INDIAN INSTITUTE OF TECHNOLOGY, DELHI

FACIAL EXPRESSION RECOGNITION

A THESIS SUBMITTED

FOR THE PARTIAL FULFILLMENT OF BACHELORS DEGREE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IIT DELHI

By-

Gaurav Ahuja – 2011CS10216

Rakshit Gautam – 2011CS10245

Under supervision of Prof. K.K. Biswas

CERTIFICATE

This is to certify that the thesis titled “Facial Expression Recognition” submitted by Gaurav Ahuja (Entry Number- 2011CS10216) and Rakshit Gautam (Entry Number- 2011CS10245) to the Indian Institute of Technology, Delhi for the award of the degrees of Bachelor of Technology in Computer Science and Engineering is a bona fide record of work carried out by them under my guidance and supervision at Department of Computer Science and Engineering at Indian Institute of Technology, Delhi.

Prof. K. K. Biswas

(Supervisor)

Department of Computer Science and Engineering

Indian Institute of Technology, Delhi

New Delhi – 110016

Date -

ACKNOWLEDGMENT

We are really grateful to IIT Delhi for giving us the opportunity to study and work with the prestigious department of Computer Science and Engineering.

We would like to thank Prof. K.K. Biswas for guiding us through the project, helping us not only on the technical aspects but also sorting out the non-technical aspects. His enthusiasm, guidance and encouragement have been invaluable to this work. We are really fortunate to have had the opportunity to work with him at IIT Delhi.

Also, we would like to thank Prof. Prem Kalra, Prof. Subhashis Banerjee and Dr Subodh Kumar for reviewing this work as members of evaluation committee and raising questions that improved our thought process throughout the project.

We are also grateful to Bharat Jangid sir for useful discussions and also Parul mam for supporting us.

We would like to thank Prof. Jeffery Cohn for the use of Cohn Kanade dataset and Dr Frank Wallhoff for the use of FEED dataset.

Gaurav Ahuja **Rakshit Gautam**

2011CS10216 **2011CS10245**

ABSTRACT

Facial Expressions carry lots of information about social behaviors, mental states of individuals which can be used in a big way for bridging the gap between humans and computers. Our aim of this work is to create a real time illumination invariant expression recognition system. In the process, we have analyzed the appearance-based and geometric features that capture the information of facial expression.

We compute features from the face images and use machine learning techniques for classification. Initial work has been done on still images from the FEED dataset. Later, we have moved onto temporal features and a more standard Cohn-Kanade dataset.

Table of Contents

Chapter 1

1. Introduction and literature survey

1.1 Introduction

Facial expression can be defined as temporally deformed facial features generated by contraction of facial muscles. In our daily life, facial expressions explain a lot regarding human behavior and emotions. Human most expressively explain their feelings through facial expressions. Expressions provide the best possible signals to understand human emotions and hence play a vital role in human interactions.

Humans have the ability to express and even understand the feelings through facial expressions. Thus, human can interact and make sense of communications by proper consideration of the emotions involved. Computers on the other hand, lack the ability to make any sense of the emotional aspect of communication.

Researchers have shown the presence of Universal Facial Expressions representing happiness, sadness, anger, fear, disgust and surprise. Our work mainly focusses on expressions observed in common scenarios which are happiness, sadness, anger and surprise.

Certain factors that make facial expression recognition, a challenging problem are illumination variations, skin tone variations among subjects, head motion (in plane and out of plane), complex movements of facial components and difference in expressions across subjects. Also, certain expressions like anger and sadness may have similar appearances for some subjects. Some of the expressions may appear to be neutral faces.

1.2 Literature Survey

The topic of Facial Expression recognition has been widely researched upon. In the late 1900's Ekman and Friesen found the existence of 'Universal Facial Expressions' representing happiness, sadness, fear, anger, disgust and surprise [21,22]. Ekman and Friesen developed Facial Expression Coding System which linked facial expressions to muscle movements. Using this system, facial expressions could be coded by decomposition into Action Units and temporal segments. But, this process was very time consuming [23].

Authors of [13] used Naïve Bayes classifier and Gaussian Tree-Augmented Naive Bayes (TAN) classifiers to learn the dependencies among different facial motion features and used Hidden Markov Models to segment live videos and find expressions. Authors of [26] use optical flow to track expressions. Major techniques used for classification include Template Matching with Chi Square statistic [24], SVM [7, 8, 24], Linear Discriminant

Analysis [8], Linear Programming [8], Naïve Bayes [13], Neural Networks [29], Deep Neural Networks [25], K Nearest Neighbor techniques.

