Automatic Facial Expression Recognition: Space-Classifier Combinations

by

Humayra Binte Ali

Thesis
submitted to Flinders University
for the degree of

Doctor of Philosophy

CSEM School
Faculty of Science and Engineering
20 November, 2017

My beloved Parents, Shovon, Nameera, Ayesha, and Afra.

Contents

List of Figures

List of Tables

Abbreviations

FER: Facial Expression Recognition.

PCA: Principal Component analysis.

ICA: Independent Component analysis.

NMF: Non-negative matrix factorization.

HOG: Histogram of Oriented Gradients.

ED: Euclidian Distance.

SVM: Support Vector Machine.

ELM: Extreme Learning Machine.

ELM: Kernel- Extreme Learning Machine Kernel.

SIFT: Scale-invariant feature transform.

CK dataset- Cohn-Kanade Facial Expression dataset.

JAFFE dataset- Japanese Female Facial Expression Dataset.

N-CV: Nested Cross-Validation

RK-CV: Repeated Cross-Validation

Summary

Facial expression recognition is a broad research domain in machine learning. Principal Component analysis (PCA), Independent Component analysis(ICA), Non-negative matrix factorization (NMF) and HOG (Histogram of Oriented Gradients) are well-established techniques for image analysis. In this thesis, we propose a facial expression recognition system, which is based on NMF, HOG, PCA and ICA for feature extraction. For classification, we implement Euclidian Distance (ED), Support Vector Machine (SVM) and ELM (Extreme Learning Machine) classifiers. Every feature has been passed to each of the classifiers to find the performance of the feature and classifier combinations. As we are using PCA, ICA and NMF which are mainly applied for dimension reduction and HOG works as SIFT descriptors, we will use the term 'space' for the feature extraction processes. Altogether we investigate the performance of sixteen space and classifier combinations to make a comparison of the FER system. Our proposed approach has been tested on both CK and JAFFE datasets.

There is a considerable debate over whether it is best to use whole or part based image analysis. So in our proposed system, we implement the FER system using both whole face and part face based recognition systems. In our experimental setup, first, we detect the three face parts (eyes, nose and mouth) using cascaded object detection by setting regions using a systematic trial and error basis.

For the extraction of facial features, we apply the commonly used PCA and ICA with the more plausible NMF and also the SIFT (Scale-invariant feature transform) descriptor like feature, HOG. As PCA, ICA and NMF work by reducing the total feature space, so in this thesis, we will consider the features produced by PCA, ICA, NMF and HOG as 'Space'. The classifiers we implement here are the following: Eu-

clidian Distance (ED), Support Vector machine (SVM), Extreme Learning Machine (ELM) and Extreme Learning Machine Kernel (ELM-Kernel). As every Space is fed to every classifier, so the total comparison is among sixteen space+classifier combinations. These space-classifier combinations are, PCA+ED, PCA+ELM, PCA+ ELM kernel, PCA+SVM, ICA+ED, ICA+ELM, ICA+ ELM kernel, ICA+SVM, NMF+ED, NMF+ELM, NMF+ ELM kernel, NMF+SVM, HOG+ED, HOG+ELM, HOG+ ELM kernel as HOG+SVM.

Potentially a subset of all the three facial parts (eyes, nose and mouth) of the face is better in terms of processing time and accuracy for identifying an expression. To prove whether three facial parts can perform better to express any certain emotions or vice versa, we implement a 3X10-fold R-K cross-validation, where 'R' is for repeated cross-validation. From the investigation, it is proved that for some space-classifier combinations three main facial parts perform better than the full face based FER and also vice versa. Also our prediction is any subset of the three facial parts can still perform better. To analyze this issue, we carefully design a 10x10 Nested Cross-Validation (N-CV) approach to tune the space-classifier combinations for each subset of the facial parts and also for the full face. We analyzed the results in the Result Analysis chapter.

We use a set of three facial regions and ensure each part is of similar size. For our proposed RK-CV method we segment the faces into three regions, eyes, nose and mouth and we consider all these three face parts to classify expressions. For the N-CV approach, we take the features for the whole face, eyes, nose, mouth, nose+ mouth, eyes+ mouth, eyes+nose, and eyes+nose+mouth. These features are made for all the seven basic expressions.

The recognition rate can be seen to be much better using the whole face decomposition and comparison, but this comes at an increased computational cost. We formulate a table which shows the influence of different facial parts for emoting a specific expression. To validate our results, we tested each expression individually, projecting it onto the whole set feature spaces trained against the whole training dataset, which has a mixture of all seven expressions.

Publications

A. Journals

1. Ali, H. B. and Powers, D. M. W. (2013). Fusion Based FastICA method: Facial Expression Recognition, *Journal of Image and Graphics, Vol.2,No. 1.pp.1-7, June2014. Doi:10.12720/joig.2.1.1.-7.*

2. Ali, H. B. and Powers, D. M. W.(2014). Facial Expression Recognition Based on WAPA and OEPA FastICA, *International Journal of Artificial Intelligence and Applications (IJAIA), Vol. 5, No. 3, May 2014.*

3. Jia, X., Zhang, Y., Powers, D.M. W., Ali, H. B. (2014). Multi-Classsifier Fusion Based Facial expression Recognition Approach, *KSII Transaction on Internet and Information System, 2014.*

4. Ali, H. B., Powers, D. M. W., Jia, X. and Zhang, Y.(2015). Extended Non Negative Matrix Factorization for Face and Facial Expression Recognition, *International Journal of Machine Learning and Computation, V5N2, April, 2015.*

B. Conferences

1. Ali, H. B. and Powers, D. M. W.(2014). Multi-Feature Fusion based Non Negative Matrix Factorization: Facial Expression Recognition from Imaging Sensors. *MLSDA, December 2, 2014, Gold Coast, Australia. Doi:10.1145/2689746.2689753. Published in ACM.*

2. Ali, H. B. and Powers, D. M. W.(2015). Face and Facial Expression Recognition: Fusion based PCA vs. NMF. *ICAART , Lisbon, Portugal, 2015.*

3. Ali, H. B. and Powers, D. M. W.(2013). Facial expression recognition based on Weighted All Parts Accumulation and Optimal Expression-specific Parts Accumulation. *DICTA 2013, Hobart, Tasmania,page 229-235.*

4. Ali, H. B., Powers, D. M. W., Leibbrandt, R., and Lewis, T. (2011). Comparison of Region Based and Weighted Principal Component Analysis and Locally Salient ICA in Terms of Facial Expression Recognition. *81-89. SNPD 2011, Sydney. Published in Springer.*

5. Ali, H. B. and Powers, D. M. W.(2014). Fusion Based FastICA method: Facial Expression Recognition, *ICCCV, London,UK, June, 2014.*

Declaration

I declare that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge it does not contain any material previously being published or written by any other person except where due reference is made in the text.

Humayra Binte Ali

Acknowledgment

All praise and glory to the Almighty for helping me to complete the Ph.D. work and the thesis. I came for this study in spite of all the difficulties in a new country and the memories left behind, my mental strength and perseverance along with the Almighty's support in the name of the good fortune helped me all these years' research and living in Australia. Then I am grateful to my respected and beloved parents and husband for everything.

Thanks to CSEM school, Flinders University for giving me a scholarship to undertake this Ph.D. study.

Professor Dr. David M W Powers, the person to whom I am ever grateful for his sincere guidance –as a teacher and as a friend. His enthusiasm, knowledge and above all his sincere support and understanding for all matters I received throughout the whole time.

Dr. Richard Leibbrandt, Dr. Trent Lewis, and fellow students of CSEM school, Flinders University for fruitful discussions, support and encouragements. Thank you all.

Finally, I thank my parent-in-laws, all my siblings, my daughters, relatives and friends, my well-wishers for their encouragement and support throughout the whole duration of my research.

Chapter 1

Introduction

1.1 Face and Facial Expression Recognition

Faces provide some of the most prominent biometric traits characterized by uniqueness and robustness. For this reason Face Recognition has caught the attention of researchers in the domain of person identification, speaker recognition, intruder detection, security enhancement as well as in other domains of computer vision, psychology, and physiotherapy. Face Recognition covers both the area of Face Identification and Face Verification. Face Identification means to find the identity of a given person out of a pool of N persons (1 to N matching) and this Face Identification is widely used in video surveillance, information retrieval, video games and some other human-computer interaction areas. On the other hand, Face Verification establishes the process of confirming or denying the identity claimed by a person (1 to 1 matching). To verify access control into computer or mobile device or building gate, and digital multimedia data access control, Face Verification techniques are needed. At the same time, there are many applications where facial expression recognition is more important than only face detection and recognition. As an example, facial expression recognition is applicable when pain estimations for patients is needed by observing the movement of facial features. Another example is human-machine interaction; like online chat conversation or online teaching where users or students expression is needed to make the conversation more realistic and fruitful. Also, analysis of facial expression is needed in long time

vehicle travel to detect whether the driver is sleepy or active during the time of driving for the safety reason. In recent times, the applications of facial expression recognition have been steadily increasing. Thus automatic recognition of facial expression has become a broad research domain of increasing significance.

In this research work, our focus is facial expression recognition (FER). As expression,we consider here the basic emotions; like anger, disgust, fear, happy, sad and surprise. **So for our work, facial expression recognition and facial emotion recognition considers the same meaning.** The most challenging part in these areas are to recognize facial expressions with a minimum time requirement and with minimum error rate. Our proposed approach and programming and mathematical analysis will focus on these constraints of minimum time requirement and with minimum error.

1.2 Background of the study

Automatic emotion recognition has been attracting the attention of researchers from several areas including computer vision, psychology, behavioral science, computer games and medicine (Pantic and Rothkrantz, 2000). But it is really a hard problem to recognize facial expression with a very high accuracy (Kapoor and Picard, 2001), (Picard, 1997), (Izard, 1979) and (Cottrell and Metcalfe, 1991). There are lots of challenges and critical issues in the domain of facial expression recognition.

As described in the previous section, facial expression recognition is playing very important role in machine learning and computer vision area. During human-to-human interactions; perception and decision-making play a very important role. And this interaction, perception and decision making occur due to change of persons' emotional expression or affective states. But this change of expression is inaccessible to computing systems unless we provide computers to understand the human expression. So without this, human-computer interaction has become a predominantly one-way interaction where a user needs to directly request computer responses. Effective natural human-computer interaction becomes hard in many applications as computers become integrated into everyday objects. In some cases, users need to be able to interact naturally with computers exactly the way interpersonal face-to-face interaction takes place.

When computers or machines will recognize human faces as well as understand human expression then we can get more feedback from machines. The ability to detect and track users expression or emotional expression or affective states has the potential to allow a computing system to initiate communication with a user based on not only the user's command but also the perceived needs of the user within the context of the user's actions. And then human-computer interaction can become more users friendly and natural. Emerging technological advances are enabling and inspiring the research field of affective computing, which aims at allowing computers to express and recognize affect (Picard, 1997). These are because research in social psychology [(Boyle et al., 1994), (Stephenson et al., 1976), (Matsumura et al., 1997), (Ekman and Davidson, 1994), (Pantic and Rothkrantz, 2000), (Ekman, 1979), (Ekman, 1982a), (Ekman, 1982b), (Ekman and Friesen, 1971), (Ekman and Friesen, 1976)] suggests that facial expressions play a major role in the human-human interactions and provide a very strong cue about finding the level of interest (Matsumura et al., 1997).

Facial expression recognition system generally consists of three steps, like face detection, feature extraction and classification. Machine learning researchers are using many algorithms for feature extraction and classification. Feature extraction is the process, which extracts the relevant information from a face image for a particular task. Feature extraction process can be performed in two different ways:

1 Take the facial features from the whole face to collect information for classifications of facial expression.

2 Divide the face into several sub-parts and process each to get an information that can be used as classification input.

A large number of machine learning algorithms can be used in the area of facial expression recognition. There is a considerable debate whether full face based FER or part based FER is more accurate. Among all these algorithms, we can divide it into two broad categories, like; appearance based approaches and geometrical feature-based approaches.

In the domain of appearance based feature extraction, subspace projection techniques are often used in computer vision problem as an efficient method for both dimension

reduction and finding the direction of the projection with certain properties. Usually, the face image is considered to lie in a high-dimensional vector space. The subspace projection techniques represent a facial image as a linear combination of low-rank basis images. The popular subspace projection techniques are PCA, ICA, NMF and FDA. Subspace projection algorithms work by creating low-rank basis images and project the original image as a linear combination of low-rank images. By projecting they employ feature vectors consisting of coefficients of the reduced components. Figure 1.1 depicts the algorithm for subspace projection step by step.

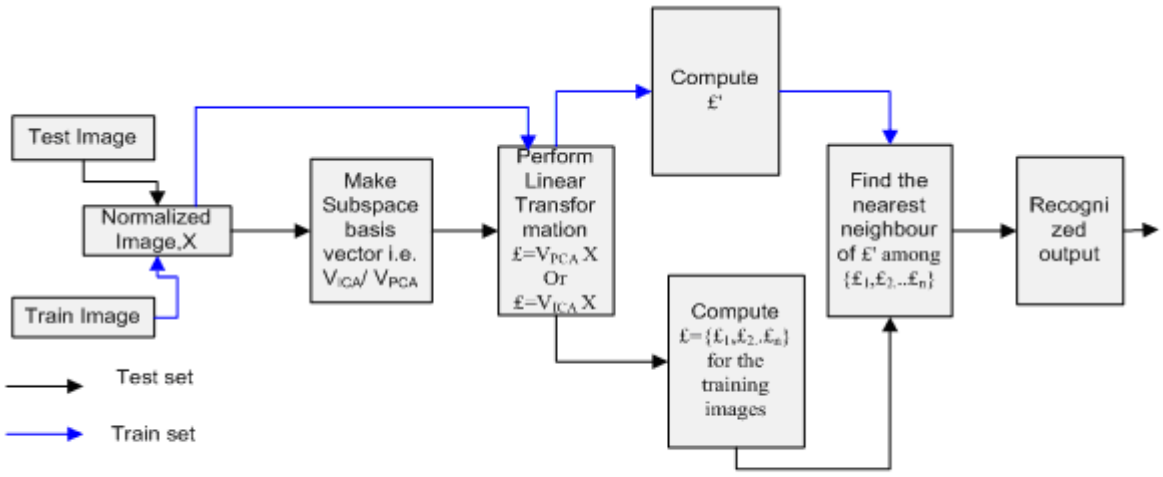

Figure 1.1: Subspace projection technique.

In the context of face expression recognition , we attempt to find some basis vectors in that space serving as much as important directions of projection in a low-rank image subspace. These subspace projection algorithms have been used in the Facial Expression Recognition area over the last ten years in the work of (Buciu et al., 2003), (Frank and Noth, 2003b), (Frank and Noth, 2003a), (Pentland, 1987), (Kolenda et al., 2002), (Uddin et al., 2009), (Chen and Kotani, 2008).

Redundancy in the sensory input contains structural information about the environment (Marian et al., 2002). PCA and ICA are most well-known methods for redundancy as well as finding useful components for attaining distinguishable properties. It has been argued in (Barlow, 1989) that such redundancy provides knowledge and that the role of the sensory system is to develop factorial representations in which these dependencies are separated into independent component (ICs) and such repre-

sentations are advantageous for encoding complex objects that are characterized by high-order dependencies. Similarly, these representations have potential as a general coding strategy for the visual system (Atick and Redlich, 1992). For feature extraction from the face and facial expression images, most of the early research works did experiment by extracting useful features using Principal Component Analysis (PCA). PCA is a second-order statistical method to derive the orthogonal bases containing the maximum variability in an unsupervised manner that provides global image features. It is also commonly used for dimension reduction. In (Donato et al., 1999) and (Ekman and Friesen, 1978), the authors employed PCA as one of the feature extractors to solve facial expression recognition with the Facial Action Coding System (FACS) and in (Cohn, 1999) same procedure is applied for face recognition. Our previous work in (Ali and Powers, 2013) shows applying PCA on face parts rather the whole face give more accuracy when using euclidian distance as a classifier. Lately, Independent Component Analysis (ICA) has been extensively utilized for face and facial expression recognition tasks due to its ability to extract local facial features.

As much of the information that distinguishes different facial expressions and different face styles stay in the higher order statistics of the images (Chen and Kotani, 2008), ICA is a better choice for face recognition as well as FER than PCA. Basically, ICA is a generalization of PCA that seeks the independencies of the image features (Hyvarinen et al., 2001), (Karklin and Lewicki, 2003). In (Bartlett et al., 1999), Bartlett et al. extracted the local image representations for the facial expression coding using ICA to classify 12 facial expressions referred to FACS. In (Chuang and Shih, 2006), Chao-Fa and Shin utilized ICA to extract the IC features of facial expression images to recognize the Action Units (AU) in the whole face as well as the lower and upper part of the faces. However, the FAU-based works mostly focus on the successful extraction of FAUs not the recognition of emotions derived from facial expression changes. Also, they encounter the limitation of AUs due to the fact that the separate facial expressions do not directly draw the comparisons with human data (Calder et al., 2000). Later on, in (Buciu et al., 2003), Buciu et al. proposed ICA for the emotion-specified FER where ICA was applied on the Japanese female facial expression database. In (Bartlett et al., 2002), Bartlett et al. again introduced ICA on the PCs for face recognition in two different architectures where the first architecture finds the spatially local basis images

and the second one the factorial face codes. They showed that both the architectures outperform PCA. Applying ICA on the PC features is usually recognized as Enhanced Independent Component Analysis (EICA) (Liu, 2004). In (Liu, 2004), Liu applied EICA for content-based face image retrieval using more than thousand frontal face images from the FERET database (Phillips et al., 1998).

In some recent research, Scale-invariant feature transform (SIFT) and a histogram of oriented gradients(HOG), are also used as effective feature descriptors [(Luo et al., 2007), (Albiol et al., 2008)]. The underlying methods of HOG have similarity with scale-invariant feature transform descriptors, shape contexts and edge orientation histograms. It is mainly computed based on a dense grid of uniformly spaced cells and to enhance the accuracy it applies overlapping local contrast normalization.

Two researchers at the French National Institute for Research in Computer Science and Automation (INRIA), Navneet Dalal and Bill Triggs first described HOG descriptors at the 2005 Conference on Computer Vision and Pattern Recognition (CVPR) (Dalal and Triggs, 2005). In this work, they focused on pedestrian detection in static images, although since then they expanded their tests to include human detection in videos, as well as to a variety of common animals and vehicles in static imagery. In the area of facial expression recognition, HOG has been successfully implemented in several works [(Lemaire et al., 2013), (Dahmane and Meunier, 2011), (Zhang et al., 2013)].

Facial expression recognition system uses two types of technologies, image processing and pattern recognition. Image processing is used for face detection and feature extraction process and pattern recognition is the process of classifying patterns of facial expression by learning the classifier. The following figure 1.2 depicts the steps. So far we have discussed the image processing step. Now we will give a short discussion on facial expression classification.

Facial expression classification can be thought of as pattern recognition problems in machine learning. The information extracted from the feature the extraction process is given to the classifier as input vectors. The first step of the classifier is to train the system based on the extracted features. Then the learned classifier is applied on a test set to recognize the accuracy or the systems performance. There are several classifiers which are extensively used in machine learning especially in facial expression

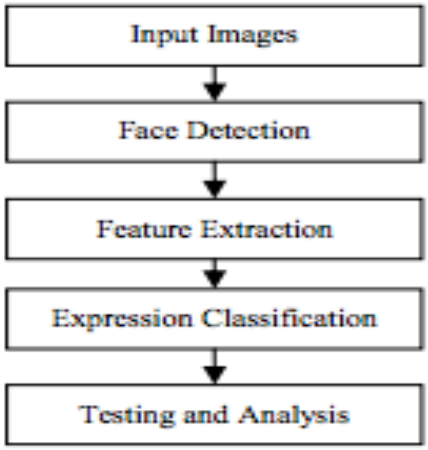

Figure 1.2: Basic Flow Chart of Facial Expression Recognition.

recognition area, like; neural networks, support vector machines, extreme learning machine, nearest neighbor classifier, linear classifiers, vector norm etc. Other classifiers that have been widely used are AdaBoost classification with a selection of several weak classifiers, and its variation, like; GentleBoost classification. Sometimes, two or more classifiers are combined to achieve better results. Although there are many classifiers have been used in machine learning and image analysis area, still, the shortcomings are also huge when it comes to the point of recognizing facial expression. The neural network classifier requires large training samples and many adjustable parameters as well as it requires a large amount of training time. The performance of AdaBoost classifier depends on the weak classifier selection.

Some research works have successfully used Support Vector Machine(SVM) with several feature extraction processes. SVM is a non-linear data-processing tool, which has been successfully used in many fields such as face recognition, databases learning, identity verification and text categorization because of it avoiding the problems of over learning. Although SVM's operational speed is low as it needs huge computational resources. For an example, (Niu and Qiu, 2010) used SVM with WPCA and PPCA for facial expression recognition and received 88.25% and 84.75% accuracy using simple train-test method. The work of (Zhang et al., 2013) used the single kernel and multi-kernel SVM where the feature extraction process is HOG and LBPH. This work benchmarked their proposed approach on Bosphorus database using nested cross-validation and received the accuracy from 70% to 80% range.

(Huang et al., 2004) proposed extreme learning machine (ELM) which is an improved feedforward neural network that can randomly generate weights and thresholds and arranges only the numbers of neurons in the hidden layer. The ELM performs well in both regression and classification problems. (Liu et al., 2015) used ELM with Gabor filter and 2D-PCA for FER and achieved a reasonable recognition rate.

In recent years, video timing characteristics of facial expression recognition research has become a hot topic. In the work of (Song and Bao, 2016), Bezier Curve has been used for feature extraction and non-linear function fitting has been used for 3D feature extraction and also classification. They received average 93.2% accuracy on their proposed video dataset.

In the review paper work (Corneanu et al., 2016), the progress of Facial Expression Recognition research based on RGB, 3D, thermal and multimodal approaches has been depicted thoroughly.

Some very recent work on facial expression analysis, researchers used the Trajectory-Pooled Fisher Vector Descriptor. In this work (Liu and Yin, 2017), an individual video is modelled as a improved fisher vector aggregated by local and global trajectory features. Gaussian mixture models are constructed based on the features extracted training video. With GMM, the test video can be instantiated by fitting to GMM, the corresponding improved fisher vector is built to model test video. They got more than 70% accuracy on NVIE and MMSE facial expression datasets. Previously the same researchers used temperature changes and head motions for facial expression analysis (Liu and Yin, 2015).

Some other recent works have show their interest to analyze game based facial expression recognition. In the work of (Sawyer et al., 2017), researchers used FACS coding system for FER and achieve a good recognition rate.

