation of the original problem that was solved above, a further cost ξ for the violation of the constraints is taken into account together with a trade-off parameter C to account for/ to give weight to the trade-off for the penalty. Geometric interpretation of ξ is shown on Figure 14.

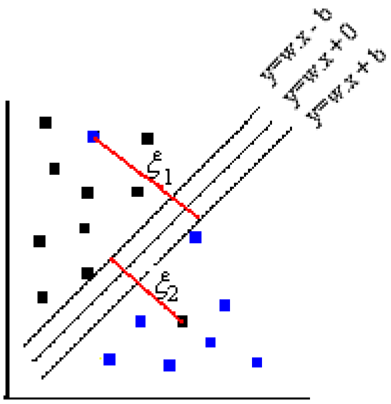


Figure 14. Support hyper-planes and cost values for a non-separable 2 class problem.

The new constraints are introduced in the following way:

$w.x+b \geq 1-\xi_i$ for $y_i = +1$ and $w.x+b \leq -1+\xi_i$ for $y_i = -1$; where $\xi_i \geq 0 \forall i$. These two constraints again can be combined as: $y_i(w.x-b)-1+\xi \geq 0,\ \xi_i \geq 0,\ \forall i$.

The resulting new problem follows:

Minimize $\frac{1}{2}|w|^2 + C\sum_i \xi_i$, subject to $y_i(w.x-b)-1+\xi \geq 0,\ \xi_i \geq 0,\ \forall i$.

The primal Lagrangian for this problem is now:

$$L(w,b,\xi,\alpha,\mu) = \frac{1}{2}|w|^2 + C\sum_i \xi i - \sum_{i=1}^{N} \alpha_i [y_i(w.x_i - b) - 1 + \xi i] - \sum_{i=1}^{N} \xi_i \mu_i$$

And the KKT conditions here are:

1) $\partial wLp = 0 \ -> w - \sum_i \alpha_i y_i x_i = 0$

2) $\partial bLp = 0 \ -> \sum_i \alpha_i y_i = 0$

3) $\partial \xi Lp = 0 \ -> C - \alpha_i - \mu_i = 0$

4) $y_i(w.x-b)-1+\xi_i \geq 0$

5) $\alpha_i \geq 0$

6) $\xi_i \geq 0$

7) $\mu_i \geq 0$

8) $\alpha_i[y_i(w.x_i - b) - 1 + \xi_i] = 0$

9) $\alpha_i \mu_i = 0$

By making use of the KKT conditions and duality, the problem is converted into the dual Lagrangian:

Maximize $L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j x_i . x_j$ subject to $\sum_i \alpha_i y_i = 0;\ 0 \leq \alpha_i \leq C\ \forall i$

It can be observed that the resulting problem is the same quadratic programming problem with the one on the separable case, however now that $\alpha_i = C - \mu_i$; $\mu_i \geq 0$; α_i obtains the form of $0 \leq \alpha_i \leq C$ [28]. Similar to the one on the separable case, the resulting problem again only depends on x through the inner products $x_i . x_j$.

It should also be emphasized that in both of the separable and non-separable cases, the classification function f is: $f(x) = w.x + b = \sum_i \alpha_i y_i x_i .x + b$

7.1.1.3 The Kernel Trick

One of the important characteristics of the Support Vector Machines is their ability to perform classification by constructing a hyper-*plane* that classifies the data optimally.

In the preceding parts, the final aim was to carry out the maximization of the dual Lagrangians having the form of $L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i .x_j$ subject to some constraints; and it was indicated that the problems were depending on x only through the inner products $x_i .x_j$. If the above described mapping of the data into a richer feature space is defined by $x -> \phi(x)$, the classification function, which was $f(x) = w.x + b = \sum_i \alpha_i y_i x_i .x + b$,

comes out to be equal to $f(x) = w.\phi(x) + b = \sum_i \alpha_i y_i \phi(x_i) \phi(x) + b$.

Here, the multiplication of $\phi(x_i)\phi(x)$ is defined as the kernel, $k(x_i, x)$. This can further be plugged into the Lagrangian equations to come up with the formulation of $L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j k(x_i .x_j)$.

Different kernels map differently distributed data (data originally having polynomial, exponential etc distribution shapes) into proper high dimensions on which data then can be separated optimally through using hyper-planes. Polynomial and Radial Basis Function (RBF) kernels are examples for some commonly used kernels. A polynomial Kernel is defined as $k(x, x') = (x.x'+1)^d$ and an RBF as $k(x, x') = \exp(-\gamma | x - x |^2)$. The Figure 15 and Figure 16 explain the concept of kernel mapping together with an example of an RBF map.

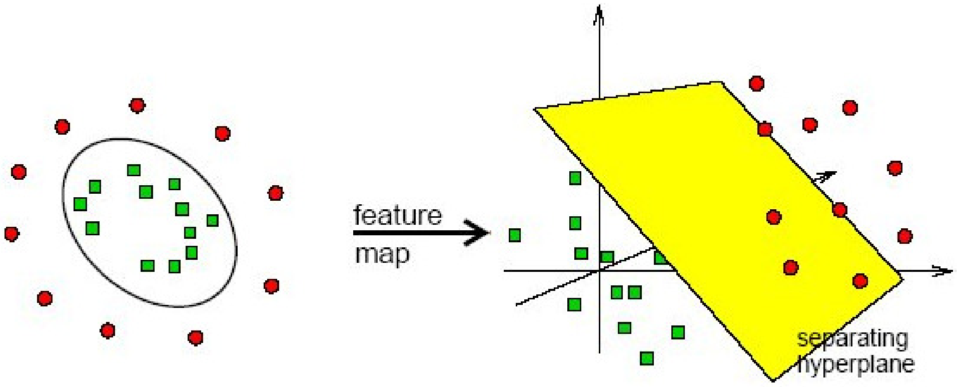


Figure 15. Feature mapping in SVMs [30]

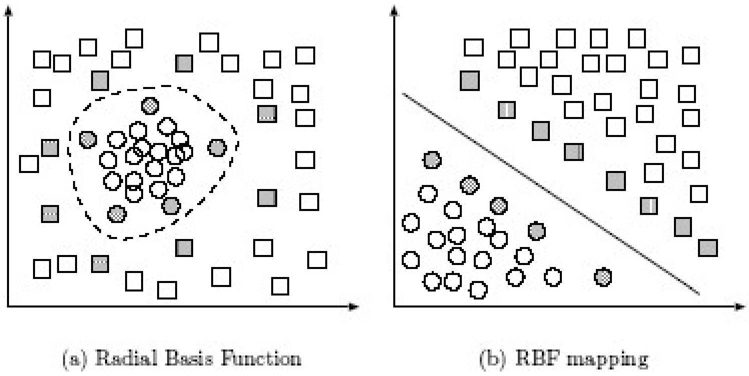


Figure 16. SVM kernel mapping: Left original space, right transformed feature space [30]

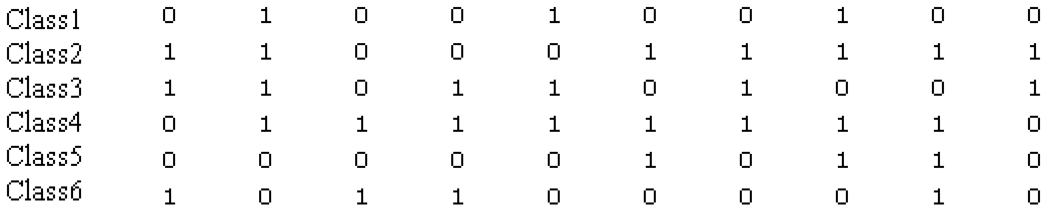
7.2 Multi Class Classification

Above described Support Vector Machines Method is mentioned to work on binary classification problems. There are some methods on which the 1vs. all binary classification schemes and outputs & weights of SVM's are used and combined in different ways such as majoty voting [31], to end up with multi-class classification results. However, in the multiple class classification part of the project, a different method, Error Correcting Output Codes (ECOC) is used. In the project, this method is made to combine the previously described Adaboost feature selection method and the binary SVM classification together with bootstrapping; although there are many combinations that can be used in future work.

7.2.1 Error Correcting Output Codes

The ECOC algorithm depends on the so-called distributed output code matrix. In this matrix, every row represents the unique codeword (a binary string of n) of a class that is a member of the multi-class problem. Therefore the number of rows (r) is equal to the number of classes, whereas the number of columns has experimental sizes depending on the number of classes / rows.

An example ECOC matrix for a 6 class problem with 10 columns can be examined on Figure 17.



Class1	0	1	0	0	1	0	0	1	0	0
Class2	1	1	0	0	0	1	1	1	1	1
Class3	1	1	0	1	1	0	1	0	0	1
Class4	0	1	1	1	1	1	1	1	1	0
Class5	0	0	0	0	0	1	0	1	1	0
Class6	1	0	1	1	0	0	0	0	1	0

Figure 17. An example ECOC matrix for a 6 class problem

Training stage: For each of the n columns, a binary classifier, which is SVM in the project, has been trained. The binary classification here can be interpreted in the following way: Within a single column, the rows having the value 1 are combined together to create a *superclass* and the same scheme applies to the rows having the value 0. Therefore, the patterns that are members of any of the classes having value 1 are trained as positive and the patterns that are members of any of the classes having value 0 are trained as negative examples in the binary SVM. This is repeated for each of the n columns and in the final stage we end up with n trained SVM functions.

Test Stage: To come up with the codeword of a new pattern, each of the n already trained SVMs are applied on it, and the resulting binary values are combined together to create the binary codeword of this test pattern. This codeword is then compared to each of the r codewords and the classification is done in favour of the class whose codeword has the closest distance, according to a distance measure to the codeword of the test pattern [32]. The usual way of measuring distance is through making use of Hamming distance. Hamming distance between a pair of same-length-codewords is defined as the number of positions for which the corresponding bit values are different.

One of the advantages of error correcting code is the method's ability to correct bit errors up to a degree. If the minimum Hamming distance between any pair of codewords is d, then at least $\lfloor d-1/2 \rfloor$ single bit errors can be corrected. For example, if the minimum Hamming distance is nine, and the total number of single bit errors between the test codeword and the true codeword is $\lfloor d-1/2 \rfloor = 4$, then the distance between the test codeword and the codeword that it is closest to just after the true one will be 9-4=5; and therefore the mapping will be done in favour of the true codeword.