The features used are either geometric features or appearance based features. Geometric features capture information about the movement of certain reference points, relative distance between specific points on the face, shapes of facial components such as lips, eyes, mouth etc. In appearance based features the texture changes are considered. Appearance based features include Local Binary Patterns [7, 8, 24], Gabor Wavelets [24], Local Directional Patterns. Active Shape models [30], Gaussian Mixture models involve use of geometric features.

Authors of [28] show that use of Gabor wavelets during spatial texture analysis yields better results. But Gabor wavelet analysis is time and memory consuming. Local Binary Pattern features perform stably and robustly over a useful range of low resolution images. Local Binary Pattern features are more illumination invariant as compared to Gabor features. Authors of [8] learn the most discriminative Local Binary Pattern features (called as Boosted-LBP) using Adaboost technique and use them to obtain highest accuracy. Authors of [29] used edge detection techniques to generate features for the Neural Network.

Authors of [30] used Active Shape model for computing facial expression change in video sequence. ASM localizes the feature points in first frame and tracks them in later frames. One can feed the displacements in facial features into a SVM classifier [31]. For dynamic texture analysis, researchers have used Volume Local Binary Patterns, Rotational Invariant Local Binary Patterns, Local Binary Patterns in Three Orthogonal Planes [7], Local Gabor Binary Patterns in Three Orthogonal Planes [28].

Chapter 2

2 Facial Expression Recognition

For recognizing facial expression in an image, firstly, the face present in the image is detected. Features are extracted from the detected face. These features are used by machine learning algorithm to predict the facial expression.

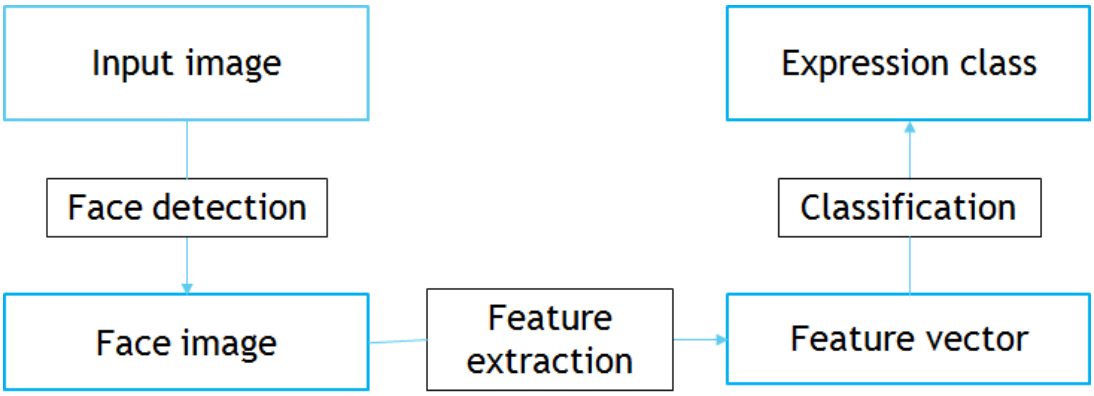

Figure 1 Facial Expression Recognition method

The steps in facial expression recognition are discussed below:

2.1) Face detection
2.2) Feature extraction
2.3) Classification

2.1 Face Detection

For a real time facial expression system, all the steps which are involved in expression recognition should be fast. Face detection is the first step in facial expression recognition. Viola Jones face detector is the first real time frontal face detector [1] [2].

In Viola Jones face detection, a small sub-window is chosen. HAAR features are computed in the sub-window. Using these features, the sub-window is classified as face or non-face using a cascade classifier. This is repeated for all sub-windows in the image to find the face [1] [2].

In a 24X24 sub-window, 162,336 HAAR features are present. It is computationally expensive to compute all the features on each sub-window. Adaboost [3] algorithm is

used for selection of subset of features which are able to discriminate between faces and non-faces.

Cascade classifier [1] is a multi-stage classifier. At each stage of classification, a classifier predicts whether the current sub-window is face or non-face. If the sub-window is a face, it is passed to the next stage of classifier. If the sub-window is a non-face, there is no need to pass the sub-window to further stages of classifier. Cascade classifier is computationally cheap because it rejects non faces early in the classification and hence not a lot of computation is done for non-faces. Viola Jones face detector has very low false positive rate [1] [2].