In this research work, our focus is automatic face recognition from frontal faces. We are concerned here with the seven basic facial expressions identified by happy, sad, fear, surprise, anger, disgust and neutral. Our main contribution here is to improve the accuracy of expression detection as well as to reduce the computation time. Our proposed algorithm shows these successes on Cohn-Kanade and JAFFE facial expres-

sion datasets. We used here PCA, ICA, NMF and HOG techniques and applied on whole faces and facial parts as well. To train and classify the facial expressions, we used here euclidian distance (ED), support vector machine (SVM), extreme learning machine(ELM) and extreme learning machine kernel (ELM Kernel). Altogether we have used 4x4= 16 feature-classifier combinations on full face and on facial parts to recognize the facial expressions as well as to analyze the comparison of full face and part face based approaches. Our proposed approaches and results will be discussed in the corresponding sections.

1.3 Facial Expressions

To express emotion, mood and attitude people communicate through speech, facial expression and body language (Qvarfordt and Zhai, 2005) and (McNeill, 1992). In our work, we can say emotion recognition in terms of verbal (speech) and non-verbal (facial expression and body languages like postures, eye gaze, and head motions) correlation detection. The face has been called the most important perceptual stimulus in the social world (Frith and Cohen, 1987) with infants as young as 3 months able to discern facial emotion (Charlesworth and Kreutzer, 1973). So visual feature extraction means mainly facial feature extraction.

At 1968, (Mehrabian, 1968) pioneered research into the role of non-verbal communication. He published findings indicating that spoken words only account for 7% of the what a listener perceives; the remaining 93% of what a listener comprehends originates from the body language and tone employed in the delivery of the word. In the research work (Mehrabian, 1968), Mehrabian determined that when judging someone's affective state, people mainly rely on facial expression and vocal intonations.

Also, Paul Ekman's research shows the same understanding as Mehrabian. For example, Paul Ekman and Wallace V. Friesen in 1976 and 1978 mentioned that face reveals a strong evidence for the fruitfulness of the systematic analysis of facial expression. Based on this they proposed FACS to understand the facial expressions The interconnections of facial expression with the rest of human psychology are mostly complicated and hard to elucidate (Ekman and Rosenberg, 1997).

Some psychology research also shows that adult children are also able to understand facial expression to detect the emotion. Although it is still highly controversial regarding the course of development of the ability to discriminate and recognize facial expressions.(Field et al., 1982) reported that 2-day old neonates could distinguish between the facial expressions of happiness, sadness and surprise, and could reciprocate by producing the same expressions in response. Between the ages of 8 and 18 months, infants become able to use the mechanism of social referencing (looking at the mothers face in order to gain information about the environment) to guide their behavior (Leibbrandt, 2000). This indicates some ability on the part of these infants to gain access to the meaning of facial expressions.

Research shows that vocal tone is another important modal to recognize expressions. But in this research work, our focus is only on facial features to recognize expressions during human-machine interaction.

1.4 Universality of Emotions

The formal study of emotion in human (and animal) behavior has a long history of the early observational work of (Darwin, 1872) up to the recent emergence of Affective Science (Davidson et al., 2003) as a cohesive discipline. Over the period, three main categories of a psychological model of human emotion have emerged.

The earliest discrete theories of emotion (stemming from Darwins work) hypothesized the existence of a small number of basic emotions, such as happiness, sadness, fear, anger, surprise and disgust (Ekman, 1999). In such theories, it is supposed that these emotions are based on specific psychological response patterns to external stimuli (Wallis et al., 2006).

Specifically, the basic emotions common to most discrete-emotions theories are anger, sadness, fear, enjoyment and disgust, with some theories also including interest, surprise, shame and guilt (Izard, 1994) and others adding contempt, awe and embarrassment (Ekman, 1992). This still leaves a great many other emotion words in English that do not refer to one of the basic emotions, such as jealousy, scorn, hope etc.

Discrete-emotion theorists argue that the basic emotions are better thought of as emotion families, and so, for instance, pride or relief could be said to belong to the enjoyment family. Also, (Izard, 1994) points out that distinct words are often used for the same emotion when it occurs with different cognitive concomitants, so that regret can be described as sadness combined with thoughts of a different course of action that one should have taken, while grief is sadness combined with thoughts of the loss of a loved one. Lastly, some emotion words refer to complexes or patterns of emotion that occur together or in rapid succession, so that, for instance, jealousy could be said to be made up of a pattern of experiencing anger, fear and sadness (Ekman, 1992).

According to some studies, specific emotions like happiness, fear, sadness, hostility, guilt, surprise and interest are considered discrete in that they are assumed to be unique experiential states that stem from distinct causes (Izard, 1977); some even consider these emotions to be basic [i.e. that they are present from birth and have distinct adaptive value: (Izard, 1992); (Stein and Oatley, 1992)].

This research work is based on basic emotions like happiness, sadness, fear, anger, disgust, surprise and neutral.

Chapter 2

Feature Extraction Algorithms

2.1 Principal Component Analysis

Principal Component Analysis is a linear dimensionality reduction technique: it transforms the data by a linear projection onto a lower-dimensional space that preserves as much data variation as possible. Here's a simple example of projecting 2D points into 1 dimension in Figure 2.1.

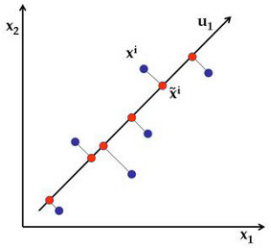

Figure 2.1: PCA example: 1D projection of 2D points in the original space

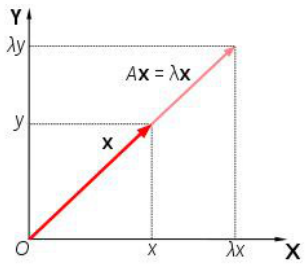

Figure 2.2: Illustration of a vector that satisfies is simply rescaled (not rotated) by.

Consider the equation $A\mathbf{v} = \lambda\mathbf{v}$. This states that the result of multiplying matrix A with a vector is the same as multiplying that vector by the scalar λ. A graphical representation of this is shown in the illustration below Figure 2.2. All non-zero vectors that satisfy such an equation are called eigenvectors of A, and their respective λ are

called eigenvalues. "Eigen" is a German term, meaning "characteristic" or "peculiar to", which is appropriate for eigenvectors and eigenvalues because to a great extent they describe the characteristics of the transformation that a matrix represents. In fact, for every real-valued, symmetric matrix we can even write the matrix entirely in terms of its eigenvalues and eigenvectors. Specifically, for an m x m matrix A we can state $A = V\Lambda V^T$, where each column of V is an eigenvector of A and $\Lambda$is a diagonal matrix with the corresponding Eigen values along its diagonal. This is called the Eigen-decomposition of the matrix.

Now we want to finally give PCA algorithm.

- Given
$$D = x^1, ..., x^n. \tag{2.1}$$
- Compute
$$\bar{x} = \frac{1}{n}\sum_i x^i \tag{2.2}$$
and
$$\sum = \frac{1}{n}\sum_{i=1}^{n}(x^i - \bar{x})(x^i - \bar{x})^T. \tag{2.3}$$
- Find the k eigenvectors of equation(3)with largest eigenvalues:
$$U_1, ..., U_k \tag{2.4}$$
These are called principal components
- Project
$$Z^i = ((x^i - \bar{x})^T U_1, ..., (x^i - \bar{x})^T U_k) \tag{2.5}$$

Note that we only need the top eigenvectors not all of them, which is a lot faster to compute.

2.2 Independent Component Analysis

The Principal Component Analysis (PCA), performed by the Karhunen-Loeve transform, produces features $y(i), i = 0, 1, ..., N$, that are mutually uncorrelated. The solution obtained by the KL transform solution is optimal when dimensionality reduction

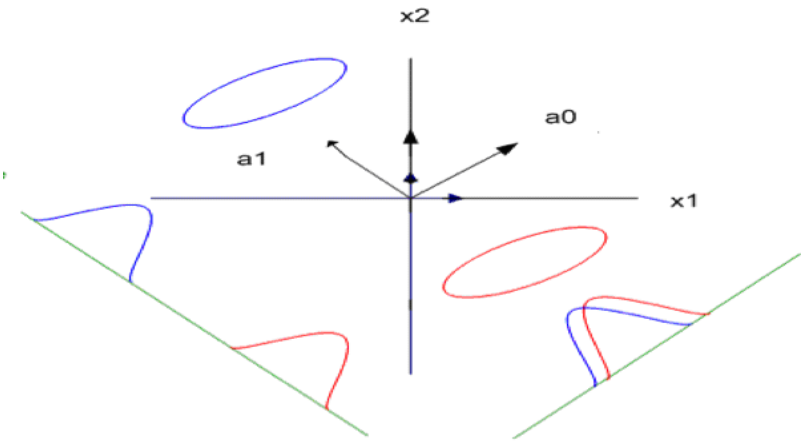

Figure 2.3: The KL transform is not always best for pattern recognition. In this example, projection on the eigenvector with the larger eigenvalue makes the two classes coincide. On the other hand, projection on the other eigenvector keeps the classes separated (Theodoridis and Koutroumbas, 2009).

is the goal and one wishes to minimize the approximation mean square error. However, for certain applications, such as the one illustrated in Figure 1, the obtained solution falls short of the expectations. In contrast, the more recently developed Independent Component Analysis (ICA) theory tries to achieve much more than simple decorrelation of the data (Hyvarinen et al., 2001),(Hyvarinen and Oja, 2000). Then ICA task is defined as follows: Given the set of input samples x, determine an N x N invertible matrix W such that the entries $y(i), i = 0, 1, ..., N-1$, of the transformed vector

$$y = Wx \tag{2.6}$$

are mutually independent. The goal of statistical independence is a stronger condition than the uncorrelatedness required by the PCA. The two condition are equivalent only for Gaussian random variables.

Searching for independent rather than uncorrelated features gives the mean of exploiting a lot more information, hidden in the higher order statistics of the data. As the example of (Figure 2.3) suggests, constraining the search by digging information in the second order statistics only results in the least interesting, for our problem, projection direction, that is, that of $a0$. However, $a1$ is, no doubt, the most interesting direction from the class separation point of view. In contrast, employing ICA can unveil from the higher order statistics of the data the piece of information that points a1 as the

most interesting one. Furthermore, searching for statistically independent features is in line with the way nature builds up the cognitive maps of the outside world in the brain, by processing the (input) sensory data (Theodoridis and Koutroumbas, 2009). (Barlow, 1989), in the so called, Barlow's hypothesis, suggests that the outcome of the early processing performed in our visual cortical feature detectors might be the result of a redundancy reduction process. In other words, the neural outputs are mutual as statistically independent as possible on the received sensory messages.

Before we proceed to develop techniques for performing ICA, we need to be sure that such a problem is well defined and has a solution and under what conditions. To this end, let us assume that our input random data vector x is indeed generated by a linear combination of statistically independent and stationary in the strict sense components (sources), that is,

$$x = Ay \tag{2.7}$$

The task now is under what conditions a matrix, W, can be computed so as to recover the components of y from equation 2.7, by exploiting information hidden in x. Usually, A is known as the mixing and W as the demixing matrix, respectively. All independent components $y(i)$,$i = 1, 2, ..., N$ with the possible exception of one, must be non-Gaussian. A second condition is that matrix A must be invertible. In the more general case where A is a nonsquare l x N matrix, then l must be greater than N and A must be of full column rank. In other words, in contrast to PCA which can always be performed, ICA is meaningful only if the involved random variable are non-Gaussian. Indeed, for Gaussian random variables independence is equivalent to uncorrelatedness and PCA suffices. From a mathematical point of view, the ICA problem is ill-posed for Gaussian Processes. Indeed, if we assume that the obtained Independent Components $y(i), i = 1, 2, ..., N$, are all Gaussian, then a linear transformation of them by any unitary matrix will also be a solution (Theodoridis and Koutroumbas, 2009). PCA achieves a unique solution by imposing a specific orthogonal structure onto the transformation matrix.

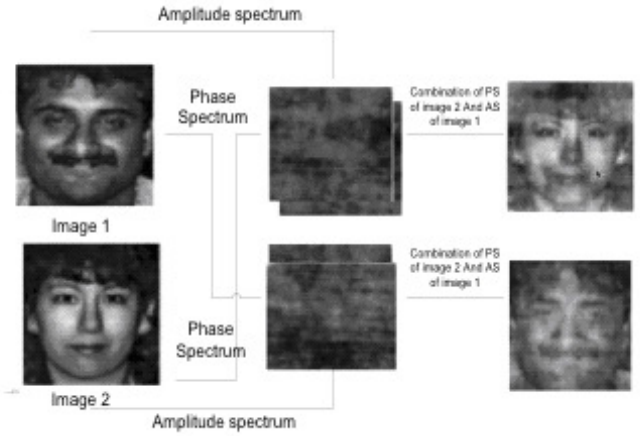

Figure 2.4: [PS: Phase Spectrum, AS:Amplitude Spectrum] The phase spectrum of the image 2 and Amplitude spectrum of image 1 produces the blurred image of image 2. The phase spectrum of the image 1 and Amplitude spectrum of image 2 produces the blurred image of image 1 replicated from (Marian et al., 2002).

2.2.1 ICA as Feature Extraction

In a specific task like face recognition, the most important information may be hidden in higher order statistics, not just the second order statistics. An example would be phase and amplitude spectrums. Amplitude Spectrum of an image is captured by Second-order statistics and the phase spectrum is hidden in the higher order statistics. The phase spectrum includes a great deal of information and also applies the human perception.

The above figure (Figure 2.4)shows that the more information pertaining the human eye is the phase spectrum.

2.2.2 ICA Algorithm

2.2.3 ICA By Maximization Of Non-Gaussianity

One of the easy and simple principles for estimating the model of ICA is based on maximization of non-Gaussianity. According to central limit theorem, the distribution of a sum of independent random variables tends towards a gaussian distribution, under certain conditions. Estimating the independent components can be accomplished by

finding the right linear combinations of the mixture variables, since we can invert the mixing as.

$$S = A^{-1}X \tag{2.8}$$

Thus to estimate one of the independent components, the linear combination of x_i can be considered. So it becomes

$$Y = b^T X = b^T AS \tag{2.9}$$

Hence if b were one of the rows of A^{-1} , this linear combination $b^T X$ would actually equal one of the independent components. But in practice, we cannot determine such b exactly because we have no knowledge of matrix "`[A]`" , but we can find an estimator that gives a good approximation. In practice, there are two different measures of Non-Gaussianity.

2.2.4 Kurtosis

The classical measure of non-gaussianity is kurtosis or the fourth order cumulant. It is stated by

$$Kurt(y) = Ey^4 - 3(Ey^2)^2 \tag{2.10}$$

As the variable y is assumed to be standardized it can say

$$Kurt(y) = Ey^4 - 3 \tag{2.11}$$

Hence the kurtosis is simply a normalized version of the fourth moment Ey^4. For the gaussian case, the fourth moment is equal to 3 and hence $Kurt(y) = 0$. Thus for gaussian variable kurtosis is zero but for the non-Gaussian random variable, it is non-zero.

2.2.5 Negentropy

Negentropy is another very important measure of non-Gaussianity. To obtain a measure of non-Gaussianity that is zero for a Gaussian variable and always non negative for a non-Gaussian random variable, we can use a slightly modified version of the

definition of differential entropy called negentropy. Negentropy J is defined as

$$J(y) = H(y_{gauss}) - H(y) \tag{2.12}$$

Where y_{gauss} is a gaussian random variable of the same covariance matrix as y.

2.2.6 Negentropy in terms of Kurtosis

The gaussian variable has the largest entropy among all the random variables. So that the negentropy for the random variables will always be positive and it is zero if and only if it is a gaussian variable. Moreover, the negentropy has an extra property that it is invariant for invertible transformation. But the estimation of negentropy is difficult, as it would require an estimate of the probability density function(pdf). Therefore in practice negentropy is approximated by using higher order moments.

$$J(y) \simeq \frac{1}{12}E(y^3)^2 + \frac{1}{48}kurt(y)^2 \tag{2.13}$$

In order to increase the robustness, another approach is to generalize the higher order cumulant approximation again the random variable y is assumed to be standardized. So that it uses expectations of general non-quadratic functions. As a simple case, we can take any two nonquadratic functions $G1$ and $G2$ such that $G1$ is odd and $G2$ is even and we obtain the following approximation[2].

$$J(y) \simeq K_1(EG_1(y))^2 + K_2(EG_2(y)) - EG_2(U)^2 \tag{2.14}$$

2.2.7 Fast Fixed Point Algorithm for ICA (FastICA)

We have implemented FastICA algorithm using three different kernels, like a hyper tangent, gaussian and cubic. The experiment shows that the gaussian kernel needs the less time than the other two and so that we choose the Gaussian kernel for further analysis. The algorithm of FastICA is depicted here. Firstly assume that we have a collection of pre-whitened random vector x. Using the derivation of the preceding section, the following steps show the fast fixed-point algorithm for ICA.

- Step 1: Take a random initial vector $w(0)$ of norm 1. Let $k = 1$.

- Step 2: Let $w(k) = E(x(w(k-1)^T x)^3) - 3w(k-1)$. The expectation can be estimated using a large sample of vectors.
- Step 3: Divide $w(k)$ by its norm.
- Step 4: $|w(k)^T w(k-1)|$ is not close enough to 1, let $k = k + 1$

The final vector $w(k)$ given by the algorithm equals one of the columns of the (orthogonal) mixing matrix "`[B]`" . In the case of blind source separation, this means that $w(k)$ separates one of the non-Gaussian source signals. The most important property of this algorithm is that a very small number of iterations, usually 5-10, seems to be enough to obtain the maximal accuracy allowed by the sample data set. This is due to the cubic convergence property of the algorithm.

2.3 Non-negative Matrix Factorization

Many machine learning research shows that Non-negative matrix factorization (NMF) is a useful decomposition for multivariate data like face and facial expression recognition. According to research studies (Lee and Seung, 2009) it is clear that NMF can be understood as part based analysis as it decomposes the matrix only into additive parts. This factorization technique of NMF is completely different of Principal Component Analysis (PCA) or Vector Quantization (VQ) in terms of the nature of the decomposed matrix. PCA and VQ work on holistic features whereas NMF decomposes a part-based representation of matrix (Lee and Seung, 2009). Here we apply NMF and PCA on whole faces and on different facial parts. PCA, ICA, VQ, NMF all these subspace learning techniques reduces the dimension and make a distributed represented in which each facial image can be approximated using a linear combination of all or selected basis images.

The factorization problem can be written like this,

$$X \approx W.H \tag{2.15}$$

where $X \in R^{MxN,\geqq 0}$

This is similar to the PCA or ICA initialization. In the above equation, R defines the low-rank dimensionality. Here $[W]$ and $[H]$ are quite unknown; $[X]$ is the known input source. Now we have to estimate the two factors. We have to start with random $[W]$ and $[H]$. Columns of $[W]$ will contain vertical information about $[X]$ and the horizontal information will be extracted in the rows of $[H]$. NMF does additive decompositions and parts make this decomposition. We first have to define the cost functions to solve an approximate representation of the factorization problem of $X \approx W.H$. By using some measure of distance between two non-negative matrices $[P]$ and $[Q]$, such cost functions can be constructed. The square of the Euclidian distance between the matrices $[P]$ and $[Q]$, is one fruitful measure.

$$||P - Q||^2 = \sum_{i,j} (P_{ij} - Q_{ij})^2 \tag{2.16}$$

The above equation is lower bounded by zero and absolutely vanishes if and only if $[P] = [Q]$. To define the cost function, another useful representation is,

$$D(P \parallel Q) = \sum_{i,j} (P_{ij} log \frac{P_{ij}}{Q_{ij}} - p_{ij} + Q_{ij}) \tag{2.17}$$

In the above equation, when $\sum_{i,j} P_{ij} = \sum_{i,j} Q_{ij} = 1$, the above Kullback-Leibler or relative entropy reduces. Here $[P]$ and $[Q]$ can be regarded a normalized probability distribution. Now, following the cost function of equation (2), we have to define it for the input matrix $[X]$ and the non-negative decomposed matrix $[W]$ and $[H]$. If we do that, the cost function would be,

$$\parallel V = WH \parallel^2 \tag{2.18}$$

The main goal is now to reduce the distance $||V - WH||$. To do that, first we have to initialize $[W]$ and $[H]$ matrix. Then we apply the multiplicative update rule, which is described in the paper of (Lee and Seung, 2009). They claim and prove that the multiplicative update rules minimize the Euclidean distance $||P - Q||$ and also the divergence, $D(P||Q)$ is decreasing when multiplicative update rule is applied. In our programming here, we use the Euclidian distance as a cost function and apply the multiplicative update rule to minimize the distance. The rules are defined below,

$$H_{p\beta} \leftarrow H_{p\beta} \frac{(W^T V)_{p\beta}}{(W^T W H)_{p\beta}} \tag{2.19}$$

$$W_{\alpha p} \leftarrow W_{\alpha p} \frac{(V H^T)_{\alpha p}}{(W H H^T)_{\alpha p}} \tag{2.20}$$

According to the mathematical analysis, if we use the equation 2.19 and 2.20 to decrease the Euclidian distance $||V - WH||$, the distance $||V - WH||$ converges. Our experimental analysis also shows that and we get a significant output on facial expression dataset.

2.4 Histogram of Oriented Gradients

In the area of Image Processing and Computer Vision, HOG (histogram of oriented gradients) has been successfully used in recent years. It has been successfully implemented in pattern recognition as a feature descriptor. The underlying method of HOG has similarity with scale-invariant feature transform descriptors, shape contexts and edge orientation histograms. It is mainly computed based on a dense grid of uniformly spaced cells and to enhance the accuracy it applies overlapping local contrast normalization.

Two researchers at the French National Institute for Research in Computer Science and Automation (INRIA), Navneet Dalal and Bill Triggs first described HOG descriptors at the 2005 Conference on Computer Vision and Pattern Recognition (CVPR) (Dalal and Triggs, 2005). In this work, they focused on pedestrian detection in static images, although since then they expanded their tests to include human detection in videos, as well as to a variety of common animals and vehicles in static imagery.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions [Wikipedia]. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The final descriptor is then the concatenation of these histograms. To enhance the increased accuracy, the local histograms are made

contrast-normalized. This normalization results in better invariance to changes in illumination and shadowing. HOG descriptor is invariant to geometric and photometric transformations due to its functionality of operating on locally spaced cells. But it is not invariant to object orientation.

