CHAPTER ONE

INTRODUCTION

1.1 General Review

Humans interrelate with each other mainly through speech, also through body gestures, to highlight a certain part of speech and display of emotions. In an effort to understand and categorize emotions, psychologists and engineers have tried to analyze facial expressions, vocal emotions, gestures, and physiological signals.

From the video images acquired from built-in cameras, and from speech waveforms collected from on-board microphones, this information can be used to teach computers to recognize human emotions. A normal two-way interaction between the human and the computer through multiple modalities is represented in Figure 1.1. In this diagram, one of the inputs to the computer is a video, from which gaze, posture, gestures, and facial and lip movements can be extracted. Computers may learn to recognize gestures, postures, facial expressions, eye contact. Similarly, speech and voice (audio) through the microphone may convey linguistic as well as paralinguistic information. On the output side, the computer may appear in the form of an "agent"- a computer-animated face. This "agent" can speak to the human through a created speech and display corresponding facial and mouth movements on the screen.

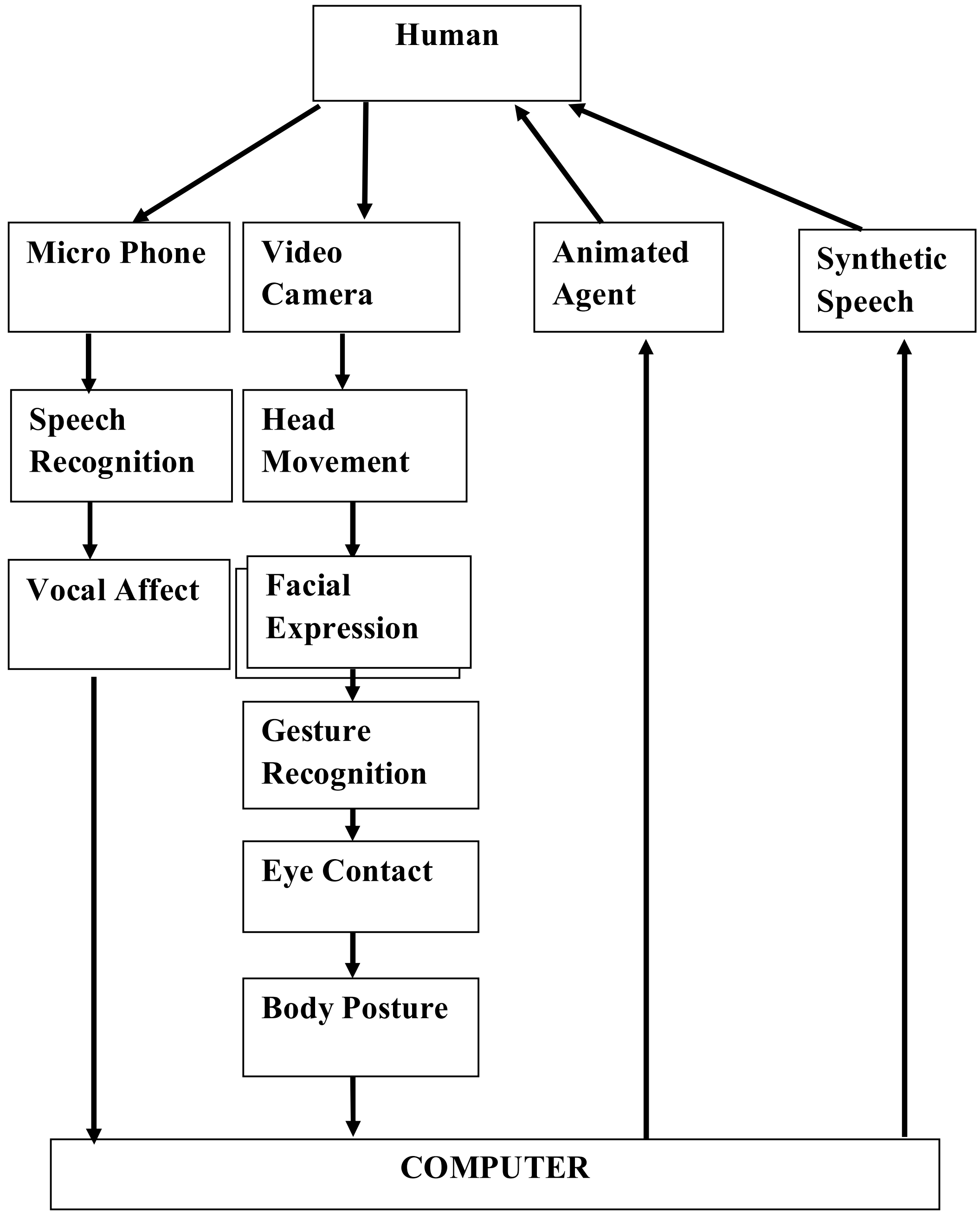


Fig.1.1 Multimodal human-computer interaction

Even if they are not openly presented in the Figure 1.1, some other modalities such as tactile or physiological signals can also be used in concurrence with the video and audio signals. The term “Emotional expression” means any outward expression that arises as a response to some stimulus event. These may include typical expressions such as a “smile” to show that one is happy, or to show one likes what one sees.

1.2 Problem Statement

Facial expression provides sensitive cues due to emotional response, and plays a major rule in human interaction and non-verbal communication. It can complement verbal communication or convey complete thoughts by itself.

The pose of the subject, the position of the face can cause some parts of the face such as eyes and nose occluded in the captured image (Yang et al., 2002)[1].

The system should be invariant to different lightening conditions and distraction as glasses, change in hair style, facial hair, moustache, and beard. The features like beard, moustache, and glasses may not result in the correct facial expression because these features are subjects to change.

1.3 Objectives

The main objective is to design facial expression model using Gabor Wavelet and Neural Networks.

Specific objectives are to:

1. recognize the different types of expressions of a face.

2. implement Gabor Wavelet and Artificial Neural Networks (ANN) for classification of emotions from the facial expressions.

1.4 Methodology

In this research, a systematic approach has been developed to train ANN topology with different algorithms for emotion classification. The algorithms are as follows:

1. Single layer network (SLN).
2. Back propagation Algorithm (BPN).
3. Cerebellar Model Articulation Controller (CMAC).

Working Details of implemented Architecture (Figure 1.2):

Step 1: Input facial image.

Step 2: Important features are extracted.

Step 3: The features are trained using the implemented SLN/BPN/CMAC and final weights are stored in a database.

Step 4: In the testing process, step 1 and step 2 are adopted. The extracted features are processed with final weights of the SLN/BPN/CMAC, to get an output in the output layer of the SLN/BPN/CMAC.

Step 5: The output is compared with a threshold value, to decide the category to which the particular emotion the facial expression belongs.

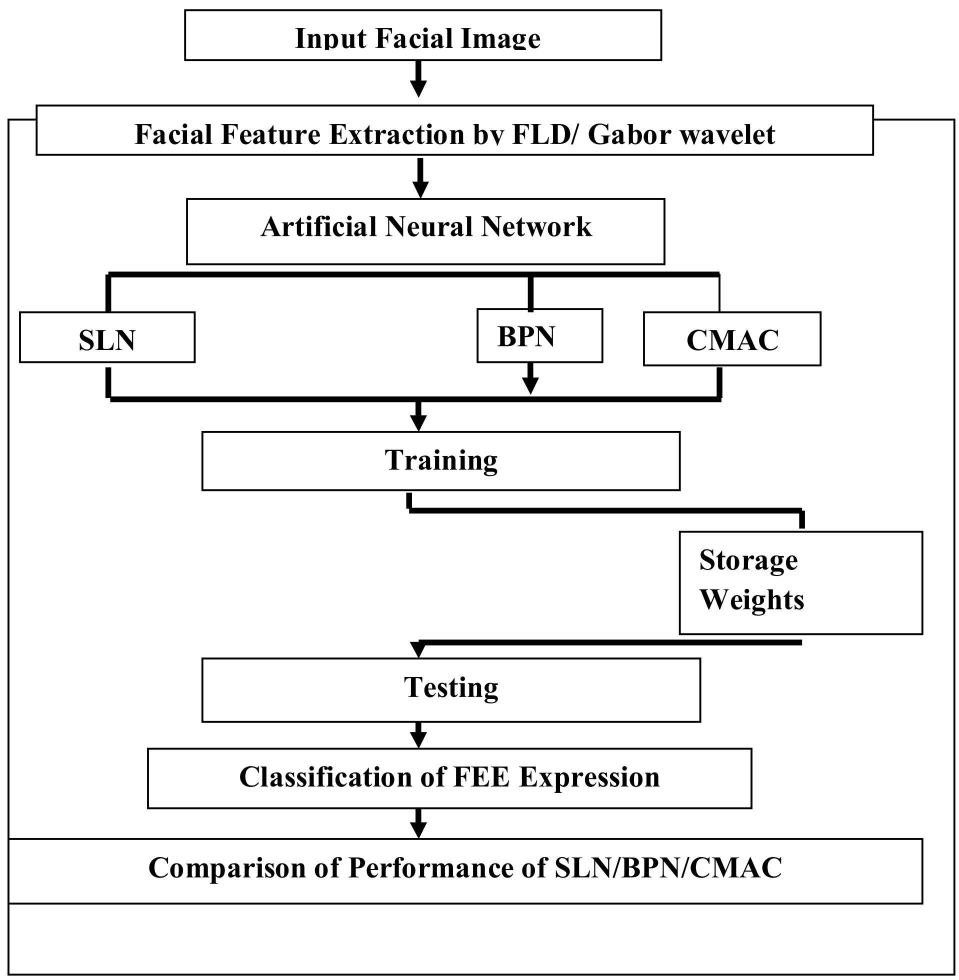


Fig.1.2 Proposed Architecture for Facial Emotion Expression Classification

1.5 The Contribution

Contributions achieved are to:

1. minimize difficulties in facial emotion expression (FEE) recognition due to the variation of facial expression across the human population.
2. analyze facial expression feature extraction methods.
3. compare the performance of facial expression classification scheme.
4. The development of a universal anatomical model for faces across different cultures, age groups and demographical origins is also a different task.

1.6 Motivation for the Research

This work addresses the problem of detecting the facial expressions by analyzing the facial images.

The work is motivated to reduce the intensive computations in the previous work to achieve better accuracy and results.

1.7 Thesis Layout

Chapter one presents introduction. Previous studies are presented in chapter two. Theoretical backgrounds are described in chapter 3. Chapter four describes the proposed system. Results are discussed in chapter five. Chapter six describes discussion in chapter seven. Finally conclusions and recommendations are presented in chapter six.

CHAPTER TWO

PREVIOUS STUDIES

2.1 Introduction

Enrichment of knowledge becomes possible only through literature search on any topic of interest. It is true even in the case of 'Emotional Classification using Facial Expression". Emotion recognition through the computer-based of facial expression has been an active and interesting area of research in the literature survey for a long time. This chapter reviews the concepts of emotion, cultural differences in recognition, cross cultural differences. It extensively discusses the literature on the various classification methods through the tabular column.

2.2 Facial Expression Recognition Studies

Matthew et al., 2006, decoded emotions by merely observing other individuals communicate via touch. Study 1: they asked whether participants can communicate three classes of emotions via touch. In study 1, 212 participants are communicated, from United States. Study 2: they partially replicated Study 1 with a sample from a different culture-Spain-and a limited set of emotions, anger, fear, happiness, sadness, disgust, love, gratitude, and sympathy. In study 2, 58 participants are communicated, from Spain. Study 3:

people can discern emotion in tactile behavior without the experience of being touched. In study 3, 114 participants are communicated. Through the touch, they could decide anger, fear, disgust, love, gratitude, and sympathy. The author concluded that humans can communicate several emotions with touch. [2].

Elaine Fox et al., 2007, presented effectiveness of facial processing by means of a visual search task. 1. When displays contained the same faces, people were slower in detecting. 2. When displays contained a discrepant face people were faster in detecting. Experiment 1 was to investigate the face-in-the-crowd. (Forty-five (31 female, 14 male) Undergraduate students at the University of Essex) Experiment 2 as same as experiment 1 but in stimulus displays were presented for 800msec. Thirty undergraduate students at the University of Essex (19 female, 11 male). In Experiment 3, change in face orientation. Detection of angry facial expressions is fast and efficient, in five experiments, the existence of the "face-in-the-crowd" effect was confirmed with simple schematic facial expressions. In Experiments 1 and 2 angry (or sad/angry) faces in neutral crowds were found more efficiently than happy faces in neutral crowds. [3].

Levine, 2007, reviewed interactions between emotion and attention. The network models of emotional processes dealt with positive and negative emotion. The author deals with emotions such as fear, joy, sadness, and anger.

The author also explained the essential contributions of emotion to intelligent behavior and the importance of quantitative theories and models to the interdisciplinary enterprise of understanding the interactions of emotion, cognition, and behavior. [4].

Abigail Marsh et al., 2007, described the recognition of fear to predict actual pro-social or antisocial behavior in an experimental setting. They used Fear facial images only. In the 3 studies, the authors verified the expectation. Individuals who recognize fear more accurately will behave more pro-socially. In Study 1- participants who identified fear more accurately. In Studies 2 and 3, participant's capacity, to identify the fear expression predicted pro-social behavior. [5].

Manuel et al., 2008, investigated salient visual features capture attention methods and detection of emotional facial expressions. Saliency was greatest for the faces (happy) and regions (mouth) that were fixated earlier and detected faster. The facial features are responsible for the detection, and the advantages are located in the mouth rather than the eye region. Twenty-four psychology undergraduates (18 women, 6 men; from 19 to 23 years of age) participated.

Karolinska Directed Emotional Faces pictures are used in the current study. Faster detection of happy and, to a lesser extent, surprised and disgusted faces was found both under upright and inverted display conditions. Inversion slowed down the detection of these faces less than that of others (fearful, angry, and sad). [6].

Hillel Aviezer et al., 2008, explained the Current theories of emotion perception. By monitoring eye movements, demonstrate that characteristic fixation patterns are determined.16 undergraduate students (9 females, 7 males) are selected for testing. After execution, the recognition rates are disgust: 92%, anger: 94%, sadness: 96%, and fear: 92%. Repeated measures analysis of variance (ANOVA) and all pair-wise comparisons confirmed that accuracy did not differ significantly between contexts. [7]

Hatice Gunes et al., 2010, discussed the problem domain of affect sensing using a dimensional approach and explored the current state-of-the-art in continuous, dimensional affect recognition. The author discusses the theories of emotion, expression and perception of emotions, facial expression, bodily expression, audio, bio-potential signals, thermal signals. He also discussed about spontaneous expressions, data acquisition, data annotation, problem domain, durations of emotion, emotion intensity, baseline problem, dimensionality, fusion. [8].

Ayesha Butalia et al, 2010, explained that an integrated system covering facial expression, gesture and body postures is capable of improving the emotion recognition rate outside the rates of the single systems. Context based recognition can be added so as to resolve the ambiguity involved in different scenarios. They explained about rough sets, Adaptive Neuro Fuzzy Inference Systems, context based approach towards facial recognition (SVM, Bayesian networks). Emotion recognition done by the methods of artificial neural network (ANN), fuzzy set, support vector machine (SVM), hidden Markov model (HMM), and based on the facial or audio features, and the recognition rate often arrives at 64% to 98%. [9].

Ravi et al., 2011, gave a comparative study and analysis of facial expression recognition technology along with its progressive growth and developments. Ekman and Friesen described that the simple static images were not enough to describe subtle changes, so facial EMG has come into picture. Judgment based, and sign vehicle based approaches find their way for accurate face detection. [10].

Seyedarabi et al., 2007 made a comparative study on RBF classifier and FIS classifier. Yongmian et al., 2008, developed a new technique to improve 3D face pose and facial expression instantaneously from a video sequence in real time. [11], [12]

2.3 Classification Based On Some Selected Algorithms:

Here are summary of some classification studies based on some selected methods including: algorithm used, features extraction, accuracy (detection accuracy, false positive ratio, false negative ratio, equal error rate) and database (number of images, number of Expressions, name of the expressions).

2.3.1 Classification Studies Based on Neural Network:

Renu et al., 2010**,** presents a novel approach for the detection of emotions using the cascading of Mutation Bacteria Foraging optimization and Adaptive Median Filter in highly corrupted noisy environment. The approach involves removal of noise from the image by the combination of MBFO & AMF and then detects local, global and statistical feature form the image. The Bacterial Foraging Optimization Algorithm (BFOA), as it is called now, is currently gaining popularity in the community of researchers, for its effectiveness in solving certain difficult real-world optimization problems. The results so far show the approach to have a promising success rate. An automatic system for the recognition of facial expressions is based on a representation of the expression, learned from a training set of pre-selected meaningful features. However, in reality the noises that may embed into an image document will affect the performance of face recognition algorithms. As a first the author investigates the emotionally intelligent computers which can perceive human emotions. They test four emotions namely anger, fear, happiness along with neutral from database in noisy environment of salt and pepper. Very high recognition rate has been achieved for all emotions along with neutral on the training dataset as well as user defined dataset. The proposed method uses cascading of MBFO & AMF for the removal of noise and Neural Networks by which emotions are classified. The overall accuracy for testing based on lips, eyes, eyes & lips are 84%, 92%, 92% respectively. [13]

Li, 2010, Facial expression is one way humans convey their emotional states. Accurate recognition of facial expressions via image analysis plays a vital role in perceptual human computer interaction, robotics and online games. He focuses on recognizing the smiling from the neutral facial expression. He proposed a face alignment method to address the localization error in existing face detection methods. The smiling and neutral facial expressions are differentiated using a novel neural architecture that combines fixed and adaptive non-linear 2-D filters. The fixed filters are used to extract primitive features, whereas the adaptive filters are trained to extract more complex features for facial expression classification. The proposed approach is evaluated on the JAFFE database and it correctly aligns and crops all images, which is better than several existing methods evaluated on the same database. The system achieves a classification rate of 99.0% for smiling versus neutral expressions. [14].

Khandait et al., 2011, provided an approach to the problem of automatic facial feature extraction from a still frontal posed image and classification and recognition of facial expression and hence emotion and mood of a person is presented. Feed forward back propagation neural network is used as a classifier for classifying the expressions of supplied face into seven basic categories like surprise, neutral, sad, disgust, fear, happy and angry. For face portion segmentation and localization, morphological image processing operations are used. Permanent facial features like eyebrows, eyes, mouth and nose are extracted using SUSAN edge detection operator, facial geometry, edge projection analysis. Experiments are carried out on JAFFE facial expression database and gives better performance in terms of 100% accuracy for training set and 95.26% accuracy for test set. [15].

Krishna Kishore and Varma, 2011, described a Hybrid Emotional Neural Network (HENN) for classification of emotions from Facial expressions. The novelty of this work is that along with the parameters of the feed forward neural network, i.e. the learning rate, momentum, new parameters such as anxiety and confidence is taken as emotional parameters are from Gabor Wavelet and are used to update emotional parameters of the network. An improved Back propagation algorithm is used for training of the proposed neural network. Features are extracted from facial expressions by applying Gabor wavelet and Discrete Cosine Transform (DCT). Both the feature sets are high dimensional so Principle Component Analysis (PCA) are used to reduce the dimensionality of features. Then Wavelet fusion technique is used to fuse the features. The fused features are used to train the neural network. The classification efficiency of the proposed method was tested on static images from the Cohn-Kanade database. The results of the proposed network were compared with the standard Feed Forward Neural Network and Radial Basis Neural Network. They also make a detailed comparison of different fusion techniques along with wavelet fusion, as well as different Neural Network classifiers. Classifications varied across different databases from 75% to100%.[16].

Mahesh Kumbhar et al., 2012, Facial expression recognition has potential applications in different aspects of day-to-day life not yet realized due to absence of effective expression recognition techniques. Their research discusses the application of Gabor filter based feature extraction in combination feed forward neural networks (classifier) for recognition of seven different facial expressions from still pictures of the human face. The study presented here gives simple method in facial expression recognition. The study presented here gives 60% to 70 % recognition of facial expression for the entire database of JAFEE. In this study the Japanese Female Facial Expression (JAFFE) database used which contains expressers that expressed expressions. The present study proves the feasibility of computer vision based facial expression recognition for practical applications like surveillance and human computer interaction. [17].

2.3.2 Classification Based on Support Vector Machine and Hidden Markov Model

Littlewort et al., 2002, presented initial results from the application of an automated facial expression recognition system to spontaneous facial expressions of pain. In this study, 26 participants were videotaped under three experimental conditions: baseline, posed pain, and real pain. The real pain condition consisted of cold pressure pain induced by submerging the arm in ice water. The goal was to:

(1) assess whether the automated measurements were consistent with expression measurements obtained by human experts, and

(2) develop a classifier to automatically differentiate real from faked pain in a subject-independent manner from the automated measurements. He employed a machine learning approach in a two-stage system. In the first stage, a set of 20 detectors for facial actions from the Facial Action Coding System operated on the continuous video stream. These data were then passed to a second machine learning stage, in which a classifier was trained to detect the difference between expressions of real pain and fake pain. Naïve human subjects tested on the same videos were at chance for differentiating faked from real pain, obtaining only 49% accuracy. The automated system was successfully able to differentiate faked from real pain. In an analysis of 26 subjects with faked pain before real

pain, the system obtained 88% correct for subject independent discrimination of real versus fake pain on a 2-alternative forced choice. Moreover, the most discriminative facial actions in the automated system were consistent with findings using human expert FACS codes. [18].

Taskeed Jabid et al., 2010, proposed local feature descriptor, Local Gaussian Structural Pattern (LGSP), for face recognition that encodes the directional information of the face's textures (the texture's structure), producing a more discriminant code than other state-of-the-art methods. Each micro-pattern's structure is computed by using a derivative-Gaussian compass mask, which is more robust against challenging illumination and noisy conditions, and encoded by using the principal directions. Consequently, the compass mask helps distinguishing among similar structural patterns. Moreover, our descriptor encodes more information by using different resolutions of the compass mask. Thus to process a face, they divided it into several regions and extract the distribution of the LGSP features from them. Then, he concatenated these features into a feature vector, and used it as face descriptor. He performs several experiments in which our descriptor showed consistent results under age, illumination, expression, and noise variations. [19]

Nirmala and Wahida, 2012, stated that facial expression analysis is an attractive, challenging and important field of study in facial analysis. It is important applications include many areas such as human–computer interaction, human emotion analysis, biometric authentication, exhaustion detection and data-driven animation. For successful facial expression recognition, the first step is to arrive at an appropriate facial representation from original face image which is a crucial step. The work, empirically evaluate facial representation using statistical features from the Local Binary Patterns, Simplified local binary mean and Mean based weight matrix for person-independent facial expression recognition. Multiclass SVM is applied systematically for classification. The Japanese female database JAFFE is used for the experiment. Extensive experiments shows that statistical features derived from LBP are effective and efficient for facial expression recognition. Further improved and best results are obtained with SLBM and MBWM features extracted using Multiclass Support Vector Machine classifiers. Neutral can be recognized with high accuracy (90–98%), while the recognition rates for Fear and Sadness are much lower (79–88%). The seven universal expressions are recognized with 100% accuracy. 91.35% accuracy with the combination of 10 subjects images using all the 7 classes and 92.18% using 6 classes of expressions. [20]

Suvashis Das and Kochi, 2012, stated that the problem of emotion prediction from the face is twofold. First, it requires that the facial Action Units (AUs)1 and their intensities are identified and second interpreting the recorded

AUs and their intensities as emotions. Their work focuses on developing an accurate model to predict emotions from Facial Action Coding System (FACS) coded facial image data based on a Hidden Markov Model (HMM) approach. The novelty of this work is:

1) A new and more accurate model for emotion prediction from AU data is proposed by assigning a set of N HMMs to every AU where N is the number of emotions they consider while conventional studies have assigned at most one HMM per AU or lesser like 6 emotion specific HMMs for the entire set of AUs [3-6]. Assigning N HMMs per AU takes away the errors that might creep in due to non-consideration of the insignificant or non-present AUs by calculating separately the probability contributions towards each emotion by every single AU in the entire AU set which is used later to calculate the mean probability for each emotion considering all AUs together.