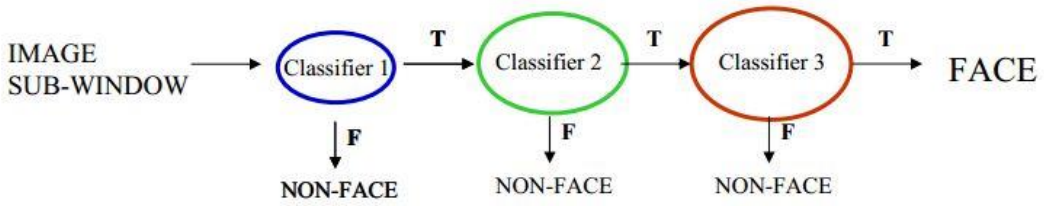

Figure 2 Cascade Classifier

We have used Viola Jones face detector [1] [2] for detecting face in an image. The input image is converted to grayscale and then Viola Jones face detector is applied to the grayscale image. Viola Jones face detector always detects a square face in the image. The detected square face is cropped and resized to a resolution of 150X150. From this resized face, 17 columns from left and 16 columns from right are truncated. The final resolution of the face is 150X117.

Figure 3 Face Detection and Normalization

2.2 Feature Extraction

After detecting the face, discriminative and informative features have to be extracted from the face so that machine learning algorithms can predict the facial expressions. Features should minimize the within class variations and maximize the between class variations.

For a real time facial expression recognition system, feature extraction should be a fast step. Gabor features [4] [5] [6] produce good results for expression recognition, but are computationally expensive. Local Binary Pattern (LBP) features [9] are fast to compute and also produce good results for expression recognition.

2.2.1 LBP Features

Local Binary Pattern: For computing the LBP of a pixel, the pixels which are in the neighborhood of current pixel are compared with respect to the current pixel. The neighbors whose intensity is less than the current pixel are labelled '0' and the rest are labelled '1'. Then the neighbors are traversed in a circular manner to generate a binary vector. The decimal value of this binary vector is known as LBP value of the current pixel.

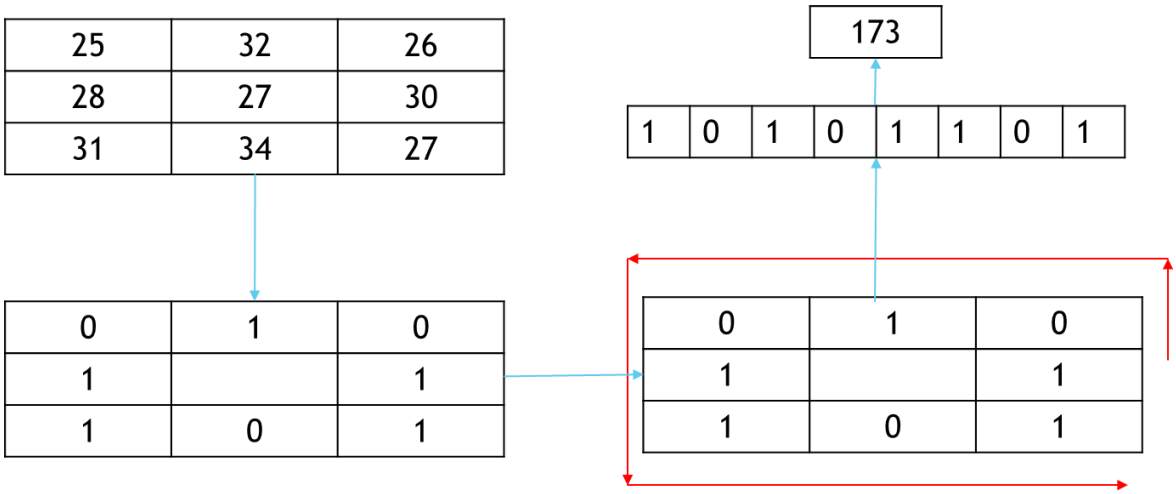

Figure 4 Local Binary Pattern of a pixel

When 'n' neighbors a pixel are used for generating LBP, length of binary vector obtained is 2^n. LBP value ranges from $[0,\ 2^{n-1}]$. When $n = 8$, LBP value ranges from $[0{,}255]$.

Local Binary Pattern (LBP) Histogram: In an image, LBP value of every pixel is computed and a frequency histogram of LBP values is plotted. This is known as LBP histogram.

When 8 neighbors are used for calculating LBP value, LBP histogram is a two dimensional vector of length 256.