Therefore, the design of the error correcting output code matrix is quite important for the classification results. As a result of the above mentioned bit error correction ability, the power of the ECOC depends highly on the row separation. So, each codeword should be well-separated in Hamming distance from each of the others [32]. Also, to create a good ECOC; the Hamming distance between a column and each of the others, and the distance between any column and the complement of each of the others should be large enough so that the deterministic learning functions per each column are uncorrelated from each other. The un-correlation brings about less correlated mistakes and therefore less simultaneous errors in individual bit positions. There are some methods such as Exhaustive Codes and Randomized Hill Climbing search techniques to end up with qualified ECOC matrices.

It has been stated that deterministic learning functions per each column had better be uncorrelated from each other so that we can end up with uncorrelated mistakes and therefore less simultaneous errors in individual bit positions. Due to the fact Support Vector Machines, which are being used in the project as the binary classifiers for each column, consist of deterministic learning functions, *bootstrapping* has also been applied in order to strengthen the un-correlation of the training data.

7.2.1.1 Bootstrapping

Bootstrapping depends on constructing a number of resamples of the training dataset through applying random sampling with replacement. That means, N many samples are randomly obtained from the whole original training dataset by replacement and finally these N samples become the new training set. Experimentally, 1/3 of the original data set is examined to be left out after bootstrapping has been applied.

This procedure is executed on the training sets of each *superclass* within runs of learning through columns. And as intended, less correlation between individual columns has been obtained.

7.2.1.2 Feature Extraction and Selection

The experiments for the multi-class classification part of the project has been applied on 32 by 32 gray scale normalized Cohn-Kanade dataset images and Gabor wavelets have been used as the features. As on a single image, there are 40960 Gabor features that are obtained by making use of 8 frequencies and 5 orientations, Adaboost as a feature selection algorithm has to be used before the binary Support Vector Machines classification is applied on each column. (It is called to work on the extracted Gabor features of the already bootstrapped data.)

7.2.1.3 ROC

Receiver operating characteristics (ROC) graphs are 2-D graphs that are used for visualizing, organizing and selecting classifiers according to their performance [33]. The Y axis of an ROC graph represents the *true positive* and the X axis represents the *false positive* rate of a classifier.

Namely, the True Positive Rate = Positives correctly classified / Total positives

the False Positive Rate (False Alarm Rate) = Negatives incorrectly classified / Total negatives.

A classifier which has got a performance point in an ROC space to the northwest of another is therefore considered a better classifier as it has got a higher true positive and a lower false positive rate.

ROC *curves* on the other hand are created through changing the value of a parameter that affects the decision of a classifier in a smooth manner (just like a decision threshold). The changing values of the parameter therefore create different points in the ROC space and these points are assumed to combine to make up a curve.

In the project ROC curves are used to determine the overall performance of some systems and are applied to in order to find out some parameters of the systems.

8 EXPERIMENTS

Various experiments have been carried out on the Cohn-Kanade frontal face database together with additional supplementary datasets from UCI Machine Learning Repository. The images in the Cohn-Kanade dataset have been coded using FACS, which describes subject's expression in terms of action units [34].

There are two main parts within this section: First part is giving information about the binary classification experiments, and the second part about the multi-class classification ones. Various experiments, aiming to compare different methodologies, have been applied on each part.

Section 8.1 includes the binary classification scheme experiments:

In section 8.1.1, comparison of the two Adaboost algorithms, Adaboost as a classifier without feature selection (Ada); and with feature selection (AdaFs) have been carried out. AdaFs, which is found to be giving out better classification results, has therefore been compared with AdaFs + SVM combination in 8.1.2. In AdaFs + SVM, features are firstly selected from the data by AdaFs and the classification itself is carried out by SVMs. Here AdaFs + SVM is observed to be better in terms of recognition rates.

In section 8.1.3, two other feature selection methods; Forward Feature Selection and Branch & Bound algorithm, combined with SVM classification are compared with the yet best system, AdaFs + SVM. Although revealing similar results to the Forward Feature selection method, AdaFs is selected to be the algorithm to be used in the further stages of the project as the feature selection method. This is because in some other work carried out on Facial Action Units Classification [12] [35] Adaboost is claimed to be giving out good results; and therefore combinations of it with not-yet-used and already-used feature extraction and classification techniques in literature are expected to provide good results in the end, for action unit classification.

Adaboost's being a good technique for feature selection might rely on the facts that: 1) It makes use of the classification information obtained from the classifier while doing the feature selection. 2) By boosting the classifier in each run, it ends up with the best discriminating feature. For example, in the project, while applying Forward Feature selection and Branch and Bound algorithms, just the rule of 1-nearast neighbour classification criteria is used to select features in each run. There doesn't exist a classifier which is boosted in each run and therefore which will supply us with better discriminating features. 3) There is experimental evidence that Adaboost as a classifier is working quite well on some similar research areas. [16]

Section 8.1.4 contains the repetition of the experiments, which gave out satisfactory results in UCI MLR, on the Cohn Kanade Action Unit coded frontal face dataset. Namely, the comparisons between Ada, AdaFs and AdaFs + SVM are carried out on the images which were normalized without the application of histogram processing. The Haar Wavelet coefficients are used as features in this section, and the number of selected features is also another parameter that is tested. Due to its high level of accuracy, AdaFs + SVM is selected to be the "1 vs. all" classification scheme for the AU recognition as a result. Although similar classification error rates come up after using 100 and 200 features, the number of features to be used in the system is determined to be 100, as the training and run times are decreased in a sensible amount.

In section 8.1.5, the selected system of AdaFs + SVM is this time tested on images, on which histogram processing was also applied. As a result of the positive affect of histogram processing, rest of the experiments are decided to be applied on histogram-processed-normalized images.

In order to see the effect of the image size on the overall performance of this system, experiments are made on the down-sampled, 16 by 16 images of AU 2 in section 8.1.6. The results for the new images do not come out as good and therefore down-sampling is discarded in the end.

Although the recognition rates acquired in the system of AdaFs + SVM applied on Haar Wavelet coefficients extraction are sucessful enough, new features are chosen to be used in section 8.1.7. The reasons that required trial of the extraction of these new features can be stated as follows:

1. Whitehill and Omlin's work [12] of the same system without feature selection (al-

though there is not enough information about the normalization part and the parameters used in the extraction and the classification parts) with Gabor wavelet features was expressed to be giving out quite unsatisfactory results. However, doing the implementation of the Gabor features and using them in the already implemented system together with feature selection could have improved the performance of the current system.

2. Though revealing good results, Haar Wavelet coefficients depend on the differences between some determined pixel intensities. Therefore, when applied to a real system consisting of taking a picture of a face and going through the steps in an automated way, the results came out to be of poorer quality than the ones obtained by applying the system on Cohn-Kanade dataset (which consisted of manually cropped and taken under similar lighting conditions images). Even 1-2 pixel shifts that occured due to non-robust localizations of the eye centres were found to be affecting the overall results in a sensible way; therefore some other feature extraction techniques had to be considered.

In order to avoid the above mentioned disadvantages of Haar Wavelet coefficients, Haar-like features; which implement the idea of using the change in the contrast values between adjacent rectangular groups of pixels, instead of the intensity values of individual pixels, are taken into consideration in section 8.1.7.1. Also these features' ability to get easily scaled and rotated is the second characteristic that makes them appealing to be used as the new features. Thirdly, it has to be indicated that these features are sensitive about the information that are carried on edges, boundaries and regions of motion on which the difference between the pixel values are important.

Surprisingly, the resulting classification rates here are not as successful. However, when the system with Haar Wavelet coefficients is applied *on real life applications*, the results also come out to be as unsatisfactory as the ones obtained by Haar-Like features applied *on the Cohn-Kanade set*. This also means that, the difference between the results of the system of Haar-like features applied on the Cohn-Kanade dataset and on a real life application is supposed to be small.

In Section 8.1.7.2, Gabor wavelets are used as the features. Results are noted for the usage of different parameters for the composition of wavelets. After obtaining satisfactory recognition rates through making use of 100 features, having fewer features (by using the same parameters) is also tested. This is because it would be better to use as few features as possible due to the restrictions in the training time and testing run time of SVM.

It has been concluded that both the number of features used and also the parameters selected for Gabor wavelets play important roles in the final classification results, and 100 features are giving out the optimal results together with the indicated / detected wavelet parameters. Experiments are also tested on 16 by 16 down-sampled images, and also on 1/10th down-sampled feature space; and the classification rates are not as good. However, the down-sampled images or features might be used in real-time applications where speed is of big importance.

The best results of the binary classification experiments are obtained through the final system of normalization with histogram processing on 32 by 32 images + Gabor Wavelet coefficients extraction + AdaFs + SVM.

Detailed discussion about the performance of Gabor Wavelets can be found on section 8.1.7.2.1.

Finally, section 8.1.7.2.2 gives information about the ROC graphs that are used to select the regularization parameter, c, of the SVM's to be used in the system and the real life application.

Section 8.2 includes the multi-class classification scheme experiments:

In this part of the experiments, in order to carry out multi-class classification of the Action Units, Error Correcting Output Codes (ECOC) are used. As the system of Gabor features extraction + feature selection by AdaFs + SVM classification was found to be giving out the best binary classification results in Section 8.1, the components of this system are inherited to be used in ECOC. Namely, as was explained in detail in Section 7.2.1, the strategy used in here consists of applying bootstrapping for each *superclass* per column; followed by accomplishing feature selection by Adaboost, and using Support Vector Machines for training and classification (per each column again). As mentioned, Gabor Wavelet coefficients are the features used.

Set of experiments are applied on three different sets of data, one having 12, the other 4 and the last one having 3 classes.

In Section 8.2.1, all the possible combinations of the upper face Action Units that exist on the Cohn-Kanade dataset images are used to form up 12 classes. The classes are created to be mutually exclusive. Another constraint is the minimum number of data each class has, which is 5.