2) A percentage score of each emotion that composed the face of a subject is predicted rather than to just identify the lead or prominent emotion from the maximum probability considerations as exhibited my majority of similar researches.

3) Discuss the gender differences in the depiction of emotion by the face.

This improvement in the model yielded better results (around 97%). [21].

2.3.3 Classification Based on Genetic Algorithm:

Amir Jamshidnezhad and Jan, 2012, stated that the genetic algorithms (GAs) are the evolutionary learning process, which are applied to complex optimization problems to find the optimum results from other results. The GAs to achieve the proper results in the iteration process selects the fitter solutions from a solution space. In this process, the parent selections, crossover and mutation operations have the important roles to find the optimum results. The honey bees mating process has been modeled to modify the GAs learning process. For the purpose of illustration, the learning process with the proposed GA, the fuzzy membership functions were tuned with the proposed GA. In the proposed hybrid model, the core of expression recognition system is the fuzzy rule based system to classify the facial expressions recognition. Therefore, the proposed genetic algorithm called queen bee algorithm (QBA) is used with the purpose of making better performance in learning process to improve the accuracy and robustness of the system. To evaluate the system performance, images from Cohn-Kanade database were used to obtain the best functions parameters. Results showed that the membership functions under the training process have tuned properly while the accuracy of classification with optimized parameters illustrated the rate of 98% in the train process. [22]

2.3.4 Classification Based on Facial Action Coding System:

Pantic and Rothkrantz, 2000, discussed the expert system called Integrated System for Facial Expression Recognition (ISFER), which performs recognition and emotional classification of human facial expression from a still full-face image. The system consists of two major parts. The first one is the ISFER Workbench, which forms a framework for hybrid facial feature detection. Multiple feature detection techniques are applied in parallel. The redundant information is used to define unambiguous face geometry containing no missing or highly inaccurate data. The second part of the system is its inference engine called HERCULES, which converts low level face geometry into high level facial actions, and then this into highest level weighted emotion labels. Recognition rate achieved by the system was 89.6%.[23]

Fasela and Juergen, 2003, mentioned that the Automatic Facial Expression Recognition has been one of the latest research topic since 1990's.There have been recent advances in detecting face, facial expression recognition and classification. There are multiple methods devised for facial feature extraction which helps in identifying face and facial expressions. They surveyed some of the published work since 2003 till date. Various methods are analyzed to identify the Facial expression. The research also discusses about the facial parameterization using Facial Action Coding System (FACS) action units and the methods which recognizes the action units parameters using facial expression data that are extracted. Various kinds of facial expressions are present in human face which can be identified based on their geometric features, appearance features and hybrid features. The two basic concepts of extracting features are based on facial deformation and facial motion. This article also identifies the techniques based on the characteristics of expressions and classifies the suitable methods that can be implemented. [24]

2.3.5 Classification Based on Fuzzy Logic:

Ayesha Butalia et al., 2012, stated that Facial expressions play an important role in interpersonal relations as well as for security purposes. The malicious intensions of a thief can be recognized with the help of his gestures, facial expressions being its major part. This is because humans demonstrate and convey a lot of evident information visually rather than verbally. Although humans recognize facial expressions virtually without effort or delay, reliable expression recognition by machine remains a challenge. A picture portrays much more than its equivalent textual description. Along this theory, she asserts that although verbal and gestural methods convey valuable information, facial expressions are unparalleled in this regard. In sustenance to this idea, a facial expression is considered to consist of deformations of facial components and their spatial relations, along with changes in the pigmentation of the same. This work envisages the detection of faces, localization of features thus leading to emotion recognition in images. Total Accuracy is 71.7%.[25].

Mayur and Dhopte, 2012, presented an approach to recognize human face expression and emotions based on some fuzzy pattern rules. Facial features for this specially eye and lips are extracted an approximated into curves which represents the relationship between the motion of features and change of expression. This work focuses the concepts like face detections, skin color segmentation, face features extractions and approximation and fuzzy rules formation. Conclusion based on fuzzy patterns never been accurate but still intension is to put more accurate results. [26]

2.3.6 Classification Based on General Methods:

Ayarpadi et al., 2012, proposed an integrated face recognition system that is robust against facial expressions by combining information from the computed intra-person optical flow and the synthesized face images. The main aim of this proposed work is to improve the accuracy of the face recognition system using the multiple training images. Making recognition more reliable under uncontrolled lighting conditions is one of the most important challenges for practical face recognition systems. He tackles this by combining the strengths of robust illumination normalization. Using synthesized images they could improve the accuracy of the face recognition system in the proposed system.[27]

Georgia Sandbach et al., 2012, mentioned that Automatic facial expression recognition constitutes an active research field due to the latest advances in computing technology that make the user's experience a clear priority. The majority of work conducted in this area involves 2D imagery, despite the problems this presents due to inherent pose and illumination variations. In order to deal with these problems, 3D and 4D (dynamic 3D) recordings are increasingly used in expression analysis research. In this research he surveyed the recent advances in 3D and 4D facial expression recognition. He discuss developments in 3D facial data acquisition and tracking, and present currently available 3D/4D face databases suitable for 3D/4D facial expressions analysis as well as the existing facial expression recognition systems that exploit either 3D or 4D data in detail. Finally, challenges that have to be addressed if 3D facial expression recognition systems are to become a part of future applications are extensively discussed. [28]

2.3.7 Classification Based on Vector Methods:

Murthy and Jadon, 2009, mentioned that making Computer Systems to recognize and infer facial expressions from the user image is a challenging research topic. A method of facial expression recognition, based on Eigens faces is presented. The approach identified the user's facial expressions from the input images, using a method that was modified from eigen face recognition. He have evaluated his method in terms of recognition accuracy using two well-known Facial Expressions databases, Cohn- Kanade facial expression database and Japanese Female Facial Expression database. The experimental results show the effectiveness of our scheme. The accuracy rates are increased by 80% for happiness and by 23, 25 % in case of surprised and sad respectively. Best-recognized category is surprise 83 % followed by happiness 80%. [29]

Mandeep Kaur and Rajiv, 2010, 2011, presented a new idea for detecting an unknown human face in input imagery and recognizing his/her facial expression. The objective of this research is to develop highly intelligent machines or robots that are mind implemented. A Facial Expression Recognition system needs to solve the following problems: detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification. The universally accepted five principal emotions to be realized are: Angry, Happy, Sad, Disgust and Surprise along with neutral. Principal Component Analysis (PCA) is implemented with Singular value decomposition (SVD) for Feature Extraction to determine principal emotions. The experiments show that the proposed facial expression recognition framework yields relatively little degradation in recognition rate due to facial images wearing glasses or loss of feature points during tracking. [30][31]

Ayesha Butalia et al., 2012, stated that Non-verbal communication may be used to enhance verbal communication or even provide developers with an alternative for communicating information. Emotion or Gesture recognition is been highlighted in the area of Artificial Intelligence and advanced machine learning. Emotion or gesture is an important feature for an intelligent Human Computer Interaction. This work is a literature survey which reveals with the research work already dealt with in this area. Facial expression has been concluded as the most important part involved in it. Even Facial features are also distinguished out of which eyes and mouth is probably more prominent. Neural networks are the widely used. Approaches towards Rough Fuzzy definition can be probably resolve the complexity. Context based recognition can be added so as to resolve the ambiguity involved in different scenarios. [32]

2.3.8 Classification Based on Statistical Method:

Feng et al., 2005, proposed a novel approach to recognize facial expressions from static images. First, the local binary patterns (LBP) are used to efficiently represent the facial images and then the linear programming (LP) technique is adopted to classify seven facial expressions—anger, disgust, fear, happiness, sadness, surprise, and neutral. Experimental results demonstrate an average recognition accuracy of 93.8% in the JAFFE database, which outperforms the rates of all other reported methods in the same database. [33]

Rachael et al., 2009, stated that central to all human interaction is the mutual understanding of emotions, achieved primarily by a set of biologically rooted social signals evolved for this purpose—facial expressions of emotion. Although facial expressions are widely considered to be the universal language of emotion, some negative facial expressions consistently elicit lower recognition levels among Eastern compared to Western groups. Here, focusing on the decoding of facial expression signals, they merge behavioral and computational analyses with novel spatiotemporal analyses of eye movements, showing that Eastern observers use a culture-specific decoding strategy that is inadequate to reliably distinguish universal facial expressions of "fear" and "disgust." Rather than distributing their fixations evenly across the face as Westerners do, Eastern observers persistently fixate the eye region. Using a model information sampler, he demonstrated that by persistently fixating the eyes, Eastern observers sample ambiguous information, thus causing significant confusion. The results question the universality of human facial expressions of emotion, highlighting their true complexity, with critical consequences for cross-cultural communication and globalization. [34]

CHAPTER THREE

THEORETICAL BACKGROUND

3.1 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is an abstract simulation of a real nervous system that contains a collection of neuron units, communicating with each other via axon connections. Such a model bears a strong resemblance to axons and dendrites in a nervous system. Due to this self-organizing and adaptive nature, the model offers potentially a new parallel processing paradigm. This model could be more robust and user-friendly than the traditional approaches. ANN can be viewed as computing elements, simulating the structure and function of the biological neural network. These networks are expected to solve the problems, in a manner, which is different from conventional mapping. Neural networks are used to mimic the operational details of the human brain in a computer. Neural networks are made of artificial 'neurons', which are actually simplified versions of the natural neurons that occur in the human brain. It is hoped, that it would be possible to replicate some of the desirable features of the human brain by constructing networks that consist of a large number of neurons. A neural architecture comprises massively parallel adaptive elements with interconnection networks, which are structured hierarchically.

Artificial neural networks are computing elements, which are based on the structure and function of the biological neurons. These networks have nodes or neurons, which are described by difference or differential equations. The nodes are interconnected layer-wise or intra-connected among themselves. Each node in the successive layer receives the inner product of synaptic weights with the outputs of the nodes in the previous layer. The inner product is called the activation value. The activation value is passed through a non-linear function.

The function of a neuron is shown in Figure 3.1. When the vectors are binary or bipolar, hard-limiting non-linearity is used. When the vectors are analog, a squashed function is used. Some of the squashed functions are sigmoid (0 to 1), tanh (-1 to +1), Gaussian, logarithmic and exponential. A network with two states of a neuron (0 or 1, and –1 or 1) is called 'discrete', and the same with a continuous output is called 'analog'. If, in a discrete network at a particular time 't', the state of every neuron is updated, the network is said to be synchronous. If the state of only one neuron is updated, the network is said to be asynchronous. A network is feed forward, if there is no closed chain of dependence among neural states. The same network is feed backward, if there is such a closed chain. When the output of the network depends upon the current input, the network is static (no memory). If the output of the network depends upon past inputs or outputs, the network is dynamic (recurrent). If the interconnection among neurons changes with time, the network is adaptive; it is called non-adaptive. The synaptic weight updating of the networks can be carried out by supervised methods, or by unsupervised methods, or by fixed weight association networks methods. In the case of the supervised methods, inputs and outputs are used; in the unsupervised methods, only the inputs are used; and in the fixed weight association networks methods, inputs and outputs are used along with pre-computed and pre-stored weights.

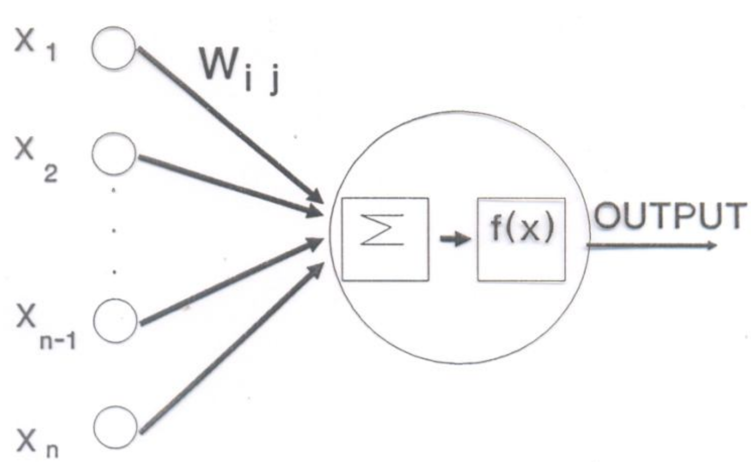


Fig.3.1 Operation of a neuron

Some of the supervised learning algorithms are the Perceptrons, decision-based neural networks, adaptive linear element (ADALINE), multilayer perceptron, temporal dynamic models and hidden Markov analysis. The various unsupervised learning algorithms are neo-cognition, self-organizing feature map, competitive learning, adaptive resonance theory (ART) and the principal component analysis. The fixed weight networks are Hamming net, Hopfield net and the combinatorial optimization. The total pattern recognition system constitutes instantiation space, feature extraction, training the network, and the testing the network.

The development of artificial neural networks was first reported in the early forties by McCullah and Pitts, 1943 [35]. In this model, a neuron fires if the sum of its excitatory inputs exceeds its threshold. This happens, as long as it receives no inhibitory input. Using this model, it is possible to construct a network that can compute any logical function. Rosenblatt, 1961 [36], found that the McCulloh-Pitts model was unbiological. In order to overcome the deficiencies in the McCulloh-Pitts model, he found out a new model, namely, the perceptron model, which could be utilized to learn and generalize. Further, he investigated several mathematical models, which included competitive learning or self-organization, and forced learning which is somewhat similar to reinforcement learning.

In addition to the above two types of learning, the concept of supervised learning was developed and incorporated in the adaptive linear element model (ADALINE). The ADALINE was found by Widrow and Hoff, 1960 [37].

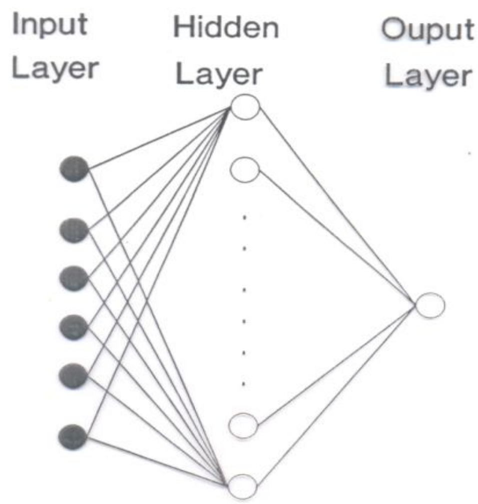


Fig.3.2 Multilayer network

The ADALINE is a single neuron, which uses a method to descend the gradient of the error, by using the supervised learning. The ADALINE is a linear neuron, and it is helpful to discriminate the patterns, which are linearly separable. The concept of multi-layer ADALINEs or multi-layer network was developed for patterns which were not linearly separable. The structure of a multi-layer network is shown in Figure 3.2. The training of the multi-layer network was first explained by Werbos, 1974 [38], as back-propagation algorithm (BPA) in his Ph.D. dissertation. His work did not become popular.

Rumelhart, 1996 [39], and his group published the parallel processing, a two-volume collection of studies on a broad variety of neural network configurations. Through these books, the concept of back-propagation algorithm became popular for training a multi-layer network.

Lippmann, 1987 [39], briefed the concept of different algorithms in his tutorial, and he still made neural networks more popular.

Much work has been carried out, with respect to the number of hidden layers, the number of hidden nodes in the hidden layer, methods of representing the patterns, training the network with initial random weights at different ranges, types of error criteria used and selection of patterns. Even though the training procedure for the neural network is unique and problem-oriented, it is sufficient to have one hidden layer for most of the problems solved by supervised training. Sietsma and Dow, 1991 [41], have analyzed the various training strategies with more than one hidden layer and finally claimed that one hidden layer was sufficient. Chester, 1990 [42], has claimed better performance for the network with two hidden layers. The number of nodes in a hidden layer should be, neither too many, nor too few. Too many nodes in the hidden layer will result in the oscillation of the mean squared error (MSE) around a particular value without any convergence; or sometimes the network converges to one of the local minima. Similarly, too few a number of nodes in the hidden layer will sometimes be just suitable, only to learn the training patterns, but generalization of the network is not possible. Therefore, it is necessary that there should be a way to find out the optimum number of nodes in a hidden layer. Hirose et al., 1991 [43], have adapted a different approach, by using an algorithm based on MSE to estimate the same. To overcome the difficulty of analyzing the number of nodes in the hidden layer, Weymaere and Martens, 1991 [44], have used Gaussian function in the hidden nodes and sigmoid function in the output nodes. Fujita, 1992 [45], has analyzed hidden unit function.

In most of the supervised training methods, the patterns are presented in a pre-determined sequence in a cycle. Normally, the order of presentation of the patterns is maintained in all the cycles. Ridgway, 1962 [46] found in his thesis that cyclic presentation of patterns could lead to cyclic adaptation. These cycles would cause the weights of the entire network to cycle, by preventing convergence. Various error criteria have been tried by Zaki, 1964 [47] and Walach and Widrow, 1984 [48], for better convergence of the network. Quantization of the weights and training BPA has been analyzed by Shoemaker et al., 1991 [49]. Analysis of BPA with respect to mean weight behavior was done by Bershad et al., 1993 ,[50].

In reality, most of the patterns are not linearly separable. Non-linear classifiers are used for pattern classification, in order to achieve good seperability. The multiplayer network is a non-linear classifier, since it uses hidden layer. In addition to multiplayer network, polynomial discriminant function (PDF) is also a non-linear classifier. In the PDF, the input vector is pre-processed similar to the suggestions by Specht, 1990 [51].

Normally, neural networks are used for classify patterns by learning from samples. Different neural networks paradigms employ different learning rules. In some way, all these paradigms determine different pattern statistics from a set of training samples. Then, the network classifies new patterns on the basis of these statistics. The BPN uses steepest descent method, which is slow and linear in convergences. This algorithm may get stuck in local minima.

3.1.2 Advantages of ANN over other classical methods

Unlike statistical estimators, ANN estimates a function without a mathematical model of how outputs depend functionally on the inputs; it learns from samples and offers robust and adaptive processing capabilities, by adopting adaptive learning and self-organization rules. It enhances the network approximation classification and has noise-immunity capabilities. It employs many processing units enhanced by extensive interconnectivity.

3.1.3 Areas of applications of ANN

The application domains of ANN are in association / clustering / classification, pattern recognition, regression / generalization, and in optimization. The different areas of applications are in computer vision, signal/image processing, speech / character recognition, remote sensing and in controls. Recent applications of ANN include the machining processes, process planning, inventory control, shape based classification, machine fault diagnosis and cold forging.

3.1.4 Multi-Layer Perceptron:

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function.

The universal approximation theorem for neural networks states that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multi-layer perceptron with just one hidden layer. This result holds only for restricted classes of activation functions, e.g. for the sigmoidal functions.

Multi-layer networks use a variety of learning techniques, the most popular being ***back-propagation***. Here the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles the network will usually converge to some state where the error of the calculations is small. In this case one says that the network has *learned* a certain target function. To adjust weights properly one applies a general method for non-linear optimization task that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason back-propagation can only be applied on networks with differentiable activation functions.

In general the problem of teaching a network that performs well, even on samples that were not used as training samples, is a quite subtle issue that requires additional techniques.

This is especially important for cases where only very limited numbers of training samples are available. The danger is that the network over-fits the training data and fails to capture the true statistical process generating the data. Computational learning theory is concerned with training classifiers on a limited amount of data. In the context of neural networks a simple heuristic, called early stopping, often ensures that the network will generalize well to examples not in the training set.

Other typical problems of the back-propagation algorithm are the speed of convergence and the possibility to end up in a local minimum of the error function. Today there are practical solutions that make back-propagation in multi-layer perceptron the solution of choice for many machine learning tasks.

A very basic example for implementing the XOR function is as shown in figure 3.3.

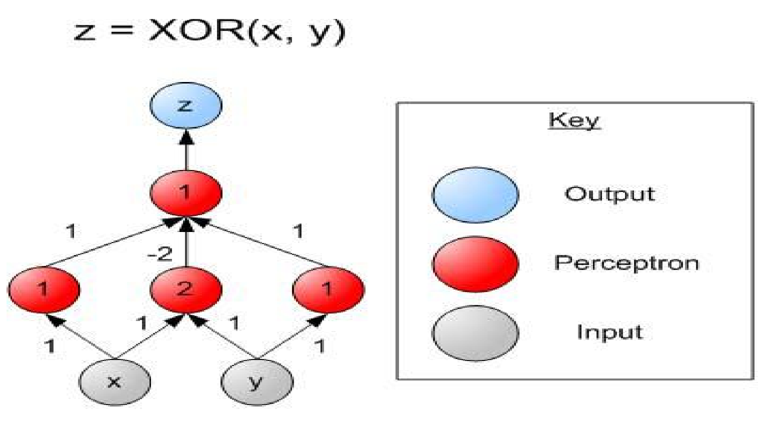


Figure 3.3 An example implementation of XOR function using MLP

The MLP neural network [52] has feed forward architecture within input layer, a hidden layer, and an output layer. The input layer of this network has N units for an N dimensional input vector. The input units are fully connected to the ***I*** hidden layer units, which are in turn, connected to the ***J*** output layers units, where ***J*** is the number of output classes.

A Multi-Layers Perceptron (MLP) is a particular of artificial neural network [53]. We will assume that we have access to a training dataset of *l* pairs (x_i, y_i) where x_i is a vector containing the pattern, while y_i is the class of the corresponding pattern. In our case a 2-class task, y_i can be coded 1 and -1.

3.1.5 Single Layer Feed Forward Network

A neural network in which the input layer of source nodes projects into an output layer of neurons but not vice-versa is known as single feed-forward or acyclic network. In single layer network, 'single layer' refers to the output layer of computation nodes as shown in Figure 3.4.

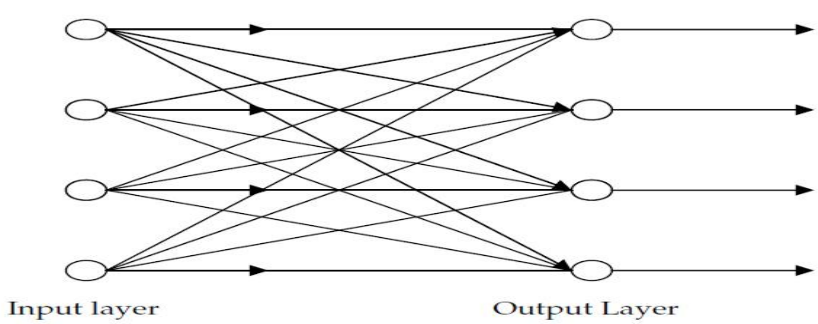


Fig.3.4 A Single layer feed forward network

3.1.6 The CMAC Neural Network

The Cerebellar Model Articulation Controller (CMAC) is a type of neural network based on a model of the mammalian cerebellum. It is also known as the Cerebellar Model Arithmetic Computer. It is a type of associative memory as defined by Horvath, 2007 [54].

The CMAC was first proposed as a function modeler for robotic controllers by James Albus, 1975 [52], but has been extensively used in reinforcement learning and also as for automated classification in the machine learning community. CMAC computes a function $f(x_1,\ldots.x_n)$, where n is the number of input dimensions. The input space is divided up into hyper-rectangles, each of which is associated with a memory cell. The contents of the memory cells are the weights, which are adjusted during training. Usually, more than one quantization of input space is used, so that any point in input space is associated with a number of hyper-rectangles, and therefore with a number of memory cells. The output of a CMAC is the algebraic sum of the weights in all the memory cells activated by the input point.