2.4.1 Algorithm Implementation

Gradient Computation

The first step of Histogram of Oriented Gradients is to compute the gradient values. The most common method is to apply the 1D centered point discrete derivative mask in both the horizontal and vertical directions. In this method, the grayscale images need to be filtered with the following filter kernels.

$$D_y = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

and

$$D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

So, for an image I, x and y derivatives can be obtained by applying the convolution operation: $I_x = I * D_x$ and $I_y = I * D_y$. The magnitude of the gradient is,

$$|G| = \sqrt{I_x^2 + I_y^2}$$

and the orientation of the gradient is given by,

$$\theta = \arctan \frac{I_x}{I_y}$$

Orientation Binning

The second step of the algorithm is preparing the cell histograms. Based on the gradient values, derived in the previous subsection, each pixel in the cell casts a weighted

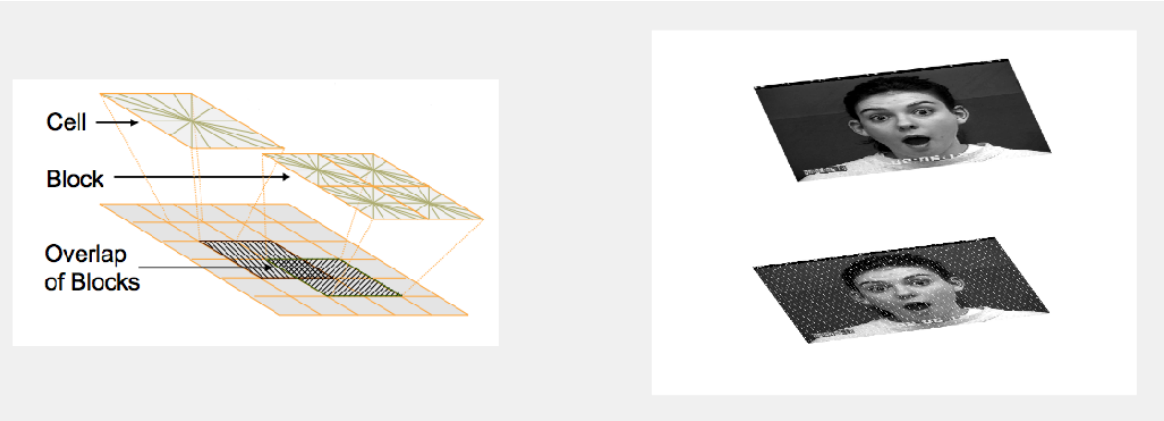

Figure 2.5: Block Normalization Process for HOG

vote for an orientation histogram channel. Depending on the gradient values, whether it is 'signed' or 'unsigned', the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees. As for the vote weight, gradient magnitude or some function of gradient magnitude, like the square or square root or some clipped version of the magnitude, is applied for pixel contribution.

Descriptor Blocks

To make the blocks robust for illumination and contrast changes, the gradient strength must be locally normalized. For this, the cells need to be grouped together to make it larger and spatially connected blocks. Then the final HOG descriptor contains the vector of the components of the already normalized cell histograms from al the block regions derived in the second step. There are basically two types of block geometrics, like rectangular R-HOG and circular C-HOG.

Block Normalization

Block normalization can be done following several methods. If v can be though of as a non-normalized vector containing all histograms in a given block, $||v_k||$ is its k norm for k=1,2, and e are some small constant, whose value does not normally has any effect on the results. Then the final normalization can be obtained by following any of the ways stated below:

$$L2 - norm : f = \frac{v}{\sqrt{||v||_2^2 + e^2}}$$

$$L1 - norm : f = \frac{v}{||v||_1 + e}$$

$$L1 - sqrt : f = \sqrt{\frac{v}{||v||_1 + e}}$$

In this thesis, we have used HOG as a feature extraction process to analyze the performance of seven-class facial expression recognition system. We have used Euclidian Distance (ED), Support Vector Machine (SVM) and Extreme Learning Machine(ELM) as the classifier with the extracted HOG features from the facial expression faces images.

Chapter 3

Classifier

3.1 Support Vector Machine

Support vector machines as well as support vector networks are supervised learning models for data classification and regression analysis. In machine learning, computer vision and pattern recognition area SVM has been very widely and successfully applied for classifying dichotomous and multi-class data problems. The genesis of support vector machines is from the statistical learning theory of Vapnik (Kumar, 2004), (Vapnik, 1995). Primarily the concept of SVM was introduced to solve binary classification problems applying supervised learning. The learning consists of solving a quadratic optimization problem, where the error surface is free of any local minimum and has global optimum(Begg et al., 2005). To find the optimal separating hyperplane using SVM, it is necessary to consider only a subset of the training points, called support vectors (Foody and Mathur, 2004) and (Cao and Tay, 2003). The SVMs are able to determine the optimal separating hyperplane efficiently and they are generally known to produce good classification accuracy even with small training sets (Cao and Tay, 2003).

For linear classification problem, an SVM model makes some data points of a specific class in the feature space and mapped in the way so that the points from the different data classes are divided by a clear gap that is as wide as possible. New data point are then mapped into that same space and predicted to belong to a class based on which

side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification problem using the kernel trick which works by implicitly mapping their inputs into high-dimensional feature spaces.

Given a set of training $(x_i, t_i)_{i=1}^{N}$, where $x_i \in R^d$ and represents the n-dimensional input feature vector and $t_i \in [-1, +1]$ is target output. then the decision function is given by (Foody and Mathur, 2004), (Osuna et al., 1997), (Gunn, 1998);

$$f(x, w, b) = sign(w.x + b) \tag{3.1}$$

The optimal separating hyperplane can be determined by solving the following optimization problem (Begg et al., 2005), (Osuna et al., 1997), (Gunn, 1998);

minimize $\Phi(w) = \frac{1}{2}\|W\|^2$

subject to

$$d_i(w.x + b) \geq 1, i = 1, 2, .., N \tag{3.2}$$

(Haykin, 1999) demonstrates to apply the Lagrangian solve the optimization problem. For the linearly non-separable patterns, the optimization problem can be reformulated as follows (Osuna et al., 1997) , (Chang and Lin, 2001);

minimize

$$\Phi(w, \Xi) = \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{N}\zeta_i \tag{3.3}$$

subject to

$$d_i(w.x + b) \geq 1 - \zeta_i, i = 1, 2, .., N \tag{3.4}$$

where $\zeta_i \geq, i = 1, 2, .., N$

In the above equation, C is the regularization parameter. To extend the above approach into non-linear decision boundaries, the same input space has to be transformed to the high dimensional Euclidian space H, like; $\Phi : R^n \longrightarrow H$ [(Begg et al., 2005), (Haykin, 1999), (Osuna et al., 1997)]. In this high feature space the input vector x is termed as $\phi(x)$. The training algorithm depends on functions of the form $\phi(x).\phi(x_i)$.

The kernel function K should be employed in this way: $K(x, x_i) = \phi(x).\phi(x_i)$. According to [(Osuna et al., 1997), (Gunn, 1998), (Burges, 1998)], the decision function should be,

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i d_i k(x, x_i) + b) \tag{3.5}$$

where $\alpha_i \geq, i = 1, 2, .., N$ are called Lagrange multipliers. One of the main kernel function is gaussian radial basis and is given by

$$K(x, y) = e^{\frac{\|x-y\|^2}{2\sigma^2}} \tag{3.6}$$

In the above equation, σ^2 is kernel parameter or width. The values of two parameters, C and σ^2 influence the classification accuracy of SVM classifier. The binary nature of SVM can be extended to multiclass classification problems. This can be done either by constructing and combining several binary classifiers or by directly considering all data in one optimization formulation (Hsu and Lin, 1998). There are three main mechanisms to solve multiclass classifier problems by combining binary SVMs: One-against-all, One-against-one, and Direct acyclic graph. In our proposed system, we evaluated One-against-one multiclass SVM.

3.1.1 One Against One Multiclass SVMs

The classical SVM method can only classify binary classes: there are several ways to change it into multiclass classification, like; construct a complicated multiclass constrain optimization SVMs, use C-SVMs and one-against-rest approach, use C-SVM and one-against-one approach. We choose the one-against-one approach because of its convenience and shorter training set comparing the one-against-rest approach.

For training data from the ith and jth classes, the following two class classification problem should be solved:

$$min_{w^{ij}, b^{ij}, \zeta_i^{ij}} \frac{1}{2}(w^{ij})^T (W^{ij}) + C \sum_t (\zeta^{ij})_t \tag{3.7}$$

subject to constraints: if x_t belongs to the ith class:

$$(w^{ij})^T \phi(x_t + b^{ij}) \geq 1 - (\zeta^{ij})_t \tag{3.8}$$

Onthe other hand, if x_t belongs to the jth class:

$$(w^{ij})^T \phi(x_t + b^{ij}) \leq 1 - (\zeta^{ij})_t \tag{3.9}$$

and:

$$(\zeta^{ij})_t \geq 0 \tag{3.10}$$

Now the test geometric feature vector x_i, taken from the above mathematics, is the input to the 7*(7-1)/6 SVMs. By applying the voting method, each binary classification contributes to the total votes. Which input vector has the maximum number of votes for being in a particular class will be in that class. In case that if two classes have identical votes, although it is not a good strategy, the multiclass SVM select the one with the smallest index label.

3.2 Extreme Learning Machine

In recent years, Extreme Learning Machine (ELM), proposed by [(Huang et al., 2004), (Huang et al., 2006b), (Huang et al., 2006a), (Huang and Chen, 2007)] is attracting more and more attention because of its outstanding performance in training speed, predicting accuracy and generalization ability [(Huang et al., 2012), (Huang et al., 2010), (Lan et al., 2010a), (Lan et al., 2010b)]. The learning speed of feedforward neural networks is in general far slower than required and it has been a major bottleneck in their applications for past decades (Huang et al., 2006b). According to (Huang et al., 2006b), there may be two reasons behind this situation,(first) the slow gradient-based learning algorithms are extensively used to train neural networks, and (second) all the parameters of the networks are tuned iteratively by using such learning algorithms. Where Extreme Learning Machine (ELM) works for single hidden layer feed-forward neural networks (SLFNs) which randomly chooses hidden nodes and analytically determines the output weights of SLFNs. (Huang et al., 2006b) also claimed very large

complex application prove that ELM can produce a good performance in most cases and can learn thousands of times faster than conventional popular learning algorithms for feedforward neural networks (Huang et al., 2004). Especially it is shown that ELM tends to outperform Support Vector Machine (SVM) in both regression and classification applications with much easier implementation (Huang, 2014). Our purpose here is to analyze the performance of (Huang et al., 2004) ELM on facial expression recognition system as it is a problem of a multi-class classifier.

3.2.1 Brief overview of ELM

ELM is a kind of single hidden layer feed-forward neural networks (SLFN) and it has the universal approximation property (Huang et al., 2006a). In order to have SLFNs work as a universal approximation, one may simply randomly choose hidden nodes and then just train the output weights linking the hidden and the output layer (Jia et al., 2016). Let x be the input, the output of an SLFN with hidden nodes can be represented by

$$O_L(x) = \sum_{i=1}^{L} \beta_i F(m_i, n_i, x), m_i \epsilon R^d, b_n \epsilon R \tag{3.11}$$

where β_i is the weight vector connecting the ith hidden nide to the output nodes, $F(m_i, n_i, x)$ is the output of the ith hidden node, m_i and n_i are the relative parameters of hidden nodes. For additive hidden nodes with activation function f(x), the output of the ith hidden node is given by

$$F(m_i, n_i, x) = f(m + i.x + n_i) \tag{3.12}$$

where m_i is the weight vector connecting the input layer to the ith hidden node and n_i is the bias of the ith hidden node.

If an SLFN with L hidden nodes can approximate N samples (m_k, t_k) with zero error, where $m_k \epsilon R^d$ is the input signal feature vector, and $t_k \epsilon R^d$ is the output target value or category label, it means that there exists β_i, m_i, n_i such that

$$O_L(x_k) = \sum_{i=1}^{L} \beta_i F(m_i, n_i, x_k) = t_k, k = 1, ..., N \tag{3.13}$$

The above equation can be written in a matrix format as

$$\mathbf{H}\beta = \mathbf{T} \tag{3.14}$$

where

$$\mathbf{H} = \begin{bmatrix} F(m_1, n_1, x_1) & ... & F(m_L, n_L, x_1) \\ . & & . \\ . & ... & . \\ . & & . \\ F(m_1, n_1, x_N) & ... & F(m_L, n_L, x_N) \end{bmatrix}_{NxL}$$

,

$$\beta = \begin{bmatrix} \beta_1^T \\ . \\ . \\ . \\ \beta_L^T \end{bmatrix}_{Lxm}$$

and

$$\mathbf{T} = \begin{bmatrix} t_1^T \\ . \\ . \\ . \\ t_N^T \end{bmatrix}_{Nxm}$$

The above equation then can be considered as a linear system and training the SLFN is simply equivalent to find a least squares solution of the linear system. It is proved that the following equation is the unique smallest norm least squares solution to learn and obtain output weight β of this linear system (Huang et al., 2006b):

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{3.15}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the hidden layer output matrix, in which the weight and bias parameters m_i and n_i are randomly asigned.

As stated in (Huang et al., 2012), the orthogonal projection method can be used in ELM: $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ if $\mathbf{H}^T\mathbf{H}$ is non-singular or $\mathbf{H}^{\dagger} = (\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$ if $\mathbf{H}\mathbf{H}^T$ is non-singular. According to the ridge regression theory (Hoerl and Kennard, 1970), it was suggested that a positive value $\frac{1}{\lambda}$ is added to the diagonal of $\mathbf{H}\mathbf{H}^T$ or $\mathbf{H}^T\mathbf{H}$ in

the calculation of the output weights β. The resultant solution is stabler and tends to have better generalization performance. Then the equation 3.15 becomes,

$$\beta = \mathbf{H}^T(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T} \tag{3.16}$$

Then the corresponding output function of ELM becomes:

$$f(x) = h(x)\beta = h(x)\mathbf{H}^T(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T} \tag{3.17}$$

3.2.2 Kernel based ELM

As explained in (Huang et al., 2012), ELM also has a kernel implementation version. ELM normally tries to randomly guess an appropriate hidden layer and uses much bigger hidden layer than needed and potentially prunes back to the minimum necessary. This can be preceded or replaced by a kernel, but if only a kernel is used the correct one must be known to the user in order to be used rather than guessed and trained by a random weighting, pseudo inversion and/or pruning process. The point is that if we already know what a good non-linear kernel is we dont need to guess.

Then one can define a kernel matrix for ELM as follows, where the hidden layer feature mapping $h(x)$ should not be guessed by the users;

$$\Omega_{ELM} = \mathbf{H}\mathbf{H}^T : \Omega_{ELMi,j} = h(x_i).h(x_j) = K(x_i, x_j) \tag{3.18}$$

Then the output of the ELM function that mentioned in equation 3.17, can be re-written as

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{H}^T(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T)^{-1}\ \mathbf{T} = \begin{bmatrix} K(x, x_1) \\ . \\ . \\ . \\ K(x, x_N) \end{bmatrix} (\frac{\mathbf{I}}{\lambda} + \Omega_{ELM})^{-1}\mathbf{T} \tag{3.19}$$

This is a single-step kernel version, where the training set $\aleph = (x_i, t_i)|x_i \epsilon R^d, t_i \varepsilon R^m, i = 1, ..., N$ and kernel= $K(u, v)$ (Huang et al., 2012).

3.3 Euclidian Distance Classifier

Here we use euclidian distance to take the minimum distance from the feature subspace. Euclidian distance is the shortest distance between two points on a plane is a straight line and is known as Euclidean distance as shown in the following equation and is a non-parametric classifier. In the Euclidean distance, the metric difference of each feature of query and database image is squared which effectively increases the divergence between them.

$$d_{Euc} = (A, B) = \sqrt{\sum_{k-1}^{m} |A_k - B_k|^2} \tag{3.20}$$

In many machine learning data matching areas, euclidian distance classifier (EDC) has been proven a successful classifier. For an example, EDC performs as well as or superior to the sample LDF (linear discriminant function) , even for nonspherical covariance configurations (Marco and Turner., 1987).

Chapter 4

Image Pre-Processing

Due to the camera configuration, light and shadow, facial pose, background structure, images need to be normalized. The Cohn-Kanade database greatly varies in illumination and contrast. For illumination adjustment, we did Contrast Adjustment and Histogram Equalization. This chapter covers illumination adjustment as well as face and facial parts detection.

4.1 Illumination Adjustment

The direction where the individual in the image has been illuminated greatly affects face recognition success. A study on illumination effects on face recognition showed that lighting the face bottom up makes face recognition a hard task. So obviously, in that case, facial expression recognition becomes harder too. The CK dataset varies greatly in image brightness. Contrast Adjustment is applied on very light images. JAFFE dataset also varies in lighting and shadow. First, we manually separate the very dark and very light images. Then we apply contrast adjustment procedure on very light images and Histogram Equalization on very dark images. We are the first to separate the internal procedure for contrast adjustment and histogram equalization. For various unknown or technical reasons, when we apply contrast adjustment procedure on very dark images, the images distort. On the other hand, when we apply histogram equalization on very light images, then also the images loss its information.

So from this investigation and also from prediction, we apply contrast adjustment procedure on very light images and Histogram Equalization on very dark images. This process saves the images from being distorted.

4.1.1 Contrast Adjustment

An image lacks contrast when there are no sharp differences between black and white. Brightness refers to the overall lightness or darkness of an image. To change the contrast or brightness of an image, we perform contrast stretching. In this process, pixel values below a specified value are displayed as black, pixel values above a specified value are displayed as white, and pixel values in between these two values are displayed as shades of gray. The result is a linear mapping of a subset of pixel values to the entire range of grays, from black to white, producing an image of higher contrast. The following figure shows this mapping.

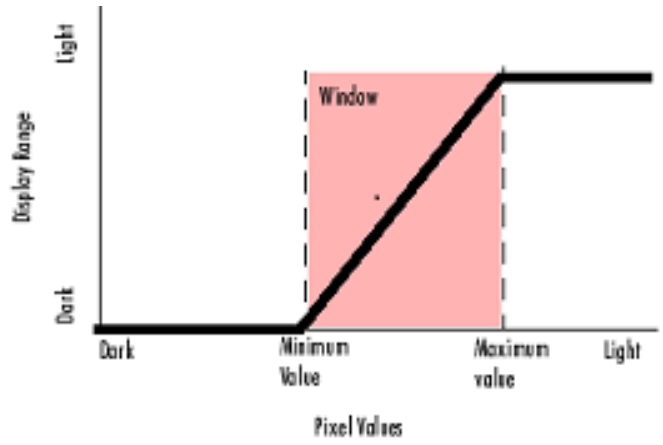

Figure 4.1: How to adjust pixel values in Contrast Adjustment Procedure.

In the Contrast Adjustment procedure, we adjust the contrast of an image by linearly scaling the pixel values between upper and lower limits. Pixel values that are above or below this range are saturated to the upper or lower limit value, respectively.

Mathematically, the contrast adjustment operation is described by the following equation, where the input limits are $[low_{in}, high_{in}]$ and the output limits are $[low_{out}, high_{out}]$.

$$Output = low_{out} + (Input - low_{in})\frac{high_{out} - low_{out}}{high_{in} - low_{in}} \tag{4.1}$$

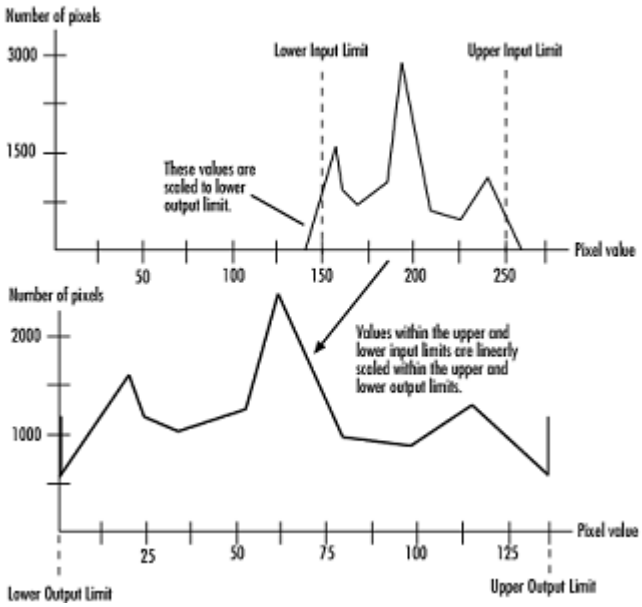

Figure 4.2: Contrast Adjustment Procedure.

4.1.2 Histogram Equalization

Histogram equalization is applied in order to improve the contrast of the images. The peaks in the image histogram, indicating the commonly used gray levels, are widened, while the valleys are compressed. Histogram equalization is a technique for adjusting image intensities to enhance contrast. The process of histogram equalization enhances the contrast of images by transforming the values in an intensity image, or the values in the color map of an indexed image so that the histogram of the output image approximately matches a specified histogram.

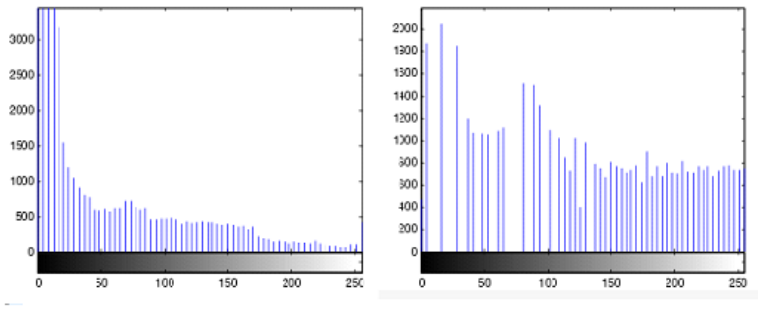

Figure 4.3: (left) Histogram of a very dark image and (right) the same image with equal histogram.

Two histogram plots are given in the above Figure. The histogram on the left is before histogram equalization (between 6-250) is applied and the one on the right

is after histogram equalization is applied. After separating the very dark images, histogram equalization is applied to the JAFFE and extended cohn-kanade databases automatically.

When a desired histogram is supplied, histogram equalization procedure chooses the gray scale transformation T to minimize

$$|cum_1(T(j)) - cum_0(j)|, \tag{4.2}$$

where the cumulative histogram of A is defined by cum_0, the cumulative sum of the histogram for all intensities j is defined by cum_1. There is a constraint that T must be monotonic and $cum_1(T(a))$ cannot overshoot $cum_0(T(a))$ by more than half the distance between the histogram counts at a. The above minimization process is subject to this constraint. Histogram equalization procedure uses the transformation $b = T(a)$ to map the gray levels in X or the colormap to their new values.

4.2 Face Detection

Paul Viola and Michael Jones proposed the Viola-Jones object detection framework in 2001. This object detection algorithm provides competitive object detection rates in real-time, which is approximately 15 times faster than any previous approach. It was motivated primarily by the problem of face detection, although it can be trained to detect a variety of object classes. This face detection framework claimed that they have three main contributions. In this thesis, we will briefly discuss these three contributions. The first contribution of the Viola-Jones face detection framework is a new representation of image called integral image, which allows the features computations very quickly.