As dealing with a 12 class problem is a complicated issue whose results never appear to be very satisfactory, the results found here are poor. Therefore, the problem is made to be divided into a smaller one. However, before that, 1 vs. all - binary classification results are also examined for AU1, AU2, AU4, AU5, AU6 and AU7 within this section. For a specific Action Unit, this is accomplished by checking all the patterns including this unit (either as single or in a combination), and by calling a mapping to be correct, if any checked pattern is mapped to a class which includes that Action Unit.

Apart from the fact that dealing with a 12 class problem in ECOC is a complex issue, the effect of the number of the training patterns on classification should also be considered. It can be examined that in the 12 class problem the best results are obtained by Class 11 and Class 3, and that these classes are ones with the maximum number of patterns. Also, the worst results are obtained by Class 1 and Class 10, which have only 7 and 6 patterns inside. Therefore, in section 8.2.2, the classification task is experimented on the 4 classes containing the maximum number of patterns. Namely: Class 11, Class 3, Class 7 and Class 2.

In section 8.2.2, the error rates obtained are better than the ones of the 12 class case. However, main reasons for some misclassifications are interpreted to come out due to the low number of patterns in Class 2, and to the fact that two of the classes; Class 2 and Class 7, contain the same action unit, Action Unit 7, which might cause confusion. Because of this, the experiments in section 8.2.3 are decided to be applied on the 3 class problem, which excludes Class 2.

The important factors leading to the decrease in the overall classification error rates of Class 3 and Class 7 in the 3-class-case experiments can be listed as follows:

1. The classes in this last run all have exclusive Action Units. To be precise, one might observe that no single Action Unit occurs in two classes at the same time.
2. The number of classes is decreased.
3. The classes having the maximum number of patterns are worked out. The least number of patterns in the 3-class case is 47, while it is 5 in the 12-class case.

Finally in Section 8.2.4, detailed discussion about ECOC can be found.

8.1 Binary Classification Case

First section of this chapter is devoted to the experiments carried out for the “1 vs. all” classification scheme, starting with applications to datasets on UCI Machine Learning Repository, and followed by the usage of Cohn-Kanade Action Unit coded dataset.

8.1.1 Comparisons of Two Adaboost Algorithms on UCI Machine Learning Repository Datasets

In order to compare the performances of Ada and AdaFs; smaller datasets like Breast Cancer Wisconsin (Breast-w), Sonar Mines vs. Rocks (Sonar), and Pima Indians Diabetes (Diabetes) which were taken from the UCI Machine Learning Repository have been used. The Breast Cancer Wisconsin dataset consists of 699 data with 10 features whereas the Sonar, Mines vs. Rocks data set consists of 208 data with 10 features, and the Pima Indians Diabetes of 768 data with 8 features.

In this experiment, the number of features chosen by AdaFs was half of the number of total features within each dataset. Table 1 indicates the error percentage results obtained through 10-fold cross validation.

	AdaFs	Ada
Breast-w	4.6 (4 feat.s out of 9 selected)	4.09
Diabetes	26.78 (4 feat.s out of 8 selected)	28.18
Sonar	14.76 (30 feat.s out of 60 selected)	19.62

Table 1. Recognition error rates of AdaFs vs. Ada, applied on UCI MLR

Both in the training and test sets of Diabetes and Sonar, error rates were lower when AdaFs was used instead of Ada, whereas both of the rates for Breast-w dataset were quite close to each other. Due to these rates and also its ability to carry out feature selection at the same time with classification, AdaFs was considered to be superior to Ada.

8.1.2 Comparison of AdaFs with AdaFs + SVM, on UCI MLR Datasets

For SVM, Radial basis and polynomial type of kernels were used, and the ones for which the least error rates were obtained, were noted down. The Breast Cancer-w and the Sonar datasets were observed to be better separated when projected into a higher dimensional space with RBF kernels, whereas the Diabetes dataset was linearly better separated. It was also observed that changing the regularization parameter ‘c’ between 0.1 and 1 affected the classification rates between 1 and 3 percent on the three datasets. The values of c in Table 2 are the ones which revealed the lowest overall error percentages.

10 fold cross validation results on Table 2 show that AdaFs + SVM revealed 5-10% better classification rates than AdaFs; however, time trade-off should as well be considered according to the dataset type, as the second method came out to be slower in cases of large datasets.

	AdaFs.	AdaFs + SVM
Breast-w	4.6 (4 feat.s out of 9 selected)	3.5 (3 / 9 feats, radial basis SVM c=0.7)
Diabetes	26.78 (4 feat.s out of 8 selected)	22.53 (4 / 8 feats, polynomial d=1 SVM c=0.7)
Sonar	14.76 (30 feat.s out of 60 selected)	11.06 (30 / 60 feats, radial basis SVM c=0.6)

Table 2. Recognition error rates of AdaFs vs. AdaFs+SVM, applied on UCI MLR

8.1.3 Comparisons of Feature Selection Techniques that are Combined with SVM Classification

Table 3 shows the comparison of the three feature selection methods, Feature Forward Selection AdaFs and Branch & Bound algorithms, all combined with Support Vector Machines classification. One of the important observations comes out to be the number of features’, which gave out the best classification results, being the same in all of the feature selection methods.

The kernels that were used in SVM for the least error classification results vary per each data set and also per feature selection method. For example, the linear separation of the Diabetes dataset was giving out the best recognition results in cases of AdaFs and Branch & Bound feature selection methods; but in case of Forward Feature Selection, it was best separated by a polynomial degree two kernel. Also, Breast-w and Sonar datasets were better separated when radial basis kernels were used, whereas this wass not the case for Diabetes, as indicated.

Both in the Forward Feature Selection and the Brach & Bound feature selections, 1-Nearest Neighbor leave-one-out classification performance has been used (The object is simply assigned to the class of its nearest neighbour by making use of Euclidian distance).

Forward Feature Selection + SVM		
Breast-w	3.22	(4 feat.s out of 9, radial basis SVM c=0.8)
Diabetes	22.53	(4 / 8 feats, polynomial deg.2 SVM c=0.8)
Sonar	11.54	(30 feat.s out of 60, radial basis SVM c=0.3)
AdaFs + SVM		
Breast-w	3.5	(3 / 9 feats, radial basis SVM c=0.7)
Diabetes	22.53	(4 / 8 feats, polynomial deg.1 SVM c=0.7)
Sonar	11.06	(30 / 60 feats, radial basis SVM c=0.6)
Branch & Bound + SVM		
Breast-w	3.51	(4 feat.s out of 9, radial basis SVM c=0.8)
Diabetes	25.91	(4 feat.s out of 8, polynomial deg.1 SVM c=0.3)
Sonar	11.06	(30 feat.s out of 60, radial basis SVM c=0.4)

Table 3. Recognition error rates of Three Different Feature Selection Methods + SVM

As it can be analyzed via Table 3, AdaFs and Forward Feature selection methods were giving out slightly superior results than Branch and Bound algorithm in the three datasets experimented.

8.1.4 Classification Technique Selection on the Cohn-Kanade Dataset

So as to apply the comparisons of techniques on the “1 vs. all” classification scheme of the Action Units, the Haar Wavelet Coefficients of the Upper Action Units AU1 and AU2 were extracted and used as features.

The experiments on the Cohn-Kanade dataset were carried out on the cropped “right eye” image regions, and all the results were obtained by applying 10-fold cross validation on the dataset consisting of 32 by 32 normalized gray scale images. Here, only the first two parts of the normalization process; namely 5.1 Eye Region Crop & Pupil Localization, and 5.2 Rotation & Size Normalizations, have been applied to the images on the training and test sets. Histogram Processing is used on the following sections and its affects on the classification is measured then.

It should be indicated that, 10-fold cross validation has not been applied on the feature selection parts of these binary classification experiments. Due to performance and timing constraints, it has been assumed that the features selected from one single run will remain mostly the same for the consecutive runs of the 10-fold cross validation applied on the whole system. Validation on SVM has been applied properly.

In the feature selection part, 100 and 200 out of 32x32=1024 features were selected (10%, 20%). In SVM classification, regularization parameter c was constrained to be such that: 0.0030 < c < 0.0040; and the kernel used was of polynomial degree 1, due to the fact that these parameters were supplying us with the least overall error rates for SVM.

-100 features out of 1024 Haar Wavelet Coefficients were selected by feature selection on AdaFs:

	Ada	AdaFs	AdaFs + SVM
AU1	17.11	12.47	10.81
AU2	6.74	6.83	5.26

-200 features out of 1024 Haar Wavelet Coefficients were selected by feature selection on AdaFs:

	Ada	AdaFs	AdaFs + SVM
AU1	17.11	12.11	13.1
AU2	6.74	6.26	5.6

Table 4. Recognition error rates obtained by 3 classification methods on Cohn-Kanade Dataset

As Adaboost classification without feature selection was detected to have a weaker performance than AdaFs and AdaFs + SVM in both 100 and 200 features cases; only these two methods were tried on the entire set of upper face action units: AU1, AU2, AU4, AU5, AU6 and AU7; for 100 and 200 features. Results can be found on Table 5.

-100 features out of 1024 Haar Wavelet Coefficients were selected by feature selection on AdaFs

	AdaFs	AdaFs + SVM
AU1	12.47	10.81
AU2	6.83	5.26
AU4	16.87	11.85
AU5	6.51	5.2
AU6	16.3	11.43
AU7	15.58	11.78

-200 features out of 1024 Haar Wavelet Coefficients were selected by feature selection on AdaFs

	AdaFs	AdaFs + SVM
AU1	12.11	13.1
AU2	6.26	5.6
AU4	15.19	13.51
AU5	6.6	5.1
AU6	15.12	11.23
AU7	14.38	12.68

Table 5. Recognition error rates for AdaFs vs AdaFs + SVM on Cohn- Kanade Dataset

In both of cases where 100 and 200 features were used, Support Vector Machines classification was observed to perform better than Adaboost classification 95% of the time.

Also, in order to check if 300 features would give such an increase in the classification performance that the trade off between complexity and accuracy would be worth having, the final experiment was repeated for 300 features for AU1. In Table 6, the classification results for AdaFs and AdaFs + SVM can be examined for 100, 200 and 300 features for AU1.