The basic operation of a two-input two-output CMAC network is illustrated in figure (3.4a). It has three layers, labeled L1, L2, and L3 in the figure. The inputs are the values y_1 andy_2. Layer 1 contains an array of "feature detecting" neurons z_{ij} for each input y_i. Each of these outputs one for inputs in a limited range, otherwise they output zero (figure 3.4b).

For any input y_i a fixed number of neurons (n_a) in each layer 1 array will be activated (n_a= 5 in the example). The layer 1 neurons effectively quantize the inputs. Layer 2 contains $n_v$association neurons a_{ij} which are connected to one neuron from each layer 1 input array (z_{1i}; z_{2j}).

Each layer 2 neuron outputs 1.0 when all its inputs are nonzero, otherwise it outputs zero—these neurons compute the logical AND of their inputs. They are arranged so exactly n_a are activated by any input (5 in the example).

Layer 3 contains the $n_x$output neurons, each of which computes a weighted sum of all layer 2 outputs, i.e.:

$$(x)_i = \sum w_{ijk} a_{jk} \qquad \textbf{(3.1)}$$

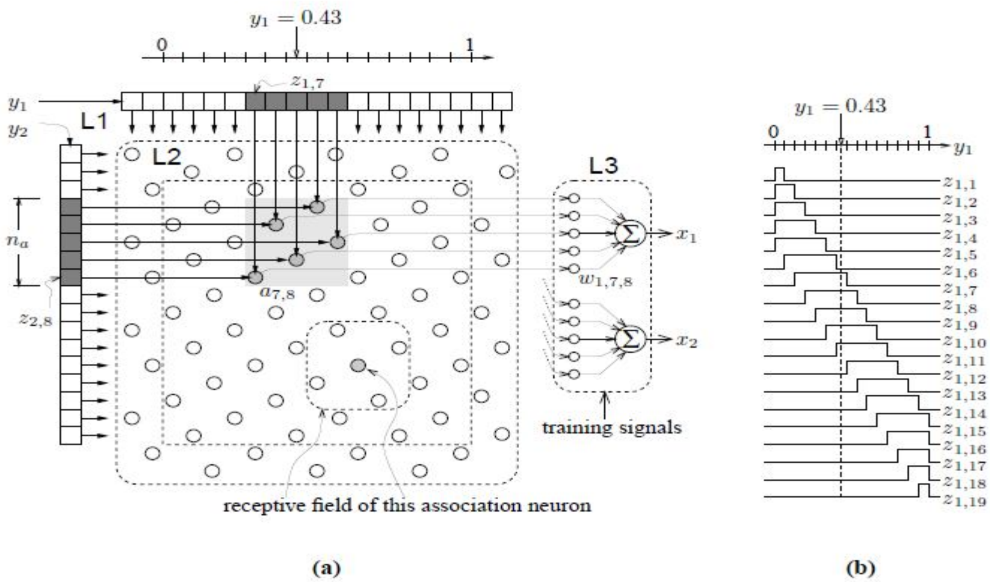


Fig.3.5 (a) An example two-input two-output CMAC, in neural network form (n_y = 2, n_a= 5, n_v= 72, n_x= 2).

(b) Responses of the feature detecting neurons for input 1.

The parameters w_{ijk} are the weights which parameterize the CMAC mapping (w_{ijk} connects a_{jk} to output i). There are n_x weights for every layer 2 association neuron, which makes n_v n_x weights in total.

Only a fraction of all the possible association neurons are used. They are distributed in a pattern which conserves weight parameters without degrading the local generalization properties too much. Each layer 2 neuron has a receptive field that is n_a x n_a units in size, i.e. this is the size of the input space region that activates the neuron.

The CMAC was intended by Albus to be a simple model of the cerebellum. Its three layers correspond to sensory feature-detecting neurons, granule cells and Purkinje cells respectively, the last two cell types being dominant in the cortex of the cerebellum. This similarity is rather superficial (as many biological properties of the cerebellum are not modeled) therefore it is not the main reason for using the CMAC in a biologically based control approach. The CMAC's implementation speed is a far more useful property.

It is apparent that, conceptually at least, the CMAC deserves to be called a neural network. However, actual software implementations are much more convenient if the CMAC is viewed as a lookup table. [52]

The CMAC has the following properties:

1) **Limited input space:** Each CMAC input has a minimum and maximum value, beyond which the weight tables do not hold any weight values. Thus the designer needs to know beforehand what the probable input ranges are.
2) **Piecewise constant:** The CMAC input to output mapping is piecewise constant because of the quantized input. In other words the mapping contains many discontinuous steps. This may seem likely to result in a

poor mapping, but in fact it is not a great disadvantage in many applications. If the input resolutions are chosen to be large enough then the discontinuities will be small. There is a tradeoff, however, because more input resolution results in a larger virtual weight space.

3) **Local generalization:** In contrast to the multi-layer perceptron (MLP), the CMAC has local generalization (LG). This means that when each data point is presented to the CMAC training algorithm, only a small region of the mapping around that point is adjusted. This occurs, of course, because only the weights which affect the output are adjusted, and each of those weights can only influence a small area of the mapping.
4) **Training sparsity:** For a small number of fixed training points the generalization performance is inferior to the MLP, as the inter-point regions are hardly ever perfectly interpolated. As the inter-point space decreases below the size of the LG region, the generalization becomes better. For a large number of "randomly" spaced training points (perhaps generated by some function or process) the CMAC is a good functional approximator.
5) **Training interference:** Consider training a CMAC at two points, first A then B, with some learning rate α. If point B is within the LG area of point

A it can make the mapping error at point A worse. This effect is called training interference

6) **Multidimensional inflexibility:** A single input CMAC has enough weight parameters so that it can generate an independent output for each quantized input. In standard CMACs with more than one input this cannot be done, because the association neurons (or weight table entries) are distributed rather sparsely in the quantized input space.

The CMAC neural network has the following advantages:

1) The mapping and training operations are extremely fast. The time taken is proportional to the number of association units.
2) The algorithms are easy to implement.
3) Local generalization prevents over-training in one area of the input space from degrading the mapping in another (unless there are too few physical weights).

It has the following disadvantages:

1) Many more weight parameters are needed than for, say, the multi-layer perceptron.
2) The generalization is not global, so useful interpolation will only occur if there are enough training points—points further apart than the local generalization distance will not be correctly interpolated.

3) The input-to-output mapping is discontinuous, without analytical derivatives, although this can be remedied with higher order CMACs [53].

4) Selection of CMAC parameters to prevent excessive hash collision can be a large design problem. If there are a limited number of physical weights available then it is difficult to do without knowledge of the input signal coverage, so trial-and-error must usually be used. If physical weights are plentiful then it is acceptable to use far more than are really necessary and not worry about hash collisions.

Despite these problems, the CMAC is extremely useful in real-time adaptive control because of its speed.

Far greater processing power would be required when using, say, an MLP network where every weight is involved in every mapping and training operation.

3.2 Gabor Wavelet

Dennis Gabor (Gábor Dénes) was a Hungarian physicist who is most notable for inventing holography, an achievement for which he later received the Nobel Prize in Physics in 1971. It was in 1946 that the first time frequency wavelets (Gabor wavelets) were introduced by Dennis Gabor [55], who at that time was researching into communication theory.

Gabor wavelet transformation has been used in various kinds of signal and pattern processing/analysis areas both in spatial and in frequency domains and is found to give out satisfactory results in application areas such as texture segmentation, fingerprint recognition and face recognition. The characteristics of Gabor wavelets such as their ability to get easily adjusted for detailed localization in spatial and frequency domains [56] and the similarity of their frequency and orientation representations to those of the human visual system have made them popular for particular usage areas and as a result they have been found to be revealing satisfactory results in the application areas mentioned above.

Gabor wavelet representation of face images derives desirable features gained by spatial frequency, spatial locality, and orientation selectivity. These discriminative features extracted from the Gabor filtered images could be more robust to illumination and facial expression changes. [57]

Gabor showed that there exists a ”quantum principle” for information: the conjoint time-frequency domain for 1D signals must necessarily be quantized so that no signal or filter can occupy less than a certain minimal area in it. This minimal area, which reflects the inevitable trade-off between time resolution and frequency resolution, has a lower bound in their product, analogous to Heisenberg’s uncertainty principle in physics. He discovered that Gaussianmodulated complex exponentids provide the best trade-off. The original Gabor elementary functions, in the form proposed by Gabor [55], are generated with a fixed Gaussian while the frequency of the modulating wave varies. These are equivalent to a family of‘ ”canonical” coherent states generated by the Weyl-Heisenberg group.

3.2.1 The Gabor Filter Bank

A two-dimensional Gabor function g(x,y) and its Fourier transform G(u,v) can be written as:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\frac{x^2}{\sigma_x{}^2}+\frac{y^2}{\sigma_y{}^2}\right]+2\pi jWx} \quad \textbf{(3.2)}$$

$$G(u,v) = exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u{}^2} + \frac{v^2}{\sigma_v{}^2}\right]\right\} \quad \textbf{(3.3)}$$

where (W) is the frequency of a sinusoidal plane wave along the x-axis, and σ_x *and* σ_y are the space constants of the Gaussian envelope along the and axes, respectively. $\sigma_u = {}^1/_{2\pi\sigma_x}$ and $\sigma_v = {}^1/_{2\pi\sigma_y}$. Filtering a signal with this basis provides a localized frequency characterization. Filters with arbitrary orientations can be obtained by a rigid rotation of the x-y coordinate system.

$$\acute{g}(x,y) = g(\acute{x},\acute{y}) \quad \textbf{(3.4)}$$

Where

$$\acute{x} = x\cos\theta + y\sin\theta, \acute{y} = -x\sin\theta + y\cos\theta \qquad \textbf{(3.5)}$$

and θ is the rotation angle.

The Gabor filter bank is designed to cover the entire frequency spectrum [5], [27]. In other words, the Gabor filter set is constructed such that the half-peak magnitude of the filters in the frequency spectrum touches each other. This results in the following formulas to compute the filter parameters σ_u and σ_v.

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{S-1}}, W = a^m U_l \qquad \textbf{(3.6)}$$

$$\sigma_u = \frac{(a-1)W}{(a+1)\sqrt{2\ln 2}} \qquad \textbf{(3.7)}$$

$$\sigma_v = \tan\left(\frac{\pi}{2K}\right)\left[W - \frac{(2\ln 2)\sigma_u{}^2}{W}\right] \times \left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u{}^2}{W^2}\right]^{-\frac{1}{2}}$$

(3.8)

Where U_l and U_h denote the lower and upper center frequencies of interest. $m \in \{0,1,..,S-1\}$ and $n \in \{0,1,..,K-1\}$ are the indices of scale and orientation, respectively. K is the number of orientations and S is the number of scales. [56].

3.2.2 Gabor Feature Extraction

Gabor filter can capture salient visual properties such as spatial localization, orientation selectivity, and spatial frequency characteristics. Considering these excellent capacities and its great success in face detection and face recognition, they choose Gabor features to represent the face image.

The Gabor representation of an image is the convolution of the image with the Gabor filter. Based on the Gabor representations, a feature vector is formed. In our experiment they use the Gabor filters with the following parameters: five scales $v \in \{0\ 1\ 2\ 3\ 4\}$ and eight orientations $u \in \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\}$.

3.3 Facial Tracking to Locate Face

Face tracking can be measured by the algorithm that analyzes video frames. Then find out the position of moving faces within the each frame. For each tracked face, three steps are involved. Initialization, tracking, stopping. Many methods used a face detector for initialization of their tracking processes. There have been studies on the profile pose face detectors. They suffered from the false-detection problem. Choudhury et al., 2003 [58], described that instead of a fixed threshold value to initialize the face tracker, used two face probability maps. One used for frontal views and another one for profiles. All local maxima in this map is chosen as face candidates. The face probabilities of which are transmitted throughout the temporal sequence. The non-faces are identified and eliminated by probabilities if they either go to zero/remain low. To represent an intermediate head pose, information from the two face probability maps is combined the experiments revealed that the probabilistic detector was very accurate than a traditional face detector. It could handle head movements covering ±90 degrees out-of-plane rotation.

After initialization, before tracking a face, one should choose the features to track The exploitation of color is one of the common choices because it is invariant to facial expressions, scale, and poses changes Boccignone et al., 2005 [59]. Certain type of processed video, which depend on, a learning set dedicated to a Color based face trackers. It might not work on unknown videos with changing illumination conditions. They are more robust to various illuminations and occlusions.

Tong et al., 2007 [60], described facial features extraction from Eyes, nose, mouth. Even though the overview of important points allows for tracking of different kinds of objects, minus any face specific knowledge. This method's power to distinguish between the target and clutter. It might not be sufficient to deal with background noise. Facial features always support tracking of high level facial information. Little use when the video is of poor quality. Most of the facial feature based face trackers have been tested using non-broad cast video.

Instead of choosing only a few features to track, an appearance based/feature less tracker matches an observation model of the whole facial appearance with the input image. Li et al., 2006 [61], uses a multi-view face detector to identify and track faces from various poses. Also, the face based observation model, a head model is also considered representing the back of the head. The face is not always trackable, so this model is basically on the idea that a head can be an object of interest. To handle occlusions due to out-of-plane head rotations exceeding ±90 degrees, a comprehensive particle filter is used to fuse these two sets of information.

Face tracking systems typically use a motion model during the tracking procedure, which defines how the image of the target might change for dissimilar possible face motions. If the face to be a planar object, then the corresponding motion model can be a 2D transformation, Affine transformation or homography, of a facial image. Certain research treats the face as a rigid three dimensional object. The resulting motion model outlines aspects depending on three-dimensional position and orientation. A face is truly both three dimensional and deformable. Some systems try to model faces, and the image of the face can be covered with a mesh. A sophisticated geometry and texture face model and Ahlberg 2004 [62], Dornaika and Davoine 2006 [63]. By the position of the nodes of the mesh, the motion of the face is defined. A motion model will give accurate results, if the quality of the video is high. A geometry and texture models are less false face detections than a two dimensional transformation model. Three dimensional based and mesh based face trackers involved in a clear appearance, high resolution and a limited pose variation. These requests cannot be fulfilled in the case of broadcast video. Almost three dimensional based, mesh-based face trackers are only tested on non-broad cast video like webcam video.

In case of tracking errors, not able to stop a face track. So, stopping procedure establishes a major lack for the face tracking algorithms. Arnaud, et al., 2005 [64], proposed an approach that uses an object tracker for face tracking. A stopping criterion based on the eye tracker to lighten drifting. The 2 positions of the tracked eyes are matched with the tracked face position. The tracking process stops if neither of the eyes is in the face region (Drifting is determined). Mostly mesh-based or top-down trackers are expected to be able to avoid drifting.

3.3.1 Challenges in Face Tracking

The important challenges in face tracking methods are variations of pose, lighting, facial deformations, occlusion, clutter and facial resolution. Pose and illumination variations frequently lead to loss of track. The familiar methods for dealing with illumination variations were presented in Hager and Belhumeur, 1998 [65]. The authors described the movement of the image points, using a parameterized function by considering illumination variation also modifying by the brightness constancy of optical flow.

Koterba et al., 2005 [66], considered the illumination invariant three-dimensional tracking within the Active Appearance Model (AAM). The method requires training images to build the model. So the result depends on the quality and variety of data. Three-dimensional model based motion estimation algorithms are the generally robust to pose variations, but lack of illumination.

Xu and Roy-Chowdhury, 2007 [67], proposed a model based face tracking method that was very tough to both pose and lighting changes. This was completed through an analytically resulting model for relating the appearance of a face in terms of its pose, the lighting, shape and surface reflectance.

Tracking through Facial Deformations is another puzzling problem. In tracking faces lot of change of expressions.

Terzopoulos and Waters, 1993 [68], methods were used by so many researchers for tracking, recognition and reconstruction.

In compare to this model based approach, Black and Yacoob, 1995 [69] offered data driven approach for tracking and recognition of non-rigid facial motion. More recently, the three-dimensional morphable model by Blanz and

Vetter, 2003 [70] have been pretty popular in synthesizing various facial expressions. It entails that it can be used for tracking by posing the problem as estimation of the synthesis parameters.

Occlusion and Clutter are the most tracking problems. It affects the performance of the face trackers. One of the strong tracking approach, is the use of particle filters used by Arulampalam et al, 2002 [71]. It can recover from a loss of track. It gives a high enough number of particles and observations. Since occlusion and clutter remain important problem in the design of highly robust face tracking systems.

Facial resolution is another problem in tracking. Low resolution will obstruct the performance of face tracking algorithm.

Zhao et al., 2003 [72], recognized in video based face recognition that low resolution to be very important problem. High resolution of faces is also a challenging problem because of over detailed facial features that need to be displayed accurately.

Dedeoglu et al., 2006 [73], proposed a method for face super resolution using AAMs.

Followed by interpolation, high resolution requires recording of multiple images. These 2 stages are considered separately because the recording is acquired through a tracking procedure followed by high resolution.

Yu et al., 2007 [74], proposed the high resolved texture values are feeding back in the nth frame for tracking the $(n+1)^{th}$ frame. This improves the tracking and also improves the high resolution output. Issues of stability and convergence could be an interesting area for future work.

3.4 Fisher's Linear Discriminant Function

Emotional facial feature extraction is the process of converting a face image into a feature vectors. The feature vector carrying characteristics of the face. This vector is used as the basis for emotional expression classification. This vector for the emotional expression recognition must have all the essential features for the classification.

After the detection of face, Holistic Approach or Analytic Approach is used to extract the features: In Holistic methods, raw facial image is exposed for feature extraction. Though, in analytic, some important facial features are detected. Here, they used Holistic Approach, as that it means they send a raw image. There are three types of feature extraction techniques are used. Namely, Feature-based method, Appearance based method and Hybrid method. The geometric features like facial points or shapes of facial components or spatial locations are used by feature-based methods for e.g. FACS. The colors, colors layouts or textures of the facial skin including wrinkles and furrows are used by the appearance-based method. Local feature based methods. The geometric and appearance facial features are used by hybrid methods. The facial emotional features are extracted either from the entire image or from regions. The whole image feature is simpler. While region based representation of emotional images is proved to be more accurate.

Pantic, 2000 [23] identified 3 basic problems in facial expression analysis approach. It deals with face detection in a facial image/image sequence, data extraction from facial expression, and classification of facial expression. The systems assume the presence of a full frontal face view in the image. Some knowledge of the global faces location. To give the correct location of the face, Viola, 2001 [75], for rapid face detection, use the Adaboost algorithm to, fully permit a search sub window above the image at multiple scales.

Essa, 1997 [76], to extract motion blobs from image sequences, the spatial and temporal filtering together with thresholding is used. To detect the presence of a face, the blobs are then evaluated using the Eigen faces method (Turk 1991 [77]). Principal component analysis (PCA) to calculate the distance of the experiential region from a face space of 128 sample images. The Person Spotter system (Steffens 1998 [78]) used spatio temporal filtering and stereo disparity in pixels. In the video, it tracks the bounding box of the head, so selecting image regions of interest. It then works skin color and convex region detectors to check for face presence in these regions.

Littlewort 2002, [18] used a bank of 40 Gabor wavelet filter at different scales and orientations to achieve convolution for data extraction.

Lades 1993 [79], extracted a jet of magnitudes of complex valued responses at different locations in a lattice executed on an image.

Cohn, 1998 [80], physically localize feature points in the first frame of an image sequence. Then used hierarchical optical flow to track the motion of small windows which is surrounded these points across frames. To represent the extracted expression information, the displacement vectors for each landmark between the first and the highest frame is used. In the last step of expression analysis, expressions are classified. The most widespread approaches are based on six basic emotions: anger, disgust, fear, joy, sorrow and surprise as said by Ekman 1999 [56]. The Facial Action Coding System (FACS), developed by Ekman and Friesen 1978 [81], a combination of 44 facial movements called Action Units are used for expressions. Whereas great progress has been made in automatically classifying according to FACS (Tian 2001 [82]). A fully automated FACS based approach for video wanted to develop.

Dailey 2000 [83], used a six unit. Neural network with a Single layer is used to classify six basic emotion categories extracted from static images.

3.4.1 Principles of Transformation

The transformation of a set on n-dimensional real vectors onto a plane is called a mapping operation. The result of this operation is a planar display. The mapping operation can be linear or non-linear. Fisher, 1936 [84], has developed a linear classification algorithm

Hong and Yang, 1991 [85], have used a method for constructing a classifier on the optimal Discriminant plane, by using minimum distance criterion for multiclass classification for less of patterns.

Foley, 1972 [86], had discussed the method of considering the number of patterns and feature size.

Gallinari et al., 1991 [87], has discussed the relations between Discriminant analysis and multiplayer Perceptrons. A linear mapping is used to map an n-dimensional vector space. Many methods are available under both linear and non-linear mappings.

Siedlecki, Siedlecka and Skalansky, 1988 [88], have given an overview of mapping techniques. The mapping of the original vector ‘X’ onto a different vector ‘Y’ on a plane is completed by a matrix transformation, which is given by equation (3.9)

$$Y = AX + b \tag{3.9}$$

Where

$$A = \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \end{bmatrix} \tag{3.10}$$

b is a 2-dimensional vector,

φ_1 is a projection vector (also called a Discriminant vector) and

φ_2 is another projection vector (also called a Discriminant vector).

The 2-dimensional vector does not introduce any relevant information, but it is given for the generality of the equation 3.10. The steps involved in the linear mappings are:

Step 1: Computation of the Discriminant vectors φ_1 and φ_2. This is exact for a particular linear mapping algorithm.

Step 2: Computation of the planar images of the original data points: this is common for all linear mapping algorithms.

Some of the linear mapping algorithms are

i) Principal component mapping, Kittler and Young, 1973 [89].

ii) Generalized declustering mapping, Sammon, 1970 [90], Fehlauer and Eisenstein, 1978 [91], and Gelsema and Eden, 1980 [92].

iii) Least squared error mapping, Mix and Jones,1982 [93] and

iv) Projection pursuit mapping, Friedman and Turkey, 1974 [94]

3.5 Summary

This chapter has presented introduction to algorithms which have been proposed in this research work. Chapter 4 presents the implementation of the proposed algorithms.

CHAPTER FOUR

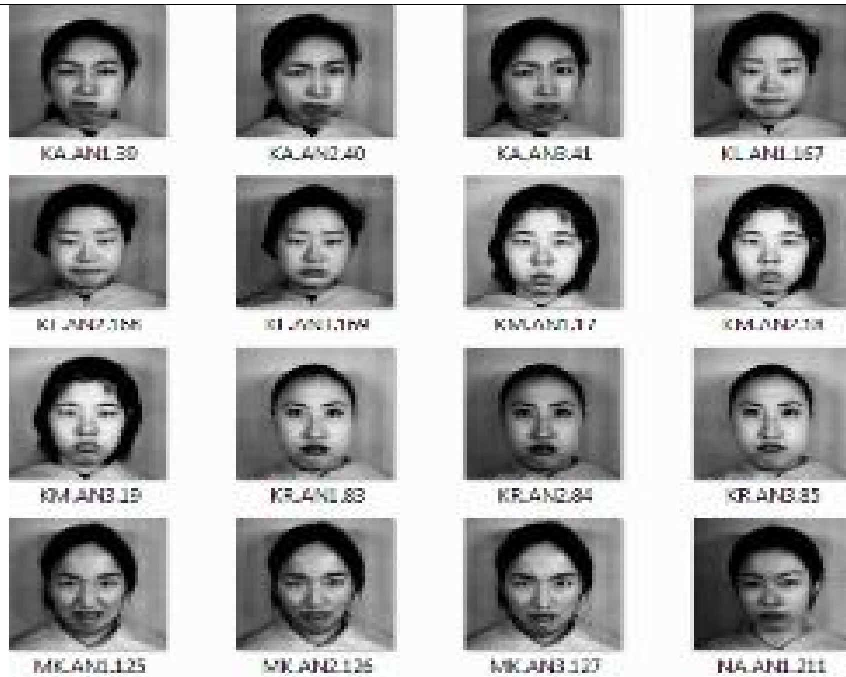
THE PROPOSED SYSTEM (METHODOLOGY)

4.1 Introduction

This chapter presents the emotion images extracted from facial image . The person is asked to express his/her feelings to different situations. The possible expressions are considered for this research work. In addition facial expressions are collected from standard database.