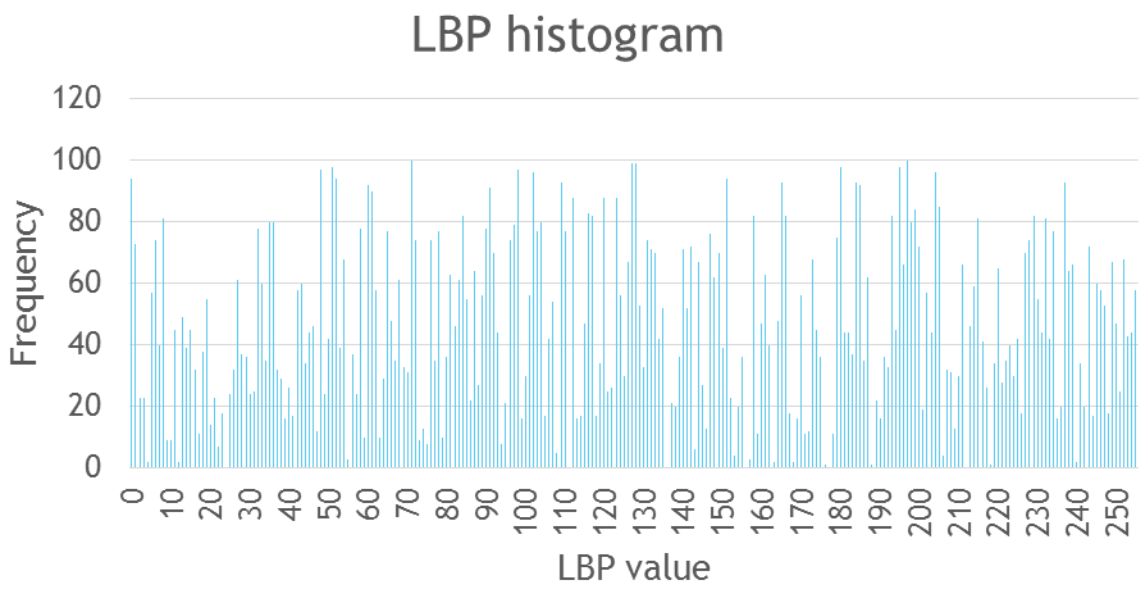

Figure 5 An example of LBP histogram

After detecting the face form an image using the face detection step, LBP features are extracted from the face image in the following steps:

- Divide the face image with resolution RXC into windows of size $wrXwc$

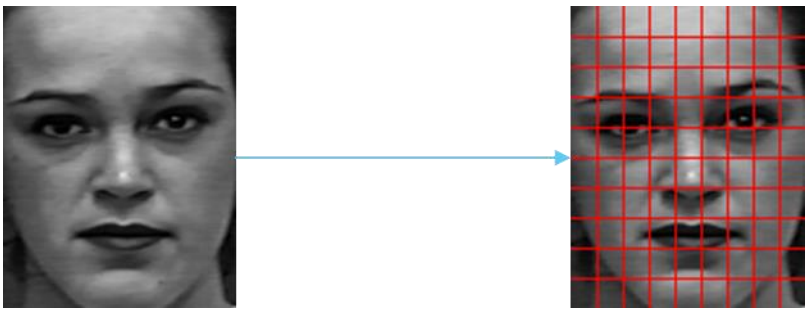

Figure 6 Face divided into windows

- Compute LBP histogram for each window
- Concatenate the histograms of each window to form a feature vector

Length of LBP feature vector:

- The face image is divided into $wrXwc$ windows
- Length of LBP histogram of each window $= 256$
- Length of feature vector $=$ Number of windows $*$ Length of feature vector of each window $= wr * wc * 256$

LBP features were used on FEED dataset and results obtained are discussed in section 4.1.

2.2.2 LDP Features

Local Directional Pattern (LDP): For computing the LDP value of a pixel, Kirsch masks in 8 directions are applied to the pixel. Replace top 'k' values and replace them with '1'. Replace all the other '$8-k$' values with '0'. The decimal value of the obtained binary vector is known as the LDP value of the pixel.