	100 features	200 features	300 features
AdaFs	12.47	12.11	12.9
AdaFs + SVM	10.81	13.1	13.9

Table 6. Recognition error rates for AdaFs and AdaFs+SVM, for 100, 200 and 300 features

It can be observed that using 300 features caused a decrease rather than an increase in the classification performance.

8.1.5 Effect of Histogram Processing on Classification

The effect of histogram processing on the performance of the system has been recorded for AU1, AU2 and AU7 in this section, on the AdaFs + SVM system with 100 selected features.

	AdaFs + SVM on images which were normalized without histogram processing	AdaFs + SVM on images which were normalized with histogram processing
AU1	10.81	9.0
AU2	5.26	4.5
AU4	11.85	12.00
AU5	5.2	3.12
AU6	11.43	11.6
AU7	11.78	12.9

Table 7. Effect of histogram processing on recognition error rates

From Table 7, the overall effect of the second step of normalization, histogram processing, can be stated to be positive on average, although for some AU’s, about 1% decrease might be observed.

8.1.6 Effect of Image Size on AdaFs + SVM Classification, on the Normalized & Histogram Processed Images, via Using Haar Wavelet Features

Although the recognition error rate for AU2 on 32 by 32 images was 4.5%, the new rate on 16 by 16 images appears to be equal to 11.2% which is 2.5 times as much. The decrease in the accuracy might be commented to come out as a result of loss in important detail in the images due down-sampling.

8.1.7 Other Feature Extraction Techniques Used in the System:

The comparison of Whitehill and Omlin's results with the AdaFs + SVM system applied on Haar Wavelet coefficients extraction is given in Table 8.

	Gabor coefficients extraction + SVM, without feature selection (Whitehill and Omlin's system)	Haar wavelet coefficients extraction + AdaFs + SVM
AU1	22.01	9.0
AU2	11.71	4.5
AU4	13.35	12.0
AU5	5.92	3.12
AU6	12.2	11.6
AU7	6.14	12.9

Table 8. Comparison of the recognition error rates obtained by Whitehill & Omlin and the current system

8.1.7.1 Extracting and Using Haar-Like Coefficients in the System:

While extracting the Haar-Like coefficients on 32 by 32 normalized and gray scaled images, the width and the height of the smallest rectangles used in the implementation of Haar-Like features (refer to section 4.2) were determined to be one fifth of the width / height of the image. This was in order to prevent having features that were obtained from regions which were as small as 1-2 pixels and therefore prevent having pixel differences as features that wouldn't be robust to image shifts etc.

The overall classification results of the system consisting of extraction of Haar-Like features + AdaFs + SVM applied on AU1, AU2 and AU7 can be examined through Table 9. For convenience, the previous results obtained by the system using Haar Wavelet coefficients are also indicated in the table.

It should be indicated that these are the least error results which were obtained by making use of Support Vector Machines with a polynomial degree 1 kernel and having c such that: $0.0030 < c < 0.0045$. It has been detected that in case of the usage of radial basis functions as kernels in SVM, the classifier becomes unable to do the classification properly and assigns all the data into a single class, all either being 0 or 1.

	Haar-Like coefficients extraction + AdaFs + SVM	Haar wavelet coefficients extraction + AdaFs + SVM
AU1	17.05	9
AU2	10.5	4.5
AU7	19.13	12.9

Table 9. Comparison of the error rates of the systems with Haar-Like and Haar wavelet coefficients

The results are unsatisfactory compared to the ones found when Haar Wavelet coefficients were used.

8.1.7.2 Extracting and Using Gabor Wavelet Coefficients in the System:

While extracting the Gabor features, there are many parameters that have to be carefully worked out (refer to section 4.3). To summarize, the Gabor Wavelet function which is equal to $g(x,y)=K\exp(-\pi(a^2(x-x_0)_r{}^2+b^2(y-y_0)_r{}^2))\exp(j(2\pi(u_0x+v_0y)+P))$ in spatial domain and to $g(x,y)=K\exp(-\pi(a^2(x-x_0)_r{}^2+b^2(y-y_0)_r{}^2))\exp(j(2\pi F_0(x\cos w_0+y\sin w_0)+P))$ in polar coordinates, had the following parameters:

K : The scale parameter of the magnitude of the Gaussian envelope.

a,b : The scale parameters of the x and y axis of the Gaussian envelope.

θ: Parameter for the rotation angle of the Gaussian envelope.

(x_0,y_0) : Peak value of the Gaussian envelope occurs at (x_0,y_0)

(u_0,v_0) : Sinusoidal carrier's Cartesian spatial frequencies. Is equal to (F_0,w_0).

P : Phase of the sinusoidal carrier

For facial expression recognition and also face recognition, the most common used values of the phase were: $\pi\frac{i}{8};i=0,...,7$ and therefore these values have been used in the project. In case of frequencies F , it was mentioned that the upper limit due the Nyquist theory would be 0.5 Hertz and having $\sqrt{2}$ as the spacing between the kernels in the frequency domain has been indicated to be giving out experimentally good results on previous research. Therefore the values for F were taken as: $\frac{f\max}{\sqrt{2}^{u}}, u = 0,...,4$ where $f\max = 0.5$. Also, as explained in section 5.3.1, it has been considered in the experiments that $w_0 = \theta$.

The parameters left out to be depended on trials are now the scale parameters of the x and y axis of the Gaussian envelope a, b and the size of the filter.

Table 10 shows the classification error rates of the system depending on Gabor features with parameters $a = 1/50, b = 1/40$ and the width and the height of the filter size was equal to the half of the width and height of the image to be filtered (Table also shows the previous results obtained after the extraction of Haar Wavelet coefficients as features for comparison issues). These parameters were the ones revealing the best classification results together with SVM having a polynomial degree 1 kernel and regularization parameter c: 0.0030 < c < 0.0040.

	Gabor wavelet coefficients extraction + AdaFs + SVM	Haar wavelet coefficients extraction + AdaFs + SVM
AU1	5.20	9
AU2	3.3	4.5
AU4	8.11	12.0
AU5	2.49	3.12
AU6	8.32	11.6
AU7	8.52	12.9

Table 10. Recognition error rates obtained by Gabor and Haar wavelet coefficients + AdaFs + SVM

Table 11 indicates the results achieved after using only 7 features; and though less accurate, they can be used in applications in which increased speed is required as a trade off for accuracy.

	Gabor wavelet coefficients extraction + AdaFs + SVM for 7 features	Gabor wavelet coefficients extraction + AdaFs + SVM classification for 100 features
AU1	10.74	5.20
AU2	4.71	3.3
AU4	13.75	8.11
AU5	6.3	2.49
AU6	19.5	8.32
AU7	12.75	8.52

Table 11. Difference between the error rates, for 7 and 100 selected Gabor features + SVM

Also for convenience, Table 12 shows the classification error rates for AU2 through making use of different parameters and again by having 100 and 7 features. It can be seen that these results are no better than the ones that were found by using the original parameters (right most column).

for AU2	a=1/2 b=1/2 filter size = 1/8th of the image size	a=1/40 b=1/40 filter size = same as the image size	a=1/50 b=1/40 filter size = half of the image size (original parameters)
100 feat.s	9.98	10.60	3.3
7 feat.s	12.68	16.63	4.71

Table 12.Various error rates for AU2, by changing parameters of the system.

Next, in order to see the effect of the size of images on the overall performance of the same system, AU 2 images which were down-sampled to 16 by 16 size were used. Although the previous recognition error rate was 3.3% for AU2, the new rate came out to be equal to 4.5% when the same system parameters are used. Although it is not a huge one, the decrease in the accuracy might again be considered to appear as a result of loss in important detail in the images due down-sampling.

And lastly, so as to decrease the training time, every 1 out of 10 Gabor wavelet coefficients of AU 2 were taken and the new sub-sampled feature space was given to the Adaboost classifier, which was to select 100 features from that space. As a result, it was examined that there was a loss about 2% on the recognition rates.

8.1.7.2.1 Some Comments about the Final System with Gabor Wavelets Extraction

-It was surprising to see that classification of Gabor features revealed better results than Whitehill and Omlin's results [Table 8]. This might be due to the reasons that 1.) They haven't exploited feature selection whereas here it was done so. 2.) The normalization step and the parameters they have used might differ in various ways than the ones used in this project.

In order to see if the first reason was important, SVM was applied on AU2 Gabor wavelets without feature selection and an error rate of 25% was found. Consequently, we see that there are some redundant features which decrease the actual classification rate that can be eliminated via using a number of selected features, and Omlin and Whitehill's method is lacking this. Also, selecting features supplies us with a great decrease in the training and run times.

-The results obtained were not only better than ones Whitehill and Omlin got, but also than the results of the previous systems consisting of Haar Wavelet Coefficients. This might be due to the fact that Gabor wavelets have better characteristics such as being more robust to pixel intensity and lighting changes than Haar Wavelets (Haar wavelets were discussed with respect to lacking these characteristics in the preceding sections).

Gabor wavelets are formed up by a convolution of a Gabor filter and an input image, and therefore they are also meaningful in the frequency domain. Convolution in the time domain is equal to multiplication in the frequency domain and the peak response of the Fourier transform of the Gabor filter is at the spatial frequency (u, v) of the complex sinusoidal. (u is taken to be equal to v most of the time.)

It was observed for all of the Action Units that the first 10 out of 100 features used for classification, have been extracted by using Gabor filters with mostly low complex sinusoidal frequencies. In other words, the percentage of the number of high frequencies used in the classification over the low ones is equal to 1/3. All the possible values of F used in the extraction were 0.125, 0.176, 0.25, 0.35 and 0.5 in ascending order ($F = \frac{f\max}{\sqrt{2}^u}, u = 0, \ldots, 4$ where $f\max$ = 0.5); and the selected features were examined to have the magnitude of frequency, F, equal to 0.125 or 0.176 in their Gabor filter's complex sinusoidal most of the time. As the peak response of the Gabor filter's Fourier transform is on and around the spatial coordinates of F, the smaller F is, the more the filter behaves like a low pass filter and vice versa. The mostly used low frequency components in the classification is therefore an indication of the fact that useful frequency bands for facial action units' recognition are the low frequency bands in images. However, here future work can take place to reveal why Gabor features do not use much edge information.