4.2 Database used in Emotion Expressions

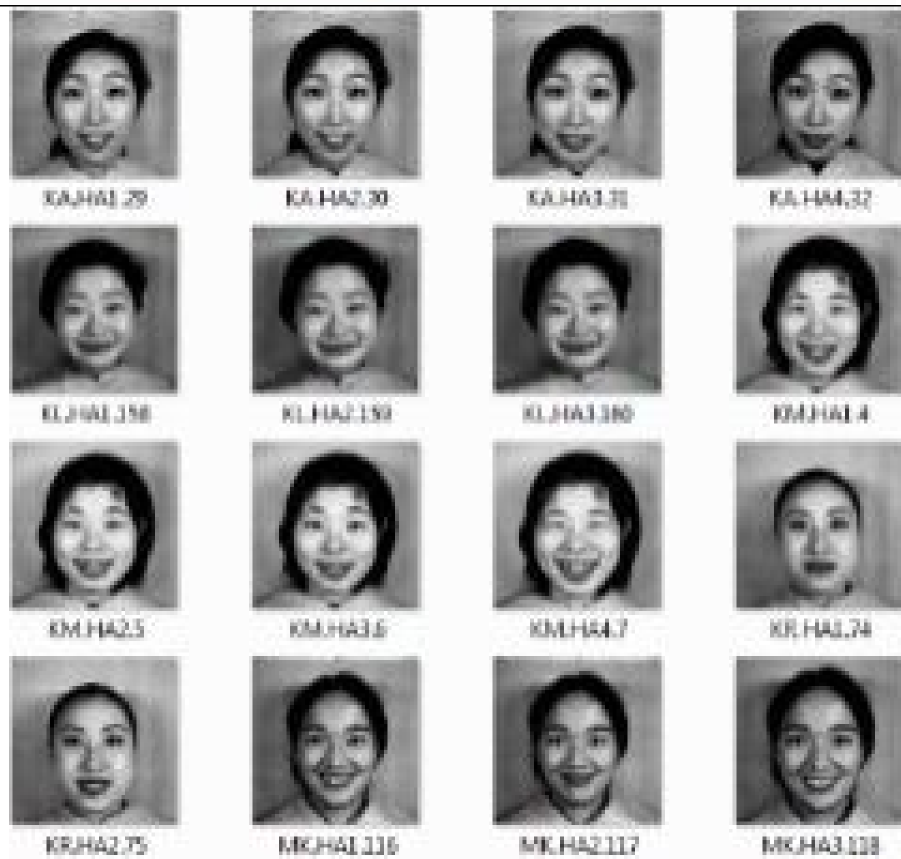
Table 4.1 presents emotion expression for anger, disgust, fear, happy, sad and surprise obtained from videos. The videos have been shot on a female who had expressed reactions through her face when particular statements are read aloud. Ten expressers posed 3 or 4 examples of each of the six basic facial expressions (happiness, sadness, surprise, anger, disgust, fear), and a neutral face for a total of 213 images of facial expressions. The actual names of the subjects are not revealed but they are referred with their initials: KA, KL, KM, KR, MK, NA, NM, TM, UY and YM. [33]

Expression name	Sample of expression	
Angry		
Disgust		

Fear


KA.FE1.45
KA.FE2.46
KA.FE3.47
KA.FE4.48
KL.FE1.174
KL.FE2.175
KL.FE3.176
KM.FE1.23
KM.FE2.24
KM.FE3.25
KR.FE1.89
KR.FE2.90
KR.FE3.91
MK.FE2.131
MK.FE2.132
MK.FE3.133

Happy


KA.HA1.29
KA.HA2.30
KA.HA3.31
KA.HA4.32
KL.HA1.158
KL.HA2.159
KL.HA3.160
KM.HA1.4
KM.HA2.5
KM.HA3.6
KM.HA4.7
KR.HA1.74
KR.HA2.75
MK.HA1.116
MK.HA2.117
MK.HA3.118

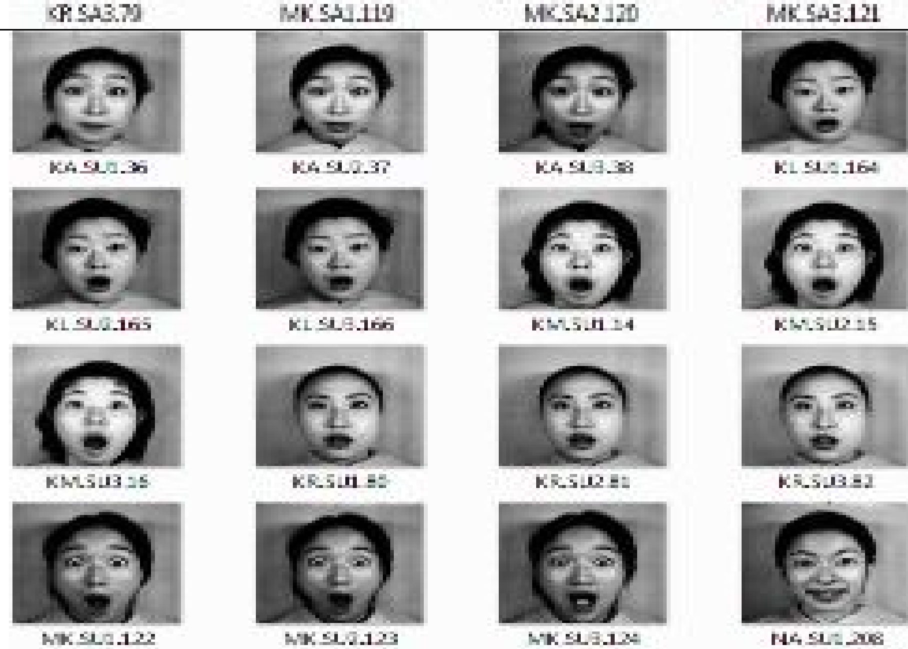
Sad	
Surprise	

Table 4.1 sample frames of expressions

4.3 Schematic Diagram for Emotion Recognition

The overall architecture for emotion recognition is presented in Figure 4.1

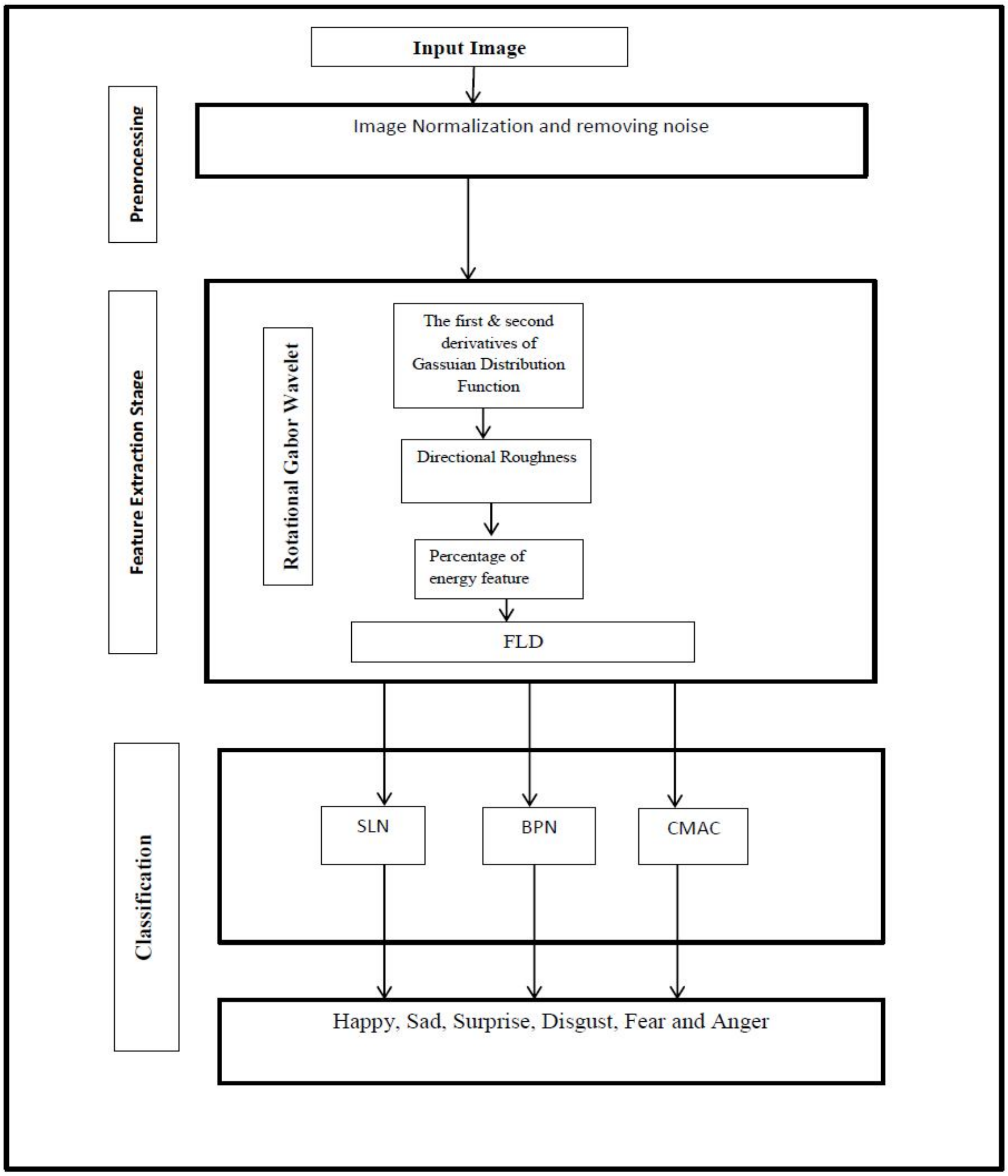


Fig.4.1 Schematic Diagram for Emotion Recognition

Step 1: Input facial image.

Step 2: Input each facial image to Gabor Wavelet, then, Fisher's Linear Discriminant function and obtain features.

Step 3 Train the SL/BPN/CMAC and store final weights.

Testing ANN for identifying emotions

Step 4: Read features from FLD and process with SL/BPN/CMAC.

Step 5: Get output in the output layer.

Step 6: Compare the output with threshold to decide the type of emotion.

The image acquisition stage reads the digital image. Preprocessing is used to normalize the image and remove noise. Wavelet scaling of image is done to reduce the amount of time taken for segment and classification. Removal of noise is done using Gaussian smoothing function. Feature extraction deals with extracting attributes those results in quantitative information.

4.4 Preprocessing

Preprocessing is done for the removal of noise from image using Gaussian smoothing function. In designing Gaussian filters the mask weights are computed directly from the Gaussian distribution, given by equation (4.1)

$$\phi(x, y, s) = \exp\{ \frac{-x^2 + y^2}{2s^2} \} \qquad (4.1)$$

where

x, y - directions , s - scale

An overlapping moving window size of NxN is used for preprocessing. The coefficient value at the center of the mask is made equal to one by a suitable multiplication factor. When performing the convolution, the output pixel values must be normalized by the sum of the mask weights to ensure that regions of uniform intensity are not affected.

4.5. Feature Extraction

Feature extraction extracts attributes from an image that results in quantitative information to produce a description about the image. It is the basis for differentiating one class of object from other. Roughness is a result of abrupt changes or transition located at the object edges. Two stages are involved in extracting the features using Directional wavelet.

1. The extraction of directional roughness features and
2. Percentage of energy features

The directional roughness features results in high quality facial segmentation performance. The use of percentage of energy feature retains the important properties of fractal dimension based features like insensitivity to absolute illumination and contrast. The percentage energy feature computed using exponential wavelets is used for segmenting different features in a given image by using k-means algorithm.

4.5.1. Rotational Wavelet Gabor Filter

Wavelet transform extract both the time (spatial) and frequency information from a given signal, and the tunable kernel size allows it to perform multi-resolution analysis. Gabor functions provide the optimal resolution in both the time (spatial) and frequency domains. Gabor wavelet transform has both the multi-resolution and multi-orientation properties.

The Gabor wavelet is used as the discrete wavelet transform with either continuous or discrete input signal, while there is an intrinsic disadvantage of the Gabor wavelets which makes this discrete case beyond the discrete wavelet constraints: the 1-D and 2-D Gabor wavelets do not have orthonormal bases. If a set of wavelets has orthonormal bases, the inverse transform could be easily reconstructed by a linear superposition, and this wavelet transform provides a complete representation. The non-orthonormal wavelets provide a complete representation only when they form a frame. The non-orthonormal properties are not much required, if the Gabor wavelets are used for feature extractions.

Texture segmentation plays an important role in recognizing and identifying a material, type of characteristic for particular image. Wavelets are employed for the computation of single and multi-scale roughness features because of their ability to extract information at different resolutions. Features are extracted in multiple directions using directional wavelet obtained from partial derivative of Gaussian distribution function. The first and second derivative wavelets are used to obtain the features of the facial image at different orientations like 0°, 45°, 90° and 135° and scales such as 1, 2 and 4.

Segmentation procedure partition an image into constituent object that is used to find the regions of interests using K-means algorithm. Figure 4.9 presents flow chart for facial image segmentation using Rotational Wavelet Gabor Filter.

The implementation of facial segmentation steps are as follows:

Step 1: Read a facial image.

Step 2: Calculate the first partial derivative w_0, w_{90}.

Step 3: Calculate second partial derivative $w_{(0,90)}$, $w_{(0,0)}$, $w_{(90,90)}$.

Step 4: Find directional roughness feature by using arithmetic averaging.

Step 5: Calculate total energy.

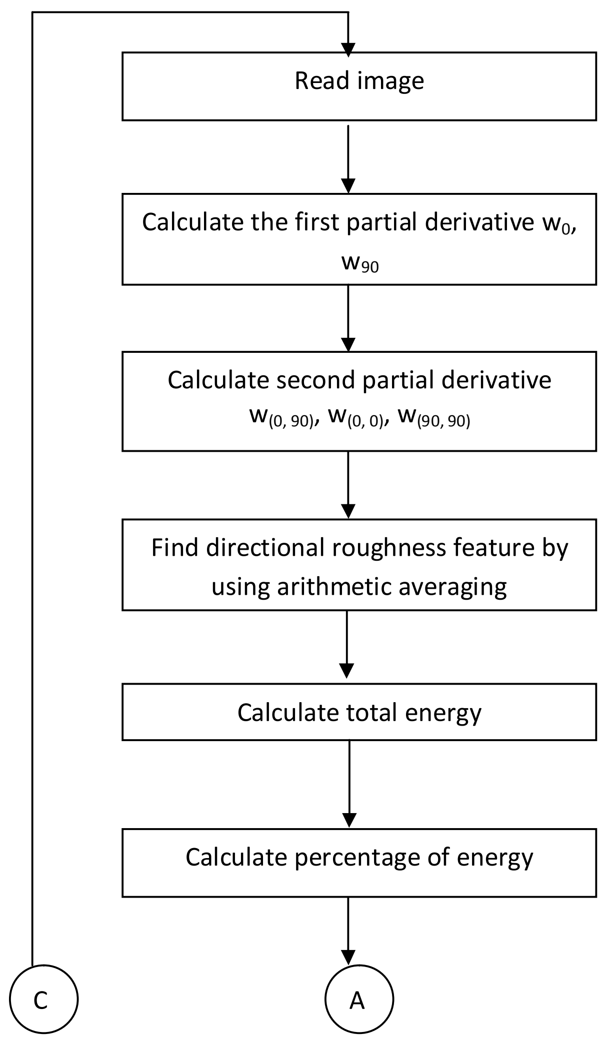
Step 6: Calculate percentage of energy.

Step 7: Done for three orientations?

Step 8: If No, then go to step 1.

Step 9: If Yes, Apply k-means algorithm.

Step 10: Initialize no. of clusters, no. of iterations, energy value.



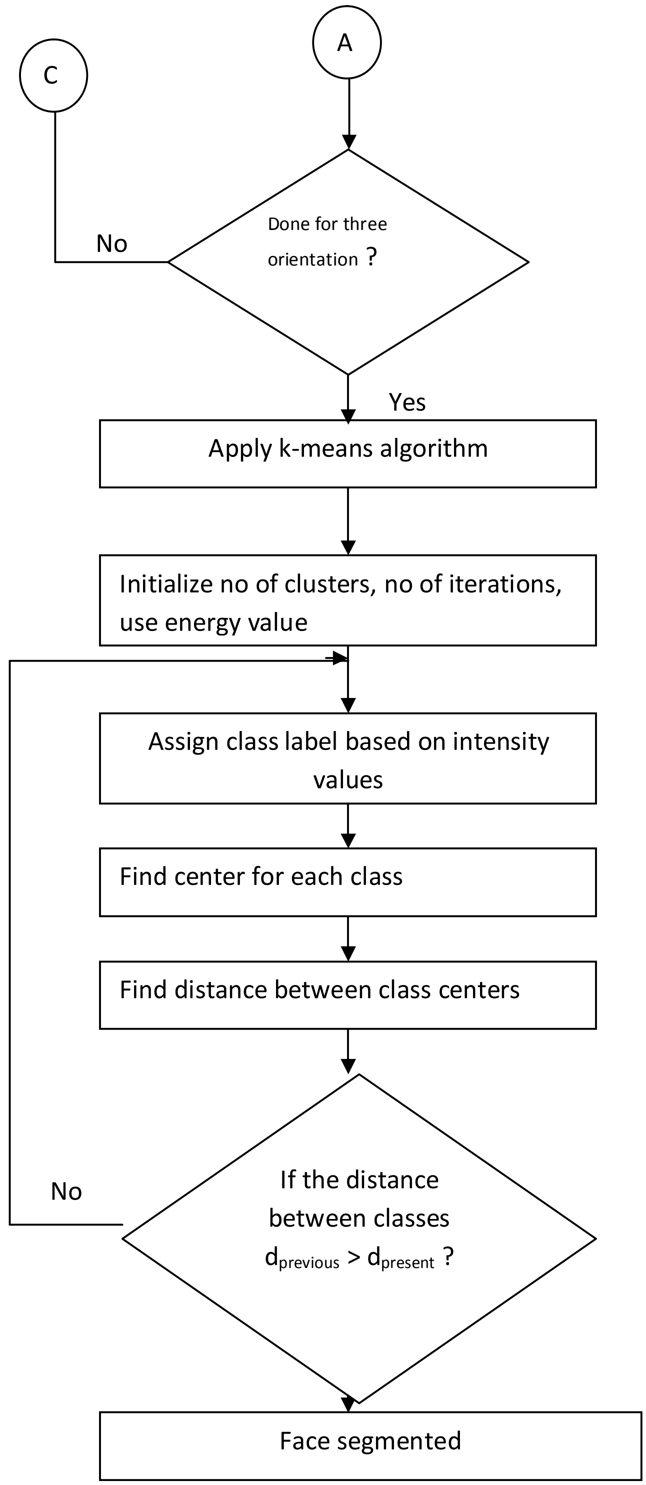


Fig.4.2 Flow chart of facial image segmentation

Step 11: Assign class label based on intensity values.

Step 12: Find center for each class.

Step 13: Find distance between class centers.

Step 14: Is the distance between classes dprevious > dpresent ?

Step 15: If No, go to step 11.

Step 16: If yes, facial image is segmented.

4.5.2 The Directional Roughness Features ($R^{\theta}_{s^i}$)

A wavelet from the exponential wavelet family is used for the computation of roughness features. The 2-D Gaussian smoothing function from equation (4.1) is partially differentiated with respect to x and y to calculate the first order partial derivatives of the smoothing function along x-axis and y-axis.

Along x-direction (0°)

$$\mathbf{W_0(x,y,s) = \frac{\partial\phi(x,y,s)}{\partial x} = \frac{-x}{s}\exp\{\frac{-x^2+y^2}{2s^2}\}} \tag{4.2}$$

Along y-direction (90°)

$$\mathbf{W_{90}(x,y,s) = \frac{\partial\phi(x,y,s)}{\partial x} = \frac{-y}{s}\exp\{\frac{-x^2+y^2}{2s^2}\}} \tag{4.3}$$

Gradient component or the filtered version of the original image in direction 0° and 90° is calculated by convolving the original image f(x,y) with the partial derivative filters along x and y directions.

$$\mathbf{W0(x,y,s) * f(x,y) = \frac{\partial\phi(x,y,s)}{\partial x} * f(x,y)} \qquad \textbf{(4.4)}$$

$$\mathbf{W_{90}(x,y,s) * f(x,y) = \frac{\partial\phi(x,y,s)}{\partial x} * f(x,y)} \qquad \textbf{(4.5)}$$

The filtered version of the original image along other directions other than 0° and 90° is calculated by using the linear combination of equations 4.4 and 4.5 and is given in equation (4.6)

$$\mathbf{W_0(x,y,s) * f(x,y) = [W_0(x,y,s) * f(x,y)]\cos\theta + [W_{90}(x,y,s) * f(x,y)]\sin\theta} \qquad (4.6)$$

where

θ - is the directional angle

Similarly, the second order partial derivatives of the smoothing function is calculated along the directions (0^o, 90^o), (0^o, 0^o) and (90^o, 90^o). The second order partial along the direction (0^o, 90^o) is obtained by partially differentiating W_0 (x, y, s) with respect to x as given in equation (4.7)

$$\mathbf{W_{0,90}(x,y,s) = \frac{\partial^2\phi(x,y,s)}{\partial x\partial y} = \frac{xy}{s^4}\exp\{\frac{-x^2+y^2}{2s^2}\}} \qquad \textbf{(4.7)}$$

The second order partial along the direction (0^o, 0^o) is obtained by partially differentiating W_0 (x, y, s) with respect to x as given in equation (4.8)

$$\mathbf{W_{0,0}(x,y,s) = \frac{\partial^2 \phi(x,y,s)}{\partial^2 x} = (\frac{x^2}{s^4} - \frac{1}{s^2})\exp\{\frac{-x^2+y^2}{2s^2}\}} \quad \textbf{(4.8)}$$

The partial along the direction (90^o, 90^o) is obtained by partially differentiating W_{90} (x, y, s) with respect to y as given in equation (4.9)

$$\mathbf{W_{90,90}(x,y,s) = \frac{\partial^2 \phi(x,y,s)}{\partial^2 y} = (\frac{y^2}{s^4} - \frac{1}{s^2})\exp\{\frac{x^2+y^2}{2s^2}\}} \quad \textbf{(4.9)}$$

The filtered version of the original image along other directions is obtained by the linear combination of the equations 4.7, 4.8 and 4.9 as given in equation (4.10)

$$\mathbf{W_{\theta,\theta+90}(x,y,s) * f(x,y) = [W_{0,90}(x,y,s) * f(x,y)] * \cos 2\theta + 0.5 * \{[W_{0,0}(x,y,s) * f(x,y)] - [W_{90,90}(x,y,s) * f(x,y)]\}\sin 2\theta} \quad \textbf{(4.10)}$$

The gradient component along any directions is found by using the equations 4.6 and 4.10. The two wavelet transforms of a function $f(x,y)$ at scale s and direction θ are calculated as given in equation (4.11)

$$\mathbf{W_1 T_f^{\theta}(x,y,s) = W_0(x,y,s) * f(x,y)} \quad \textbf{(4.11)}$$

$$\mathbf{W_2 T_f^{\ \theta}(x,y,s) = W_{\theta,\theta+90}(x,y,s) * f(x,y)} \quad \textbf{(4.12)}$$

where

$W_1 T_f^{\theta}(x,y,s)$ – is the first derivative wavelet

$W_2 T_f^{\theta}(x,y,s)$ – is the second derivative wavelet

The directional roughness features is obtained by finding the arithmetic average in a 9 x 9 window for the two wavelet transforms given in the equations 4.11 and 4.12 with different orientations and scales. The wavelet with the maximum is selected as given in equation (4.13)

$$R_{s^i}^{\theta} \approx< \max |W_i T_f^{\theta}(u,v,s)| >_{NxN} \qquad \textbf{(4.13)}$$

<> NxN – arithmetic average in an N×N window

i = 1, 2 (1 and 2 derivative wavelet)

4.5.3 Percentage of Energy Feature

The effect of roughness depends on the relative texture energy between different directions. The roughness features are weighted with the percentage of textural energy existing in the corresponding direction. The energy computed in direction θ using an s-scale wavelet is given by equation (4.14).

$$\mathbf{E_{s^i}^{\theta} =<| W_i T_f^{\theta}(x,y,s) |>_{NXN}} \qquad \textbf{(4.14)}$$

Total Energy

The total energy at scale s is obtained from equation (4.15)

$$E_{s^i}^{total} = \sum_{\theta} <| W_i T_f^{\theta}(x,y,s) |>_{NxN} \qquad \textbf{(4.15)}$$

Percentage of Energy

The percentage of energy feature computed in direction θ and scale s is given by equation:

$$Per_{s^i}^{\theta} = \frac{E_s^{\theta}}{E_s^{total}} - \qquad (4.16)$$

where

E_s^{θ} is the energy computed in direction θ

E_s^{total} is the total energy

The percentage of energy $Per_{s^i}^{\theta}$ is insensitive to the absolute image illumination because energy is computed using exponential wavelets where the dc component is removed. It is also insensitive to contrast changes, because E_s^{θ} and E_s^{total} is multiplied by a constant multiplicative term. The schematic flow of the extracting energy feature from the facial image is given in figure 4.3.