4.2.1 Features and Integral Image

Being inspired from the research outcomes for using features rather than the pixels, this face detection framework uses rectangular features to identify faces. This process mainly uses three kinds of features. The difference between the sums of the

pixels within two rectangular regions is the numeric value of a two-rectangle feature. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangle feature performs the summation within two outside rectangles subtracted from the sum in a center positioned rectangle. Finally, a four-rectangle feature computes the difference between diagonal pairs of the rectangles. By applying integral

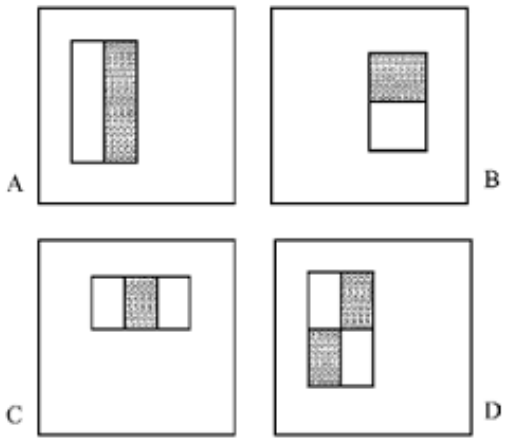

Figure 4.4: (left) Example rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles is subtracted from the sum of pixels in the gray rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature, Replicated from (Viola and Jones, 2004).

image, which is an intermediate representation for the images, rectangular features can be computed very quickly. The integral image at location x, y contains the sum of the pixels above and to the left of x, y, which is:

$$ii(x,y) = \sum_{x' \leqq x, y' \leqq y} i(x',y') \tag{4.3}$$

where $ii_{(x,y)}$ is the integral image and $i_{(x,y)}$ is the original image. The following equations are the recurrences:

$$t(x,y) = t(x,y-1) + j(x,y) \tag{4.4}$$

$$j(x,y) = jj(x-1,y) + t(x,y) \tag{4.5}$$

where $t(x, y)$ is the cumulative row sum, $t(x, -1) = 0$, and $jj(-1, y) = 0)$ the integral image can be computed in one pass over the original image. By applying this integral image, feature extraction process speeds up very rapidly.

4.2.2 AdaBoost Learning Algorithm

The second contribution of (Viola and Jones, 2004) is a simple and efficient classifier that is built by selecting a small number of important features from a huge library of potential features. This classifier is built using the AdaBoost learning algorithm (Freund and Schapire, 1995). Boosting performs a classification technique, which processes by combining weak learners into a constructive ensemble classifier. Adaboost iteratively combines the classifiers from a linear combination of the weak classifiers. At first, it gives equal weight to each training example. Then it performs the iterative training procedure by raising the weights of misclassified sample by the associated weak learner. Thus it increases the speed as well as accuracy of the feature selection process.

4.2.3 Cascaded Classifier

Combining successively more complex classifiers in a cascade structure is the third major contribution of the Viola-Jones face detection framework. This cascaded classifier quickly discards background regions and spends more computation on face-like regions and thus it increases the detection rate to a great extent. A cascade structure does it job as recursive and degenerated decision tree. At each stage, either the process denies a specific feature and the process stops, or the classifier accepts the feature and forwards it to the immediate next stage. The training classifier applying Adaboost structures the inner stages in the cascade classifier. Research shows that the chain of classifiers is structurally more complex as well as possess low false positive rates.

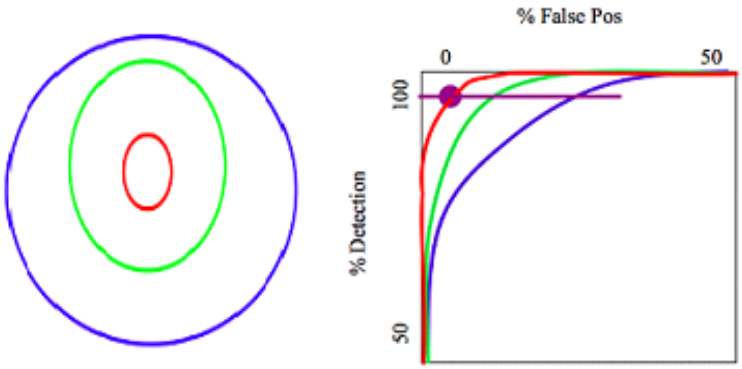

Figure 4.5: Cascade of Classifiers (left) and ROC curve shows how the accuracy is improving by the cascaded architecture.

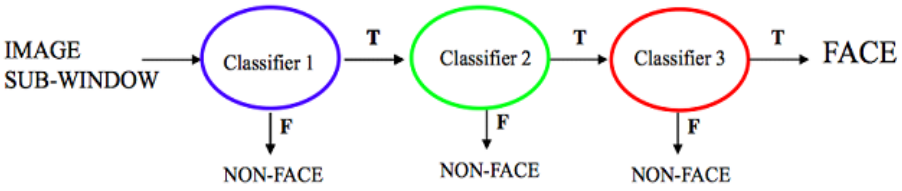

Figure 4.6: A diagram of the cascaded classifier. A pool of classifiers is applied to every sub-window. The initial classifier eliminates a large number of negative examples with very little pro- cessing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing the num- ber of sub-windows have been reduced radically. Further processing can take any form such as additional stages of the cascade (as in our detection system) or an alternative detection system (Viola and Jones, 2004).

Chapter 5

Experimental Setup

5.1 Dataset

We benchmark our results on the Cohn-Kanade (CK) and JAFFE datasets. We used both the datasets for facial expression recognition.

The Cohn-Kanade dataset has a variable number of images per expression. For each expressed emotion we have a sequence from neutral face images. In the CK dataset, there are 110 images from anger expression, 102 images from disgust, 152 images from fear, 101 images from happy, 110 images from neutral, 110 images from sad and 100 images from surprised expression. In total, we have 785 images in CK dataset. In this dataset, Sixty-five percent were female, 15 percent were African-American, and three percent were Asian or Latino. The observation room was equipped with a chair for the subject and two Panasonic WV3230 cameras, each connected to a Panasonic S-VHS AG-7500 video recorder with a Horita synchronized time-code generator. One of the cameras was positioned directly in front of the subject, and the other was located 30 degrees to the right of the subject. Then the images were digitized into 640 by 480 or 490 pixel arrays with 8-bit precision for grayscale values.

In the JAFFE dataset, each of the ten subjects posed for 3 or 4 examples of each of the six basic or distinctive facial expressions (happiness, sadness, surprise, anger, disgust, fear) as well as a neutral face expression. Altogether JAFFE has 219 facial images and we use 210 images which include 30 images from each expression (30x7=210 images).

In the JAFFE set, each subject took pictures of herself while looking through a semi-reflective plastic sheet towards the camera. To create even illumination on the frontal face tungsten lights were used and to decrease the back reflection a box enclosing the region between the plastic sheet and the camera was used.

The following two figures are a portion of the two datasets, the CK and the JAFFE dataset, which we feed in our proposed repeated and nested k-fold cross-validation based Facial Expression Recognition system. We apply contrast adjustment procedure on very light images and Histogram Equalization on very dark images on both datasets before doing face detection. This process makes all the images of same contrast and brightness.

Figure 5.1: Cohn Kanade dataset.

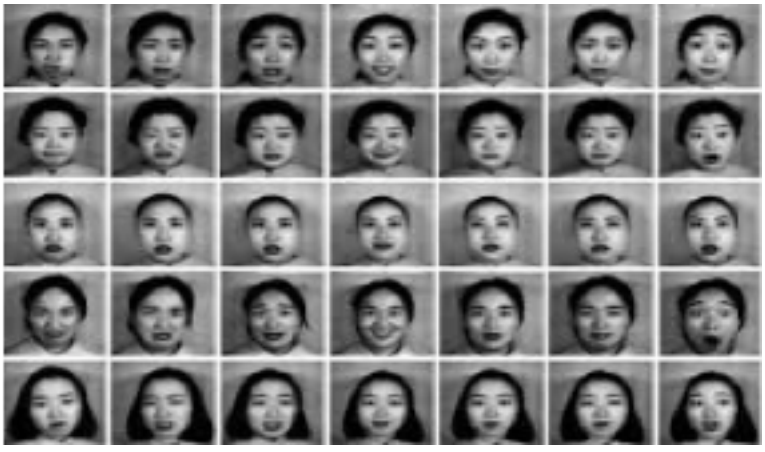

Figure 5.2: Data from JAFFE dataset.

5.2 Face and Facial Parts Detection

The CK dataset varies greatly in image brightness. For image pre- normalization procedure, first, we use Contrast Adjustment to enhance the image from very light images. Then to improve the contrast of the very dark images we apply Histogram Equalization. In CK dataset, the background is large with all the face images. For face detection, we apply the Viola-Jones algorithm (Paul and Jones, 2001). The face detection algorithm gives a green bounding box as shown in figure 5.4. The same procedures are applied to JAFFE dataset.

The three major steps of **face detection** methods are stated below.

- Integral images for fast feature evaluation.
- Boosting for feature selection.
- Attentional cascade for fast rejection of non-face windows.

For eyes, nose and mouth detection, we applied cascaded object detector with region set on already detected frontal faces (Fig. 5.4). This cascade object detector with proper region set can identify the eyes, nose and mouth. Our proposed model for facial parts detection determines the locations and sizes of human facial parts by extracting region of interests and then apply Viola-Jones object detector. This region set property limits the facial parts search area. Thus this proposed model reduces the computation time to a great extent as well as this model increases the detection rate than the straight Viola-Jones algorithm.

Basically, this object detection algorithm uses a cascade of classifiers to efficiently process image regions for the presence of a target object. Each stage in the cascade applies increasingly more complex binary classifiers, which allows the algorithm to rapidly reject regions that do not contain the target. If the desired object is not found at any stage in the cascade, the detector immediately rejects the region and processing is terminated. The model for **facial parts detection** region set has been shown in figure 5.3.

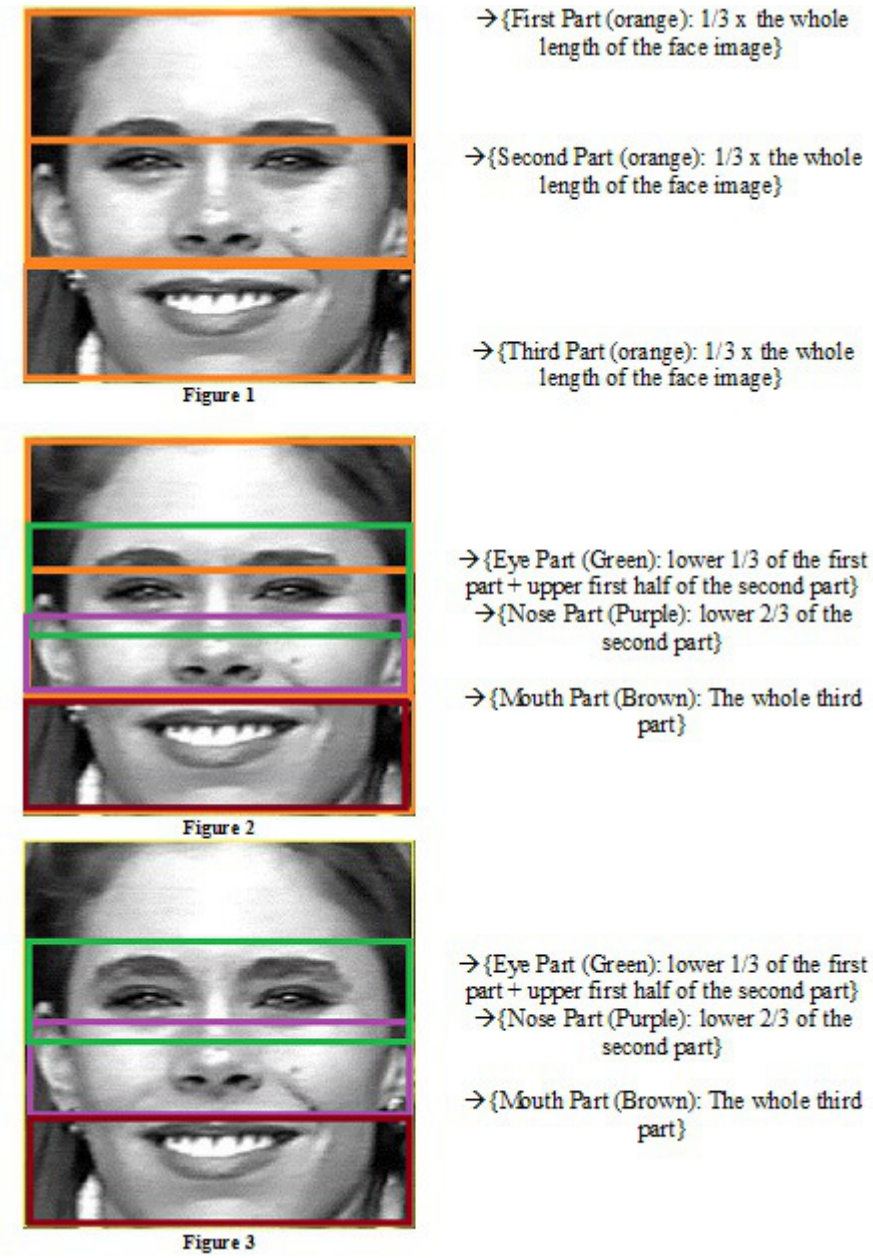

Figure 5.3: Proposed model for the region of interest for facial parts detection.

We perform face and facial parts detection on 785 images from seven expression classes in CK dataset and 210 images from the same seven expression classes on JAFFE dataset. As shown in figure 5.3 the **facial parts detection** area are divided into three regions interest calculated using the following equations.

- Step 1: First we apply the straight Viola-Jones face detection algorithm on our facial expression dataset with the large background. The outcome is the detected face surrounded by an orange bounding box in figure 5.3. Then we have divided

the detected face by three regions. Figure 1 of figure 5.3, shows the equally divided three sections.

- Step 2: In figure 2 of figure 5.3, we divided the model into new three parts again. Where

  Eye part (Green)= Lower 1/3 of the first part from the previous step + upper 1/2 part of the second part from the previous step.

  Nose part (Purple)=Lower 2/3 of the second part.

  Mouth part (Brown)= The whole third part which is the 1/3 of the whole detected face area.

- Step 3: Then we apply the Viola-Jones eye-detector in the Eye part, Viola-Jones nose-detector in the nose part, Viola-Jones mouth-detector in the mouth part. The facial parts detection outcome is depicted in following figure 5.4.

Figure 5.4: Face and Facial Parts Detection.

5.3 Cross-Validation: Splitting Train and Test Data

5.3.1 Holdout

The conventional holdout strategy is applied to measure the classifier performance when the amount of data for training and testing is limited. For example, the holdout

method holds one-third of the data for testing and use the remaining two-thirds for training. The training corpus is only used to train the model, while the testing corpus is only used to estimate the performance of the model.

In the holdout method, this is uncertain whether a sample, partitioned for testing and training, is representative or not. To overcome this issue, each class in the full dataset should be represented in about the right proportion in the training and testing sets. If all examples with a certain class were omitted from the training set it is hardly expected a classifier learned from that data to perform well on examples of that class, and the situation would be exacerbated by the fact that the class would necessarily be overrepresented in the test set because none of its instances made it into the training set. The solution from this obstacle is Stratification, which ensures that the random sampling is done in a way that guarantees that each class is properly represented in both training and test sets. While it is generally well worth doing, stratification provides only a primitive safeguard against uneven representation in training and test sets (Witten and Frank, 2005).

Research studies (Efron and Gong, 1983), (Efron and Tibshirani, 1994), (Steyerberg, 2008), (Stone, 1974) has shown that holdout validation is statistically inefficient because much of the data is not used to train the prediction model. Moreover, an unfortunate split of the training and testing corpora may cause the performance estimate of holdout validation to be misleading. To reduce the bias and variance research works of (Tarvo, 2008), (Nagappan et al., 2008), (Zimmermann and Nagappan, 2008) suggested and proves that holdout validation method should be applied in a repeated fashion (Tantithamthavorn et al., 2015).

The holdout method is repeated multiple times with crossover from the within and heldout sets. On the otherhand, in cross-validation data has to be split into a fixed number of folds or partitions. The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made (Schneider, 1997).

5.3.2 Cross-Validation

Cross-validation extends the idea of holdout validation by repeating the splitting process several times. A cross-validation estimate is a random number that depends on the division into folds. For an example, if the number of folds is n, then the dataset is split into n approximately equal partitions; each, in turn, is used for testing and the remainder is for training. Which means, $(n-1)/n$ of the data for training and $1/n$ for testing. This is called n-fold cross validation and if stratification is adopted as well, then it is called stratified n-fold cross validation.

k-fold Cross-validation

k-fold cross validation is one way to improve over the holdout method (Schneider, 1997). The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once and gets to be in a training set $k-1$ times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set k different times.

In k-fold cross validation we create a k-fold partition of our samples such as $k-1$ sets of observations are used as training examples and $n-k$ sets of observations are used as training data iteratively. The advantage of k-fold Cross- validation is that all the examples in the dataset are eventually used for both training and testing.

Leave-one-out cross-validation

Leave-one-out cross validation is k-fold cross validation taken to its logical extreme, with k equal to n, the number of data points in the set. That means that n separate

times, the function approximator is trained on all the data except for one point and a prediction is made for that point (Schneider, 1997). As before the average error is computed and used to evaluate the model. The evaluation is given by leave-one-out cross-validation error (LOO-CV) is good, but at first pass, it seems very expensive to compute. This method not only fully utilizes the available data, but also eliminates the influence of choices of random pairing. Thus, the LOO-CV is a nearly unbiased and reliable method. The computational requirement for the LOO seems daunting at a glance since it requires n training cycles for evaluating a single parameter (Shao et al., 2015).

10-fold cross-validation

According to (Witten and Frank, 2005), extensive tests on numerous different datasets with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up. (Witten and Frank, 2005) commented that the standard way of predicting the error rate of a learning technique given a single, fixed sample of data is to use stratified ten-fold cross-validation. The data is divided randomly into 10 parts in which the class is represented in approximately the same proportions as in the full dataset. Each part is held out in turn and the learning scheme trained on the remaining nine-tenths; then its error rate is calculated on the holdout set. Thus, the learning procedure has executed a total of 10 times on different training sets (each set has a lot in common with the others). Finally, the 10 error estimates are averaged to yield on overall error estimate. Although these arguments are by no means conclusive, and debate continues to rage in machine learning and data mining circles about what is the best scheme for evaluation, 10-fold cross validation has become the standard method in practical terms(Witten and Frank, 2005). It is also proved from different tests that the use of stratification improves results slightly. So for the limited data, stratified 10-flod cross validation is a standard method. In some cases 5 or 20- fold cross validation is likely to be almost good.

However, a single ten-fold cross-validation might not be enough to get a reliable error estimate. Different ten-fold cross-validation experiments with the same learning

scheme and dataset often produce different results because of the effect of random variation in choosing the folds themselves (Witten and Frank, 2005). Although stratification can reduce the variation but it does not eliminate it entirely. According to (Witten and Frank, 2005), to measure an accurate error rate, the standard method is 10 times 10-flod cross-validation and average the results. This involves invoking the learning algorithm 100 times on datasets that are all nine-tenths the size of the original. To enhance a good measure of performance is a computation-intensive undertaking.

While the cross-validation technique is known to be nearly unbiased, some studies [(Isaksson et al., 2008), (Braga-Neto and Dougherty, 2004)] find out that it can produce unstable results for small datasets. To overcome this shortcoming and to improve the variance of cross-validation results, the entire cross-validation process can be repeated several times.

For our proposed system, for facial expression recognition, we implemented our system based on 3x10- fold cross validation. For any systems performance, execution time is an issue, so we use 3x10 fold cross validation rather than 10x10 or more outer folds. and we have reasonable results and in some context, it outperforms some predominant FER systems. Moreover, we also perform 10x10 fold Nested Cross-Validation (NCV).

Chapter 6

Proposed Approach

Automatic face and emotion recognition has been attracting the attention of researchers from several areas including computer vision, psychology, behavioral science, computer games and medicine (Pantic and Rothkrantz, 2000). But it is really a hard problem to recognize face and facial expression with a very high accuracy (Kapoor and Picard, 2001), (Picard, 1997), (Izard, 1979) and (Cottrell and Metcalfe, 1991). There are lots of challenges and critical issues in the domain of face and facial expression recognition.

Among the critical challenges, there is one of the main concerns over whether it is best to use whole or part based image analysis. Sometimes different facial parts of the face are optimal in terms of processing time and accuracy than holistic faces for identifying an expression. We propose here,

- Repeated Cross-Validation based approach for FER system using Whole face and three main facial parts (eyes, nose and mouth).
- Nested Cross-Validation based approach for FER system to find the best space-classifier combinations on the whole face, three main facial parts and all possible combinations of the facial parts.

Many machines learning kinds of research show that Principal Component Analysis (PCA), Independent Component Analysis(ICA) and Non-negative matrix factorization (NMF) are useful decompositions for multivariate data like face and facial expression

recognition. Principal Component Analysis is a linear dimensionality reduction technique: it transforms the data by a linear projection onto a lower-dimensional space that preserves as much data variation as possible. The Principal Component Analysis (PCA) is performed by the Karhunen-Loeve transform produces features, that are mutually uncorrelated. The solution obtained by the KL transform solution is optimal when dimensionality reduction is the goal and one wishes to minimize the approximation mean square error. However, for certain applications, the obtained solution falls short of the expectations. In contrast, the more recently developed Independent Component Analysis (ICA) theory tries to achieve much more than simple decorrelation of the data. According to research studies (Lee and Seung, 2009) it is clear that NMF can be understood as part based analysis as it decomposes the matrix only into additive parts. PCA, ICA, NMF all these subspace learning techniques reduces the dimension and make a distributed represented in which each facial image can be approximated using linear combinations of all or selected basis images. Although the underlying method of the histogram of oriented gradients (HOG) is not similar to PCA, ICA, and NMF. It has been successfully used in In the area of Image Processing and Computer Vision in recent years. HOG has been successfully implemented in pattern recognition as a feature descriptor. The underlying methods of HOG have similarity with scale-invariant feature transform descriptors, shape contexts and edge orientation histograms. As we are using PCA, ICA and NMF which are mainly applied for dimension reduction and HOG works as SIFT descriptors, we will use the term 'space' for the feature extraction processes. Altogether we investigate the performance of sixteen space and classifier combinations to make a comparison of the FER system.

We benchmark our system on CK and JAFFE dataset using full face and four facial parts. For our feature extraction, we use HOG, PCA, NMF and ICA and as classifiers, we apply Euclidian Distance (ED), Extreme Learning Machine(ELM), Extreme Learning Machine Kernel (ELM Kernel) and Support Vector Machine (SVM).