One possible explanation might be that although convolution is taken into account, the edge information might vary quite a lot in each image even if they belong to the same class; and therefore this information may not be useful, and maybe redundant to be used by SVM in the classification. It should also be indicated that, although not as much, high frequency components were also used in the classification (one third of the time).

-Convolution with a filter having size of 1/5 of the original image is one of the key ideas that increase the robustness of Gabor features to shifts and variances, as we are not making use of the sindividual pixel values from now on.

-Increased robustness of Gabor features (compared to Haar Wavelet coefficients) to shifts and lighting variations shows its positive effects when the real life application is taken into consideration. Gabor features are examined to give out better results than Haar features in this application that consists of taking the photo of a test face, applying normalization, extraction, selection and classification tasks all in automated ways.

-Although in the binary classification case the speed was not taken into consideration while deciding about the facial extraction method to use, the complexity in the training time of Gabor wavelet coefficients should be noted. Whitehill and Omlin showed in their previous research that [12], extraction of Gabor wavelet coefficients is 300 times as much as the Haar ones. This is due to the facts that the extraction of Gabor wavelet coefficients needs the convolution process, and that there are 10960 features extracted from a 32 by 32 images, whereas there are only 1024 in case of Haar.

-In Table 13, results of 3 more systems that have proposed solutions to the Action Units classification are shown, together with the Whitehill and Omlin's method and the system that was implemented in the project. Although all of the systems differ in many ways, this table is presented just to give an overall view of the literature.

	Research of Barlett, Littlewort, et al [36] Gabor wavelet extraction + Ada (no feat. selection) *On Cohn-Kanade Dataset*	Research of Barlett, Littlewort, et al [36] Gabor wavelet extraction + SVM (no feat. selection) *On Cohn-Kanade Dataset*	Research of Kaapor, Qi and Picard [10] Tracking of fiducial points & shape parameters in templates + SVM *On CMU Dataset*	Research of Whitehill and Omlin [12] Gabor wavelet extraction + SVM (no feat. selection) *On Cohn-Kanade Dataset*	Research of Whitehill and Omlin [12] Haar wavelet extraction + AdaFs (feature selection with classification) *On Cohn-Kanade Dataset*	System Implemented in the Project Gabor wavelet extraction + Adaboost Fs + SVM *On Cohn-Kanade Dataset*
AU1	8	7	27	22.01	17.17	5.20
AU2	12	4	7.4	11.71	6.74	3.3
AU4	11	11	19	13.35	14.77	8.11
AU5	8	8	33.3	5.92	5.61	2.49
AU6	7	6	N/A	12.2	6.61	8.32
AU7	12	13	0	6.14	11.69	8.52

Table 13. Upper face action unit recognition error rates obtained by some researchers

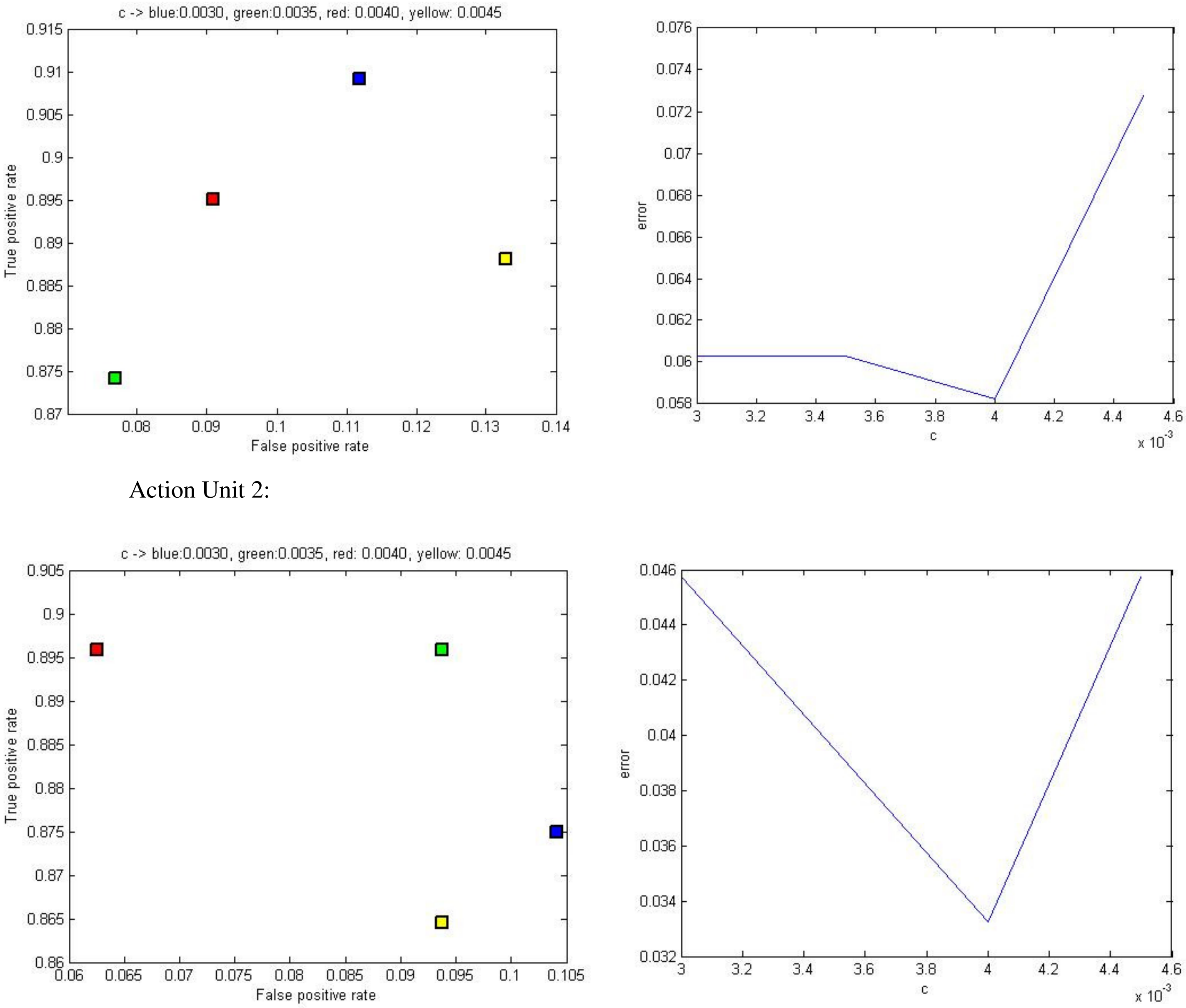
8.1.7.2.2 ROC Graphs

Although Receiver Operating Characteristic (ROC) graphs can be used to visualize and select classifiers according to their performance and should be examined in each system for each classifier separately, this project mainly used them to decide about the regularization parameter, c, of the Support Vector Machines classifier for the real life application. The real life application is implemented to decide if a specific Action Unit is on or off on any given image, and is making use of the best final system which consists of Gabor feature extraction, Adaboost feature selection and Support Vector machines classification.

It has been found through the accomplished experiments that, having the c value between 0.0030 and 0.0045 is supplying us with the least classification errors. However, apart from the error rates, in order to see the performance of SVM from a different angle and to decide which c value to use in the implementation of the application, true and false positive rates are examined on ROC graphs in this section.

Figure 18 shows the ROC graphs of SVM's for Action Unit 1 and Action Unit 2 obtained by thresholding the regularization parameter 'c', and the corresponding error rate graphs. The parameters used for the whole system are the ones from which the best classification results in Figure 18 were achieved.

Action Unit 1:



Action Unit 2:

Figure 18. ROC graphs for Action Unit 2, obtained by thresholding the regularization parameter, c.

It was indicated that a classifier which has got a performance point in an ROC space to the northwest of another is considered a better classifier as it has got a higher true positive and a lower false positive rate. When this fact together with the error rates is taken into account, regularization parameter c's value being equal to 0.0040 can be considered as optimum. Therefore, this value was used in the implemented application.

Although the overall error percentages supply us with important information about a classifier, the true and false positive rates should always be examined and interpreted as well. The design of specific systems needing to meet requirements like having the true positive rate above a certain threshold etc should therefore rely on ROC graphs.

Another thing to note here is that, although the boundary of c was changed smoothly, we didn't end up with a smooth ROC *curve*. The points were rather individual and seemed to be independent; the reasons for these can be addressed in future work.

8.2 Multi-class Classification

In the second section of Chapter 8, multi-class classification experiments applied on the Cohn-Kanade Action Unit coded dataset are presented. On all the experiments carried out in the following parts, 10-fold cross validation is used for both the feature selection and the classification parts.

8.2.1 12 class case:

When 12 classes were first taken into account, it was known experimentally that selection of the number of columns between 50 and 100 was revealing satisfactory results; that's why 'n', the number of columns, was selected to be 75. In case large number of columns such as 75, there is theoretical and experimental evidence that the ECOC capability can be approximated by creating the codeword bits in a randomized manner (creating binary 1 by 50% and binary 0 by 50% chances per bit) to the capability of one that is created by Hill Climbing. Therefore, this is the method that was used in the project.

While creating the code bits randomly, the checks for the Hamming-distance-between-any-rows' being at least 30, and on average $\lfloor 75/2 \rfloor = 37$ were done; so that at least $\lfloor 30-1/2 \rfloor = 14$ and on average about $\lfloor 37-1/2 \rfloor = 18$ bit errors out of 75 bits could be corrected on the test run. This gave out about 20% error correction. Also, checks on columns, such as every new column's being different than all zeros / all ones / all the previous columns and their complements, were carried out.

Table 14 illustrates the classes, which are combinations of more than one upper face Action Units appearing at the same time, or which just consist of a single upper face Action Unit. It also shows the number of data contained in each class.