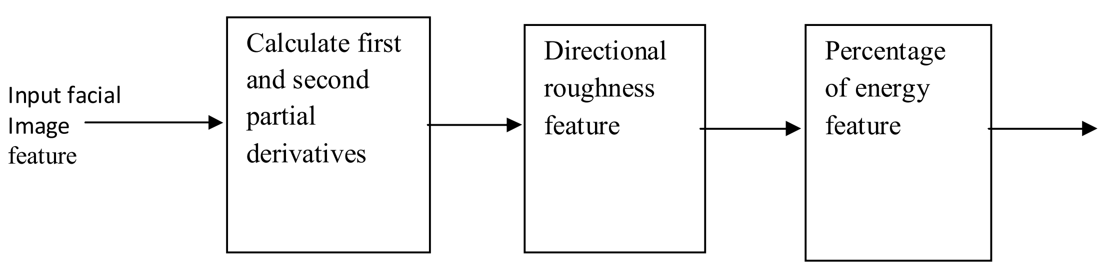


Fig.4.3 Design of feature extraction

4.6 Facial Feature Representation Using Fisher's Linear Discriminant Function

As mentioned in chapter 3 there are four methods in linear mapping algorithms Out of these four methods, generalized de-clustering optimal

Discriminant plane, based on a mapping technique of Fisher is used. The vectors φ_1, and φ_2 are discriminant vectors. The plane formed by them is the discriminant plane which is optimal.

4.6.1 Computation of Discriminant Vectors φ_1 And φ_2

The Fisher's criterion is given by

$$J(\varphi) = \frac{\varphi^T S_b \varphi}{\varphi^T S_w \varphi} \quad \textbf{(4.17)}$$

$$S_b = \sum_{i-1}^{m} P(\omega_i)(m_i - m_o)(m_i - m_o)^T \quad \textbf{(4.18)}$$

$$S_w = \sum_{i-1}^{m} P(\omega_i) E\left[(x_i - m_i)(x_i - m_i)^T / \omega_i\right] \quad \textbf{(4.19)}$$

Where

S_b is the between class matrix, and

S_w is the within class matrix which is non-singular.

Where

P (ω_i) is a priori the probability of the i^{th} pattern, generally P (ωI) = 1/m,

m_i is the mean vector of the i^{th} class patterns, I=1, 2…... m;

m_o is the global mean vector of all the patterns in all the classes.

$X = \{X_i, I=1, 2....L\}$ is the n-dimensional patterns of each class.

The Discriminant vector that maximizes J in equation 4.17 is denoted by φ_1. The vector φ_1 is found as a solution of the Eigen value problem given by

$$S_b\varphi_1 = \lambda_{m1} S_W\varphi_1 \qquad \textbf{(4.20)}$$

Where λ_{m1} is the greatest non-zero Eigen value of $S_b\ S_w^{-1}$. The eigenvector corresponding to λ_{m1} is φ_1. Another Discriminant vector φ_2 is obtained by using the same criterion of equation 4.17. The vector φ_2 should also satisfy the equation given by

$$\varphi_2^T\varphi_1 = 0.0 \qquad \textbf{(4.21)}$$

The equation 4.21 indicates that the solution obtained is geometrically independent. The Discriminant vector φ2 is found as a solution of the Eigen value problem, which is given by (4.22).

$$Q_pS_b\varphi_2 = \lambda_{m2}S_W\varphi_2 \qquad \textbf{(4.22)}$$

Where

λ_{m2} is the greatest non-zero eigenvalue of $Q_pS_bS_w^{-1}$ and Q_p is the projection matrix given by(4.23)

$$Q = I - \frac{\varphi_1}{\varphi_1^T} \frac{\varphi_1^T}{S_W^{-1}} \frac{S_W^{-1}}{\varphi_1} \quad \textbf{(4.23)}$$

Where

I is an identity matrix.

In equation 4.20, S_w should be non-singular. It is necessary that S_w should be non-singular, even for a more general discriminating analysis and generating multi orthonormal Foley and Sammon, 1975[95], Liu et al. [96], 1992, and Cheng et al., 1992 [97], vectors, If S_w is singular, S_w should be made on-singular, by using singular value decomposition (SVD) method and perturbing the matrix. Klema and Laub, 1980 [98], have discussed the SVD and the method of its computation with some applications. Sullivan and Liu, 1984 [99], has explained the SVD with application to signal processing. By using equations (4.22 and 4.23), the values of φ_1 and φ_2 discriminant vectors are obtained and they are given by:

$$\varphi_1 = \begin{bmatrix} -0.13059656 & 0739955 \\ -0.09095359 & 5816678 \\ -0.06548560 & 2401720 \\ 0.06717354 & 2644008 \\ 0.29328135 & 5542666 \\ 0.47468756 & 8145903 \\ 0.46897722 & 0521294 \\ 0.36064186 & 0260513 \\ 0.17954562 & 6038281 \\ -0.05646176 & 6225427 \\ -0.19018648 & 2893432 \\ -0.15807285 & 2973602 \\ -0.10211950 & 7177282 \\ -0.06935812 & 5399382 \\ -0.02968253 & 6209896 \end{bmatrix} \qquad \varphi_2 = \begin{bmatrix} -0.04041746 & 4909674 \\ -0.09167520 & 0730481 \\ -0.05502264 & 4452162 \\ 0.11746745 & 8321155 \\ 0.23323184 & 4182514 \\ 0.12606430 & 7529118 \\ -0.05675134 & 4343668 \\ -0.16774214 & 7406088 \\ -0.10991863 & 7599609 \\ 0.00056600 & 8138727 \\ 0.08104532 & 0736508 \\ 0.03046641 & 0830449 \\ -0.05315576 & 4472642 \\ -0.06326505 & 3687451 \\ 0.05239123 & 8811373 \end{bmatrix}$$

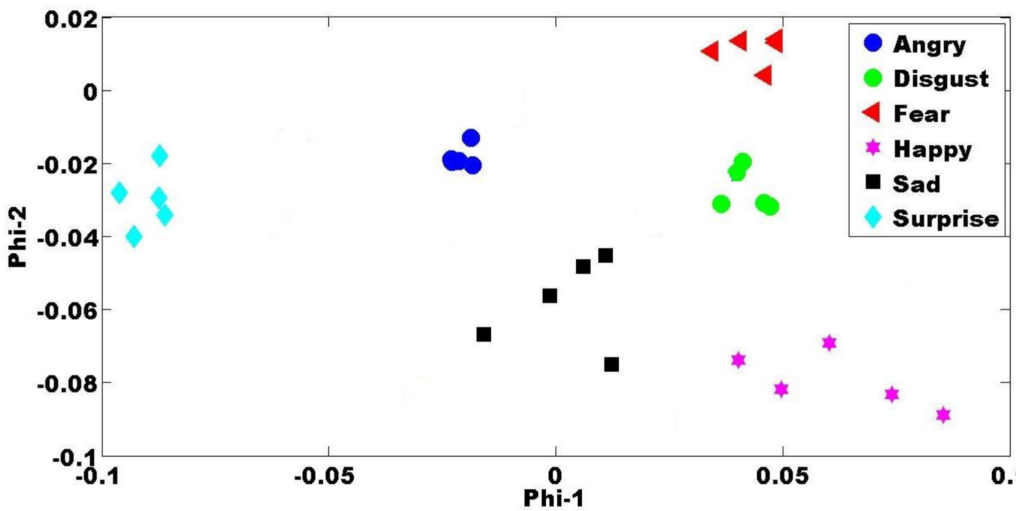


Fig. 4.4 Plot of discriminant vectors for all 6 expressions

Figure 4.4 shows plots of 6 different expressions based on the FLD output. In this figure, the expressions happy and sad are scattered and not clutters. The other expressions are cluttered.

4.6.2 Computation of 2-Dimensional Vector From The Original N-dimensional Vector

The 2-dimensinoal vectors set are denoted by Y_i. The vector Y_i is given by:

$$Y_i = (u_i, v_i) = \{X_i^T\varphi_1, X_i^T, \varphi_2\} \quad \textbf{(4.24)}$$

The vector set Y_i, is obtained by projecting the original vector 'X' of the patterns onto the space spanned by φ_1 and φ_2 by using equation 4.24.

Let X (k) $\subset$X belongs to the K^{th} class. The corresponding 2-dimensional sets of vectors are

$$y^{(k)} =(y_1^{(k)}, y_2^{(k)},\ldots\ldots\ldots\ldots,y_{Lk}^{(k)}) \quad \textbf{(4.25)}$$

$$\sum_{i-1}^{m} L_k - L$$

Where

$$y_i^{(k)} = (u_i^{(k)} , v_i^{(k)}), i = 1,2,3,\ldots\ldots, L_{k,}$$

(4.26)

The 2-dimensional vectors of the training patterns are given in Table 4.2.

Pattern No.	U	V
1	0.170027	0.26589654
2	0.186384	0.31478438
3	0.275119	0.15631626
4	0.207705	0.093517463
5	0.18044	0.27104327
6	0.180694	0.25908034
7	0.139818	0.2674465
8	0.148349	0.30747598
9	0.183	0.27899469
10	0.163267	0.27702721
11	0.145098	0.22248487
12	0.135343	0.26594515
13	0.175656	0.26430668
14	0.171221	0.24790403
15	0.191819	0.24980761

Table 4.2 Pattern used for training ANN

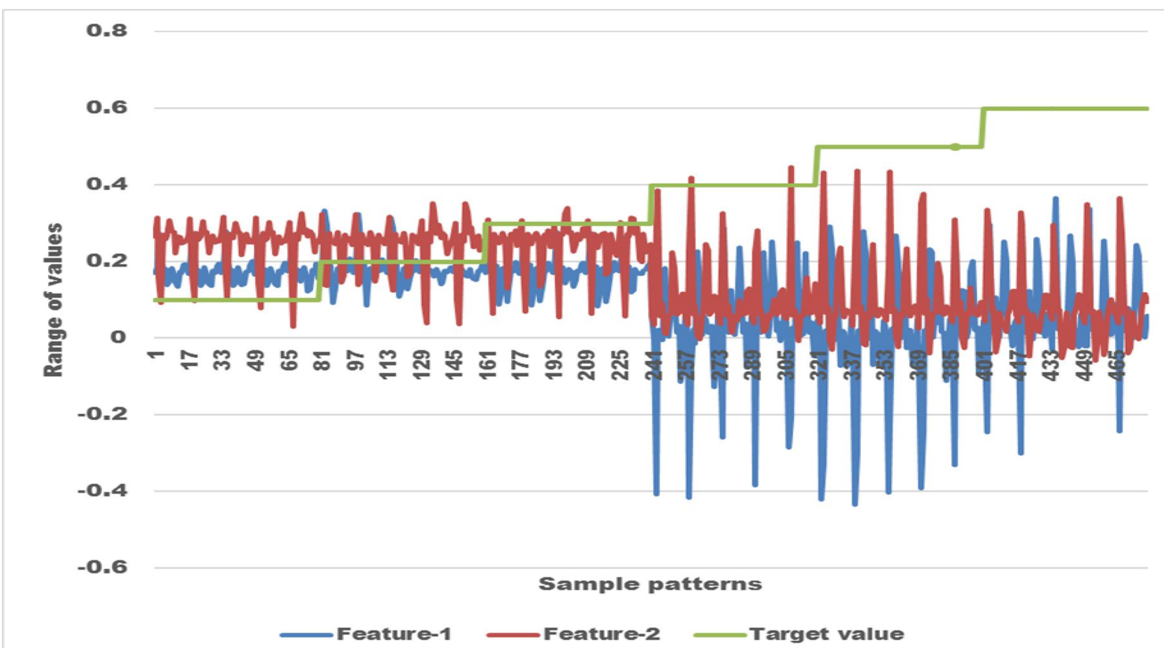


Fig.4.5 Training patterns for ANN algorithms

4.7 Normalization of the Patterns

The patterns are normalized so that the values of the features are in the range of 0 to 1, and the computational complexity is reduced. The normalization of the patterns is done by:

$$x_i = x_i / x_{max} \qquad \textbf{(4.27)}$$

Where

x_i is the value of a feature, and

x_{max} is the maximum value of the feature.

4.7.1 Selection of Patterns for Training

The numbers of classes, which are based on the classification range of the outputs, are decided. If only one output is considered, the range of classification is simple. If more than one output is considered, a combination criterion has to be considered. The total numbers of patterns are decided for each class. Out of these patterns, the number of patterns to be used for training the network is decided. The remaining patterns are used for testing the classification performance of the network. The patterns selected for training the network should be, such that they represent the entire population of the data.

The selection of patterns is done by:

$$E_i^2 = \frac{\sum_{j=1}^{nf} (x_{ij} - \bar{x}_j)^2}{\sigma_i^2} \qquad \textbf{(4.28)}$$

Where

E_i^2 is the maximum variance of a pattern,

nf is the number of features, and

$$\sigma_i^2 = \frac{\sum_{j=1}^{nf}(x_{ij} - \bar{x}_j)^2}{L} \qquad \textbf{(4.29)}$$

Where $\bar{x}_j$ Is the mean for each feature and

L Is the number of patterns

The value of E_i^2 is found for each pattern. Patterns with maximum E_i^2 are chosen from each class for training the network.

4.7.2 Training Strategies for the Network

For the network to learn the patterns, different weight updating algorithms have been developed. They are called supervised methods and unsupervised methods. Since both the inputs and outputs are considered for facial recognition, supervised learning technique has been used. The present work involves modification of existing weight updating algorithm, combination of classical method with neural network, method of training the network for more number of patterns, and training the network properly for more than two classifications.

The network functions on a supervised learning strategy. The inputs of a pattern are presented. The output of the network obtained in the output layer is compared with the desired output of the pattern. The difference between the calculated output of the network and the desired output is called error (MSE) of the network for the pattern presented. This error is propagated backwards, such that the weights connecting the different layers are updated. By this process, the MSE of the network for the pattern presented is minimized. This procedure is summed up. After presenting the last training pattern, the network is considered to have learnt all the training patterns through iterations, but the MSE is large. To minimize MSE, the network has to be presented with all the training patterns many times. There is no guarantee that the network will reach the global minimum; instead, it will reach one of the local minima. The MSE may increase, which means, divergence rather than convergence. Sometimes, there may be oscillations between convergence and divergence.

The training of the network can be stopped, either by considering MSE or by considering classification performance as the criterion. When classification performance is considered as the criterion, test patterns are presented at the end of each epoch (iteration). Once the desired performance is obtained, training of the network is stopped. When MSE is considered as the criterion, one may not know the exact MSE, to which the network has to be trained. If the network is trained till it reaches a very low MSE, over fitting of the network occurs. Over fitting represents the loss of generality of the network. That is, the network classifies only the patterns, which are used during training, and not the test patterns.

4.8 Single Layer Network:

This section presents a method of quick convergence of ANN trained by BPN algorithm. The method of weight updation is modified and the algorithm proposed is named as SLN.

In the previous section , the network was trained with analog data. The updating of the weights was done during the presentation of all the training patterns for all the iterations, till the desired MSE or the classification performance of the network for the test patterns was reached. In this section, functional update method is used. During the presentation of each training pattern, the weights of the network are updated, when a functional criterion is satisfied. Otherwise, no updating of the weights is done. The patterns are presented in the binary form.

4.8.1 Principles of SLN

The main idea of this method is that the weights of this network will be updated, only when any one of the nodes in the output layer of the network is misclassified. Even if one of the nodes in the output layer is not misclassified, no updating of weights is done. A node in the output layer is misclassified, if the difference between the desired output and the network is greater than 0.5. The flow chart for SLN with functional update is given in Figure 4.6. The number of layers and the number of nodes in the input layer are decided. The weights between layer, and Θ for the output layer, are initialized. A training pattern is presented to the input layer of the network, and the difference between the network's output and the target output is calculated for each node in the output layer. If the difference obtained for each node is greater than the value of functional criteria, a counter is incremented and the weights and thresholds are updated. If the difference of not even one node is greater than 0.5, no updating for the weights and thresholds is done.

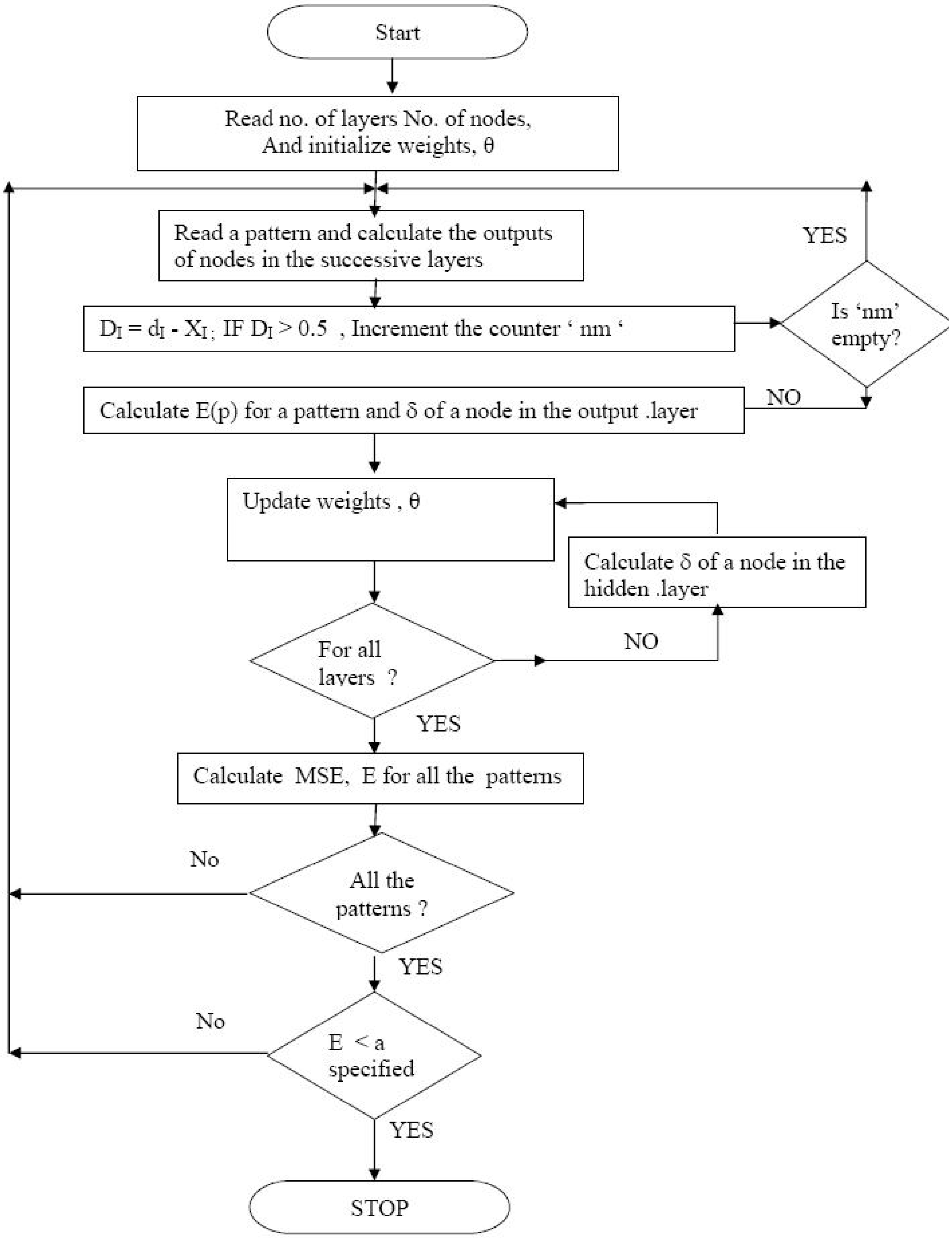


Fig.4.6 Flow-chart for SLN

The MSE of the network for each pattern is calculated, only when at least one node in the output of the network is misclassified. Remaining training patterns are presented to the network. Training of the network is stopped, when a performance index of the network is reached.

The algorithm for the functional update is as follows:

Step 1: The weights and thresholds of the network are initialized.

Step 2: The inputs and outputs of a pattern are presented to the network.

Step 3: The output of each node in the successive layers is calculated by:

$$x_i = 1/(1+\exp(\sum w_{ij}\, x_i + \theta_i)) \qquad \textbf{(4.38)}$$

Step 4: The number of nodes in the output layer, which are misclassified, are denoted by 'nm'. A node is misclassified, if it does not satisfy the equation:

$$1-\varepsilon > D \geq 0.5 \qquad \textbf{(4.39)}$$

Where

ε is the value fixed by the programmer, and

$$D = |\text{ Desired output-Network output }| \qquad \textbf{(4.40)}$$

If 'nm' is empty, i.e., not even one node satisfies equation 4.2, step 2 is adopted.

Step 5: If 'nm' is not empty, the objective function 'j' is computed by:

$$\mathbf{J} = \tfrac{1}{2}\sum \sum_{Xi \in nm} \mathbf{D}^2 \qquad \textbf{(4.41)}$$

Step 6: The weights and thresholds are updated.

Step 7: The steps (2-6) are adopted, until the total MSE of all the patterns is below a specified value.

4.9 Back-Propagation Neural Network (BPN)

The BPN uses the steepest-descent method to reach a global minimum. The flow chart for the BPN is given in Figure 4.7. The number of layers and number of nodes in the hidden layers are decided. The connections between nodes are initialized with random weights. A pattern from the training set is presented in the input layer of the network and the error at the output layer is calculated. The error is propagated backwards towards the input layer and the weights are updated. This procedure is repeated for all the training patterns. At the end of the each iteration, test patterns are presented to ANN, and the classification performance of ANN is evaluated. Further training of ANN is continued till the desired classification performance is reached.