6.1 Repeated Cross-Validation based approach for FER system using Whole face and three main facial parts (eyes, nose and mouth).

In this section, our **first objective** is to analyze the comparison of facial expression recognition based on the whole face and part based faces. In our experimental setup, first, we detect the face using Viola-Jones face detection algorithm. Then we apply PCA, ICA, NMF and HOG on whole faces for facial expression recognition to produce four feature sets. The extracted feature sets are then passed on four classifiers, like; Euclidian Distance(ED), Support Vector Machine(SVM), Extreme Learning Machine(ELM) and Extreme Learning Machine Kernel (ELM Kernel). Then the system produces twelve feature-classifier combinations which are PCA+ED, PCA+SVM, PCA+ELM, PCA+ELM Kernel, ICA+ED, ICA+SVM, ICA+ELM, ICA+ELM Kernel, NMF+ED, NMF+SVM, NMF+ELM, NMF+ELM Kernel, HOG+ED, HOG+SVM, HOG+ELM and HOG+ELM Kernel. We carefully design a 3x10 fold RCV (3x Repeated 10-fold CV), which is repeated 10 fold, cross-validation to evaluate the performance of the system and make a comparison among the space-classifier combinations.

So our **second objective** is to analyze the space-classifiers performance on three main facial parts (eyes, nose, mouth). We divide the whole face into three main facial parts and follow the same way as described above. The whole process in described in the flow chart in figure 6.1 and the associated pseudocode is in Algorithm 1 .

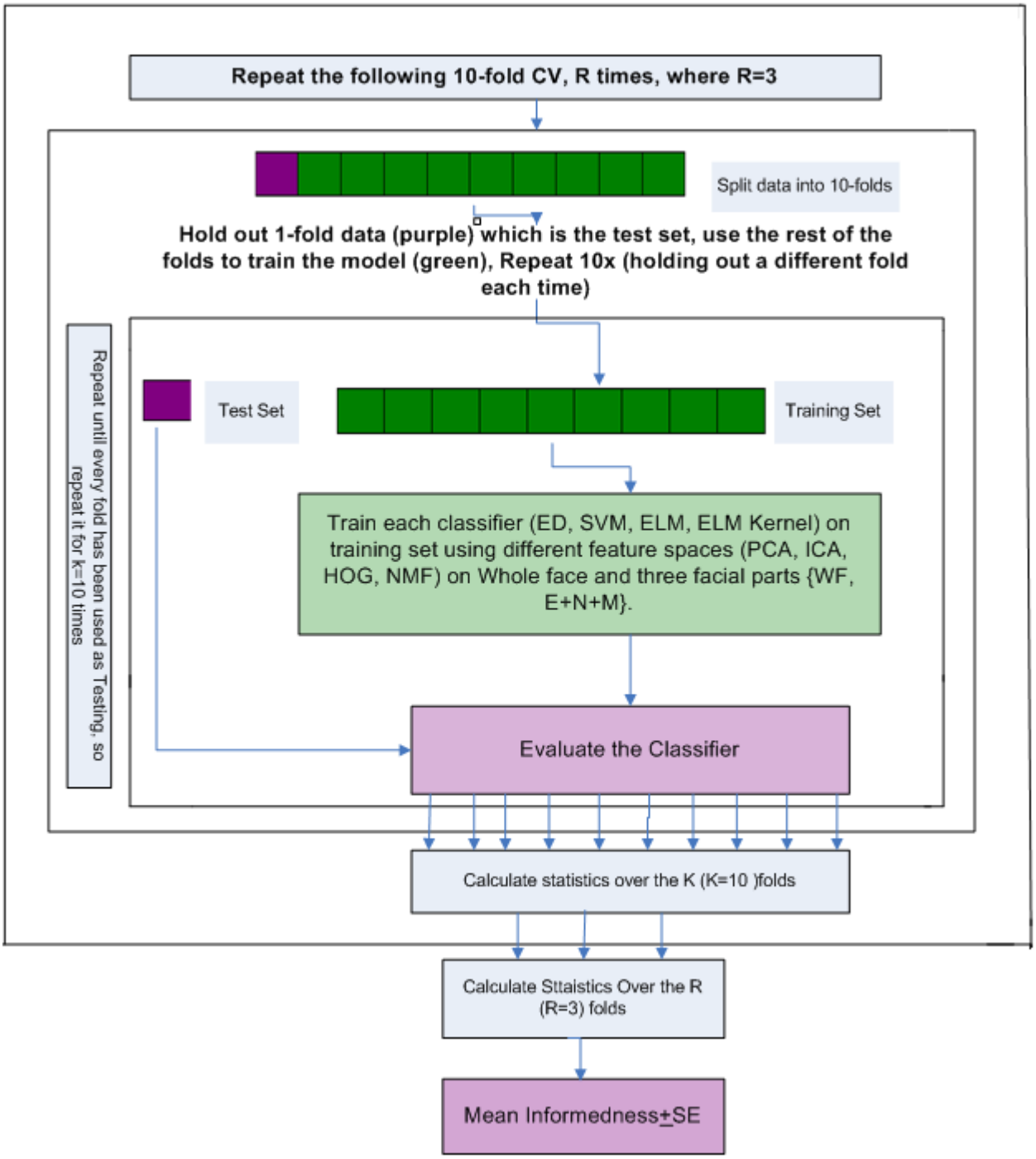

Figure 6.1: Our proposed Repeated K-fold Cross-Validation approach for FER system.

Algorithm 1 FER using Repeated K-fold Cross-Validation over specified feature sets.

0: **Initialization**

0: *Ps* ← *[(WF) and (E,N,M)]*

0: *E* ← *[Anger, Disgust, Fear, Happy, Neutral, Sad, Surprised]*

0: *C* ← *[ED, SVM, ELM, ELM Kernel]*

0: *F* ← *[NMF, PCA, ICA, HOG]*

0: **for** `<R=1 to 3>` **do**{R is for Repeated cross-validation}

0: **for** `<k=1 to 10>` **do**{k is for k-fold cross-validation}

0: Let WP be a feature extractor for $FP = F(P)$ in (k-1) data part.

0: Train FP on classifier set C in (k-1) data part.

0: Let FC be F trained input FP to predict labels E.

0: Evaluate the classifier on the held out test dataset $(1/k)$. .

0: Calculate the statistics S over k=10 folds on the powerset of (WP, FC).

0: Calculate the statistics$S$ over R=3 folds on the powerset of (WP, FC).

0: Return all S for all the combination of (WP, FC). =0

6.2 Nested Cross-Validation based approach for FER system to find the best space-classifier combinations on the whole face, three main facial parts and all possible combinations of the facial parts.

Our prediction is for some cases even less part of faces may perform better than the three facial parts. To prove these predictions, we implement a 10x10 fold N-CV based FER system. In this approach, we use whole face (WF), three facial parts (eyes, nose and mouth which we denote as (E+N+M)) and the all possible combinations of the three facial parts, which are, Eyes (E), Nose (N), Mouth (M), Eyes + Nose (E+N), Eyes + Mouth (E+M), Mouth +Nose (M+N) as facial features. then we benchmark our proposed N-CV analysis on CK and JAFFE dataset. The results are shown in the

following two tables.

Already by using three facial parts (i.e., eyes, nose and mouth), we are reducing some facial features and hence the system needs less memory for calculations. Our one the main objectives, is to increase the accuracy as well as decrease the calculation time. So we reduced some facial features, like; the top of the forehead, two sides of the cheeks and took the main three facial parts which are prominent parts to emote the basic expressions. From the analysis and prediction of some face parts may perform better than the full face or all three facial parts, we implemented a nested cross-validation (N-CV) basis FER system.

The following algorithm 2 is the pseudocode for proposed approach. Also, we provide a flow chart of this nested Cross-Validation based approach in the flow chart 6.2.

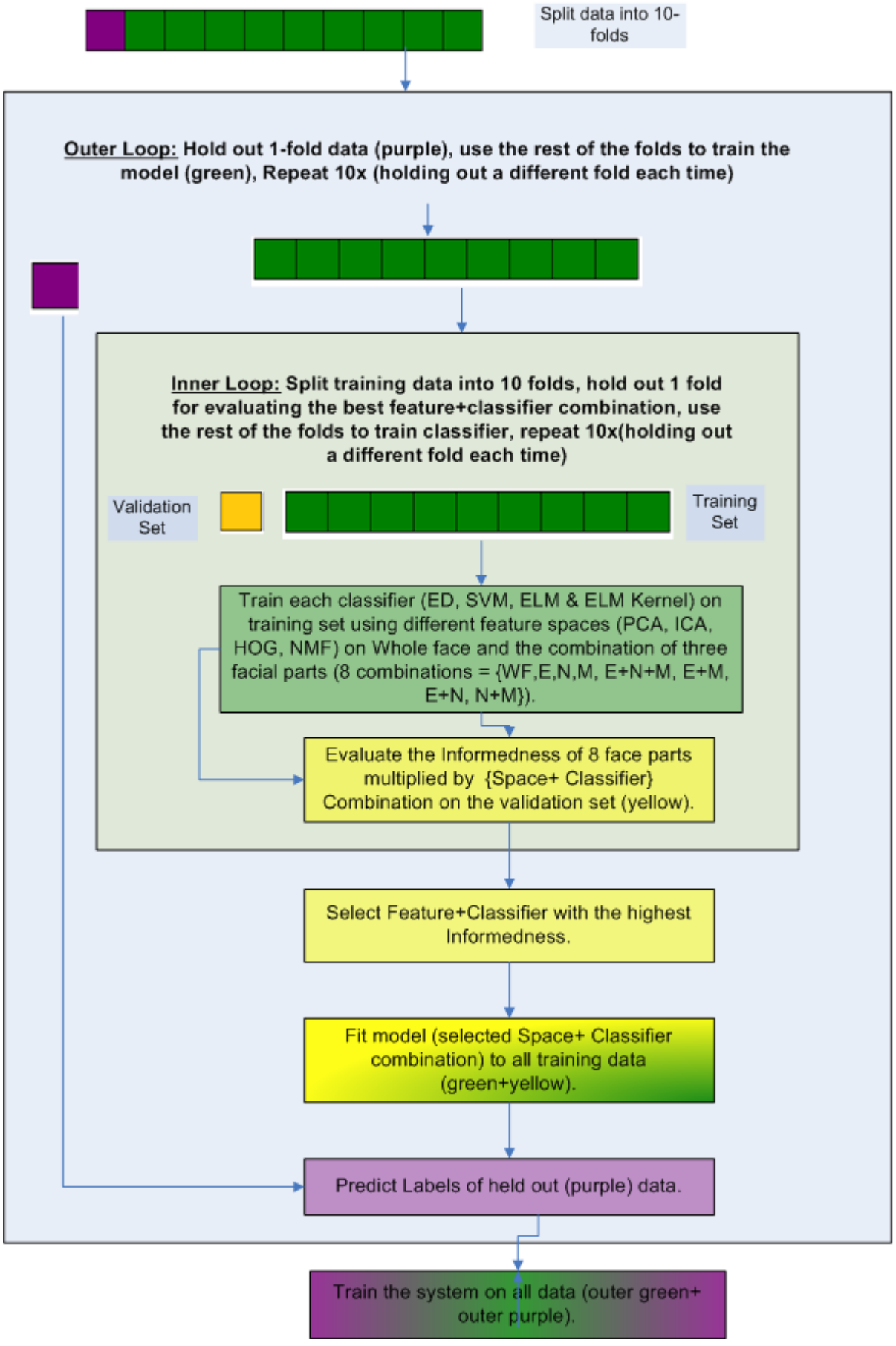

Figure 6.2: Our proposed Nested k-fold Cross-Validation approach for FER system.

Algorithm 2 FER using Nested K-fold Cross-Validation to select feature sets.

0: **Initialization**

0: $Ps \leftarrow$ *[(WF) and (E,N,M), (E), (N), (M), (E,N), (E,M), (M,N)]*

0: $E \leftarrow$ *[Anger, Disgust, Fear, Happy, Neutral, Sad, Surprised]*

0: $C \leftarrow$ *[ED, SVM, ELM, ELM Kernel]*

0: $F \leftarrow$ *[NMF, PCA, ICA, HOG]*

0: **for** each P from Ps **do**

0: **for** `<N=1 to 10>` **do**{N is for Nested cross-validation}

0: **for** `<k=1 to 10>` **do**{k is for k-fold cross-validation}

0: Let WP be a feature extractor for $FP = F(P)$ in ((k-1)-1) data part.

0: Train FP on classifier set C in (k-1) data part.

0: Let FC be F trained input FP to predict labels E.

0: Evaluate the classifier on the validation dataset.

0: Calculate the statistics S over k=10 folds on the powerset of (WP, FC).

0: Return the $argmax(F, C)$ for the highest Informedness from S.

0: Train the test set using the returned (F, C).

0: Calculate the statistics S over N=10 folds from test set.

0: Return $argmax(F, C)$ with the highest Informedness from S for each Ps. =0

In the next chapter, we investigated the performances of the sixteen feature-classifier combinations for both full face and part face based FER system. For some feature-classifier combinations, full face performs better than the part based faces where as for some the output is vice versa. Also our result shows that the performances are dataset dependent. For performance evaluation, we propose here kappa statistics, correlation and informedness besides accuracy.

Chapter 7

Facial Expression Recognition: Performance Evaluation

On a systematic trial and error basis, we set some hyperparameter values for ELM and ELM Kernel for PCA. In the case of ELM classifier, using trial and error basis, we found that for 'sigmoid activation function with 100 hidden number of neurons work better than other combinations. So we use here 'Sigmoid activation function with 100 hidden neuron numbers. The Elm Kernel was performing with very low informedness, we investigated whether the main issue is that Elm Kernel does not work properly when (near) singular. But the underlying PCA method decomposes the whole input data into three decomposed matrix [U S V], where S is the diagonal matrix of singular values and U and V are rotations into a latent space that is reduced if (near) singular. So we took the Pseudoinverse of the 'singular matrix and increases the performance to a great extent. Using the same method with PCA more generally also works better. Optimizing the hyperparameter using nested cross-validation, with ELM and ELM Kernel will be our future work.

For SVM, we use here 'Linear Kernel. Optimization of kernels is beyond the scope of this thesis. This can be one of our future contributions.

On a systematic trial and error basis, we set some of the hyperparameter values for ELM and ELM Kernel for ICA. Like PCA, we found the same combination for ICA which is 'sigmoid activation function with 100 hidden number of neurons works better

than other combinations in ELM classifier using ICA. In the case of matrix decomposition, ICA performs PCA decomposition first to get the independent components A and decorrelate weight W. Basically, the PCA U (or V) matrix is divided by the singular matrix S, which is then multiplied by the random initial weights to get the initial estimate of the decorrelated weights. So converting the diagonal matrix of the singular values into Pseudoinverse for ICA feature extraction, in the case of ELM classifier reduces the performance of the output. On the other hand in case of ELM Kernel (to use with ICA), converting the diagonal matrix of the singular values into Pseudoinverse does not change the accuracy. So we use 'pinv(S)= No for ELM and ELM Kernel while using ICA.

7.1 Performance Metrics

(Powers, 2012) claimed that the traditional evaluation measures used in Computational Intelligence (including Error Rates, Accuracy, Recall, Precision and F-measure) are of limited value for unbiased evaluation of systems, and are not meaningful for comparison of algorithms unless both the dataset and algorithm parameters are strictly controlled for skew (Prevalence and Bias).

Here we will show some evaluation techniques which are highly dependent on the assumptions made about the distributions of the dataset and the underlying populations. The author of (Powers, 2012) commented that Research in Computational Linguistics usually requires some form of quantitative evaluation. A number of traditional measures borrowed from Information Retrieval (Manning and Schutze, 1999) are in common use but there has been considerable critical evaluation of these measures themselves over the last decade or so (Entwisle and Powers, 1998), (Flach, 2003), (Ben-David, 2008a), (Ben-David, 2008b). Receiver Operating Analysis (ROC) has been advocated as an alternative by many, and in particular, has been used by (Furnkranz and Flach, 2005), (Ben-David, 2008a), (Ben-David, 2008b), (Powers, 2008) to better understand both learning algorithms relationship and the between the various measures, and the inherent biases that make many of them suspect (Powers, 2012). One of the key advantages of ROC is that it provides a clear indication of chance level

performance as well as a less well-known indication of the relative cost weighting of positive and negative cases for each possible system or parameterization represented (Powers, 2012).

Studies that measure the agreement between two or more observers should include a statistic that takes into account the fact that observers will sometimes agree or disagree simply by chance (Viera and Garrett, 2005). The kappa statistic (or kappa coefficient) is the most commonly used statistic for this purpose. A kappa of 1 indicates perfect agreement, whereas a kappa of 0 indicates agreement equivalent to chance. A limitation of kappa is that it is affected by the prevalence of the finding under observation.

There are other methods of assessing inter observer-agreement, but kappa is the most commonly reported measure in the medical literature. Kappa makes no distinction among various types and sources of disagreement. Because it is affected by prevalence, it may not be appropriate to compare kappa between different studies or populations. Nonetheless, kappa can provide more information than a simple calculation of the raw proportion of agreement (Viera and Garrett, 2005).

7.1.1 Two classes and non-negative Kappa

Kappa was originally proposed (Cohen, 1960) to compare human ratings in a binary, or dichotomous, classification task. (Cohen, 1960) recognized that Rand Accuracy did not take chance into account and therefore proposed to subtract off the chance level of Accuracy and then renormalize to the form of a probability.

$$K(Acc) = [AccE(Acc)]/[1E(Acc)] \tag{7.1}$$

This leaves the question of how to estimate the expected Accuracy, E(Acc). (Cohen, 1960) made the assumption that raters would have different distributions that could be estimated as the products of the corresponding marginal coefficients of the contingency table.

7.1.2 Accuracy

Statistics relative to (the total numbers of items in) the real classes are called Rates and have the number (or proportion) of Real Positives (RP) or Real Negatives (RN) in the denominator. In this notation, we have $Recall = TPR = TP/RP$.

Conversely statistics relative to the (number of) predictions are called Accuracies, so relative to the predictions that label instances positively, Predicted Positives (PP), we have $Precision = TPA = TP/PP$.

The accuracy of all our predictions, positive or negative, is given by $RandAccuracy = (TF + TN)/N = tf + tn$, and this is what is meant in general by the unadorned term Accuracy, or the abbreviation Acc. Rand Accuracy is the weighted average of Precision and Inverse Precision (probability that negative predictions are correctly labeled), where the weighting is made according to the number of predictions made for the corresponding labels. Rand Accuracy is also the weighted average of Recall and Inverse Recall (probability that negative instances are correctly predicted), where the weighting is made according to a number of instances in the corresponding classes.

Cohen assumes that their distribution of ratings is independent, as reflected both by the margins and the contingencies: $etp = rp * pp; etn = rn * pn$. This gives us $E(Acc) = etp + etn = (ETP + ETN)/N$. By contrast, the two raters two class form of (Fleiss, 1981) Kappa, also known as $ScottPi$, assumes that both raters are labeling independently using the same distribution and that the margins reflect this potential variation. The expected number of positives is thus effectively estimated as the average of the two raters counts so that $EP = (RP+PP)/2, andEN = (RN+PN)/2, ETP = EP2andETN = EN2$.

7.1.3 Multiclass multi-rater Kappa

(Fleiss, 1981) and others sought to generalize the (Cohen, 1960) definition of Kappa to handle both multiple class (not just positive/negative) and multiple raters (not just two, one of which we have called real and the other prediction). Fleiss in fact generalized Scotts (Scott, 1955) Pi in both senses, not Cohen Kappa. The Fleiss

Kappa is not formulated as we have done here for exposition but in terms of pairings (agreements) amongst the raters, who are each assumed to have rated the same number of items, N, but not necessarily all.

7.1.4 Powers Informedness

(Powers, 2003a) derived a further multiclass Kappa-like measure from first principles, dubbing it Informedness, based on an analogy of Bookmaker associating costs/payoffs based on the odds. This is then proven to measure the proportion of time (or probability) a decision is informed versus random, based on the same assumptions re expectation as Cohen Kappa, and we will thus call it Powers Kappa, and derive a formulation of the corresponding expectation. (Powers, 2011) further identifies that the dichotomous form of Powers Kappa is equivalent to the Gini coefficient as a deskewed version of the weighted Relative Accuracy proposed by (Flach, 2003) based on his analysis and deskewing of common evaluation measures in the ROC paradigm. (Powers, 2011) also identifies that Dichotomous Informedness is equivalent to an empirically derived psychological measure called DeltaP (Perruchet and Peereman, 2004) DeltaP (and its dual DeltaP) were derived based on analysis of human word association data, the combination of this empirical observation with the place of DeltaP as the dichotomous case of Powers Informedness suggests that human association is in some sense optimal.

(Powers, 2011) also introduces a dual of Informedness that he names Markedness and shows that the geometric mean of Informedness and Markedness is Matthews Correlation, the nominal analog of Pearson Correlation (Powers, 2012). Powers Informedness is, in fact, a variant of Kappa with some similarities to Cohen Kappa, but also some advantages over both Cohen and Fleiss Kappa due to its asymmetric relation with Recall, in the dichotomous form of (Powers, 2011), $Informedness = Recall + InverseRecall - 1 = (Recall - Bias)/(1 - Prevalence)$.

If we think of Kappa as assessing the relationship between two raters, Powers statistic is not evenhanded and the Informedness and Markedness duals measure the two directions of prediction, normalizing Recall, and Precision (Powers, 2012). In fact, the relationship with Correlation allows these to be interpreted as regression coefficients

for the prediction function and its inverse.

7.1.5 Correlation

It is often asked why we dont just use Correlation to measure. (Uebersax, 1987) Uebersax (1987), (Hutchinson, 1993) Hutchison (1993) and (Bonett and Price, 2005) Bonnet and Price (2005) each compare Kappa and Correlation and conclude that there does not seem to be any situation where Kappa would be preferable to Correlation. However all the Kappa and Correlation variants considered were symmetric, and it is thus interesting to consider the separate regression coefficients underlying it that represent the Powers Kappa duals of Informedness and Markedness, which have the advantage of separating out the influences of Prevalence and Bias (which then allows macro-averaging, which is not admissable for any symmetric form of Correlation or Kappa, as we will discuss shortly). (Powers, 2011) Powers (2007) regards Matthews Correlation as an appropriate measure for symmetric situations (like rater agreement) and generalizes the relationships between Correlation and Significance to the Markedness and Informedness Measures. The differences between Informedness and Markedness, which relate to mismatches in Prevalence and Bias,mean that the pair of numbers provides further information about the nature of the relationship between the two classifications or raters, whilst the ability to take the geometric mean (of macro-averaged) Informedness and Markedness means that a single Correlation can be provided when appropriate.