Class Number	Action Units On	Number of Data
1	Action Unit 1	7
2	Action Unit 4	26
3	Action Unit 6	65
4	Action Unit 7	5
5	Action Unit 1&2	23
6	Action Unit 1&4	20
7	Action Unit 4&7	47
8	Action Unit 6 & 7	13
9	Action Unit 1 & 4 & 7	11
10	Action Unit 1 & 2 & 4	6
11	Action Unit 1 & 2 & 5	62
12	Action Unit 4 & 6 & 7	22

Table 14. 12 ECOC classes

The parameters for Adaboost feature selections and Support Vector Machines classifications, which were applied on each column for binary classification of superclasses, were taken from the most successful "1 vs. all" binary classification scheme that was found in Section 8.1. Within that scheme:

1. Gabor features were extracted with the parameters $\theta = \pi \frac{i}{8}; i = 0,...,7$; $F = \frac{0.5}{\sqrt{2}^u}, u = 0,...,4$; $w_0 = \theta$; $a = 1/50, b = 1/40$

2. Adaboost was applied to select 100 features.

3. SVM was used to with a polynomial degree 1 kernel and the regularization parameter c being such that: 0.0030 < c < 0.0040

10-fold cross validation applied contained both the feature selection and the classification parts. However, in order to decrease the time needed for training, every 1 out of 10 Gabor features from a total of 40960 was taken to form the new feature set, to which Adaboost was to be applied in order to select 100 features.

The overall recognition/classification error rates of the multiclass task a result of the first 8 runs of 10-fold cross validation are given in Table 15. (Validation could not be completed due timing restrictions)

Class Number	Action Units On	Number of Data	Overall Classification Error
1	Action Unit 1	7	100
2	Action Unit 4	26	80
3	Action Unit 6	65	25
4	Action Unit 7	5	80
5	Action Unit 1&2	23	64
6	Action Unit 1&4	20	28
7	Action Unit 4&7	47	34
8	Action Unit 6 & 7	13	90
9	Action Unit 1 & 4 & 7	11	87
10	Action Unit 1 & 2 & 4	6	100
11	Action Unit 1 & 2 & 5	62	15
12	Action Unit 4 & 6 & 7	22	95

Table 15. Recognition error rates obtained by ECOC with 12 classes

It can be seen that the overall error rates came out to be quite unsatisfactory.

The 1 vs. all- binary classification results for AU1, AU2, AU4, AU5, AU6 and AU7 can be found on Table 16.

	Binary Classification Error Rates
AU1	21
AU2	16
AU4	24
AU5	15
AU6	23
AU7	36

Table 16. Binary classification errors for the upper face action units, by using ECOC

As expected, the results here appeared to be better than the ones for the multi-class problem, but were still not quite fine.

8.2.2 4 class case:

In case of 4 and 3 (next section) classes, exhaustive codes, which mean code matrices consisting of all possible column combinations without having identical or complementary ones, were used. Code length in exhaustive codes is equal to $2^{k-1}-1$ where k is the class number. In those problems, although the column separations were well obtained, the inter-row Hamming distance came out to be as small as 2 or 3 due small number of columns and therefore the number of single bit errors that the code could correct had the supremum which was only either $\lfloor 2-1/2 \rfloor = 0$ or $\lfloor 3-1/2 \rfloor = 1$.

The results again obtained from the first 8 runs of 10-fold cross validation for the 4 class problem in shown on Table 17.

Class Number	Action Units On	Number of Data	Overall Classification Error in the 12 Class Problem	Overall Classification Error in the 4 Class Problem
2	Action Unit 4	26	80	47
3	Action Unit 6	65	25	19
7	Action Unit 4&7	47	34	51
11	Action Unit 1 & 2 & 5	62	15	13

Table 17. Recognition error rates for the 4 class problem, by using ECOC

As expected, the overall classification errors of the 4 class problem were better than the ones of the 12 class one with an exception for Class 7.

8.2.3 3 class case:

First 8 runs of 10-fold cross validation for the 3 class problem, which consisted of the same classes of the 4 class case without Class 2, revealed the results shown on Table 18.

Class Number	Action Units On	Number of Data	Overall Classification Errors in the 12 Class Problem	Overall Classification Error in the 4 Class Problem	Overall Classification Error in the 3 Class Problem
3	Action Unit 6	65	25	19	3.5
7	Action Unit 4&7	47	34	51	17
11	Action Unit 1 & 2 & 5	62	15	13	18

Table 18. Recognition error rates for the 12, 4 and 3 class problems, by using ECOC

Here, the significant decrease in the overall classification error rates of Class 3 and Class 7 might be observed; while there was an increase in the error rate of Class 11 although not quite a lot.

8.2.4 Further Discussion About the Multi-Class Case

As a conclusion for the multi-class classification problem of the project, it should be indicated that not many methods have been tried on ECOC and as a future work; different combinations of different methods might be used and compared. Although the results obtained in the project for the multi class case were not as accurate as the ones obtained in the binary classification part, the sensible amount of increase in speed when ECOC was considered should as well be appreciated. There are also some other techniques in literature that are used to end up with multi-class decisions, by making use of the binary classifiers. They depend on different methods, such as applying all the binary classifiers to the data and using majority voting or thresholding [31], or treating each class combination as a different class and doing the training and testing as well. However it should again be noted that hardly any of them can have the speed of ECOC. Therefore, works on improving the accuracy of ECOC will have considerable importance as the aim is to end up with classification results which are both accurate and less complex in terms in timing and computation.

9 CONCLUSION

9.1 Summary

In this project, in order to classify and recognize facial expression, Facial Action Coding System has been used, and binary and multi-class classifications of face Action Units through using different methods were accomplished. Some experiments firstly used small datasets taken from UCI Machine Learning Repository in order to decide on the feature selection and classification methods. The methods that were revealing the best results were then applied on the Cohn-Kanade Action Unit coded dataset. Extraction of different features such as Haar Wavelet coefficients, Gabor wavelet coefficients and Haar-like features were used to test the accuracy of the classification methods on the Cohn-Kanade dataset.

While making the decision on the feature selection and the classification techniques to be used in the binary classification scheme, the accuracy & the correct classification rates rather than speed were mainly taken into consideration. Adaboost, which is mostly used as a classifier in the literature, has been used here as a feature selection technique. It comes out to be good technique for feature selection as it makes use of boosting and the classification information at the same time with feature selection. Forward Feature Selection and Branch & Bound algorithms were also tested on the data.

Among the various systems that have been checked on the data, the system consisting of: normalization and histogram processing of images + applying feature extraction of Gabor wavelet coefficients + selecting features by Adaboost + classification by Support Vector machines came out to be the one supplying us with the highest recognition rates. The results were even better than / challenging the ones in the literature. For example, more than 93% recognition rates were hardly obtained for AU1 in the literature whereas the rate was about 94.5% in the project. The remaining recognition rates for the other upper face action units were all found to be above 91%.

In the multi-class classification case, the aim was to make use of a different method, Error Correcting Output Codes, which would supply a great increase in speed. SVM was selected as the binary classification method used in ECOC, and Adaboost was again the feature selection method applied on the Gabor wavelet coefficients. Namely, the system which was found to be giving out the best binary classification results was used in the multi-class case with ECOC. Although the recognition rates for a large number of classes came out to be unsatisfactory, the results for smaller number of classes with large number of training data were much better. In other words, when the problem with 3 classes, all of which including more than 45 samples, was considered, the maximum recognition rate for a specific class came out to be over 96% whereas the overall recognition average was about 87%. Although the speed was not taken into account while creating the systems of the binary classification scheme; here it was considered to be very important. There are many methods in the literature, as mentioned in 8.2.4, which can do the classification in a robust however a slow way; and the aim to make the classification faster, was fulfilled.

9.2 Evaluation

The objectives of coming up with solutions to the binary and multi-class facial expression recognition problems were accomplished by making use of several systems during the project. The accuracy obtained on the recognition rates was over 91% for all the upper face action units on the Cohn-Kanade dataset. On the multi-class classification, the aim to increase speed and decrease complexity was achieved; however lower accuracy on the recognition compared to the binary case was obtained. This was in fact an expected result; however, different techniques can be used and tried on the implemented system to further increase its classification rates, while keeping the increased speed. One of the other objectives of the project was to create a system that would be invariant to environmental effects such as lighting and pose changes. Although the system currently works only on frontal face images, the changes in the illumination and also small shifts occurring due to the non-robust localization of the eye centres are found to affect the system in a minimum amount, when Gabor features and the explained normalization techniques were used. The discussion about Gabor features' characteristics can be found on section 8 and 8.1.7.2.1.

9.3 Future Work

Although the systems were studied in detail during the project, there is still some future work to be accomplished in order to improve the existing methodologies and have a deeper understanding.

For feature extraction: It was indicated that the features that were giving out the best classification results were mostly low frequency components of the Gabor wavelets. Although there were some assumptions made in section 8.1.2.7.1, further work about the usage of edge information in the classification can be done. This task might also include the visualization of the Gabor wavelets, together with the Haar-Like and Haar wavelet coefficients which were giving out the highest recognition rates.

Secondly, as Haar-Like coefficients are claimed to be more robust to shifts and lighting changes than the Haar wavelets, the difference between the results obtained on the Cohn-Kanade dataset and obtained from a real life application was supposed to be minimum. A real life implementation making use of Haar-Like coefficients might therefore be useful to check this claim.

For feature selection: Some other methods may also be considered to be used in the project. Although Adaboost has got various advantages that were mainly mentioned throughout the project, it is found to be working slow when the number of features to apply the selection is larger than

10.000. Therefore, in order to decrease the amount of time in the training stage, usage of other feature selection algorithms might as well be considered.

It should also be mentioned that, feature classification was only applied once in the binary case experiments consisting of 10-fold cross validation. Due to timing constraints, it was assumed that the features selected within a single run of the validation would remain the same for the consecutive runs too. However, as a future work, one can also check if the features selected in each run are mostly the same or not.

In addition to the binary case, the 10-fold cross validation applied on the multi-class classification contains both the feature selection and the classification parts. However, as it was indicated, every 1 out of 10 Gabor features from a total of 40960 were taken to form the new feature set, to which Adaboost would be applied for feature selection. This was again due to timing restrictions. It was assumed that through making use of the whole feature space instead of the 1/10th; better features to give out better classification results might have been chosen. A test in order to see the effect of down-sampling the feature space had been carried on Action Unit 2 through binary classification and the results were found to have a 2% decrease in accuracy. Therefore same decrease is also expected in the multi-class case.