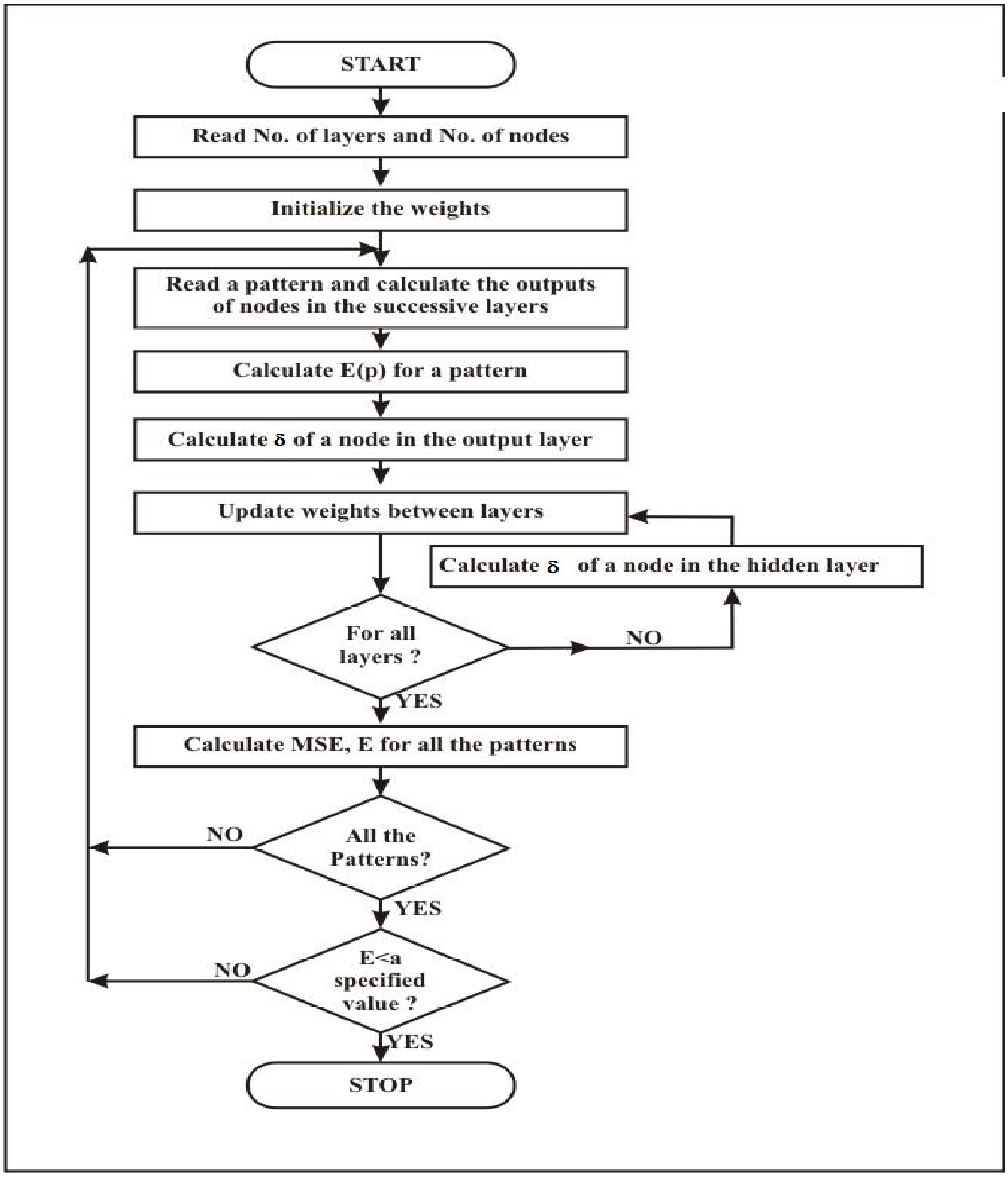


Fig 4.7 Flow chart of the back propagation algorithm

4.9.1 Forward propagation

1. The weights and thresholds of the network are initialized.

2. The inputs and outputs of a pattern are presented to the network.
3. The output of each node in the successive layers is calculated.

$$\mathbf{o_{(output\ of\ a\ node)} = 1/(1+exp(-\sum w_{ij}\ x_i\ ))} \quad \textbf{(4.30)}$$

4. The error of a pattern is calculated

$$\mathbf{E(p) = (1/2) \sum(d(p)–o(p))^2} \quad \textbf{(4.31)}$$

4.9.2 Reverse Propagation

1. The error for the nodes in the output layer is calculated

$$\mathbf{\delta_{(output\ layer)} = o(1\text{-}o)(d\text{-}o)} \quad \textbf{(4.32)}$$

2. The weights between output layer and hidden layer are updated

$$\mathbf{W(n+1) = W(n) + \eta\delta_{(output\ layer)}\ o_{(hidden\ layer)}} \quad \textbf{(4.33)}$$

3. The error for the nodes in the hidden layer is calculated

$$\mathbf{\delta_{(hidden\ layer)} = o(1\text{-}o) \sum\delta_{(output\ layer)}\ W_{(updated\ weights\ between\ hidden\ and\ output\ layer)}} \textbf{(4.34)}$$

4. The weights between hidden and input layer are updated.

$$\mathbf{W(n+1) = W(n) + \eta\delta_{(hidden\ layer)}\ o_{(input\ layer)}} \quad \textbf{(4.36)}$$

The above steps complete one weight updation.

5. Second pattern is presented and the above steps are followed for the second weight updation.

6. When all the training patterns are presented, a cycle of iteration or epoch is completed.
7. The errors of all the training patterns are calculated and displayed on the monitor as the mean squared error (MSE).

$$\mathbf{E(MSE) = \sum E(p)} \qquad \mathbf{(4.37)}$$

4.9.3 Effect of Number of Hidden Layers In BPN Convergence

The network is trained with two hidden layers. The total number of nodes used in both the hidden layers are 14. The different combinations of number of nodes in the first hidden layer and in the second hidden layer are given in Table 4.3The convergence rates of the network with two hidden layers is more when compared to that of convergence rates of the network with one hidden layer. When there is only one hidden layer and the number of hidden nodes is 10, less number of epochs is required for the network to reach lowest MSE. When there are two hidden layers with 6 nodes in the first hidden layer and 8 nodes in the second hidden layer, it requires more epochs for the network to reach required MSE. Since it requires more number of iterations for the network with more than one hidden layer, it is sufficient to have only one hidden layer.

The learning factor (η) is supposed to guide the convergence rates of the network to the desired MSE with less number of iterations. It so happens, that sometimes η will make the network to converge to the desired MSE after an increased number of iterations. For 10 nodes in the hidden layer, it requires less iterations for the network to reach MSE of 0.01 when η is 0.05.

The network is trained with threshold (θ) and without θ is used, updation of θ is done similar to weight updation. The parameter θ is used in all the layers except in the input layer. For 10 nodes in the hidden layer, it requires less iterations for the network to reach MSE of 0.01 without θ, and more iterations for the network to reach MSE of 0.01 with θ.

Table 4.3 presents number of epochs required to reach 0.01 with two hidden layers in the BPN.

No. of nodes in layer 1	No. of nodes in layer 2	No. of iterations to reach MSE of 0.01
4	10	102
5	9	68
6	8	85
7	7	74
8	6	85
9	5	95
10	4	124

Table 4.3 Number of nodes in each hidden layer

4.9.4 Fixing the Number of Nodes in the Hidden Layer

The number of nodes in the hidden layer of BPN is fixed based on the maximum number of emotion images recognized. Simulation of the number of nodes in the hidden layer is changed from 2 nodes to 20 nodes. More than 20 nodes are not simulated in the BPN training.

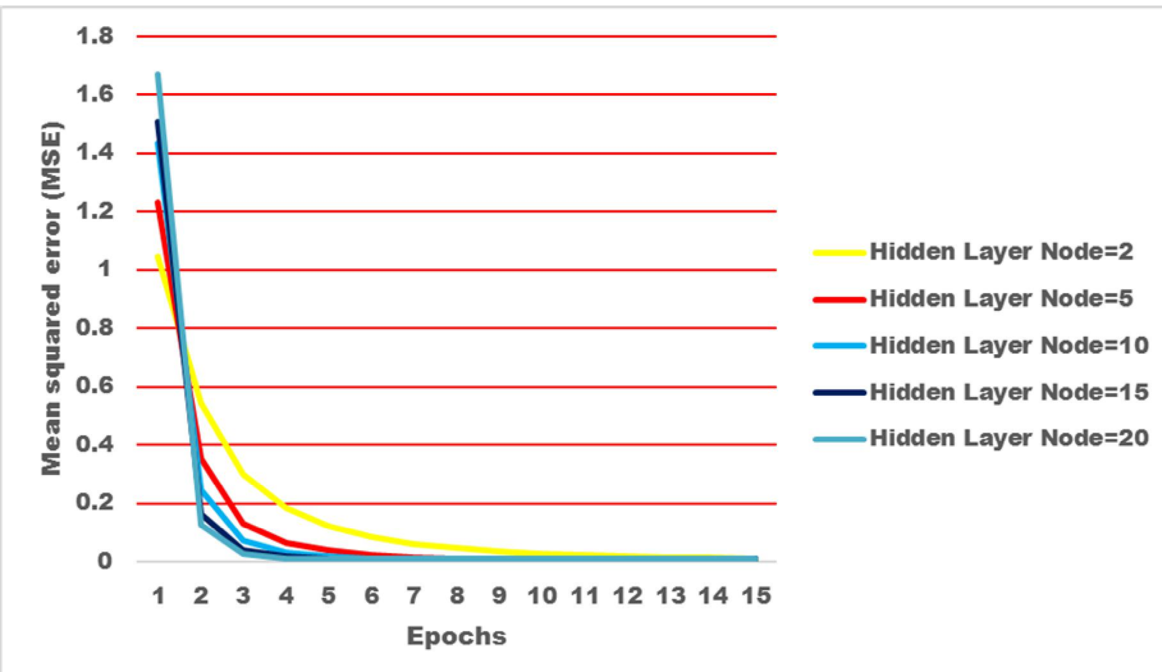


Fig.4.8a Epochs versus mean squared error

Figure 4.8a presents the convergence curve for BPN with different number of nodes used in the hidden layer for training the FLD feature obtained from the angry images. The hidden layer with 20 nodes results in more convergence of the error. Other convergence curve indicates less convergence and hence more epochs taken to reach the desired error value.

4.9.5 Emotion Identification performance

Figure 4.8b presents the number of iterations and the corresponding percentage of classification of the facial expressions for different number of nodes in the hidden layer of the BPN. When the numbers of nodes are between 10 to 20, maximum classification is achieved.

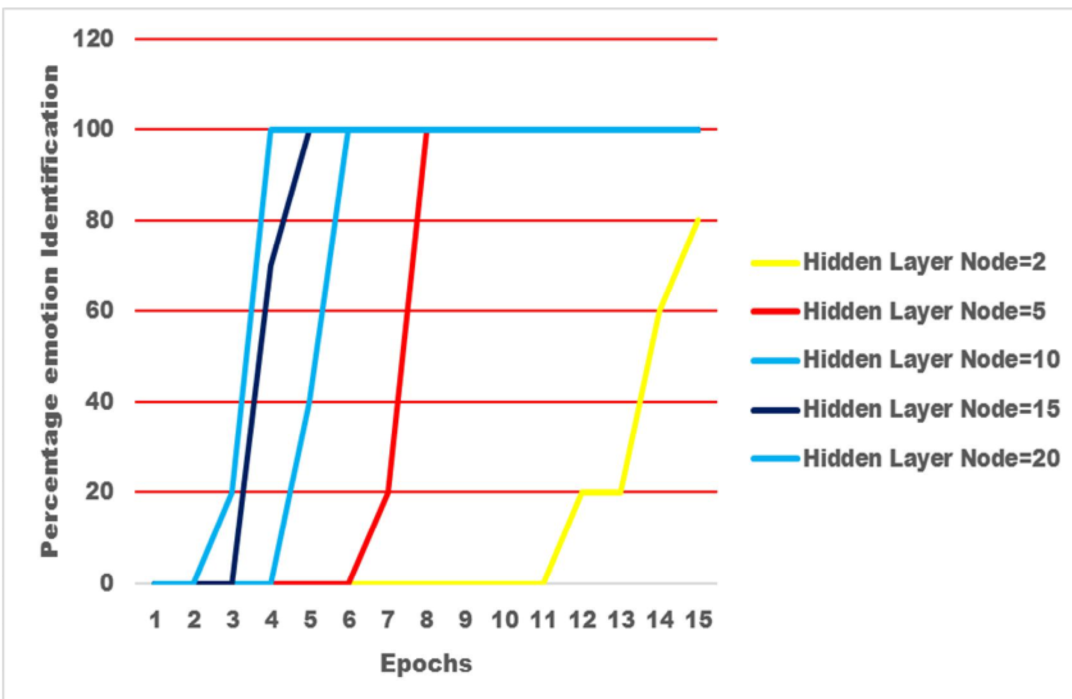


Fig.4.8b Epochs versus percentage of emotion identification

4.10 Cerebellar Model Articulation Controller (CMAC):

CMAC is an associative memory artificial neural network algorithm which maps inputs and corresponding outputs as any other network would do. However, the specialty of CMAC is that, the input pattern is further quantized in an overlapping fashion. Figure 4.9 shows the complete topology for implementing CMAC. In the testing process, input patterns are presented in the input layer of CMAC architecture. The values of the patterns are searched in the already quantized space and the corresponding space numbers of the different values of the input pattern is noted. The locations of the different spaces corresponding to the original input pattern are now treated as the input pattern for subsequent processing. Weights corresponding to the different space locations are summarized linearly to produce require outputs.

Steps involved in implementing CMAC:

Step 1: Present a pattern with two dimensional values obtained from FLD.

Step 2: quantize each value of the pattern into many ranges. For example, 0.1 is quantized into (0.01 -0.05), (>0.05 till 0.08), (>0.08 till <0.12). So, in this example, three quantization ranges are created. The value can be associated to range 3 (>0.08 till <0.12). Hence, 0.1 can be [0, 0, 1]. Similarly, if the second dimensional value obtained from FLD is 0.07, then 0.07 can be [1, 0, 0]. Hence, the two FLD value given in the input layer with two input nodes, has been converted into 6 valued patterns [0, 0, 1, 1, 0, 0]. This pattern is further used to propagate forward till the output layer of the CMAC topology. An error is calculated in the output layer which is used to propagate backwards for updating the connection weights.

Step 3: In the testing process, new pattern is presented and the quantization process is carried out. All the weight connections already updated during the training process are selectively chosen based on the presence of 1 in the quantized pattern to obtain an output in the output layer. The output is further compared with a template to identify the FEE.

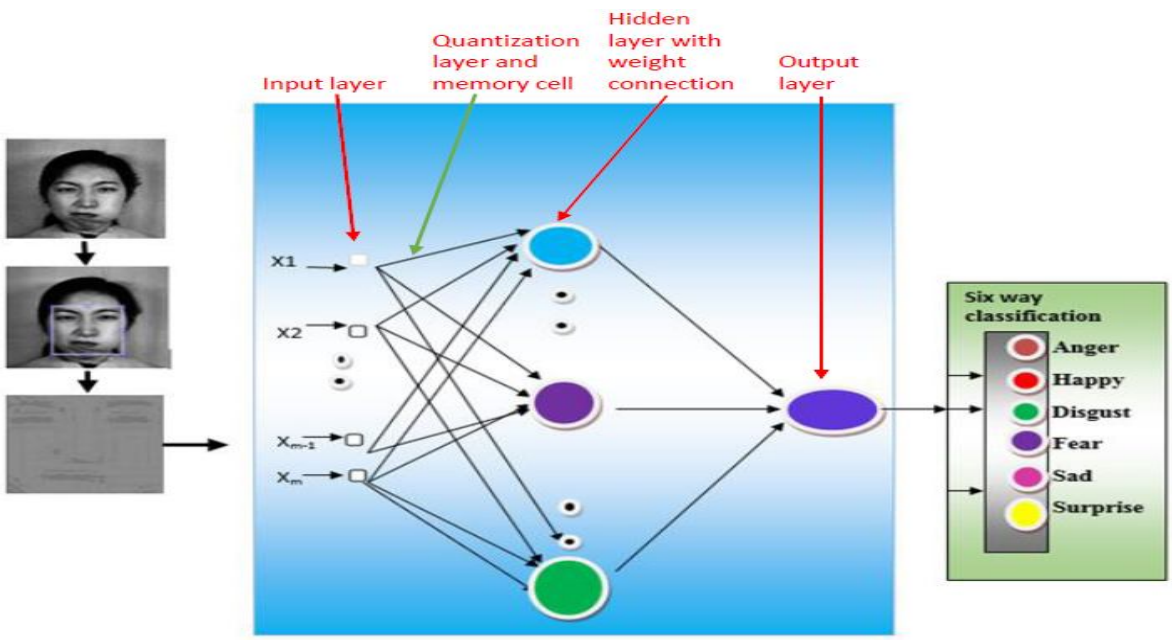


Fig.4.9 CMAC in training and testing of faces

4.11 Summary

This chapter has presented implementations of the proposed algorithms for facial expression recognition. Chapter 5 presents the results and discussion of the implemented algorithms.

CHAPTER FIVE

RESULTS

5.1 Introduction

This chapter presents the recognition performance of the proposed algorithms for identifying the type of emotions for 6 expressions: Anger, Happy, Fear, Sad, Surprise, and Disgust.

5.2 ROC for FLD/SLN/BPN/CMAC

ROC curve is a graphical plot that illustrates the performance of a system as its discrimination threshold is varied. ROC provides ample space for accommodating the performance of proposed algorithms. This space is upper diagonal of the Roc curve. It is created by plotting FPR vs. TPR. True positive rate (TPR) is the fraction of true positives out of the total actual positives.

$$Sensitivity=TPR=TP/(TP+FN)) \quad \textbf{(5.1)}$$

False positive rate (FPR) is the fraction of false positives out of the total actual negatives.

$$Specificity=FPR=FP/(FP+TN) \quad \textbf{(5.2)}$$

Where: FP—False Positive, FN—False Negative, TP—True Positive, and TN—True Negative.

FP—Frame does not contain any expression, but algorithm says expression is present.

FN—AN expression is actually present in a frame. But the algorithm says that there is no expression.

TP—True Positive, frame detected as containing expression. The expression category is correctly identified.

TN— True Negative, frame identified as not containing expression. The expression is not detected.

5.3. Accuracy of emotion expression identification

The accuracy refers to how correctly, the proposed algorithms identifies emotion expression in a frame. Different measures like precision and recall can be used to evaluate identification of emotion expression. However, in this work the emotion expression identification accuracy is expressed as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5.3}$$

Detection refers to presence of an expression.

Identification refers to classification of expression.

5.4. Angry Expression

FPR	TPR	Specificity	Sensitivity	Accuracy
0.052632	0.941176	94.73684	94.11765	94.44444
0.038462	0.96	96.15385	96	96.07843
0.055556	0.941176	94.44444	94.11765	94.28571
0.071429	0.9	92.85714	90	91.66667
0.071429	0.923077	92.85714	92.30769	92.59259
0.085106	0.916667	91.48936	91.66667	91.52542
0.071429	0.866667	92.85714	86.66667	89.65517
0.052632	0.909091	94.73684	90.90909	93.33333
0.111111	0.529412	88.88889	52.94118	65.38462
0.029412	0.9375	97.05882	93.75	96
0.018519	0.8	98.14815	80	96.61017
0.029412	0.941176	97.05882	94.11765	96.07843
0.08	0.833333	92	83.33333	90.32258
0.052632	0.888889	94.73684	88.88889	92.85714
0.090909	0.75	90.90909	75	86.66667
0.034483	0.916667	96.55172	91.66667	95.12195
0.071429	0.75	92.85714	75	88.88889
0.066667	0.875	93.33333	87.5	91.30435
0.029412	0.888889	97.05882	88.88889	95.34884
0.071429	0.5	92.85714	50	87.5
0.090909	0.8	90.90909	80	87.5
0	0.8	100	80	92.85714
0.1	0.8	90	80	86.66667
0.018519	0.954545	98.14815	95.45455	97.36842
0.026316	0.363636	97.36842	36.36364	83.67347
0.016667	0.923077	98.33333	92.30769	97.26027
0.1	0.833333	90	83.33333	87.5
0.1	0.885246	90	88.52459	88.73239
0.083333	0.5	91.66667	50	85.71429
0	0.8	100	80	86.66667
0.230769	0.9	76.92308	90	82.6087

Table 5.1 ROC for FLD

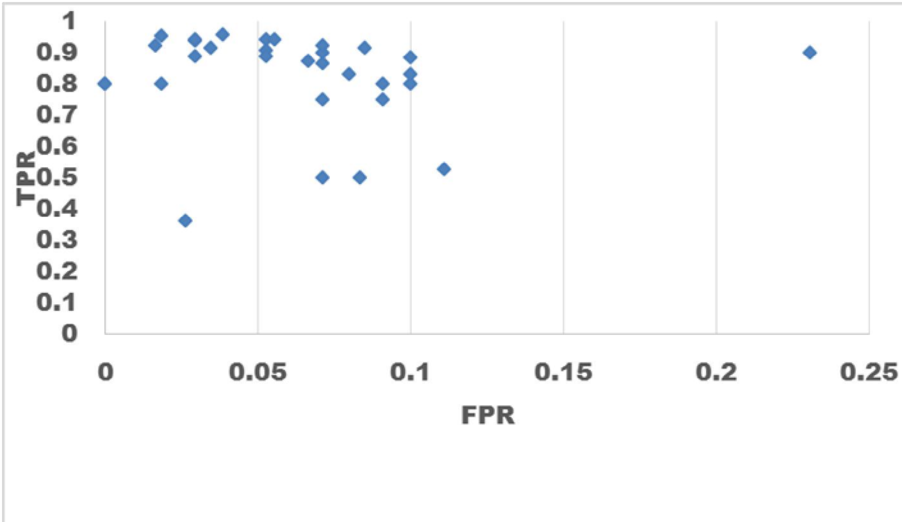


Fig.5.1 ROC for FLD

Figure 5.1 presents ROC for the performance of FLD in estimating "Angry expression". The ROC plot shows the points above the diagonal. The TPR is more and the FPR is less.