From the above discussion, we can conclude that to evaluate the classifier's performance, accuracy is a usual measurement criterion. However, due to the variability of a number of classes and bias of the systems, accuracy does not show reliable the measurement. (Powers, 2003b) first introduced the concept of informedness which is a concept of probabilistic measurement based on the decision, prediction or contingency is informed, rather than due to chance. Therefore we also adopt here informedness besides accuracy to enhance a better understanding of classifier's performance. Accuracy is calculated as the following equation which indicates the proportion of right

prediction amount from the whole sample data set.

$$Accuracy = \sum_{i=1}^{m} a_i i/N. \tag{7.2}$$

Where m is the number of expressions (here m=6) and N is the total number of images. To estimate the informedness, bookmaker is an algorithm, which calculates from a contingency table encountering the idea of betting with fair odds (Powers, 2011) and (Powers, 2012). It is shown that informedness subsumes chance-corrected accuracy estimates based on other techniques that allow for chance, including Receiver Operating Characters (ROC), Correlation and Kappa, all of which are identical when bias is matched to prevalence. Informedness calculates the probability that the program makes an informed decision versus guessing. It is calculated by the following equation.

$$Informedness = \frac{winloss}{N} \tag{7.3}$$

Where $winloss = \sum_{i \neq j}(a_{ij} * bias[j]/(prev[j] - 1)) + \sum_{i=j}(a_{ij} * bias[j]/(prev[j]))$ and $prev[i] = X_i/N$, $bias_i = Y_i/N$. For clarity $prev$= prevalence, N is the total samples in the dataset, X_i and Y_i are the derived values which are the number of samples in original and predicted set correspondingly.

In order to provide a better understanding of the results, we propose here kappa statistics, correlation and informedness besides accuracy. The datasets we use here are CK dataset, which is biased and JAFFE dataset which is not bias. Which means in CK dataset, there are 110 images from anger expression, 102 images from disgust, 152 images from fear, 101 images from happy, 110 images from neutral, 110 images from sad and 100 images from surprised expression. In total, we have 785 images in CK dataset. On the other hand, in JAFFE we have 210 images which include 30 images from each expression (30 x 7=210 images).

7.2 Facial Expression Recognition Analysis: Repeated K-fold Cross-Validation

There is a considerable debate over whether it is best to use whole or part based image analysis. Being motivated by this debate, we develop our Facial Expression Recogni-

tion (FER) system using whole face and the three main facial parts, which are; eyes, nose and mouth . For the extraction of facial features, we apply the commonly used PCA and ICA with the more plausible NMF and also the SIFT (Scale-invariant feature transform) descriptor like feature, HOG. As PCA, ICA and NMF work by reducing the total feature space, so in this thesis, we will consider the features produced by PCA, ICA, NMF and HOG as 'Space'. The classifiers we implement here are Euclidian Distance (ED), Support Vector machine (SVM), Extreme Learning Machine (ELM) and Extreme Learning Machine Kernel (ELM-Kernel). As every Space is fed to every classifier, so the total comparison is among sixteen space+classifier combinations. These space-classifier combinations are, PCA+ED, PCA+ELM, PCA+ ELM kernel, PCA+SVM, ICA+ED, ICA+ELM, ICA+ ELM kernel, ICA+SVM, NMF+ED, NMF+ELM, NMF+ ELM kernel, NMF+SVM, HOG+ED, HOG+ELM, HOG+ ELM kernel as HOG+SVM.

To prove whether three facial parts can perform better to express any certain emotions or vice versa, we implement a 3x10-fold R-K Cross-validation. From the investigation, it is proved that for some space-classifier combinations three main facial parts perform better than the full face based FER and also vice versa. From this investigation, our prediction is any subset of the three facial parts can still perform better. To analyze this issue, we carefully design a 10x10 Nested Cross-Validation (N-CV) approach to tune the space-classifier combinations for each subset of the facial parts and also for the full face. We analyzed the results in tables in the next sections.

We benchmark our system on CK and JAFFE dataset using full face and three facial parts (eyes, nose and mouth). For the N-CV approach, every possible subset of the three main facial features has been tested. So our **first objective** is to analyze the comparison of facial expression recognition system based on the whole face and part faces. Our **second objective** is to analyze which combination of features and classifiers perform better for any subset of the main facial features.

For performance evaluation, we propose here kappa statistics, correlation, and informedness besides accuracy. For ELM Kernel classifier we use 'RBF Kernel' and for SVM, we use 'Linear Kernel. Optimization of kernels for ELM Kernel and SVM is beyond the scope of this thesis. This will be one of our future contributions.

For all the performance measurement tables in this chapter, we marked the performance over 80% as bold.

7.3 Histogram of Oriented Gradients

In the area of Image Processing and Computer Vision, HOG (histogram of oriented gradients) has been successfully used in recent years [(Dalal and Triggs, 2005), (Lemaire et al., 2013), (Dahmane and Meunier, 2011), (Zhang et al., 2013)]. It has been successfully implemented in pattern recognition as a feature descriptor. The underlying method of HOG has similarity with scale-invariant feature transform descriptors, shape contexts and edge orientation histograms. It is mainly computed based on a dense grid of uniformly spaced cells and to enhance the accuracy it applies overlapping local contrast normalization.

In our proposed approach, we apply HOG with four classifiers, like; Euclidian Distance (ED), Extreme Learning Machine(ELM), Extreme Learning Machine Kernel (ELM Kernel) and Support Vector Machine (SVM). Our objective is to analyze which combination of spaces and classifiers perform better. Moreover, we compare whether full face or part face based facial expression recognition performs better applying Histogram of Oriented Gradients method.

7.3.1 Overall Performance of HOG

For our proposed 3x10 repeated cross-validation approach, we investigated the statistical measurement, like; accuracy, informedness, kappa and correlation to analyze the performance of the space-classifier combinations. The first table is for CK dataset. the second table is for JAFFE dataset. As ELM does not perform well with HOG feature. So we are interested to find the performance of ELM Kernel as well as the training and testing accuracy separately for both ELM and ELM Kernel. These are tabulated in the next table. At last, we tabulated the performance, which is informedness, of CK and JAFFE dataset together using HOG+ED, HOG+ELM, HOG+ELM kernel and HOG+SVM.

Features	Algorithm	Dataset	Informedness(%)	Accuracy(%)	Kappa(%)	Correlation(%)
Whole	HOG+ED		86.22	88.28	86.27	86.30
Face	HOG+ELM Kernel	CK	84.38	86.24	83.85	84.10
	HOG+ELM		30.32	40.63	30.83	31.43
	HOG+SVM		88.42	88.79	86.89	87.84
E+	HOG+ED		85.24	87.17	87.12	87.12
N+M	HOG+ELM Kernel	CK	73.66	77.71	73.87	74.03
	HOG+ELM		36.04	45.21	36.51	37.94
	HOG+SVM		84.50	85.25	83.10	83.89

Table 7.1: Performance Metrices of HOG based Facial Expression analysis with ED, ELM, ELM Kernel and SVM classifier on CK data using Whole Face and the Three Facial Parts.

Features	Algorithm	Dataset	Informedness(%)	Accuracy(%)	Kappa(%)	Correlation(%)
Whole	HOG+ED		89.44	90.39	87.88	89.85
Face	HOG+ELM kernel	JAFFE	85.90	87.62	85.56	86.23
	HOG+ELM		32.09	23.00	24.03	24.38
	HOG+SVM		69.42	71.90	67.22	75.74
E	HOG+ED		80.59	82.67	82.59	82.59
N+M	HOG+ELM Kernel	JAFFE	72.17	75.71	71.67	73.22
	HOG+ELM		41.90	34.08	33.00	33.97
	HOG+SVM		67.05	69.52	64.44	70.68

Table 7.2: Performance metrices of HOG based Facial Expression analysis with ED, ELM, ELM Kernel and SVM classifier on JAFFE data using Whole Face and Three Facial Parts.

As ELM amd ELM Kernel works differently with HOG feature, we are interested to investigate both the testing training accuracy for both ELM and ELM Kernel.

Dataset	Features	Algorithm	Hidden Neuron Numbers	Activation Function	Kernel Type	Testing Accuracy(%)	Training Accuracy(%)
CK	Full Face	HOG+ELM Kernel	N/A	N/A	RBF	86.24±1.50	99.47±0.00
CK	E+N+M	HOG+ELM Kernel	N/A	N/A	RBF	77.71 1.50	99.47±0.00
CK	Full Face	HOG+ELM	100	Sigmoid	N/A	40.63 ±1.79	62.30 ±2.30
CK	E+N+M	HOG+ELM	100	Sigmoid	N/A	45.21 ±2.15	66.30 ±2.30
JAFFE	Full Face	HOG+ELM	100	Sigmoid	N/A	32.09 ±2.15	80.00 ±4.00
JAFFE	E+N+M	HOG+ELM	100	Sigmoid	N/A	41.90 ±3.15	97.47 ±0.80
JAFFE	Full Face	HOG+ELM kernel	N/A	N/A	RBF	87.62±1.61	99.47±0.00
JAFFE	E+N+M	HOG+ELM Kernel	N/A	N/A	RBF	72.17±1.82	99.47±0.00

Table 7.3: Comarison of HOG based FER analysis with ELM and ELM Kernel classifier on JAFFE dataset.

Features	Algorithm	Dataset	Informedness
Whole Face	HOG+ED	CK	**86.22 ±0.17**
	HOG+ELM Kernel		**84.38 ±0.01**
	HOG+ELM		30.32 ±0.90
	HOG+SVM		**88.42 ±0.48**
E+ N+M	HOG+ED	CK	**85.24 ±0.19**
	HOG+ELM Kernel		73.66 ±0.03
	HOG+ELM		36.04 ±1.14
	HOG+SVM		**84.50 ±0.50**
Whole Face	HOG+ED	JAFFE	**89.44 ±0.50**
	HOG+ELM Kernel		**85.90 ±0.00**
	HOG+ELM		23.00 ±0.75
	HOG+SVM		69.42 ±0.00
E N+M	HOG+ED	JAFFE	**80.59 ±0.34**
	HOG+ELM Kernel		72.17 ±0.00
	HOG+ELM		34.08 ±0.55
	HOG+SVM		67.05 ±0.00

Table 7.4: Informedness of HOG based FER analysis with ED, ELM, ELM Kernel and SVM classifier on CK and JAFFE datasets.

7.4 Non-Negative Matrix factorization

Machine learning research shows that Non-negative matrix factorization (NMF) is a useful decomposition for multivariate data like face and facial expression recognition. According to research studies (Lee and Seung, 2009) it is clear that NMF can be understood as part based analysis as it decomposes the matrix only into additive parts. This factorization technique of NMF is completely different of Principal Component Analysis (PCA) or Vector Quantization (VQ) in terms of the nature of the decomposed matrix. It can be seen through the visual decomposition of both methods. Figure 7.1 shows a portion of the NMF decomposed faces.

Figure 7.1: A portion of the NMF-decomposed faces.

The next figure(Fig.7.2) shows the NMF reduced subspace of several facial parts.

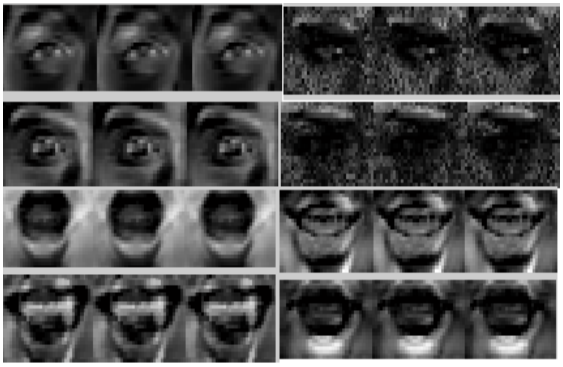

Figure 7.2: NMF decomposed facial parts.

We applied NMF with four classifiers, like; Euclidian Distance (ED), Extreme Learning Machine(ELM), Extreme Learning Machine(ELM Kernel) and Support Vector Machine (SVM). As described before, our objective is to analyse the performance of the combination of spaces and classifiers as well as the performance of full face or part face based facial expression recognition.

7.4.1 Overall Performance of NMF

For our proposed 3x10 repeated cross-validation approach, we investigated the statistical measurement, like; accuracy, informedness, kappa and correlation to analyse the performance of the space-classifier combinations. The first table is for CK dataset. The second table is for JAFFE dataset. As ELM does not perform well with NMF

feature, we are interested to find the performance of ELM Kernel. It is shown in the table that ELM Kernel also does not perform well with NMF. To have an overview of these two related classifiers (ELM without kernel and ELM kernel), we find the training and testing accuracy separately for both ELM and ELM Kernel which are tabulated in the next table. Finally, we tabulate the performance, which is informedness, of CK and JAFFE dataset together using NMF+ED, NMF+ELM, NMF+ELM kernel and NMF+SVM.

Features	**Algorithm**	**Dataset**	**Informedness(%)**	**Accuracy(%)**	**Kappa(%)**	**Correlation(%)**
Whole	NMF+ED		84.56	86.24	83.88	84.08
Face	NMF+ELM	CK	35.76	47.00	37.99	38.33
	NMF+ELM kernel		0.00	13.62	0.00	0.00
	NMF+SVM		63.01	65.95	60.03	62.17
E+N+M	NMF+ED		84.50	86.11	83.74	83.89
	NMF+ELM	CK	36.58	48.14	37.99	38.33
	NMF+ELM kernel		0.00	13.62	0.00	0.00
	NMF+SVM		57.67	61.42	54.78	57.26

Table 7.5: Performance metrices of NMF based FER analysis with ED, ELM, ELM Kernel and SVM classifier on CK dataset using Whole Face and Three Facial Parts.

Features	**Algorithm**	**Dataset**	**Informedness(%)**	**Accuracy%**	**Kappa(%)**	**Correlation(%)**
Whole	NMF+ED		82.49	84.76	82.22	82.64
Face	NMF+ELM Kernel	JAFFE	17.14	25.23	4.23	10.88
	NMF+ELM		56.67	49.86	48.18	50.02
	NMF+SVM		77.66	80.07	76.75	77.78
E+N+M	NMF+ED		71.61	74.76	70.56	71.51
	NMF+ELM kernel	JAFFE	0.00	17.14	2.01	0.00
	NMF+ELM		46.67	41.09	39.54	41.27
	NMF+SVM		63.38	67.45	62.02	64.20

Table 7.6: Performances metroces of NMF based FER analysis with ED, ELM, ELM Kernel and SVM classifier on JAFFE dataset using Whole Face and three Facial Parts.

Dataset	Features	Algorithm	Hidden Neuron Numbers	Activation Function	Kernel Type	Testing Accuracy(%)	Training Accuracy(%)
CK	Full Face	NMF+ELM	100	Sigmoid	N/A	48.14±1.31	69.00±2.00
CK	E+N+M	NMF+ELM	100	Sigmoid	N/A	47.00±1.30	70.00 ±3.00
CK	Full face	NMF+ELM kernel	N/A	N/A	RBF	13.62 ±0.46	99.50±0.50
CK	E+N+M	NMF+ELM kernel	N/A	N/A	RBF	13.62 ±0.46	99.50±0.50
JAFFE	Full Face	NMF+ELM	100	Sigmoid	N/A	56.67±3.78	99.30 ±0.44
JAFFE	E+N+M	NMF+ELM	100	Sigmoid	N/A	46.67±2.22	98.50±0.50
JAFFE	Full Face	NMF+ELM kernel	N/A	N/A	RBF	17.14±0.77	99.50±0.50
JAFFE	E+N+M	NMF+ELM Kernel	N/A	N/A	RBF	17.14±0.77	99.36±0.30

Table 7.7: Comarison of NMF based FER analysis with ELM and ELM Kernel classifier on JAFFE dataset.

Features	Algorithm	Dataset	Informedness
Whole Face	NMF+ED	CK	**84.56 ±0.80**
	NMF+ELM		36.58 ±0.58
	NMF+ELM kernel		0.00
	NMF+SVM		63.01 ±0.17
E+ N+M	NMF+ED	CK	**84.50 ±0.68**
	NMF+ELM		35.76 ±0.52
	NMF+ELM kernel		0.00
	NMF+SVM		57.67 ±0.25
Whole Face	NMF+ED	JAFFE	**82.49±0.60**
	NMF+ELM kernel		25.23±1.22
	NMF+ELM		49.86±1.20
	NMF+SVM		77.66 ±0.79
E+ N+M	NMF+ED	JAFFE	71.61 ±0.70
	NMF+ELM Kernel		0.00
	NMF+ELM		41.09±0.80
	NMF+SVM		63.38 ±0.50

Table 7.8: Informedness of HOG based FER analysis with ED, ELM, ELM Kernel and SVM classifier on CK and JAFFE datasets.

7.5 Principal Component Analysis

Principal Component Analysis is a linear dimensionality reduction technique: it transforms the data by a linear projection onto a lower-dimensional space that preserves as much data variation as possible. The following figure (Fig.7.3) is the outcome of face images (single and mixed expression dataset) from the Eigen decomposition of the datasets.

Figure 7.3: Eigen Decomposed faces.

As described before we made face into four parts and applied PCA and calculated the accuracy. We benchmarked our system on CK and JAFFE dataset using full face and three facial parts. We applied PCA with three classifiers, like; Euclidian Distance (ED), Extreme Learning Machine(ELM), Extreme Learning Machine Kernel (ELM Kernel) and Support Vector Machine (SVM). Our objective is to analyze which combination of features and classifiers perform better as well as whether full face or part face based facial expression recognition performs better applying Principal Component Analysis method.

7.5.1 Overall Performance of PCA

For our proposed full face and part face based FER system using a 3x10 repeated cross-validation, we investigated the statistical measurement, like; accuracy, informedness, kappa and correlation to analyze the performance of the space-classifier combinations. The first table is for CK dataset. the second table is for JAFFE dataset. As ELM

does not perform well with PCA feature. So we are interested to find the performance of ELM Kernel as well as the training and testing accuracy separately for both ELM and ELM Kernel. These are tabulated in the next table. At last, we tabulated the performance, which is informedness, of CK and JAFFE dataset together using PCA+ED, PCA+ELM, PCA+ELM Kernel and PCA+SVM.

Features	Algorithm	Dataset	Informedness(%)	Accuracy(%)	Kappa(%)	Correlation(%)
Whole	PCA+ED		83.77	85.72	83.26	83.62
Face	PCA+ELM Kernel ($S^{\dagger}$=Yes)[1]	CK	0.00	19.36	0.00	0.00
	PCA+ELM ($S^{\dagger}$=Yes)		32.00	41.90	31.09	31.45
	PCA+SVM		74.08	77.20	73.30	74.08
E+N+M	PCA+ED		82.93	85.27	82.75	82.94
	PCA+ELM Kernel ($S^{\dagger}$=Yes)	CK	0 .00	19.36	0.00	0.00
	PCA+ELM ($S^{\dagger}$=Yes)		21.91	34.27	23.89	24.82
	PCA+SVM		68.10	71.00	67.25	67.33

Table 7.9: performances metrices of PCA based FER analysis with ED, ELM, ELM Kernel and SVM classifier on CK using Whole Face and Three Facial Parts.

Features	Algorithm	Dataset	Informedness(%)	Accuracy(%)	Kappa(%)	Correlation(%)
Whole	PCA+ED		63.05	68.11	62.80	63.53
Face	PCA+ELM Kernel ($S^{\dagger}$=Yes)	JAFFE	85.18	85.71	84.29	85.26
	PCA+ELM ($S^{\dagger}$=Yes)		86.55	85.67	84.51	85.75
	PCA+SVM		84.58	85.71	83.33	84.67
E+N+M	PCA+ED		60.34	65.84	60.15	61.19
	PCA+ELM Kernel ($S^{\dagger}$=Yes)		76.99	78.58	74.49	76.19
	PCA+ELM ($S^{\dagger}$=Yes)	JAFFE	71.14	74.76	69.75	70.95
	PCA+SVM		71.13	73.81	69.44	70.68

Table 7.10: Performance metrices of PCA based FER analysis with ED, ELM, ELM Kernel and SVM classifier on JAFFE data using Whole Face and Three Facial Parts.

On a systematic trial and error basis, we set some hyperparameter values for ELM and ELM Kernel for PCA. In the case of ELM classifier, using trial and error basis, we found that for 'sigmoid activation function with 100 hidden number of neurons work better than other combinations. So we use here Sigmoid activation function with 100 hidden neuron numbers. **The Elm Kernel was performing with very low informedness, we investigated whether the main issue is that Elm Kernel does not work properly when (near) singular.** But the underlying PCA method decomposes the whole input data into three decomposed matrix [U S V], among these 'S

[1] $S^{\dagger}$: Pseudoinverse of the diagonal matrix of sungular values.

is a singular matrix. So we took the Pseudoinverse of the 'singular matrix and increases the performance to a great extent. Using same way, we set that For ELM Kernel, PCA works better. Optimizing the hyperparameter using nested cross-validation, with ELM and ELM Kernel will be our future work. In the table 7.5.1, for some values, we put both the testing and training accuracy using both pinv(S)=Yes and pinv(S)=No.

Dataset	**Features**	**Algorithm**	**Hidden Neuron Numbers**	**Activation Function**	**pinv(S) Type**	**Kernel**	**Testing Accuracy(%)**	**Training Accuracy(%)**
CK	Full Face	PCA+ELM	100	Sigmoid	Yes	N/A	41.90 ±2.37	55.47 ±2.00
CK	E+N+M	PCA+ELM	100	Sigmoid	No	N/A	34.27 ±1.41	41.66 ±2.00
CK	E+N+M	PCA+ELM	100	Sigmoid	Yes	N/A	44.33 ±1.20	55.47 ±2.00
CK	Full Face	PCA+ELM kernel	N/A	N/A	Yes	RBF	19.36 ±0.13	19.36 ±0.13
CK	E+N+M	PCA+ELM kernel	N/A	N/A	Yes	RBF	19.36 ±0.13	19.36 ±0.13
JAFFE	Full Face	PCA+ELM	100	Sigmoid	Yes	N/A	85.67 ±3.69	99.47 ±0.00
JAFFE	Full Face	PCA+ELM	100	Sigmoid	No	N/A	25.71 ±1.80	38.00 ±0.00
JAFFE	E+N+M	PCA+ELM	100	Sigmoid	Yes	N/A	74.76 ±2.66	98.41 ±0.00
JAFFE	Full Face	PCA+ELM kernel	N/A	N/A	Yes	RBF	85.71±2.00	98.80±0.47
JAFFE	Full Face	PCA+ELM kernel	N/A	N/A	No	RBF	15.24±0.95	99.47±0.30
JAFFE	E+N+M	PCA+ELM Kernel	N/A	N/A	Yes	RBF	78.57 ±2.38	98.80 ±0.47

Table 7.11: Comarison of PCA based FER analysis with ELM and ELM Kernel classifier on JAFFE dataset.