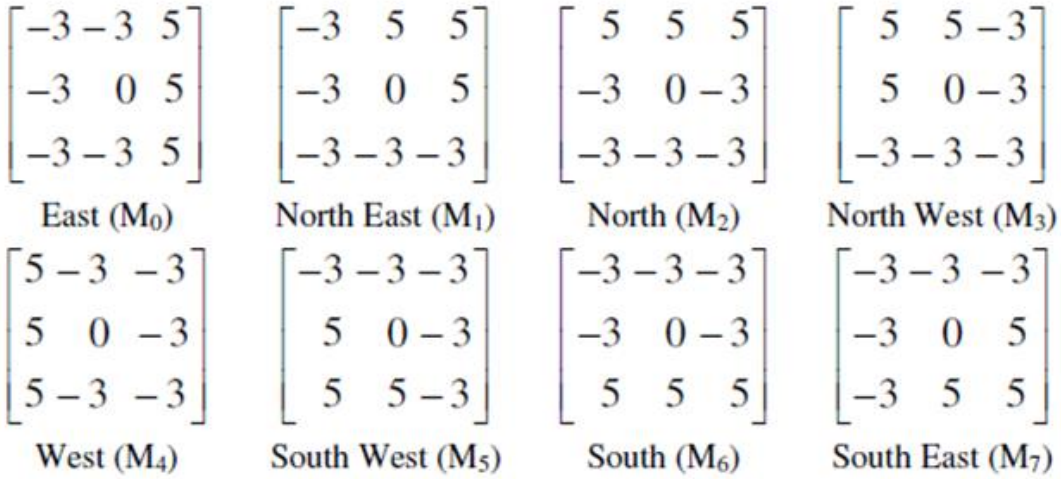

Figure 7 Kirsch edge response vectors in 8 directions. source - [32]

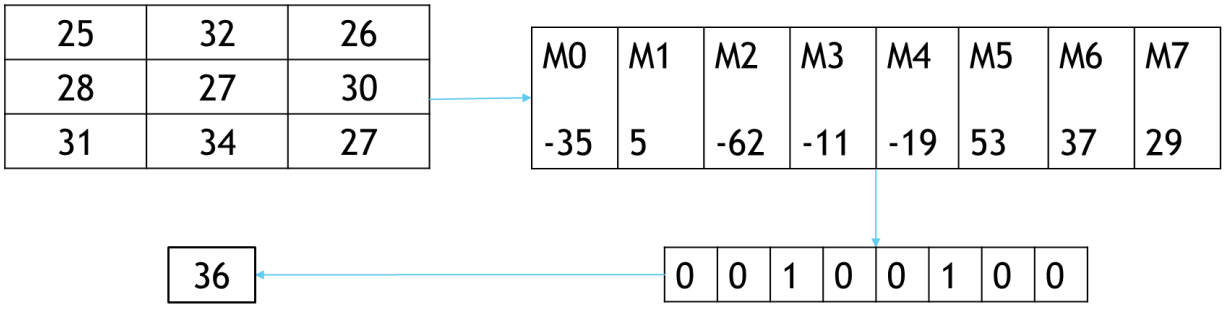

Figure 8 Local Directional Pattern for a pixel

Local Directional Pattern (LDP) Histogram: In an image, LDP value of every pixel is computed and a frequency histogram of LDP values is plotted. This is known as LDP histogram.

Similar to LBP features, LDP features are extracted from an image by dividing the image into sub-windows, computing LDP histograms for each sub-window and concatenating the LDP histograms for each window to form a feature vector.

After detecting the face form an image using the face detection step, LDP features are extracted from the face image in the following steps:

- Divide the face image with resolution RXC into $wrXwc$ windows

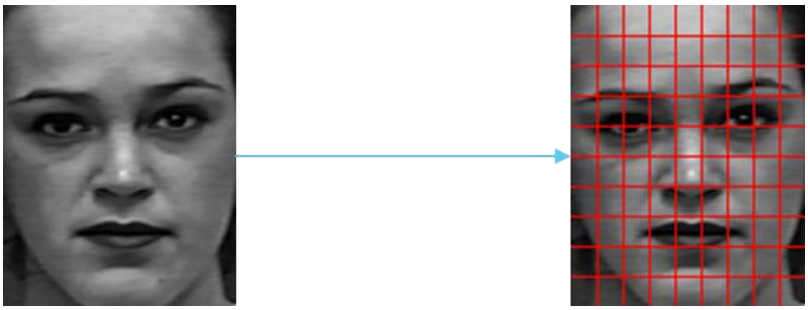

Figure 9 Face divided into windows

- Compute LDP histogram for each window
- Concatenate the histograms of each window to form a feature vector

Length of LDP feature vector:

- The face image is divided into $wrXwc$ windows
- Length of LDP histogram of each window $= 256$
- Length of feature vector $=$ Number of windows $*$ Length of feature vector of each window $= wr * wc * 256$

2.3 Classification

The features extracted from the face image are used by machine learning algorithms to predict the facial expression. SVM, Neural Networks, Adaboost, Naïve Bayes, Template based matching and Tree based methods have been tried for the expression recognition problem [8] [9] [10] [11] [12] [13] [14]. SVM is a linear classifier and is one of the most commonly used classifier in machine learning.

Since the feature vectors are very high dimensional in our problem, PCA is used for dimensionality reduction.