TPR	Specificity	Sensitivity	Accuracy
0.941176	93.75	94.11765	93.93939
0.96	95.65217	96	95.83333
0.941176	93.33333	94.11765	93.75
0.9	90.90909	90	90.47619
0.923077	90.90909	92.30769	91.66667
0.916667	97.56098	91.66667	96.22642
0.866667	90.90909	86.66667	88.46154
0.909091	93.75	90.90909	92.59259
0.9	83.33333	90	87.5
0.9375	96.77419	93.75	95.74468
0.8	98.03922	80	96.42857
0.941176	96.77419	94.11765	95.83333
0.833333	90.90909	83.33333	89.28571
0.888889	93.75	88.88889	92
0.75	87.5	75	83.33333
0.916667	96.15385	91.66667	94.73684

0.75	90.90909	75	86.66667
0.875	91.66667	87.5	90
0.888889	96.77419	88.88889	95
0.5	90.90909	50	84.61539
0.8	87.5	80	84.61539
0.8	100	80	90.90909
0.8	85.71429	80	83.33333
0.954545	98.03922	95.45455	97.26027
0.571429	97.14286	57.14286	90.47619
0.923077	98.24561	92.30769	97.14286
0.833333	85.71429	83.33333	84.61539
0.885246	85.71429	88.52459	88.23529
0.5	88.88889	50	81.81818
0.8	100	80	83.33333
0.9	70	90	80

Table 5.2 ROC for SLN

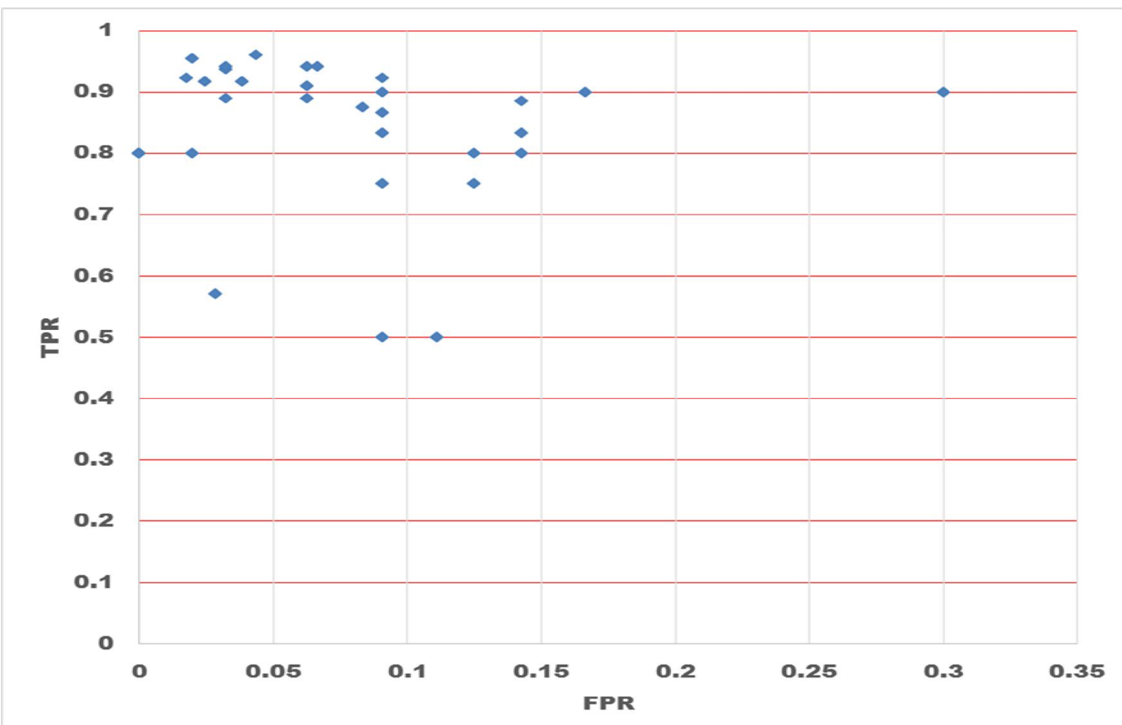


Fig.5.2 ROC for SLN

Figure 5.2 presents ROC for the performance of SLN in estimating "Angry expression". The ROC plot shows the points above the diagonal. The TPR is more and the FPR is less.

FPR	TPR	Specificity	Sensitivity	Accuracy
0.03846	0.94118	96.1538	94.1176	95.3488
0.0303	0.96	96.9697	96	96.5517
0.04	0.94118	96	94.1176	95.2381
0.04762	0.9	95.2381	90	93.5484
0.04762	0.92308	95.2381	92.3077	94.1176
0.01961	0.91667	98.0392	91.6667	96.8254
0.04762	0.86667	95.2381	86.6667	91.6667
0.03846	0.90909	96.1538	90.9091	94.5946
0.0625	0.9	93.75	90	92.3077
0.02439	0.9375	97.561	93.75	96.4912
0.01639	0.8	98.3607	80	96.9697
0.02439	0.94118	97.561	94.1176	96.5517
0.0625	0.83333	93.75	83.3333	92.1053
0.03846	0.88889	96.1538	88.8889	94.2857
0.05556	0.75	94.4444	75	90.9091
0.02778	0.91667	97.2222	91.6667	95.8333
0.04762	0.75	95.2381	75	92
0.04545	0.875	95.4545	87.5	93.3333
0.02439	0.88889	97.561	88.8889	96
0.04762	0.5	95.2381	50	91.3043
0.05556	0.8	94.4444	80	91.3043
0	0.8	100	80	95.2381
0.05882	0.8	94.1176	80	90.9091
0.01639	0.95455	98.3607	95.4545	97.5904
0.02222	0.57143	97.7778	57.1429	92.3077
0.01493	0.92308	98.5075	92.3077	97.5
0.05882	0.83333	94.1176	83.3333	91.3043
0.05882	0.88525	94.1176	88.5246	89.7436
0.05263	0.5	94.7368	50	90.4762
0	0.8	100	80	90.9091
0.15	0.9	85	90	86.6667

Table 5.3 ROC for BPN

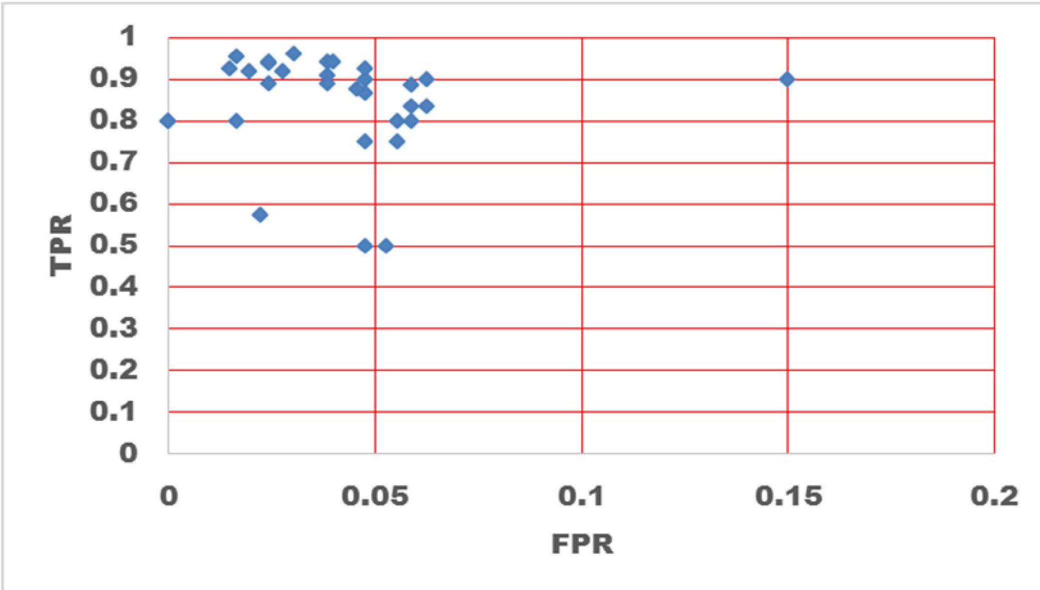


Fig.5.3 ROC for BPN

Figure 5.3 presents ROC for the performance of BPN in estimating "Angry expression". The ROC plot shows the points above the diagonal. The TPR is more and the FPR is less.

FPR	TPR	Specificity	Sensitivity	Accuracy
0.05263	0.94118	94.7368	94.1176	94.4444
0.03846	0.96	96.1538	96	96.0784
0.05556	0.94118	94.4444	94.1176	94.2857
0.07143	0.9	92.8571	90	91.6667
0.07143	0.92308	92.8571	92.3077	92.5926
0.08511	0.91667	91.4894	91.6667	91.5254
0.07143	0.86667	92.8571	86.6667	89.6552
0.05263	0.89091	94.7368	90.9091	93.3333
0.11111	0.52941	88.8889	52.9412	65.3846
0.02941	0.9375	97.0588	93.75	96
0.01852	0.8	98.1481	80	96.6102
0.02941	0.94118	97.0588	94.1176	96.0784
0.08	0.83333	92	83.3333	90.3226
0.05263	0.88889	94.7368	88.8889	92.8571
0.09091	0.75	90.9091	75	86.6667

0.03448	0.89817	96.5517	91.6667	95.122
0.07143	0.75	92.8571	75	88.8889
0.06667	0.875	93.3333	87.5	91.3043
0.02941	0.88889	97.0588	88.8889	95.3488
0.07143	0.5	92.8571	50	87.5
0.09091	0.8	90.9091	80	87.5
0	0.8	100	80	92.8571
0.1	0.8	90	80	86.6667
0.01852	0.89545	98.1481	95.4545	97.3684
0.02632	0.36364	97.3684	36.3636	83.6735
0.01667	0.92308	98.3333	92.3077	97.2603
0.1	0.83333	90	83.3333	87.5
0.1	0.88525	90	88.5246	88.7324
0.08333	0.5	91.6667	50	85.7143
0	0.8	100	80	86.6667
0.23077	0.9	76.9231	90	82.6087

Table 5.4 ROC for CMAC

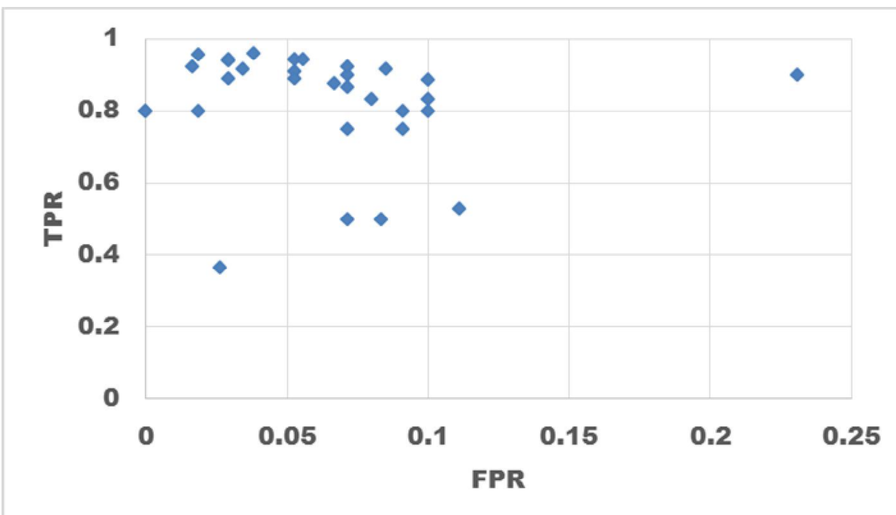


Fig.5.4 ROC for CMAC

Figure 5.4 presents ROC for the performance of CMAC in estimating "Angry expression". The ROC plot shows the points above the diagonal. The TPR is more and the FPR is less.

FLD	SLN	BPN	CMAC
91.89189	94.4444	95.3488	93.9394
94.23077	96.0784	96.5517	95.8333
91.66667	94.2857	95.2381	93.75
88	91.6667	93.5484	90.4762
89.28571	92.5926	94.1176	91.6667
90	91.5254	96.8254	96.2264
86.66667	89.6552	91.6667	88.4615
90.32258	93.3333	94.5946	92.5926
62.96296	65.3846	92.3077	87.5
94.11765	96	96.4912	95.7447
95	96.6102	96.9697	96.4286
94.23077	96.0784	96.5517	95.8333
87.5	90.3226	92.1053	89.2857
89.65517	92.8571	94.2857	92
81.25	86.6667	90.9091	83.3333
92.85714	95.122	95.8333	94.7368
84.21053	88.8889	92	86.6667
87.5	91.3043	93.3333	90
93.18182	95.3488	96	95
82.35294	87.5	91.3043	84.6154

Table 5.5 Accuracy for proposed algorithms

in identifying "Angry" expression

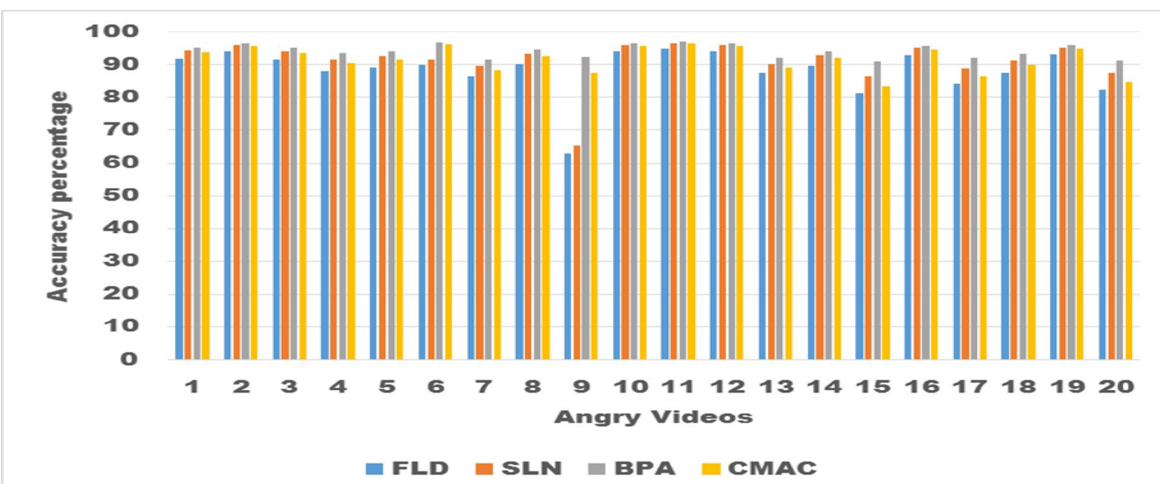


Fig.5.5 Accuracy for proposed algorithms in identifying "Angry"

expression

Figure 5.5 presents accuracy obtained for the proposed algorithms in identifying Angry expression. The accuracy of CMAC is superior when compared to that of FLD, SLN, and BPN accuracy.

FLD	SLN	BPN	CMAC
90	94.7368	96.1538	93.75
92.59259	96.1538	96.9697	95.6522
89.47368	94.4444	96	93.3333
86.66667	92.8571	95.2381	90.9091
86.66667	92.8571	95.2381	90.9091
89.58333	91.4894	98.0392	97.561
86.66667	92.8571	95.2381	90.9091
90	94.7368	96.1538	93.75
80	88.8889	93.75	83.3333
94.28571	97.0588	97.561	96.7742
96.36364	98.1481	98.3607	98.0392
94.28571	97.0588	97.561	96.7742
88.46154	92	93.75	90.9091
90	94.7368	96.1538	93.75
83.33333	90.9091	94.4444	87.5
93.33333	96.5517	97.2222	96.1538
86.66667	92.8571	95.2381	90.9091
87.5	93.3333	95.4545	91.6667
94.28571	97.0588	97.561	96.7742
86.66667	92.8571	95.2381	90.9091
83.33333	90.9091	94.4444	87.5
90	100	100	100
81.81818	90	94.1176	85.7143
96.36364	98.1481	98.3607	98.0392
94.8718	97.3684	97.7778	97.1429
96.72131	98.3333	98.5075	98.2456
81.81818	90	94.1176	85.7143
81.81818	90	94.1176	85.7143
84.61539	91.6667	94.7368	88.8889
83.33333	100	100	100
71.42857	76.9231	85	70

Table 5.6 Specificity for proposed algorithms in identifying "Angry" expression

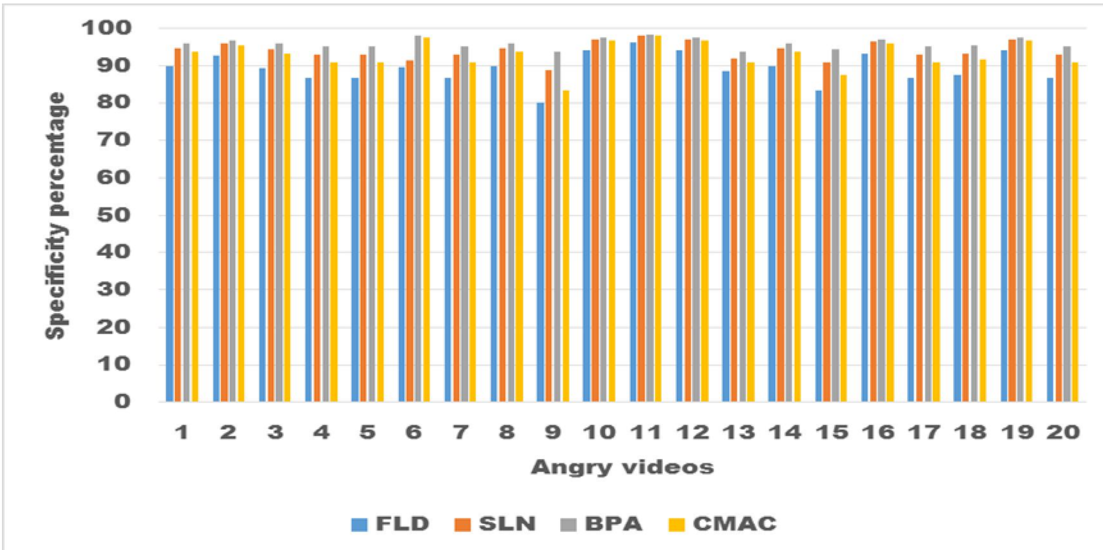


Fig.5.6 Specificity for proposed algorithms in identifying "Angry" expression

Figure 5.6 presents specificity obtained for the proposed algorithms in identifying Angry expression. The specificity of CMAC is superior when compared to that of FLD, SLN, and BPN specificity.

FLD	SLN	BPN	CMAC
94.11765	94.1176	94.1176	94.1176
96	96	96	96
94.11765	94.1176	94.1176	94.1176
90	90	90	90
92.30769	92.3077	92.3077	92.3077
91.66667	91.6667	91.6667	91.6667
86.66667	86.6667	86.6667	86.6667
90.90909	90.9091	90.9091	90.9091
52.94118	52.9412	90	90
93.75	93.75	93.75	93.75
80	80	80	80

94.11765	94.1176	94.1176	94.1176
83.33333	83.3333	83.3333	83.3333
88.88889	88.8889	88.8889	88.8889
75	75	75	75
91.66667	91.6667	91.6667	91.6667
75	75	75	75
87.5	87.5	87.5	87.5
88.88889	88.8889	88.8889	88.8889
50	50	50	50
80	80	80	80
80	80	80	80
80	80	80	80
95.45455	95.4545	95.4545	95.4545
36.36364	36.3636	57.1429	57.1429
92.30769	92.3077	92.3077	92.3077
83.33333	83.3333	83.3333	83.3333
88.52459	88.5246	88.5246	88.5246
50	50	50	50
80	80	80	80
90	90	90	90

Table 5.7 Sensitivity for proposed algorithms

in identifying "Angry" expression

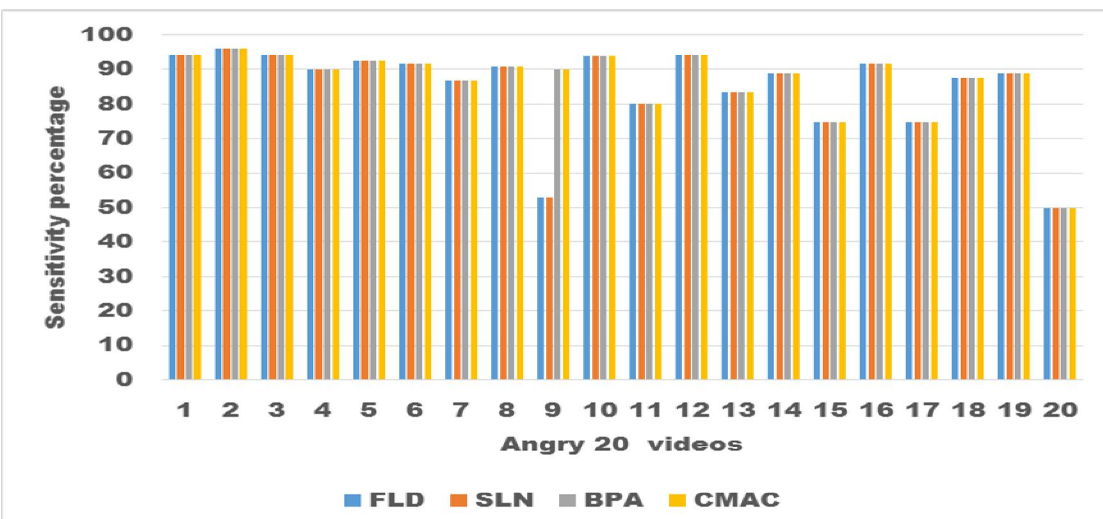


Fig.5.7 Sensitivity for proposed algorithms in identifying "Angry" expression

Figure 5.7 presents sensitivity obtained for the proposed algorithms in identifying Angry expression. The sensitivity of CMAC is superior when compared to that of FLD, SLN, and BPN sensitivity.

5.5. Accuracy of proposed algorithms in identifying the 6 expressions.

Figure 5.8 presents accuracies obtained for the proposed algorithms in identifying Angry expression. The accuracy of CMAC is superior when compared to that of FLD, SLN, and BPN accuracy.

Facial Expression	SLN	FLD	BPN	CMAC
Anger	94.1	93.8	95.1	95.1
Happy	96	96	96.2	96.3
Fear	94	93.8	95	97
Sad	92.8	94	94.3	96
Surprise	92	91.8	94	95
Disgust	94	96.1	96.4	98

Table 5.8 Comparison of accuracy for FLD/SLN/BPN/CMAC in identifying six expressions

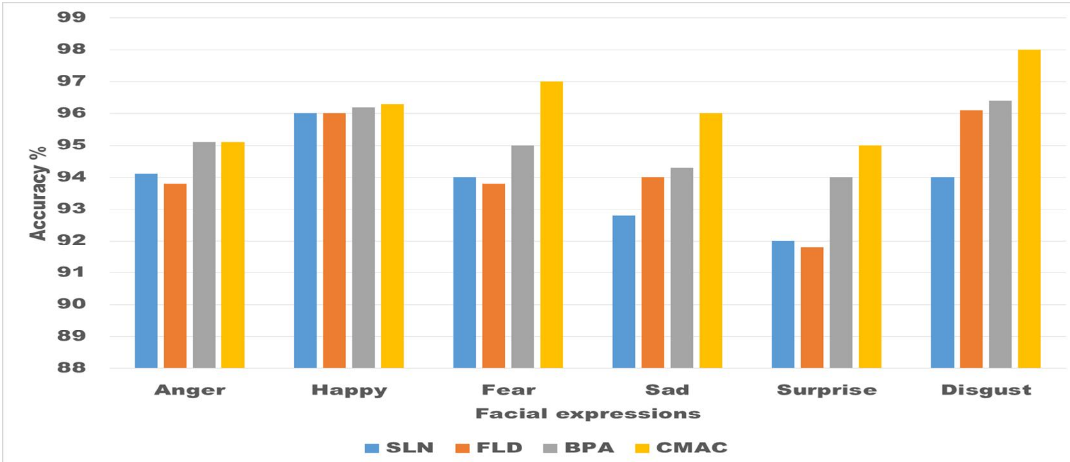


Fig.5.8 Comparison of accuracy for FLD/SLN/BPN/CMAC in identifying six expressions

5.6. Summary

This chapter has presented the expression identification performance of FLD/ SLN/ BPN/ CMAC ANN algorithms. The ROC, accuracy, sensitivity and specificity indices are presented for comparisons.

CHAPTER SIX

DISCUSSIONS

6.1. Introduction

This chapter explains how the results support the answers of questions which our thesis answered. Also gives small comparisons between our thesis and other researches done in the same area which mentioned in chapter two.

6.2. Discussions:

In this thesis, a new approach to facial expression recognition with Gabor features and three neural networks is presented. The methods use Gabor filters for extracting the feature vectors combined with FLD, also FLD was used as a classifier.

PCA can significantly reduce the dimensionality of the original feature without loss of much information in the sense of representation, but it may lose important information for discrimination between different classes. So for increase performance of this work FLD was used.