Features	Algorithm	Dataset	Informedness(%)
Whole	PCA+ED		**83.77 ±0.21**
Face	PCA+ELM Kernel	CK	0.00
	PCA+ELM		32.00 ±0.80
	PCA+SVM		74.08 ±0.98
E+N+M	PCA+ED		**82.93 ±0.22**
	PCA+ELM Kernel	CK	0.00
	PCA+ELM		21.91 ±0.55
	PCA+SVM		68.10 ±0.78
Whole	PCA+ED		63.05 ±0.63
Face	PCA+ELM Kernel	JAFFE	**85.18 ±1.30**
	PCA+ELM		**86.55±1.93**
	PCA+SVM		**84.58 ±0.08**
E+N+M	PCA+ED		60.34 ±0.89
	PCA+ELM	JAFFE	71.14 ±1.68
	PCA+ELM		76.99 ±0.52
	PCA+SVM		71.13 ±0.80

Table 7.12: Informedness of FER analysis with ED, ELM, ELM Kernel and SVM classifiers with PCA on JAFFE and CK data.

7.6 Independent Component Analysis

The Principal Component Analysis (PCA) is performed by the Karhunen-Loeve transform produces features $y(i), i = 0, 1, ..., N$, that are mutually uncorrelated. The solution obtained by the KL transform solution is optimal when dimensionality reduction is the goal and one wishes to minimize the approximation mean square error. However, for certain applications, such as the one illustrated in Figure 1, the obtained solution falls short of the expectations. In contrast, the more recently developed Independent Component Analysis (ICA) theory tries to achieve much more than simple decorrelation of the data. (Hyvarinen et al., 2001),(Hyvarinen and Oja, 2000). Figure 7.4 shows the source image, Figure 7.5 is the filtered mixed signals after Differentiation as

we performed Differentiation as a preprocessing step. Independent Components are shown in figure 7.6 and the inverse matrix is shown in figure 7.7.

Figure 7.4: Source images.

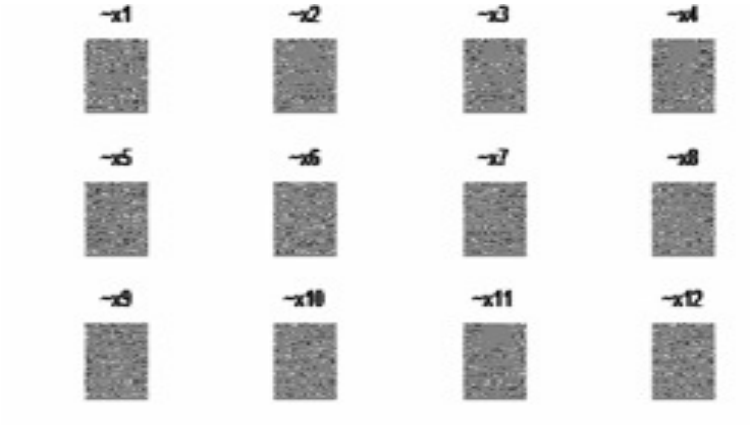

Figure 7.5: Images after Differentiation.

Figure 7.6: Independent Components.

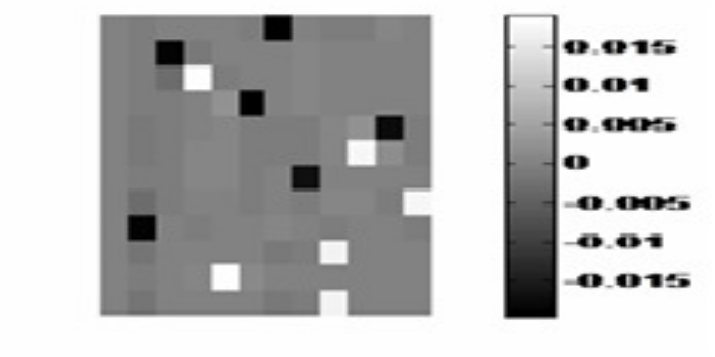

Figure 7.7: Inverse Matrix.

For our proposed experiment, we implemented FastICA with three kernels: Hyper Tangent, Gaussian and Cubic kernels. then we made a comparison among these three kernels. The following table clearly shows that the Gaussian kernel needs the less time compared to other kernels of ICA. This table 7.6 is a comparison among first ten independent component extraction time. For the next step, we choose the FastIca algorithm with Gaussian kernel.

Algorithm	**Extracted Component**	**Trial**	**Iteration**	**Elapsed** Time	**Performance** Index
FastICA	Component1	01	11	881.5s	0.1894
Kernel:Hyper Tangent	Component2	01	11		
(tanh(y))	Component3	01	94		
	Component4	01	18		
	Component5	01	25		
	Component6	01	13		
	Component7	01	13		
	Component8	01	12		
	Component9	01	22		
	Component10	01	16		
FastICA	Component1	01	11	45.11s	0.1835
Kernel:Gaussian	Component2	01	11		
$(y * exp(-y^2/2))$	Component3	01	15		
	Component4	01	16		
	Component5	01	58		
	Component6	01	32		
	Component7	01	25		
	Component8	01	13		
	Component9	01	41		
	Component10	01	12		
FastICA	Component1	01	217	397.56 s	0.1329
Kernel:Cubic	Component2	01	236		
(y^3)	Component3	01	40		
	Component4	01	20		
	Component5	01	16		
	Component6	01	19		
	Component7	01	11		
	Component8	01	19		
	Component9	01	21		
	Component10	01	12		

Table 7.13: Time Comparison among different kernels of FastICA algorithm.

We applied ICA with the same four classifiers, like; Euclidian Distance (ED), Extreme Learning Machine(ELM), Extreme Learning Machine Kernel (ELM Kernel) and Support Vector Machine (SVM). Our objective is to analyze which combination of features and classifiers perform better as well as whether full face or part face based facial expression recognition performs better applying Independent Component Analysis method. The confusion matrices are given sequentially using CK and JAFFE dataset both applying full face and then four facial parts. At the end of this section, we provided some statistical measurement, like; accuracy, informedness, kappa and correlation. As described before applied FastICA with Gaussian kernel and calculated the accuracy, kappa, informedness and correlation as performance measurement.

7.6.1 Overall Performance of ICA

For our proposed 3x10 repeated cross-validation approach, we investigated the statistical measurement, like; accuracy, informedness, kappa and correlation to analyze the performance of the space-classifier combinations. The first table is for CK dataset. the second table is for JAFFE dataset. As ELM does not perform well with ICA feature. So we are interested to find the performance of ELM Kernel as well as the training and testing accuracy separately for both ELM and ELM Kernel. These are tabulated in the next table. At last, we tabulated the performance, which is informedness, of CK and JAFFE dataset together using ICA+ED, ICA+ELM, ICA+ELM kernel and ICA+SVM.

Features	**Algorithm**	**Dataset**	**Informedness(%)**	**Accuracy(%)**	**Kappa(%)**	**Correlation(%)**
Whole	ICA+ED		79.08	81.37	78.21	78.81
Face	ICA+ELM($S^{\dagger}$=No)[2]	CK	28.34	41.53	30.18	30.67
	ICA+ELM kernel($S^{\dagger}$=No)		0	13.62	0	0
	ICA+SVM		71.40	74.27	70.86	71.65
E+N+M	ICA+ED		84.06	86.17	83.79	83.96
	ICA+ELM($S^{\dagger}$=No)	CK	43.33	30.80	31.37	32.67
	ICA+ELM Kernel($S^{\dagger}$=No)		0	13.62	0	0
	ICA+SVM		64.90	69.00	63.75	64.00

Table 7.14: Performance measurement of ICA based FER analysis with ED, ELM, ELM kernel and SVM classifier on CK data using Whole Face and Three Facial Parts.

[2] $S^{\dagger}$: Pseudoinverse of the diagonal matrix of sungular values.

Features	Algorithm	Dataset	Informedness(%)	Accuracy(%)	Kappa(%)	Correlation(%)
Whole	ICA+ED		81.96	84.40	81.78	82.30
	ICA +ELM($S^{\dagger}$=No)[3]		46.00	54.29	47.02	47.30
Face	ICA+ELMKernel($S^{\dagger}$=No)	JAFFE	19.50	17.14	4.14	9.96
	ICA+SVM		75.41	77.62	73.89	75.92
E+N+M	ICA+ED		77.80	79.32	75.89	77.23
	ICA+ELM ($S^{\dagger}$=No)	JAFFE	31.33	40.95	30.71	31.72
	ICA+ELMKerne($S^{\dagger}$=No)		0	17.14	0	0
	ICA+SVM		70.05	70.55	64.89	66.70

Table 7.15: Performance measurement of ICA based FER analysis with ED, ELM, ELM Kernel and SVM classifier on JAFFE data using Whole Face and Three Facial Parts.

On a systematic trial and error basis, we set some hyperparameter values for ELM and ELM Kernel for ICA. Like PCA, we found the same combination for ICA which is sigmoid activation function with 100 hidden number of neurons works better than other combinations in ELM classifier using ICA. In the case of matrix decomposition, ICA performs PCA decomposition first to get the independent components 'A' and decorrelate weight 'W' . Basically, the PCA U (or V) matrix is divided by the singular matrix S, which is then multiplied by the random initial weights to get the initial estimate of the decorrelated weights. So converting'singular values,S (S is the diagonal matrix of singular values), into Pseudoinverse for ICA feature extraction, in the case of ELM classifier reduces the performance of the output. On the other hand in case of ELM Kernel (to use with ICA), converting 'singular values into Pseudoinverse does not change the accuracy. So we use pinv(S)= No for ELM and ELM Kernel while using ICA. For an example, for JAFFE dataset, applying ICA+ELM with pinv(S)=yes, we get the testing and training accuracy both as 19.23. On the other hand, applying ICA+ELM with pinv(S)=No, we get the testing accuracy as 54.29 and training accuracy as 91.00. So we don't perform the pseudoinverse of the diagonal matrix of the singular values when applying ICA.

[3] $S^{\dagger}$: Pseudoinverse of the diagonal matrix of sungular values.

Dataset	Features	Algorithm	Hidden Neuron Numbers	Activation Function	pinv(S) Type	Kernel	Testing Accuracy(%)	Training Accuracy(%)
CK	Full Face	ICA+ELM	100	Sigmoid	No	N/A	41.53 ±1.81	60.38 ±1.50
CK	E+N+M	ICA+ELM	100	Sigmoid	No	N/A	43.33 ±1.78	64.38 ±2.00
CK	Full Face	ICA+ELM kernel	N/A	N/A	No	RBF	13.62 ±0.46	98.80±0.70
CK	E+N+M	ICA+ELM kernel	N/A	N/A	No	RBF	13.62 ±0.46	98.80±0.70
JAFFE	Full Face	ICA+ELM	100	Sigmoid	No	N/A	54.29 ±0.00	91.00 ±0.00
JAFFE	E+N+M	ICA+ELM	100	Sigmoid		N/A	40.95 ±2.78	96.00 ±1.50
JAFFE	Full Face	ICA+ELM kernel	N/A	N/A	No	RBF	17.14±1.05	98.80±0.70
JAFFE	E+N+M	ICA+ELM Kernel	N/A	N/A	No	RBF	17.14±1.05	98.80±0.70

Table 7.16: Comarison of ICA based FER analysis with ELM and ELM Kernel classifier on JAFFE dataset.

Features	Algorithm	Dataset	Informedness(%)
Whole Face	ICA+ED	CK	79.08 ±0.16
	ICA+ELM		28.34 ±0.80
	ICA+ELM Kernel		0.00
	ICA+SVM		71.40 ±0.80
E+N+M	ICA+ED	CK	**84.06 ±0.18**
	ICA+ELM		30.80 ±1.78
	ICA+ELM Kernel		0.00
	ICA+SVM		64.90 ±0.60
Whole Face	ICA+ED	JAFFE	**81.96 ±0.42**
	ICA+ELM		46.00 ±0.74
	ICA+ELM Kernel		19.50 ±1.70
	ICA+SVM		75.41±0.72
E+N+M	ICA+ED	JAFFE	77.80 ±0.51
	ICA+ELM		31.33 ±0.87
	ICA+ELM Kernel		0.00
	ICA+SVM		70.05 ±0.68

Table 7.17: Informedness of FER analysis with ED, ELM, ELM Kernel and SVM classifiers with ICA on JAFFE and CK data.

7.7 Facial Expression Recognition Analysis: Nested Cross-Validation

From our implementations and from the above tables, we found that for some space-classifier combinations, full face performs better where for some other space-classifier combinations, part faces performing better. Our prediction is for some cases even less part of faces may perform better than the three facial parts. Our another investigation is the performances of space-classifier combinations are data set dependent.

To prove these findings and predictions, we implement a 10x10 fold N-CV based FER system. In this approach, we use whole face (WF), three facial parts (eyes, nose and mouth which we denote as (E+N+M)) and the all possible combinations of the three facial parts, which are, Eyes (E), Nose (N), Mouth (M), Eyes + Nose (E+N), Eyes + Mouth (E+M), Mouth +Nose (M+N) as facial features. Then we benchmark our proposed N-CV analysis on CK and JAFFE dataset. The results are shown in the following two tables.

Already by using three facial parts (i.e., eyes, nose and mouth), we are reducing some facial features and hence the system needs less memory for calculations. Our one the main objectives, is to increase the accuracy as well as decrease the calculation time. So we reduced some facial features, like; the top of the forehead, two sides of the cheeks and took the main three facial parts which are prominent parts to emote the basic expressions. From the analysis and prediction of some face parts may perform better than the full face or all three facial parts, we implemented a nested cross-validation (N-CV) basis FER system.

Features	Algorithm	Dataset	Informedness(%) ±S.E.
Whole Face	HOG+SVM		**88.75 ±0.48**
Eyes	HOG+ED		**82.10 ±0.17**
Mouth	HOG+ED	CK	**88.75 ± 0.17**
Nose	NMF+ED		**84.68 ±0.70**
Eyes+Mouth	NMF+ED		**85.39 ±0.68**
Eyes+Nose	HOG+ED		**84.90 ±0.17**
Nose+Mouth	NMF+ED		**86.60 ±0.70**
Eyes+ Mouth + Nose	HOG+SVM		**83.75 ±0.48**

Table 7.18: Informedness of several facial parts using N-CV for CK dataset.

Features	Algorithm	Dataset	Informedness(%) ±S.E.
Whole Face	HOG+ED		**90.70 ±0.50**
Eyes	HOG+ED		**82.60 ±.50**
Mouth	HOG+ED	JAFFE	66.60 ±0.48
Nose	NMF+ED		77.10 ±0.70
Eyes+Mouth	HOG+ED		**87.70 ±0.48**
Eyes+Nose	HOG+ED		**83.62 ±0.48**
Mouth+Nose	HOG+ED		71.17 ±0.50
Eyes+ Mouth + Nose	HOG+ED		**82.59 ±0.34**

Table 7.19: Informedness of several facial parts using N-CV for JAFFE dataset.

From the above two tables 7.18 and 7.19, it is clear that **the performance of the space-classifiers are truly dataset dependent.** As for an example, HOG+ED is showing the highest accuracy for whole face based FER system for JAFFE dataset. On the other hand, HOG+SVM is showing the highest accuracy for whole face based FER system for CK dataset.

Second very significant finding is that, in 7.18 table, using CK dataset, we get the informedness of **88.75%** by using only the mouth feature applying HOG+SVM. This informedness is exactly same with the informedness of whole face based FER system. Moreover, the standard error of the mean is less for only mouth based FER system

with the same informedness as whole face based FER system. Less facial feature is much efficient in terms of memory usage and computational time. This is one of the very significant output of this thesis which will give a complete new direction in the area of facial expression analysis, and also in other areas of image analysis.

7.8 Comparison of our proposed Feature-Classifier Combinations with state of the art FER Systems

We count Informedness as a performance evaluator for CK and JAFFE datasets. We implement here a 3x10 fold repeated cross-validation to compare among the homogenous algorithms for FER system.

The bottom line for the performances of space+classifier combinations are, the space-classifier combinations are dataset and facial feature dependent, which means the ranking of classifier-feature combinations are different for different datasets and also different for full face and four facial combinations based analysis on the same dataset.

Now we will make a comparison of the state-of-the-art systems for facial expression recognition with our proposed approaches. We propose sixteen space-classifier combinations for two datasets, the CK and The JAFFE, for full face and three main face parts (facial features=2). Altogether it comes 16x2x2=64 comparisons for the repeated cross-validation. Again, we apply 10x10 nested cross-validation where multistage algorithms have been used which involves tuning space-classifier parameters for each subset of the face parts. To tune the space-classifier parameters, eight face parts (Full face, Eyes, Nose, Mouth, Eyes+Nose, Eyes+Mouth, Nose+mouth, (Eyes+nose+mouth)) have been used for both datasets. So it becomes the comparison among 8x2=16 combinations. Altogether all of our tables comprise 64+16= 80 comparisons. In the following table, for our proposed approaches here, we will show the space-classifier combinations of the highest performances from each space (i.e., PCA, ICA, NMF and HOG). From the nested cross-validation, we will show the accuracies over 86%. We will show here the accuracy as most of the state of the art FER systems provide the

only accuracy as their performance measurement. The following table will illustrate this.

Reference	Evaluation	Classes	Dataset	Feature	Classifier	Accuracy
(Liu et al., 2015)	Train-Test	7	CK	Gabor Filter	ELM	95%
			JAFFE	+2D-PCA		94%
(Niu and Qiu, 2010)	Train-Test	7	CK AU-Coded	WPCA	SVM	88.25%
				PPCA		84.75%
(Zhang et al., 2013)	Nested Cross validation	7	Bosphorus Dataset	HOG	C-SVM with single kernel	70.31%
				LBPH	SVM with single kernel	72.38%
				LBPH+ HOG	SimpleMKL based multiclass-SVM	76.32%
				LBPH+ HOG	HessianMKL based multiclass-SVM	80.30%
(Shan et al., 2009)	10-fold	7	CK	LBP+PCA	SVM	91.40%
(Turan and Lam, 2014)	7-fold	7	CK	PHOG	SVM	91.30%
				LPQ		95.03%
Our Proposed	3x10-fold Repeated Cross-validation	7	CK (FF[4])	PCA	ED	85.72%
			JAFFE (FF)	PCA	ELM	85.76%
Our Proposed	3x10-fold Repeated Cross-validation	7	CK (PF[5])	ICA	ED	86.17%
			JAFFE (FF)	ICA	ED	84.40%
Our Proposed	3x10-fold Repeated Cross-validation	7	CK (FF)	NMF	ED	86.24%
			JAFFE (FF)	NMF	ED	84.76%
Our Proposed	3x10-fold Repeated Cross-validation	7	CK (FF)	HOG	ED	88.28%
			CK (FF)	HOG	ELM Kernel	86.24%
			CK(FF)	HOG	SVM	88.79
Our Proposed	3x10-fold Repeated Cross-validation	7	CK (PF)	HOG	ED	87.17%
			CK (PF)	HOG	SVM	85.25%
Our Proposed	3x10-fold Repeated Cross-validation	7	JAFFE (FF)	HOG	ED	90.39%
			JAFFE (FF)	HOG	SVM	87.62%
Our Proposed	10x10-fold Nested Cross-validation	7	CK (FF)	HOG	SVM	89.50%
			CK Mouth	HOG	ED	89.50%
			CK Eyes+Mouth	NMF	ED	87.00%
			CK Nose+Mouth	NMF	ED	88.40%
Our Proposed	10x10-fold Nested Cross-validation	7	JAFFE (FF)	HOG	ED	91.50%
			JAFFE Eyes+Mouth	HOG	ED	88.60%

Table 7.20: Compariosn of our proposed approaches (the four highest feature-classifier combination from four features) with state of the art FER systems.

[4]FF: Full Face

[5]PF: Three main facial parts(eyes, nose and mouth)

Chapter 8

Concluding Remarks

As described in the Introductory chapter, facial expression recognition is playing very important role in machine learning and computer vision. During human-to-human interactions, perception and decision-making play a very important role. And this interaction, perception and decision making occur due to change of persons' emotional expression or affective states. But this change of expression is inaccessible to computing systems unless we provide computers to understand the human expression. So without this, human-computer interaction has become a predominantly one-way interaction where a user needs to directly request computer responses. Effective natural human-computer interaction becomes hard in many applications as computers become integrated into everyday objects. In some cases, users need to be able to interact naturally with computers exactly the way interpersonal face-to-face interaction takes place. The ability to detect and track users expression or emotional expression or affective states has the potential to allow a computing system to initiate communication with a user based on not only the user's command but also the perceived needs of the user within the context of the user's actions. And then human-computer interaction can become more users friendly and natural. Emerging technological advances are enabling and inspiring the research field of affective computing, which aims at allowing computers to express and recognize affect (Picard, 1997). For example, research in social psychology [(Boyle et al., 1994), (Stephenson et al., 1976), (Matsumura et al., 1997), (Ekman and Davidson, 1994), (Pantic and Rothkrantz, 2000), (Ekman, 1979), (Ekman, 1982a), (Ekman, 1982b), (Ekman and Friesen, 1971), (Ekman and Friesen,

1976)] suggests that facial expressions play a major role in the human-human interactions and provide a very strong cue about finding the level of interest (Matsumura et al., 1997).

There is a considerable debate over whether it is best to use whole or part based image analysis. Our motivation is to analyze the effect of this debate on Facial Expression Recognition system. So in our proposed approach, we implement both facial parts and whole face based approach. Facial expression recognition system generally consists of three steps, like face detection, feature extraction and classification. Machine learning researchers are using many algorithms for feature extraction and also for classification. In our experimental setup, firs,t we detect the three face parts (eyes, nose and mouth) using cascaded object detection by setting regions in a systematic trial and error basis.

For the extraction of facial features, we apply the commonly used PCA and ICA with the more plausible NMF and also the SIFT (Scale-invariant feature transform) descriptor like feature, HOG. As PCA, ICA and NMF work by reducing the total feature space, so in this thesis, we consider the features produced by PCA, ICA, NMF and HOG as 'Space'. The classifiers we implement here are Euclidian Distance (ED), Support Vector machine (SVM), Extreme Learning Machine (ELM) and Extreme Learning Machine Kernel (ELM-Kernel). As every Space is fed to every classifier, so the total comparison is among sixteen space+classifier combinations. These space-classifier combinations are, PCA+ED, PCA+ELM, PCA+ ELM kernel, PCA+SVM, ICA+ED, ICA+ELM, ICA+ ELM kernel, ICA+SVM, NMF+ED, NMF+ELM, NMF+ ELM kernel, NMF+SVM, HOG+ED, HOG+ELM, HOG+ ELM kernel as HOG+SVM. For performance evaluation, we propose here kappa statistics, correlation, and informedness besides accuracy. For ELM Kernel classifier we use 'RBF Kernel' and for SVM, we use 'Linear Kernel'. Optimization of kernels for ELM Kernel and SVM is beyond the scope of this thesis.