For classification: As dealing with large number of classes in the multi-class classification is a difficult task, the results obtained are not found to be very satisfying. Therefore, for the multi-class classification section of the project, different combinations of feature selection and classification methods, as well as different techniques to end up with the optimal code matrix, might be tried and compared. The aim should be to keep the speed of ECOC while increasing its accuracy for problems consisting of large number of classes with small number of training data.

When checking the performance of a classifier, ROC curves can be examined in deeper detail in the future. In the project, the regularization parameter c was thresholded and the corresponding ROC graph was detected. However, the graph did not have a smooth curve characteristic; the points found were rather distributed in the space. Thresholding different parameters might have revealed more important and interpretable information about the classifier’s performance.

One final thing to note that, as it was indicated, the experiments were applied on only the right eye images of the faces. Further results while taking both of the eyes into account might be examined.

REFERENCES

[1] Ekman, P. and Friesen, W.V. *Pictures of Facial Affect*. Palo Alto, Calif.: Consulting Psychologist, 1976.

[2] Izard, C.; Dougherty, L. and Hembree, E.A. *A System for Identifying Affect Expressions by Holistic Judgements*, unpublished manuscript, Univ. Of Delaware, 1983.

[3] Bartlett, M.; Hager, J.; Ekman, P. and Sejnowski T. *Measuring Facial Expressions by Computer Image Analysis*, Psychophysiology, vol. 36, pages 253-264, 1999.

[4] Bartlett, M.S; Littlewort, G.; Lainscsek, C.; Fasel, I. and Movellan, J. *Machine Learnin Methods for Fully Automatic Recognition of Facial Expressions and Facial Actions*, IEEE International Conference on Systems, Men and Cybernetics, pages 592-597, 2004.

[5] Ekman, P. and Friesen, W.V. *The Facial Action Coding System: A Technique for the Measurement of Facial Movement*, San Francisco: Consulting Psychologists Press, 1978.

[6] Mase,K. *Recognition of Facial Expression from Optical Flow*, IEICE Trans, vol. E74, no.10, pages 3474-3483, October 1991.

[7] Yacoob, Y. and Davis, L.S. *Recognizing Human Facial Expression from Long Image Sequences Using Optical Flow*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 6, pages 636-642, June 1996.

[8] Suwa, M.; Sugie N. and Fujimora K. *A Preliminary Note on Pattern Recognition of Human Emotional Expression*, Proc. International Joint Conf, Pattern Recognition, pages 408-410, 1978.

[9] Lanitis, A.; Taylor C. and Cootes T. *Automatic Interpretation and Coding of Face Images Using Flexible Models*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no.7, pages 743-756, July 1997.

[10] Kaapor, A.; Qi, Y. and Picard, R.W. *Fully Automatic Upper Facial Action Recognition*, Proceedings of the IEEE Int'l Workshop on Analysis and Modeling of Faces and Gestures, 2003.

[11] Zhang, Z. *Feature-Based Facial Expression Recognition: Sensitivity Analysis and Experiments with a Multilayer Perceptron*, Int'l Pattern Recognition and Artificial Intelligence, vol. 13, no. 6, pages 893-911, 1999.

[12] Whitehill, J. and Omlin, C. W. *Haar Features for FACS AU Recognition*, Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 2006.

[13] Tian, Y.; Kanade, T. and Cohn, Jeffrey. *Recognizing Action Units for Facial Expression Analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, February 2001.

[14] Homepage of Mikulic, E. Researcher at RMIT University, http://dmr.ath.cx/gfx/haar/.

[15] Homepage of Shah. J, Temple University, ECE Department, http://shahj.net/selected_projects/ip/project6.htm#_Toc40805719

[16] Viola, P. and Jones, M.J, Robust Real-Time Face Detection, *International Journal of Computer Vision,* 57(2): 137-154, 2004.

[17] Freund, Y. and Schapire, R. E, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *Journal of Computer and System Sciences* 55: 119-139, 1997.

[18] Donato, G.; Bartlett, M. S.; Hager, J.; Ekman, P. and Sejnowski, T.J. *Classifying Facial Actions*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 10, October 1999.

[19] Zor, C. and Tutuncu, I. *Facial Feature Extraction and Analysis & Automatic Caricature Drawing.* B.Sc. Thesis, Sabanci University, Turkey, 2007.

[20] Hsu, R.; Jain, A.K. and Abdel-Mottaleb, M. *Face Detection in Color Images,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pages 696-706, 2002.

[21] Shen, L.; Bai, Li and Fairhurst, M. *Gabor Wavelets and General Discriminant Analysis for Face Identification and Verification* Image Vision Computing, vol 25, no 5, pages 553-563, 2006.

[22] Jain, A.K and Farrokhnia, F. Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition*, 24(12):1167-1186, 1991.

[23] Lee, C.J. and Wang, S.D.; Fingerprint Feature Extraction Using Gabor Filters. *Electronics Letters,* 35(4):288-290, 1999.

[24] Zhan, Y.; Niu, D. and Cao, P. *Facial Expression Recognition Based on Gabor Wavelet Transformation and Elastic Templates Matching*, Third International Conference on Image and Graphics (ICIG'04), pages 254-257, 2004.

[25] Movellan, J. *Tutorial on Gabor Filters*, 1996.

[26] Asuncion, A. and Newman, D.J. UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html, School of Information and Computer Science, University of California, Irvine, CA, 2007.

[27] Duda, R.; Hart, P. and Stork, D.; *Pattern Classification*, Second Edition. Wiley-Intescience Publication.,USA, 2001.

[28] Welling, M. *Support Vector Machines,* Department of Computer Science, University of Toronto.

[29] Burges, C. J.C. *A Tutorial on Support Vector Machines for Pattern Recognition*, pages 1-43, Kluwer Academic Publisers, Boston.

[30] DTREG, Software For Predictive Modelling and Forecasting, http://www.dtreg.com/svm.htm

[31] Caifeng Shan, C.; Gong. S, and McOwan, P.W. *Robust Facial Expression Recognition Using Local Binary Patterns,* IEEE International Conference on Image Processing, ICIP2005, vol. 2, pages 370-373, 2005.

[32] Dietterich, T.G and Bakiri, G. Solving Multi-class Learning Problems via Error-Correcting Output Codes, *Journal of Artificial Intelligence Research 2,* pages 263-286, 1995.

[33] Fawcett, T. An Introduction to ROC Analysis, *Pattern Recognition Letters 27,* pages 861-874,2006

[34] Cohn, J.F.; Kanade, T. et al. Comprehensive database for facial expression analysis, 2000.

[35] Silapachote, P.; Karuppiah, D. R.; and Hanson, A. R. Feature Selection Using Adaboost for Face Expression Recognition*, Visualization, Imaging, and Image Processing (452)*, 2004

[36] Barlett, S. M.; Littlewort, G.; Frank, M.; Lainscsek, C.; Fasel, I. and Movellan, J. *Fully Automatic Facial Action Recognition in Spontaneous Behavior,* Proceedings of the 7th International IEEE Conference on Automatic Face and Gesture Recognition, 2006.

[37] Vezhnevets, A., GML AdaBoost MATLAB Toolbox, http://research.graphicon.ru/machine-learning/, 2007.

[38] Duin, R. P. W.; Juszczak, P.; Paclik, P.; Pekalska, E.; de Ridder, D. and Tax, D. M. J, *PRTools, A Matlab Toolbox for Pattern Recognition,* Delft Pattern Recognition Research, 2004.

[39] Moore, A. M. *Support Vector Machines Tutorial*, School of Computer Science, Carnegie Mellon University, 2001.

[40] Franc, V. and Hlavac, V. *Statistical Pattern Recognition Toolbox*, Czech Technical University, 2004.

APPENDIX 1 – USER GUIDE

In this section, brief explanations of the Matlab source codes that are provided within the CD of the project are given. Note that, not all of the codes used in the project could have been supplied due to copyright constraints.

Adaboost.m: Given the dataset together with the labels, apply the boosting and return the training error.

Bootstrap.m: Bootstrap the given data in cell structure. The resulting bootstrapped data is again supplied in a cell.

EcocMatrixForm.m: Form the ECOC matrix for the 12 class problem with 75 columns.

EcocTest.m: Apply ECOC classification test on data; work with Gabor features and already trained SVM classifiers. Directories must be changed while using.

EcocTrainingPart.m: Apply ECOC training on the data given in the specified directory.

eyeMap.m: Apply the EyeMap technique on the given image and return the eye coordinates.

featureAda10cross.m: Apply 10-fold cross validation on AdaFs and return the error rate.

FeatureAdaTrainingError_Gabor.m: Calculate the training error for Adaboost applied on extracted Gabor wavelet features and return the error rate.

FeatureAdaTrainingError_GaborRestricted.m: Calculate the training error for Adaboost applied on extracted and down-sampled (1/10th) Gabor features and return the error rate.

FeatureAdaTrainingError_HaarW.m: Calculate the training error for Adaboost applied on extracted Haar Wavelet features and return the error rate.

FeatureAdaTrainingError_HaarLike.m: Calculate the training error for Adaboost applied on extracted Haar-Like features and return the error rate.

FormGaborData.m: Form the Gabor wavelet coefficients of a single image.

FormGaborDataAll.m: Extract the Gabor wavelet features of all the negative and positive sample images stored in the directory of a specific class. Directories must be changed while using.

FormHaarDataAll.m: Extract the Haar wavelet features of all the negative and positive sample images stored in the directory of a specific class. Directories must be changed while using.

FormHaarLikeDataAll.m: Extract the Haar-Like coefficients of all the negative and positive sample images stored in the directory of a specific class. Directories must be changed while using.