We have used SVM, SVM with Gaussian kernel, nearest neighbor and Naïve Bayes classifier in our experiments.

Chapter 3

3. Facial Expression Datasets

Facial expression recognition is a machine learning problem. Training and test data are required for this problem. To compare different algorithms for facial expression recognition problem, standard and publically available datasets are a must.

There are many expression datasets available like Cohn-Kanade AU-Coded Facial Expression (CK) database [17], CK+ database [18], FG-net Facial Expressions and Emotion (FEED) database [15], Japanese Female Facial Expression (JAFFE) database [16] etc. We have used CK and FEED database in our experiments. The details of these datasets are described below.

3.1 FG-net Facial Expressions and Emotion (FEED) database

This dataset consists of videos of 19 students between the age group of 18 to 30 years. Each student preforms 6 basic expressions, namely Anger, Disgust, Fear, Happiness, Sad and Surprise. Each expression is preformed 3 times. Each video starts from a neutral pose, then the person performs an expression and finally returns to a neutral pose. The faces in the videos are nearly frontal.

Figure 10 FEED Database sample

3.2 Cohn-Kanade AU-Coded Facial Expression (CK) database

This dataset consists of 486 image sequences from 97 posers. Each image sequence consists of 9 to 60 images. The first image in an image sequence consists of neutral pose and the last image of the image sequence is the peak of an expression. Most of the images have frontal or near frontal pose.

Figure 11 Cohn-Kanade Database sample

Chapter 4

4. Facial Expression Recognition on still images

We have tried some features, classifiers and preprocessing steps recognizing facial expressions in still images. We have performed the following experiments:

- ➔ LBP features on FEED dataset and CK dataset
- ➔ LDP features on CK dataset
- ➔ Weber Normalization before computing LBP features

The experiments with their results are discussed in detail in the following subsections.

4.1 Results on FEED dataset

FEED dataset consists of videos from 19 people who perform 6 basic expressions and each expression is performed 3 times. We manually handpicked images from these videos to create an expression dataset. The expressions of Anger, Happiness, Neutral and Surprise were selected in the dataset.

Images from 15 people were used for training SVM classifier and the images from 4 people were used for testing.

The details of the dataset are tabulated below:

Table 1 FEED Dataset details

Expression	Total Images	Training Images	Testing Images
Anger	396	356	40
Surprise	620	568	52
Happiness	2399	2218	181
Neutral	1740	1616	124
Total	5155	4758	397

Three experiments were carried out on FEED dataset.

Experiment 1:

- Face images of resolution 150X117 were divided into 10X9 windows of size 15X13

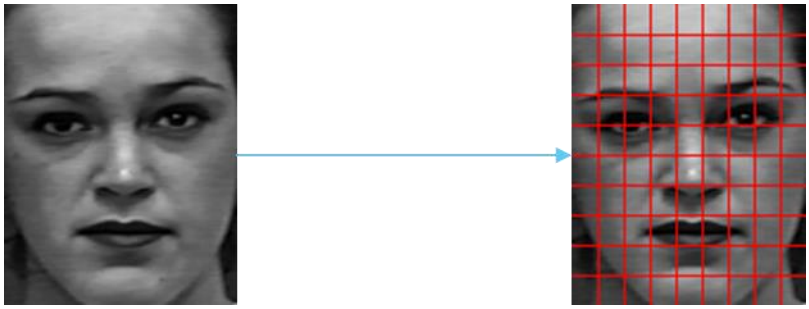

Figure 12 Face divided into 10X9 = 90 windows

- LBP histograms were computed for each window
- LBP histograms from each window were concatenated to form a feature vector
- Size of feature vector $= 9 * 10 * 256 = 23040$
- One vs. all SVM was used for classification

Results:

Accuracy using one vs. all SVM – 81.61%

Table 2 Results on FEED database using one vs. all SVM

Predicted → Actual |	Anger	Happiness	Neutral	Surprise
Anger	6 (15%)	0	5 (12.5%)	29 (72.5%)
Happiness	0	167 (92.26%)	7 (3.86%)	7 (3.86%)
Neutral	0	16 (12.9%)	102 (82.25%)	6 (4.8%)
Surprise	0	0	3 (5.7%)	49 (94.23%)

Experiment 2:

In the previous experiment, number of features is very large. To reduce the number of features, we decided to extract features from the region of face which has the most information about facial expression, eyes and mouth.