Potentially a subset of all the three facial parts (eyes, nose and mouth) of the face is better in terms of processing time and accuracy for identifying an expression. To prove whether three facial parts can perform better to express any certain emotions or vice versa, we implement a 3x10-fold R-K cross-validation, where 'R' is for repeated cross-validation. From the investigation, it is proved that for some space-classifier

combinations three main facial parts perform better than the full face based FER and also vice versa. From this investigation, our prediction is that any subset of the three facial parts can still perform better. To analyze this issue, we carefully design a 10x10 Nested Cross-Validation (N-CV) to tune the space-classifier combinations for each subset of the facial parts and also for the full face. We analyzed the results in the Chapter 7.

We use a set of three facial regions and ensure each part is of similar size. For our proposed RK-CV method we segment the faces into three regions: eyes, nose and mouth and we consider all three parts to classify expressions. We investigate that for some space-classifier combinations, the part face is better and for some other cases full face based approach is better. As for an example, in table 7.4, where HOG based performance has been shown, the full face based performance is better for both datasets and with each of the four classifiers; ED, SVM, ELM and ELM Kernel. Then, in table 7.8, where NMF based performance has been shown, for CK dataset, the full face and the part face based performance is same for three classifiers; ED, ELM and ELM Kernel. But SVM performs better on a full face for the same dataset. On the other hand, for all the four classifiers with JAFFE dataset, the full face based approach performs better than the part based faces. Again in table 7.12, where PCA based performance has been shown, the full face based PCA performs better for both datasets. Although the performance of PCA+ED is very competitive for both full face (83.77% informedness) and part face based (82.93% informedness) approach for CK dataset. Lastly, for table 7.17, where ICA based performance has been shown, on CK dataset, ICA+ED and ICA+ELM perform better in part face based systems than the full face approach. On the other hand, full face based approach with the four classifiers with ICA performs better on JAFFE dataset. **From this analysis, we can conclude that the performance of the classifiers is facial feature dependent (full face or part faces).**

For the N-CV approach, we take the features for the whole face, eyes, nose, mouth, nose+ mouth, eyes+ mouth, eyes+nose, and eyes+nose+mouth. These features are made for all the seven basic expressions.

From the N-CV analysis, it is clear that **the performance of the space-classifiers**

is truly dataset dependent. As for an example, HOG+ED is showing the highest accuracy for whole face based FER system for JAFFE dataset. On the other hand, HOG+SVM is showing the highest accuracy for whole face based FER system for CK dataset. **We can recommend that the performance of space-classifier is dependent on the datasets.**

Second, a very significant finding is that, in 7.18 table, using CK dataset, we get the informedness of **88.75%** by using only the mouth feature applying HOG+SVM. This informedness is exactly same with the informedness of the whole face based FER system. Moreover, the standard error of the mean is less for the only mouth based FER system with the same informedness as whole face based FER system. The Less facial feature is much efficient in terms of memory usage and computational time. This **88.75%** is the highest informedness from the nested cross-validation based multi-part FER system. **So we can recommend that only mouth based FER system can achieve the highest performance for facial expression recognition. Which means mouth is the most liable part for emoting a particular facial expression.** This is one of the very significant contributions of this thesis which will give a completely new direction in the area of facial expression analysis, and also in other areas of image analysis.

As stated before, tuning the hyperparameters for SVM and ELM Kernel is our possible future works. Also for ELM and ELM kernel based approach with PCA and ICA, we try the pseudoinverse of the diagonal matrix of singular values. Due to the time limit, we did not try with the unit matrix of 'S'. Exploration with unit matrix will be one of our future works too.

Bibliography

Albiol, A., Monzo, D., Martin, A., Sastre, J., and Albiol, A. (2008). Face recognition using hog–ebgm. *Pattern Recognition Letters*, 29(10):1537–1543.

Ali, H. B. and Powers, D. M. W. (2013). Facial expression recognition based on weighted all parts accumulation and optimal expression-specific parts accumulation. In *Digital Image Computing Techniques and Applications (DICTA), 2013 International Conference on. IEEE,Hobart, Tasmania,Vol.2,No. 1.pp.1-7*, page 229235.

Atick, J. J. and Redlich, A. N. (1992). What does the retina know about natural scenes? In *Neural Comput.*

Barlow, H. B. (1989). Unsupervised learning. In *Neural Computing.*

Bartlett, M. S., Donato, G., Movellan, J. R., Hager, J. C., Ekman, P., and Sejnowski, T. J. (1999). Face image analysis for expression measurement and detection of deceit. In *in Proceedings of the Sixth Joint Symposium on Neural Computation*, pages 8–15.

Bartlett, M. S., Movellan, J. R., and Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Transaction on Neural Networks*, 13(06):1450–1464.

Begg, R., Palaniswami, M., and Owen, B. (2005). Support vector machines for automated gait classification. In *IEEE Transaction on Biomedical engineering*, pages 52 (5), 828838.

Ben-David, A. (2008a). *About the relationship between ROC curves and Cohens kappa.* Engineering Applications of AI,21:874882, 2008, Sydney.

Ben-David, A. (2008b). *Comparison of classification accuracy using Cohens Weighted Kappa.* Expert Systems with Applications, 825832.

Bonett, D. G. and Price, R. (2005). Inferential methods for the tetrachoric correlation coefficient. In *Journal of Educational and Behavioral Statistics*, pages 30:2, 213–225.

Boyle, E., Anderson, A. H., and Newlands, A. (1994). The effects of visibility on dialogue in a cooperative problem solving task. *Language and Speech*, 37(1):1–20.

Braga-Neto, U. M. and Dougherty, E. R. (2004). Is cross-validation valid for small-sample microarray classification. *Bioinformatics*, 20(3):374–380.

Buciu, I., kotropoulos, C., and Pitas, I. (2003). Ica and gabor representation for facial expression recognition. In *International Conference on Image Processing*.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. In *Data Mining Knowledge*, pages Disc. 2 (2), 121–167.

Calder, A. J., Young, A. W., Keane, J., and Dean, M. (2000). Configural information in facial expression perception. *Journal of Experimental Psychology: Human Perception and Performance*, 26(02):527–551.

Cao, L. and Tay, F. (2003). Support vector machine with adaptive parameters in financial time series forecasting. In *IEEE Transaction on Neural Network*, pages 14 (6), 1506–1518.

Chang, C. C. and Lin, C. J. (2001). *LIBSVM: a library for Support Vector Machines.* Available from: ¡http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Charlesworth, W. R. and Kreutzer, M. A. (1973). *Facial expressions of infants and children. In P. Ekman (Ed.) Darwin and facial expression: A century of research in review (pp. 91-138).* Handbook of autism and pervasive developmental disorders, New York, academic press edition.

Chen, F. and Kotani, K. (2008). Facial expression recognition by supervised independent component analysis using map estimation. In *IEICE Transactions on Information and Systems*.

Chuang, C.-F. and Shih, F. Y. (2006). Recognizing facial action units using independent component analysis and support vector machine. In *Pattern Recognition*, pages 1795–1798.

Cohen, J. (1960). A coefficient of agreement for nominal scales. In *Educational and Psychological Measurement*, pages 37–46.

Cohn, F. J. (1999). Automated face analysis by feature point tracking has high concurrent validity with manual facs coding. *Psychophysiology*, 36(01):35–43.

Corneanu, C. A., Simon, M. O., Cohn, J. F., and Guerrero, S. E. (2016). Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1548–1568.

Cottrell, G. W. and Metcalfe, J. (1991). Empath: Face,gender and emotion recognition using holons. *Advances in Neural Inf Processing Systems*, 3(12):564–571.

Dahmane, M. and Meunier, J. (2011). Emotion recognition using dynamic grid-based hog features. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 884–888. IEEE.

Dalal, N. and Triggs, B. (2005). *Histograms of oriented gradients for human detection.* IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp.886-893.

Darwin, C. (1872). *The Expression of Emotions in Man and Animals.* John Murray, London.

Davidson, R., Scherer, K., and Goldsmith, H. (2003). *Handbook of Affective Sciences.* Oxford University Press.

Donato, G., Bartlett, M. S., Hagar, J. C., Ekman, P., , and Sejnowski, T. J. (1999). Classifying facial actions. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 21(10):974–989.

Efron, B. and Gong, G. (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, 37(1):36–48.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap.* CRC press.

Ekman, P. (1979). About brows: Emotional and conversational signals. *Human Ethology*, (1):169–202.

Ekman, P. (1982a). *Emotions in the Human Face.* Cambridge University Press.

Ekman, P. (1982b). *Methods for Measuring Facial Actions.* Handbook of Methods in Nonverbal Behaviour Research, Cambridge University.

Ekman, P. (1992). An argument for basic emotions. *Cognition and Emotion*, 6:169–200.

Ekman, P. (1999). *Basic Emotions.* Handbook of Cognition and Emotion, New York, dalgleish t and power m (eds.):john wiley edition.

Ekman, P. and Davidson, R. (1994). *The Nature of Emotion: Fundamental Questions*. Oxford University Press.

Ekman, P. and Friesen, W. (1976). *Pictures of Facial Affect*. Consulting Psychologists.

Ekman, P. and Friesen, W. (1978). *The Facial Action Coding System: A technique for the Measurement of Facial Movement*. San Francisco: Consulting Psychologist Press, San Francisco.

Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129.

Ekman, P. and Rosenberg, E. L. (1997). *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression using the FACS*. Oxford University Press, London.

Entwisle, J. and Powers, D. M. W. (January 1998). *The Present Use of Statistics in the Evaluation of NLP Parsers*. NeMLaP3/CoNLL98 Joint Conference, Sydney.

Field, T. M., Woodson, R., Greenberg, R., and Cohen, D. (1982). Basic emotions: Theory and measurement. *Cognition and Emotion*, 6(218):179–181.

Flach, P. A. (2003). *The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics*. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, pp. 226-233 edition.

Fleiss, J. L. (1981). Statistical methods for rates and proportions (2nd ed). In *New York: Wiley*, page 7.

Foody, G. and Mathur, A. (2004). A relative evaluation of multiclass image classification by support vector machines. In *IEEE Transaction on Geoscience Remote Sensors*, pages 42 (6), 13351343.

Frank, C. and Noth, E. (2003a). Automatic pixel selection for optimizing facial expression recognition using eigenfaces. In *DAGM*.

Frank, C. and Noth, E. (2003b). Optimizing eigenfaces by face masks for facial expression recognition. In *CAIP*.

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of online learning and an application to boosting. In *In Computational Learning Theory: Eurocolt 95, Springer-Verlag*, pages 23–37. Springer-Verlag.

Frith, U. and Cohen, S. B. (1987). *Perception in autistic children. In D. J. Cohen and A. M. Donnellan (Eds.).* Handbook of autism and pervasive developmental disorders, New York, john wiley edition.

Furnkranz, J. and Flach, P. A. (2005). *ROC n Rule Learning.* Towards a Better Understanding of Covering Algorithms, Machine Learning, 58(1):39-77 edition.

Gunn, S. (1998). Support vector machines for classification and regression, image speech and intelligent system group. In *Department of Electrical Computer Science, University of Southampton, Southampton, UK, Technical Report.*

Haykin, S. (1999). Neural networks: A comprehensive foundation. In *Pearson Education International, second Edition.*

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hsu, C. W. and Lin, C. J. (1998). A comparison of methods for multiclass support vector machines. In *Data Mining Knowledge*, pages Disc. 2 (2), 121167.

Huang, G. B. (2014). *An insight into extreme learning machines: random neurons, random features and kernels.* Cognitive Computation., pp. 1-15.

Huang, G. B. and Chen, L. (2007). *Convex incremental extreme learning machine.* Neurocomputing., Vol.70, pp.30563062.

Huang, G. B., Chen, L., and Siew, C. K. (2006a). *Universal approximation using incremental constructive feed forward networks with random hidden nodes.* IEEE Transaction on Neural Network., Vol.17(4), 879892.

Huang, G. B., Ding, X., and Zhou, H. (2010). *Optimization method based extreme learning machine for classification.* Neurocomputing., Vol. 74(1), 155-163.

Huang, G. B., Zhou, H., and Ding, X. (2012). *Extreme learning machine for regression and multiclass classification.* IEEE Transactions on Systems, Man, and Cybernetics., Vol. 42(2), 513529.

Huang, G. B., Zhu, Q. Y., and Siew, C. K. (2004). *Extreme learning machine: a new learning scheme of feed forward neural networks.* Proceedings of international joint conference on neural networks(IJCNN2004), Vol. 2, pp.985990.

Huang, G. B., Zhu, Q. Y., and Siew, C. K. (2006b). *Extreme learning machine: theory and applications*. Neurocomputing, Vol. 70(1), pp.489-501.

Hutchinson, T. P. (1993). Focus on psychometrics. kappa muddles together two sources of disagreement: tetrachoric correlation is preferable. In *Research in Nursing and Health*, pages 16(4):313–6.

Hyvarinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley and Sons.

Hyvarinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and application. In *Neural Networks*, pages 13(4–5):411– 430.

Isaksson, A., Wallman, M., Göransson, H., and Gustafsson, M. G. (2008). Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognition Letters*, 29(14):1960–1965.

Izard, C. (1979). *The Maximally Descriminative Facial Movement Coding System (MAX)*. Ph.D. thesis, University of Delaware, Newark, Delaware.

Izard, C. E. (1977). *Human emotions*. New York: Plenum.

Izard, C. E. (1992). Basic emotions relations among emotions and emotion-cognition relations. In *Psychological Review*.

Izard, C. E. (1994). Innate and universal facial expressions: evidence from development and cross-cultural research. *Psychological Bulletin*, 115:288–299.

Jia, X., Wang, R., j. Liu, and Powers, D. M. (2016). *A semi-supervised online sequential extreme learning machine method.* Neurocomputing., pp.168-178.

Kapoor, S. M. and Picard, R. (2001). Towards learning companion that recognizes affect. In *in the Proceedings of American Association for Artificial Intelligence*.

Karklin, Y. and Lewicki, M. S. (2003). Learning higher-order structures in natural images. In *Network: Computation in Neural Systems*, pages 483–499.

Kolenda, T., Hansen, L. K., Larsen, J., and Winther, O. (2002). Independent component analysis for understanding multimedia content. In *IEEE Workshop on Neural Networks for Signal Processing*.

Kumar, S. (2004). *Neural Networks: A Class Room Approach.* Cognitive Computation., New Delhi, India.

Lan, Y., Soh, Y., and Huang, G. B. (2010a). *Constructive hidden nodes selection of extreme learning machine for regression.* Neurocomputing., Vol. 73(16),3191-3199.

Lan, Y., Soh, Y., and Huang, G. B. (2010b). *Two-stage extreme learning machine for regression.* Neurocomputing., Vol. 73(16), 3028-3038.

Lee, D. D. and Seung, H. S. (2009). Learning the parts of objects by non-negative matrix factorization. In *Letters to Nature*, pages 788–791.

Leibbrandt, R. E. (2000). *The influence of semantic and syntactic information on children's internal-state interpretations of novel adjectives.* Unpublished Masters thesis, University of Essex, Colchester, United Kingdom.

Lemaire, P., Ardabilian, M., Chen, L., and Daoudi, M. (2013). Fully automatic 3d facial expression recognition using differential mean curvature maps and histograms of oriented gradients. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–7. IEEE.

Liu, C. (2004). Enhanced independent component analysis and its application to content based face image retrieval. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 34(02):1117– 1127.

Liu, P. and Yin, L. (2015). Spontaneous facial expression analysis based on temperature changes and head motions. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–6. IEEE.

Liu, P. and Yin, L. (2017). Spontaneous thermal facial expression analysis based on trajectory-pooled fisher vector descriptor. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, pages 835–840. IEEE.

Liu, Z.-T., Sui, G.-T., Li, D.-Y., and Tan, G.-Z. (2015). A novel facial expression recognition method based on extreme learning machine. In *InControl Conference (CCC), 2015 34th Chinese, IEEE*, pages 3852–3857.

Luo, J., Ma, Y., Takikawa, E., Lao, S., Kawade, M., and Lu, B.-L. (2007). Person-specific sift features for face recognition. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 2, pages II–593. IEEE.

Manning, C. D. and Schutze, H. (1999). Foundations of statistical natural language processing. In *MITPress, Cambridge, MA*.

Marco, Virgil R., D. M. Y. and Turner., D. W. (1987). The euclidean distance classifier: an alternative to the linear discriminant function. In *Advances in Neural Information Processing Systems*, pages 485–505. Communications in Statistics-Simulation and Computation.

Marian, S. B., Javier, R. M., and Terrence, J. S. (2002). Face recognition by independent component analysis. In *IEEE transaction Neural Network*.

Matsumura, K., Nakamura, Y., and Matsui, K. (1997). Mathematical representation and image generation of human faces by metamorphosis. *Electronics and Communication in Japan*, 80(1):36–46.

McNeill, D. (1992). *Hand and Mind: What Gestures Reveal About Thought*. Univ. of Chicago Press, Chicago, IL.

Mehrabian, A. (1968). *Communication without words*. Psychology today, 2(4): 53-56 edition.

Nagappan, N., Murphy, B., and Basili, V. (2008). The influence of organizational structure on software quality: an empirical case study. In *Proceedings of the 30th international conference on Software engineering*, pages 521–530. ACM.

Niu, Z. and Qiu, X. (2010). Facial expression recognition based on weighted principal component analysis and support vector machines. In *In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), IEEE*, pages V3–174.

Osuna, E., Freund, R., and Girosi, F. (1997). Training support vector machines: an application to face detection. In *iProceedings of Computer Vision and Pattern Recognition*, page pp. 130136.

Pantic, M. and Rothkrantz, L. J. M. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(12).

Paul, V. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, volume 1. Proceedings of the 2001 IEEE Computer Society Conference.

Pentland, A. P. (1987). Automatic pixel selection for optimizing facial expression recognition using eigenfaces. In *IEEE proceedings of the First International Conference on Computer Vision*.

Perruchet, P. and Peereman, R. (2004). The exploitation of distributional information in syllable processing. In *J. Neurolinguistics*, page 17:97119.

Phillips, P. J., Wechsler, H., Huang, J., and Rauss, P. (1998). The feret database and evaluation procedure for face-recognition algorithms. In *Image and Vision Computing*, page 295306.

Picard, R. W. (1997). *Affective Computing*. MIT Press, Cambridge.

Powers, D. M. W. (2003a). Recall and precision versus the bookmaker. In *Proceedings of the International Conference on Cognitive Science (ICSC-2003),Sydney Australia*, pages 529–534.

Powers, D. M. W. (2003b). Recall and precision versus the bookmaker. In *Interna- tional Conference on Cognitive Science (ICSC-2003)*, page 529534.

Powers, D. M. W. (2007/2011). Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation. In *School of Informatics and Engineering, Flinders University, Adelaide, Australia, TR SIE-07-001, Journal of Machine Learning Technologies*, pages 2:1 37–63.

Powers, D. M. W. (2008). *Evaluation Evaluation*. The 18th European Conference on Artificial Intelligence (ECAI08).

Powers, D. M. W. (Avignon France, 2012). The problem of kappa. In *13th Conference of the Euro- pean Chapter of the Association for Computational Linguistics*.

Qvarfordt, P. and Zhai, S. (2005). Conversing with the user based on eye-gaze patterns. In *Conference of Human-Factors in Computing System*.

Sawyer, R., Smith, A., Rowe, J., Azevedo, R., and Lester, J. (2017). Enhancing student models in game-based learning with facial expression recognition. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 192–201. ACM.

Schneider, J. (1997). http://www.cs.cmu.edu/schneide/tut5/node42.html.

Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. In *Public Opinion Quarterly*, pages 19,321–325.

Shan, C., Gong, S., and McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. In *Image and Vision Computing*, pages 27(6), 803–816.

Shao, Z., Er, M. J., and Wang, N. (2015). An efficient leave-one-out cross-validation-based extreme learning machine (eloo-elm) with minimal user intervention.

Song, X. and Bao, H. (2016). Facial expression recognition based on video. In *Applied Imagery Pattern Recognition Workshop (AIPR), 2016 IEEE*, pages 1–5. IEEE.

Stein, N. L. and Oatley, K. (1992). Basic emotions: Theory and measurement. *Cognition and Emotion*, 6:161–168.

Stephenson, G. M., Ayling, K., and Rutter, D. R. (1976). The role of visual communication in social exchange. *Britain Journal of Social Clinical Psychology*, 15:113– 120.

Steyerberg, E. (2008). *Clinical prediction models: a practical approach to development, validation, and updating*. Springer Science & Business Media.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147.

Tantithamthavorn, C., McIntosh, S., Hassan, A., and Matsumoto, K. (2015). An empirical comparison of model validation techniques for defect prediction models.

Tarvo, A. (2008). Using statistical models to predict software regressions. In *2008 19th International Symposium on Software Reliability Engineering (ISSRE)*, pages 259–264. IEEE.

Theodoridis, S. and Koutroumbas, K. (2009). *Pattern Recognition Book*. Academic Press, 4th edition.

Turan, C. and Lam, K. M. (2014). Region-based feature fusion for facial-expression recognition. In *In 2014 IEEE International Conference on Image Processing (ICIP), IEEE*, pages 5966–5970.

Uddin, M. Z., Lee, J. J., and Kim, T. S. (2009). An enhanced independent component based human facial expression recognition from video. In *IEEE Transactions on Consumer Electronics*.

Uebersax, J. (1987). Diversity of decision-making models and the measurement of interrater agreement. In *Psychological Bulletin*, pages 101, 140146.

Vapnik, V. (1995). The nature of statistical learning theory. In *Springer-Verlag, New York*.

Viera, A. J. and Garrett, J. M. (2005). Understanding interobserver agreement: the kappa statistic. In *Fam Med*, pages 37(5), 360–363.

Viola, P. and Jones, M. J. (2004). Robust real time face detection. *International Journal of Computer Vision*, 57:137–154.

Wallis, P., Moore, R., Fagerberg, P., Cavazza, M., and Wilks, Y. (2006). Emotion in human-agent interfaces. *Companions Consortium: State of the Art Papers*, 3.

Witten, I. H. and Frank, E. (2005). Data mining: Practical machine learning tools and techniques.

Zhang, X., Mahoor, M. H., and Voyles, R. M. (2013). Facial expression recognition using hessianmkl based multiclass-svm. In *In Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on (pp. 1-6). IEEE.*, pages 1–6.

Zimmermann, T. and Nagappan, N. (2008). Predicting defects using network analysis on dependency graphs. In *Proceedings of the 30th international conference on Software engineering*, pages 531–540. ACM.