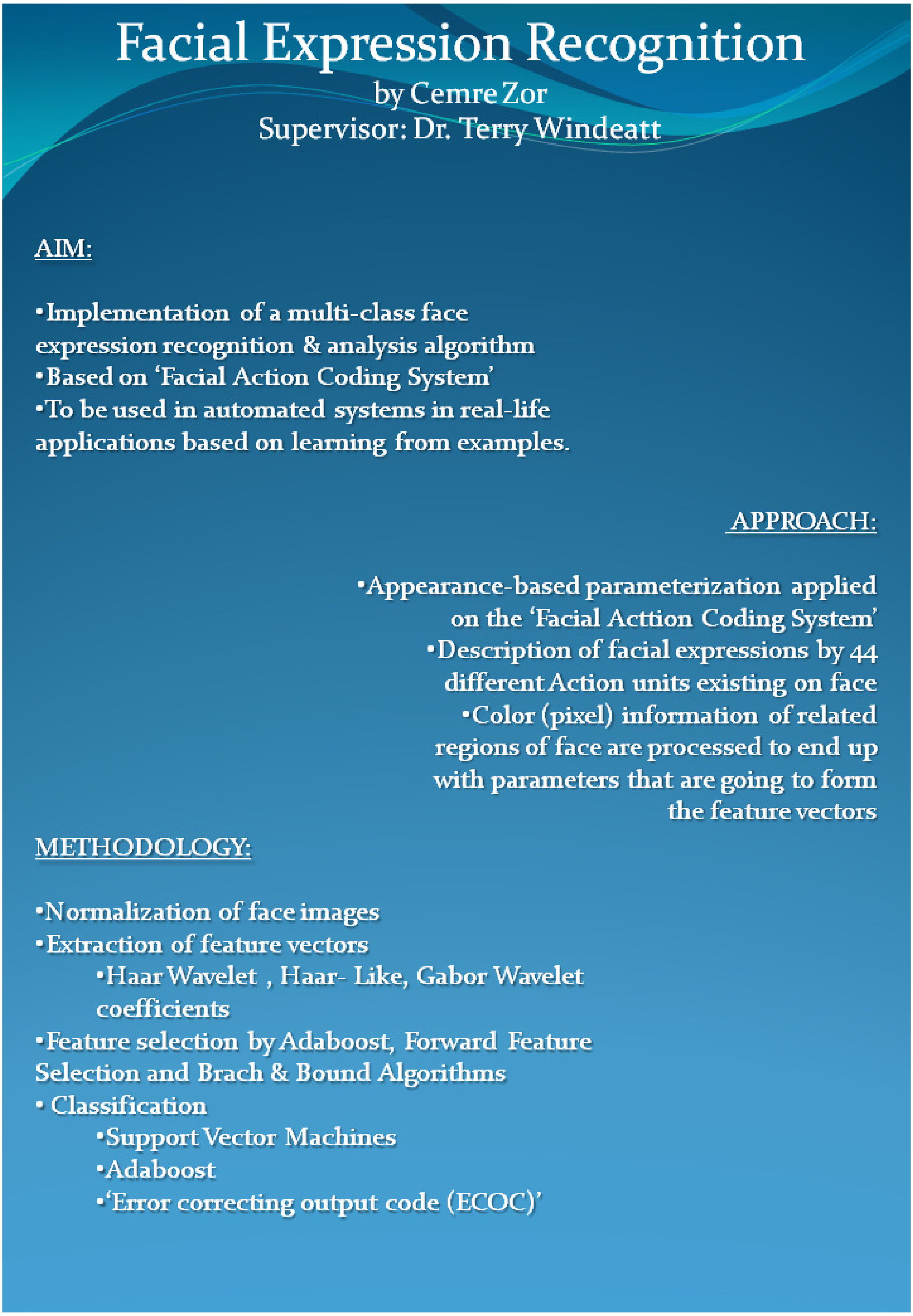
FormSelGaborDataAll.m: Create the new dataset with selected Gabor features. The selected features are obtained from one of feature selection algorithms and are stored in .mat files.

FormSelHaarDataAll.m: Create the new dataset with selected Gabor features. The selected features are obtained from one of feature selection algorithms and are stored in a list.

GaborC.m: Given an image, a specified frequency and an orientation, extract the Gabor wavelet features.
haar_like.m: Given an image, extract the Haar-Like coefficients.
Lowfft.m: Apply low pass filtering on an image.
multiGabor.m: Apply Gabor filtering on an image in 8 frequencies and 5 phases.
normalize1.m: Normalize the given images (without applying histogram processing).
normalize2.m: Apply histogram processing on the images (which are normalized through normalize1.m)
svmCrossValTemp.m: Given selected features with labels, apply 10 fold cross validation on SVM's, using PRTOOLS [38], and also calculate the true and false positive rates to plot the ROC graph while thresholding the c value.

Note: tree_node_w.m, crossvalidation.m and stumps3.m are files taken from the GML [37] and SPR [40] toolboxes to be used in the implementations of some of the above explained functions.

APPENDIX 2 – PROJECT POSTER



Facial Expression Recognition

by Cemre Zor

Supervisor: Dr. Terry Windeatt

AIM:

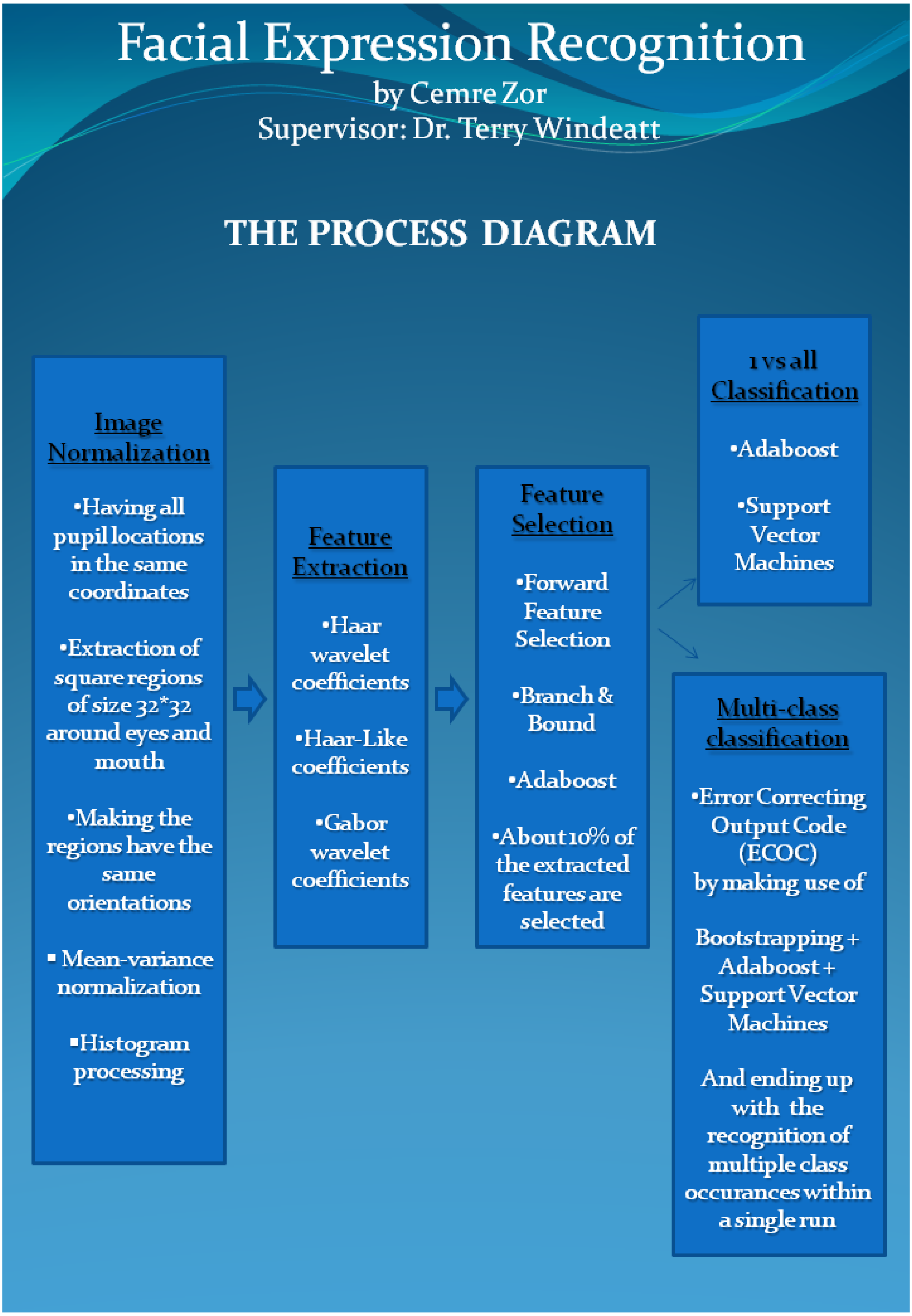
- Implementation of a multi-class face expression recognition & analysis algorithm
- Based on 'Facial Action Coding System'
- To be used in automated systems in real-life applications based on learning from examples.

APPROACH:

- Appearance-based parameterization applied on the 'Facial Acttion Coding System'
- Description of facial expressions by 44 different Action units existing on face
- Color (pixel) information of related regions of face are processed to end up with parameters that are going to form the feature vectors

METHODOLOGY:

- Normalization of face images
- Extraction of feature vectors
    - Haar Wavelet , Haar- Like, Gabor Wavelet coefficients
- Feature selection by Adaboost, Forward Feature Selection and Brach & Bound Algorithms
- Classification
    - Support Vector Machines
    - Adaboost
    - 'Error correcting output code (ECOC)'



Facial Expression Recognition

by Cemre Zor

Supervisor: Dr. Terry Windeatt

THE PROCESS DIAGRAM

Image Normalization

- Having all pupil locations in the same coordinates
- Extraction of square regions of size 32*32 around eyes and mouth
- Making the regions have the same orientations
- Mean-variance normalization
- Histogram processing

Feature Extraction

- Haar wavelet coefficients
- Haar-Like coefficients
- Gabor wavelet coefficients

Feature Selection

- Forward Feature Selection
- Branch & Bound
- Adaboost
- About 10% of the extracted features are selected

1 vs all Classification

- Adaboost
- Support Vector Machines

Multi-class classification

- Error Correcting Output Code (ECOC) by making use of

Bootstrapping + Adaboost + Support Vector Machines

And ending up with the recognition of multiple class occurances within a single run