- Face images of resolution 150X117 were divided into 10X9 windows of size 15X13
- LBP histograms were computed for windows containing eyes and mouth

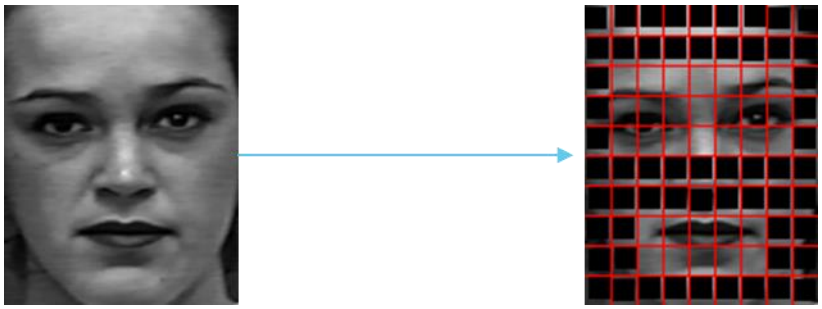

Figure 13 Figure showing the windows used for extracting features

- Features from 31 windows (not blackened windows) were used
- Size of feature vector $= 31 * 256 = 7936$
- Number of features have nearly reduced to $1/3$ of previous approach
- One vs. one SVM was used for classification

Results:

From this experiment onwards, LibSVM [19] was used for training SVM. In the previous experiment, the inbuilt SVM in matlab [20] was used.

Accuracy using one vs. one SVM [Linear Kernel] – 86.39%

Accuracy using one vs. one SVM [Gaussian Kernel] – 73.80%

Table 3 Results on FEED dataset using one vs. one SVM (Linear Kernel)

Predicted → Actual |	**Anger**	**Happiness**	**Neutral**	**Surprise**
Anger	34 (85%)	3 (7.5%)	1 (2.5%)	2 (5%)
Happiness	0	165 (91.16%)	16 (8.83%)	0
Neutral	8 (6.45%)	10 (8.06%)	106 (85.48%)	0
Surprise	1 (1.92%)	1 (1.92%)	12 (23.07%)	38 (73.07%)

Experiment 3:

To further reduce the number of features, we decided to use Principal Component Analysis on our data.

- PCA was performed on the features obtained in the previous experiment
- One vs. one SVM was used for training

Results:

Maximum accuracy of 87.91% was observed when projections on first 200 principal components were used as features.

Accuracy does not change when projections on more than 2000 principal components are used as features.

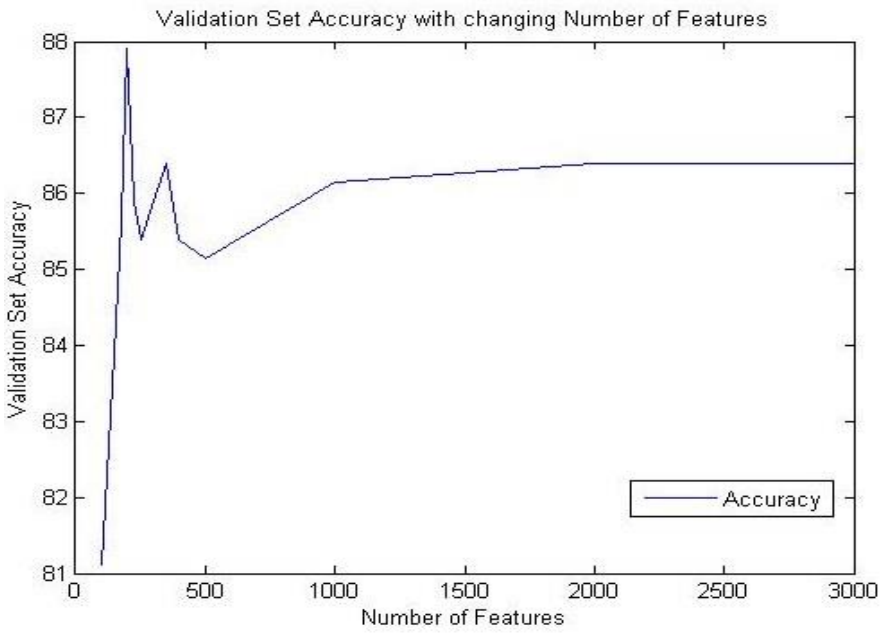

Figure 14 Test set accuracy with varying number of principal components used for training

Results Summary:

Table 4 Summary of results on FEED dataset

MODEL	ACCURACY
All windows + One vs. All SVM (Linear)	81.61%
Less Windows + One vs. One SVM (Linear)	86.40%
Less Windows + One vs. One SVM (Gaussian)	73.80%
Less Windows + PCA (200) + One vs. One SVM (Linear)	87.91%
Less Windows + PCA (2000) + One vs. One SVM (Linear)	86.40%

4.2 Comparison of LBP and LDP features

We performed three experiments on Cohn-Kanade dataset. In these experiments, size of sub-window in which the face image is divided is varied and the accuracy of LBP and LDP features are compared. LDP is considered as more stable, introduction of slight noise in an image causes less changes in LDP pattern as compared to LBP. This is because, LDP is affected by the change in gradient along the most prominent directions.