Using FLD in facial expression recognition in our thesis showed that this approach has limitations over the variations in light, size and in the head orientation. Nevertheless, this method showed very good classifications of faces. A good recognition system should have the ability to adapt over time. Reasoning about images in face space provides a means to learn and subsequently recognize new faces in an unsupervised manner, so as shown in figure 5.5 using neural networks is better than FLD.

Below are some comparison between the proposed methods and some other related methods. Extensive experimental results verify the effectiveness of our approach outperforms most of the approaches.

The proposed methods compare with Krishna Kishore, 2011, describes a novel Hybrid Emotional Neural Network (HENN) for classification of emotions from Facial expressions. Features are extracted from facial expressions by applying Gabor wavelet and Discrete Cosine Transform (DCT), then using PCA the method was tested on static images from the Cohn-Kanade database. Classifications varied across different databases from 75% to100%. While the proposed methods using the same methodology the classification rate is better for 94% to 100%, this comes from the usage of Directional Gabor wavelet because the directional roughness features results in high quality facial segmentation performance.

Nirmala Priya, Multiclass SVM is applied systematically for classification. The Japanese female database JAFFE is used for the experiment. Extensive experiments shows that statistical features derived from LBP are effective and efficient for facial expression recognition. The recognition rates for Fear and Sadness are much lower (79–88%). The proposed methods classification in the same expressions using BPN were 95% for fear and 94.3% for sadness while for CMAC were 97% for fear and 96% for sadness this because of strength of the usage of neural networks.

Amir Jamshidnezhad, using genetic algorithms (GAs), images from Cohn-Kanade database were used to obtain the best functions parameters, the rate of 98% in the train process. The proposed methods classification rate using CMAC is 98% which is the same.

Ayesha Butalia, Facial expressions using fuzzy logic the accuracy of recognition was 71.7%. The accuracy rate was between 91.8% to 98% for all proposed methods (FLD/SLN/BPN/CMAC), this also because of strength of the usage of neural networks.

Yu-Yi Liao Facial Expression Recognition using CMAC Network with Clustering Memory the accuracy rate was 93.33% while in our proposed methods using CMAC was 95.1% this because of the usage of Rotational Gabor wavelet combined with FLD.

Renu, presents a novel approach for the detection of emotions using the cascading of Mutation Bacteria Foraging optimization and Adaptive Median Filter in highly corrupted noisy environment

Feng, 2005, Experimental results demonstrate an average recognition accuracy of 93.8% in the JAFFE database. The accuracy rate was between 91.8% to 98% for all proposed methods (FLD/SLN/BPN/CMAC), this also because of strength of the usage of neural networks.

Table 6.1 shows some comparison for the accuracy rate of some selected facial expression recognition methods, which uses JAFFE database.

Method	**Accuracy**	**Proposed Methods**
Facial Expression Recognition using CMAC Network with Clustering Memory	93.33%	95.1%
Learning From Examples in the Small Sample Case: Face Expression Recognition	92.2%	93.2
Recognition of facial expressions using Gabor wavelets and learning vector quantization	87.51%	74%(recognition rate)

Table 6.1 comparison for the accuracy rate of some selected facial expression recognition methods

CHAPTER SEVEN

CONCLUSIONS AND RCOMMENDATIONS

7.1 Conclusion

This thesis has proposed and implemented three ANN algorithms to improve emotion expression identification accuracy. Six expressions Anger, Happy, Fear, Sad, Surprise and Disgust have been collected using JAFFE database.

Three features were extracted using wavelet Gabor filter for segmentation of the image. The image is further processed with Fisher's Linear Discriminant function to obtain transformed two dimensional features. These features are better to discriminate facial expression of 6 different categories considered in this research work. The Fisher's Linear Discriminant features are input to ANN algorithms such as FLD/SLN/BPN/CMAC. The ANN algorithms are learning algorithms which can estimate emotion expression better than the conventional methods features.

1. The combined CMAC provides highest emotion expression identification when compared to emotion expression identification accuracy of FLD/SLN/BPN.

2. The emotion expression identification accuracy is purely based on the type of features and the transformed features obtained from FLD are better.
3. The emotion expression identification accuracy of BPN and CMAC are almost same.
4. The emotion expression identification accuracy of SLN and FLD are not comparable with the performance of CMAC and BPN as expected

7.2 Recommendations

The thesis work has given promising hopes for the future researchers to consider the other types of CMAC for implementing in facial emotion expression identification. Also, to use Fisher's Linear Discriminant in the other types of neural networks, and the proposed methods will be tuned towards applicability in real-time. Also for increase performance in future works additional methods such as combination of PCA and FLD can be applied.

REFERENCES

[1] Yang M. H., Kriegman D. J., and Ahuja N., 2002, Detecting Faces in Images: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, pp.34-58.

[2] Matthew J. Hertenstein and Dacher Keltner, 2006, Touch Communicates Distinct Emotions, The American Psychological Association, Emotion, Vol.6, No.3, pp.528-533.

[3] Elaine Fox, Victoria Lester, Riccardo Russo, Bowles R.J., Alessio Pichler, Kevin Dutton, 2007, Facial Expressions of Emotion: Are Angry Faces Detected More Efficiently?, Cognition and Emotion, PMC, Vol.14, No.1, pp.61-92.

[4] Levine D.S., 2007, Neural network modeling of emotion, Phys. Life Rev., Vol.4, pp.37-63.

[5] Abigail A. Marsh and Megan N. Kozak and Nalini Ambady, 2007, Accurate Identification of Fear Facial Expressions Predicts Prosocial Behavior, National Institutes of Health(NIH), Emotion, Vol.7, No.2, pp.239-251.

[6] Manuel G. Calvo and Lauri Nummenmaa, 2008, Detection of Emotional Faces: Salient Physical Features mide Effective Visual Search, Journal of experimental psychology: General, Vol.137, No.3, pp.471-494.

[7] Hillel Aviezer, Ruicong Zhi, Ran R. Hassin, Jennifer Ryan, Cheryl Grady, Josh Susskind, Adam Anderson, Morris Moscovitch, Shlomo Bentin, 2008, Angry, Disgusted, or Afraid? Studies on the Malleability of Emotion Perception Association for Psychological Science, Vol.19, No.7.

[8] Hatice Gunes and Maja Pantic, 2010, Automatic, dimensional and Continuous Emotion recognition, International Journal of Synthetic Emotions, Vol.1, No.1, pp.68-99.

[9] Ayesha Butalia, Ramani A.K., and Parag Kulkarni, 2010, Emotional Recognition and towards Context based decision, International Journal of Computer Applications, Vol.9, No.3, pp.42-54.

[10] Ravi S., and Mahima S., 2011, Study of the Changing Trends in Facial Expression Recognition, International Journal of Computer Applications (0975-8887), Vol.21. No.5, pp.10-16.

[11] Seyedarabi H., and Aghagolzadeh A., Khanmohammadi S., and Kabir E., 2007, Journal of Iranian Association of Electrical and Electronics Engineers, Vol.4, No.1, pp.11-19.

[12] Yongmian Zhang, Qiaang ji, Zhiwei, Beifang Yi, 2008, Dynamic Facial Expression Analysis and Synthesis with Mpeg-4 Facial Animation Parameters, IEEE Transactions on Circuits and Systems For Video Technology, Vol.18, No.10, pp.1-15.

[13] Renu Nagpal, Pooja Nagpal and Sumeet Kaur, 2010, Hybrid Technique for Human Face Emotion Detection, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol.1, No.6, pp.91-101.

[14] Li P., Phung S.L., Bouzerdoum A., and Tivive F.H.C, 2010, Automatic Recognition of Smiling and Neutral Facial Expressions, Digital Image Computing: Techniques and Applications, pp.581-586.

[15] Khandait S.P., Thool R.C., and Khandait P.D., 2011, Automatic Facial Feature Extraction and Expression Recognition based on Neural Network, International Journal of Advanced Computer Science and Applications,Vol.2, No.1, pp.112-118.

[16] Krishna Kishore K.V., and Varma G.P.S., 2011, Efficient Facial Emotion Classification With Wavelet Fusion of Multi Features, International Journal of Computer Science and Network Security, Vol.11, No.8, pp.160-165.

[17] Mahesh Kumbhar, Ashish Jadhav and Manasi Patil, 2012, Facial Expression Recognition Based on Image Feature, International Journal of Computer and Communication Engineering, Vol.1, No.2, pp.117-119.

[18] Littlewort G., Fasel I., Stewart M., Bartlett and Movellan J. R., 2002, Fully automatic coding of basic expressions from video, Technical Report, UCSD INC MP Lab.

[19] Taskeed Jabid, Md. Hasanul Kabir and Oksam Chae, 2010, Robust Facial Expression Recognition Based on Local Directional Pattern, ETRI Journal, Vol.32, No.5, pp.784-794.

[20] Nirmala Priya G., and Wahida Banu R.S.D., 2012, Person Independent Facial Expression Detection using MBWM and Multiclass SVM, International Journal of Computer Applications, Vol.55, No.17, pp.52-58.

[21] Suvashis Das and Koichi Yamada, 2012, An HMM based Model for Prediction of Emotional Composition of a Facial Expression using both Significant and Insignificant Action Units and Associated Gender Differences, International Journal of Computer Applications, Vol.45, No.11, pp.11-18.

[22] Amir Jamshidnezhad and Md Jan Nordin 2012, An adaptive learning model based genetic for facial expression recognition, International Journal of the Physical Sciences, Vol.7, No.4, pp.619-623.

[23] Pantic M., and Rothkrantz L.J.M., 2000, Expert system for Automatic Analysis of facial Expressions, ELSEVIER, Image and Vision Computing, Vol.18, pp.881-905.

[24] Fasela B., and Juergen Luettinb., 2003, Automatic facial expression analysis: a survey, ELSIER, Pattern Recognition, Vol.36, pp.259–275.

[25] Ayesha Butalia, Maya Ingle and Parag Kulkarni, 2012, Best Emotional Feature Selection Criteria Based On Rough Set Theory, International Journal of Computer Application and Engineering Technology, Vol.1, No.2,.pp.49-71.

[26] Mayur S. Burange and Dhopte S.V., 2012, Neuro Fuzzy Model for Human Face Expression Recognition, International Journal of Engineering and Advanced Technology (IJEAT), Vol.1, No.5, pp.35-38.

[27] Ayarpadi K., Kannan E., Resmi R. Nair, Anitha T., and Srinivasan R., 2012, Face Recognition under Expressions and Lighting Variations using Masking and Synthesizing, International Journal of Engineering Research and Applications, Vol.2, No.1, pp.758-763.

[28] Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic and Lijun Yin, 2012, Static and dynamic 3D facial expression recognition: A comprehensive survey, ELSEVIER, Image and Vision Computing, Vol.30, pp.683–697.

[29] Murthy G.R.S., and Jadon R.S., 2009, Effectiveness of Eigen spaces for facial expression recognition, International Journal of Computer Theory and Engineering, Vol.1, No.5, pp.638-642.

[30] Mandeep Kaur and Rajeev Vashisht and Nirvair Neeru, 2010, Recognition of Facial Expressions with principal Component Analysis and

Singular Value decomposition, International Journal of Computer Applications (0975-8887), Vol.9, No.12, pp.36-40.

[31] Mandeep Kaur and Rajeev Vashisht, 2011, Comparative Study of Facial Expression Recognition Techniques, International Journal of Computer Applications (0975 – 8887), Vol.13, No.1, pp.43-50.

[32] Ayesha Butalia, Maya Ingle and Parag Kulkarni, 2012, Facial Expression Recognition for Security, International Journal of Modern Engineering Research, Vol.2, No.4, pp.1449-1453.

[33] Feng X., Pietikäinen M., and Hadid A., 2005, Facial Expression Recognition with Local Binary Patterns and Linear Programming, Pattern Recognition and Image Analysis, Vol.15, No.2, pp.546-548.

[34] Rachael E. Jack, Caroline Blais, Christoph Scheepers, Philippe G. Schyns and Roberto Caldara, 2009, Cultural Confusions Show that Facial Expressions Are Not Universal, Current Biology, Elsevier, Vol.19, No.18, pp.1543-1548.

[35] McCullah W.S., and Pitts W., 1943, A logical calculus of the ideas imminent in nervous activity, Bulletin of Mathematical Biophysics, Vol.5, pp.115-133.

[36] Rosenblatt F., 1961, Principles of neuro dynamics: Perceptron and the theory of brain mechanisms, Sparton Books, Washington D.C.

[37] Widrow B., and Hoff M.E., 1960, Adaptive switching circuits, IRE WESSCON Convention Record, New York: IRE, pp.96-104.

[38] Werbos P.J., 1974, Beyond regression: new tools for prediction and analysis in the behavioural sciences, Ph.D. Thesis, Department of Applied Mathematics, Hardvard University, Cambridge, Massachusetts.

[39] Rumelhart D.E., Hinton G.E., and William R.J., 1996, Learning internal representations by error propagation, in Rumelhart D.E and McCleland J.L., Parallel distribution processing: Explorations in the microstructure of cognitions, Vol.I, Foundations, M.I.T., Press, Cambridge, M.A.

[40] Lippmann R.P., 1987, An introduction to computing with neural nets, IEE Transactions On Acoustics, Speech and Signal Processing Magazine, Vol.35, No.4, pp.4-22.

[41] Sietsma J., and Dow R.J.F., 1991, Creating artificial neural networks that generalize Neural Networks, Vol.4, No.1, pp.67-79.

[42] Chester D., 1990, Why two hidden layers are better than one, IEE International Joint Conference on Neural Networks, IJCNN'90, pp.265-268.

[43] Hirose Y., Yamshita K., and Hijiya S, 1991, Back-propagation algorithm which varies the number of hidden units, Neural Networks, Vol.4, No.1, pp.61-66.

[44] Weymaere N., and Martens J.P., 1991, A fast and robust learning algorithms for feed forward neural networks, Neural Networks, Vol.4, Vol.4, No.3, pp.361-369.

[45] Fujita O., 1992, Optimization of the hidden unit function in feed forward neural networks, Neural Networks, Vol.5, No.5, pp.755-764.

[46] Ridgway W.C., 1962, An adaptive logic system with generalizing properties, Ph.D. thesis, Tech.rep.1556-1, Stanford Electron Labs., Stanford, CA.

[47] Zaki M., 1964, General error criteria, IEEE Transactions on Information Theory, Vol.IT-10, No.2, pp.94-95.

[48] Walach E., and Widrow B., 1984, The least mean fourth (LMF) adaptive algorithm and its family, IEEE Transactions on Information Theory, Vol.IT-30, No.2, pp.275-283.

[49] Shoemaker P.A., Carlin M.J., Shimbukuro R.L., 1991, Back-propagation learning with trinary quantization of weight updates, Neural Networks, Vol.4, No.2, pp.231-241.

[50] Bershad N.J., Shynk J.J., and Feintuch P.L., 1993, Statistical analysis of the single-layer-back-propagation algorithm: part-I-Mean weight behaviour, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.41, No.2, pp.573-582.

[51] Specht D.E., 1990, Probabilistic neural networks and the polynomial adaline as complimentary techniques for classification, IEEE Transactions on Neural Networks, Vol.1, No.1, pp.111-121.

[52] Albus J.S, 1975, A New Approach to Manipulator Control: the Cerebellar Model Articulation Controller (CMAC), Trans. ASME, Series G. Journal of Dynamic Systems, Measurement and Control, Vol.97, pp.220-233.

[53] Albus J.S., 1979, Mechanisms of Planning and Problem Solving in the Brain, Mathematical Biosciences, Vol.45, pp.247-293.

[54] Horváth, G. Kernel, 2006, CMAC: An Efficient Neural Network for Classification and Regression, Acta Polytechnica Hungarica, Vol.3, Issue 1, pp.

[55] Gabor D., 1946, Theory of communication. Journal of Institute of Electrical Engineers, Vol.93, pp.429-457.

[56] Ekman P., 1999, Basic emotions, In T. Dalgleish and T. Power, editors, The Handbook of Cognition and Emotion. John Wiley & Sons, Ltd.

[57] Guodong Guo and Charles R. Dyer, Learning From Examples in the Small Sample Case: Face Expression Recognition, IEEE 2007

[58] Choudhury R., Schmid C., and Mikolajczyk K., 2003, Face detection and tracking in a video by propagating detection probabilities, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, No.10, pp.1215-1228.

[59] Boccignone G., Caggiano V., Fiore G. D., and Marcelli A., 2005, Probabilistic detection and tracking of faces in video, In Proceedings of International Conference on Image Analysis and Processing, pp.687-694.

[60] Tong Y., YangWang, Zhiwei Zhu, Qiang Ji, 2007, Robust facial feature tracking under varying face pose and facial expression, Pattern Recognition, Vol.40, pp.3195–3208.

[61] Li Y., Ai H., Huang C., and Lao S., 2006, Robust head tracking with particles based on multiple cues fusion, ECCV Workshop on HCI, pp.29-39.

[62] Dornaika F., and Ahlberg J., 2004, Fast and reliable active appearance model search for 3-D face tracking, IEEE Transactions on Systems, Man and Cybernetics, Vol.34, No.4, pp.1838-1853.

[63] Dornaika F., and Davoine F., 2006, On appearance based face and facial action tracking, IEEE Transactions on Circuits and Systems for Video Technology, Vol.16, No.9, pp.1107-1124.

[64] Arnaud E., Fauvet B., Mémin É., and Bouthemy P., 2005, A robust and automatic face tracker dedicated to broadcast videos, In Proceeding Of International Conference on Image Processing, pp.429-432.

[65] Hager G.D., Belhumeur P., 1998, Efficient region tracking with parametric models of geometry and illumination, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.20, pp.1025-1039.

[66] Koterba S., Baker S., Matthews I., Hu C., Xiao H., Cohn J., Kanade T., 2005, Multi-view in fitting and camera calibration. IEEE International Conference on Computer Vision., pp.511-518.

[67] Xu Y., Roy-Chowdhury A., 2007, Integrating motion, illumination and structure in video sequences with applications in illumination-invariant tracking, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp.793-806.

[68] Terzopoulos D., and Waters K., 1993, Analysis and synthesis of facial image sequences using physical and anatomical models IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.15, pp.569–579.

[69] Black M.J. and Yacoob Y, 1995, Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion, In Proceedings of International Conference on Computer Vision, pp.374-381.

[70] Blanz V., Vetter T., 2003, Face recognition based on fitting a 3D morphable model, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.25, pp.1063-1074.

[71] Arulampalam M., Maskell A., Gordon N., Clapp T., 2002, A Tutorial on Particle Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking, IEEE Transactions on Signal Processing, Vol.50, Issue 2, pp.174-188.

[72] Zhao W., Chellappa R., Phillips P., Rosenfeld A., 2003, Face Recognition: A Literature Survey. ACM Transactions, pp.399-458.

[73] Dedeoglu G., Baker S., Kanade T., 2006, Resolution-aware fitting of active appearance models to low-resolution images, European Conference on Computer Vision, pp.115-134.

[74] Yu J., Bhanu B., Xu Y., Roy-Chowdhury A., 2007, Super-resolved facial texture under changing pose and illumination. International Conference on Image Processing, pp.553-556.

[75] Viola P., and Jones M., 2001, Robust real-time object detection, Technical Report 2001/01, Compaq Cambridge Research Lab.

[76] Essa I.A., and. Pentland A.P., 1997, Coding, analysis, interpretation and recognition of facial expressions, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.19, No.7, pp.757- 763.

[77] Turk M., and Pentland A., 1991, Eigenfaces for recognition. Journal of Cognitive Neuroscience, Vol.3, No.1, pp.71-86.

[78] Steffens J., Elagin E., and Neven H., 1998, Person spotter fast and robust system for human detection, Tracking, and recognition, In Proceedings International Conference on Automatic Face and Gesture Recognition, pp.516-521.

[79] Lades M., Vorbruggen J.C., Buhmann J., Lange J., vonder Malsburg C., Wurtz R.P., and Konen W., 1993, Distortion invariant object recognition in the dynamic link architecture, IEEE Transactions on Computers, Vol.42, pp.300-311.

[80] Cohn J. F., Zlochower A.J., Lien J.J., and Kanade T., 1998, Feature-point tracking by optical flow discriminates subtle differences in facial expressions, In Proceedings of International Conference on Automatic Face and Gesture Recognition, pp.396-401.

[81] Ekman P., and Friesen., 1978, W.V. Facial Action Coding System: Investigator's Guide, Consulting Psychologists Press.

[82] Tian Y.L., Kanade T., and Cohn. J.F, 2001, Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.23, No.2, pp.97-115.

[83] Dailey M. N., Cottrell G.W., and Adolphs R., 2000, A six-unit network is all you need to discover happiness, In Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society, Mahwah, NJ, USA, Erlbaum.

[84] Fisher R.A., 1936, The use of multiple measurement in taxonomic problems, Annals of Eugenics, Vol.7, pp.178-188.

[85] Hong Z.Q., and Yang J.Y., 1991, Optimal discriminant plane for a small number of samples and design method of classifier on the plane, Pattern Recognition, Vol.24, No.4, pp.317-324.

[86] Foley D.H., 1972, Consideration of sample and feature size, IEEE Transactions on Information Theory, Vol.IT-18, No.5, pp.681-626.

[87] Gallinari P., Thiria S., Badran F., and Fogelman-Soulie F., 1991, On the relations between discriminant analysis and multiplayer perceptron, Neural Networks, Vol.4, No.3, pp.349-360.

[88] Siedlecki W., Siedlecka K., and Skalansky J., 1988, Experiments on mapping techniques for exploratory pattern analysis, Pattern Recognition, Vol.21, No.5, pp.431-438.

[89] Kittler J and Young P.C, 1973, Approach to feature selection based on the Karhunen-Loeve expansion, Pattern Recognition, Vol.5, No.5, pp.335-352.

[90] Sammon J.W., 1970, An optimal discriminant plane, IEEE Transactions on Computers, Vol.C-19, No.9, pp.826-829.

[91] Fehlauer J., and Eisentein B.A., 1978, A declustering criterion for feature extraction in pattern recognition, IEEE Transactions on Computers, Vol.C-27, No.3, pp.261-266.

[92] Gelsema E., and Eden R., 1980, Mapping algorithms in ISPAHAN, Pattern Recognition, Vol.12, No.1, pp.127-136.

[93] Mix D.F., and Jones R.A., 1982, A dimensionality reduction technique based on a least squared error criterion, IEEE Transactions Pattern Analysis Machine Intelligence, PAMI-4, pp.537-544.

[94] Friedman J.H., and Turkey J.W., 1974, A projection pursuit algorithm for exploratory data analysis, IEEE Transactions On Computer, Vol.C-23, No.9, pp.881-890.

[95] Foley D.H., and Sammon Jr., J.W., 1975, An optimal set of discriminant vectors, IEEE Transactions On Computers, Vol.C-24, No.3, pp281-289.

[96] Liu K., Cheng Y.Q., and Yang J.Y., 1992, A generalized optimal set of discriminant vectors, Pattern Recognition, Vol.25, No.7, pp.731-739.

[97] Cheng Y.Q., Zhuang Y.M., and Yang J.Y., 1992, Optimal fisher discriminant analysis using the rank decomposition, Pattern Recognition, Vol.25, No.1, pp.101-111.

[98] Klema V.C., and Laub A.J., 1980, The singular value decomposition; its computation and some applications, IEEE Transactions on Automatic Control, Vol.AC-25, No.2, pp.164-176.

[99] Sullivan B.J., and Liu B., 1984, On the use of singular value decomposition and decimation in discrete-time band-limited signal extrapolation, IEEE Transactions On Acoustics, Speech and Signal Processing, Vol.ASSP-32, No.6, pp.1201-1212.

Appendices

List Publications