In each of the experiments,

- Face image was divided into windows
- LBP histograms were computed for each window
- LBP histograms from all the windows were concatenated to form a feature vector
- SVM classifier with linear kernel was used for training the model
- Similarly, LDP features were extracted from face images, SVM was trained and results were obtained

The experiments are listed below:

Experiment 1:

- Face image size of 150X117
- Window size of 21X18
- Features from the whole face image are taken
- Total windows $= 6 * 7 = 42$ windows
- Length of feature vector $= 256 * 42 = 10752$

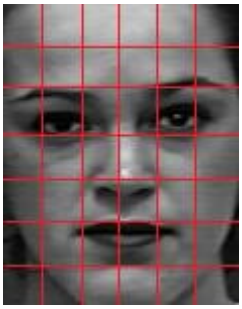

Figure 15 Windows in experiment 1

Experiment 2:

- Face image size of 150X117
- Window size of 25X22
- Features from eyes and mouth region are taken
- Total windows $= (4 * 2) + 3 = 11$ windows
- Length of feature vector $= 256 * 11 = 2818$
- Window size is large in this experiment

Figure 16 Windows in experiment 2

Experiment 3:

- Face image size of 150X117
- Window size of 13X15
- Features from eyes and mouth region are taken
- Total windows $= (7 * 3) + (5 * 2) = 11$ windows
- Length of feature vector $= 256 * 31 = 7936$
- Window size is small in this experiment

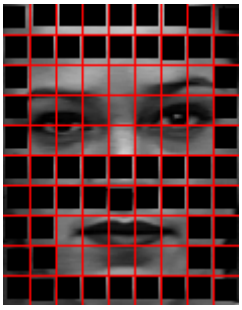

Figure 17 Windows in experiment 3

Dataset details:

The experiments were performed on Cohn-Kanade dataset. It has total 486 image sequences from 97 posers. From the image sequences of Anger, Happiness and Surprise expressions, a dataset was created. In an image sequence, the first image is a neutral image and the last image contains the peak of an expression. For creating a still image dataset, first frame from every image sequence was taken and placed in "Neutral" class. The last three images of the image sequence were placed in expression class i.e. if the image sequence was labelled Anger, the last three images of the image sequence were placed in "Anger" class.

From the total 97 posers, images from 75 posers were used for training and the images from 22 posers were used for testing.

Table 5 Dataset details

Expression →	Anger	Happiness	Neutral	Surprise
Training Images	163	204	220	194
Test Images	48	60	65	51
Total	211	264	285	245

Results:

Experiment 1:

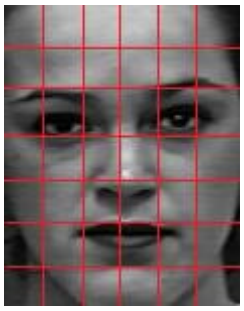

Figure 18 Windows in experiment 1

Accuracy using LBP features – 70.54%

Accuracy using LDP features – 46.87%

Experiment 2:

Figure 19 Windows in experiment 2

Accuracy using LBP features – 67.41%

Table 6 Results using LBP in experiment 2

Predicted → Actual |	Anger	Happiness	Neutral	Surprise
Anger	25 (52.08%)	0	17 (35.41%)	6 (12.5%)
Happiness	2 (3.33%)	54 (90.0%)	4 (6.67%)	0
Neutral	15 (23.08%)	0	42 (64.61%)	8 (12.31%)
Surprise	15 (29.41%)	0	6 (11.76%)	30 (46.15%)

Accuracy using LDP features – 38.39%

Table 7 Results using LDP in experiment 2

Predicted → Actual |	Anger	Happiness	Neutral	Surprise
Anger	20 (41.67%)	7 (14.58%)	15 (31.25%)	6 (12.5%)
Happiness	20 (33.3%)	14 (23.3%)	22 (36.67%)	4 (6.67%)
Neutral	16 (24.62%)	9 (13.84%)	34 (52.31%)	6 (9.23%)
Surprise	16 (31.37%)	3 (5.88%)	14 (27.45%)	18 (35.29%)

Experiment 3:

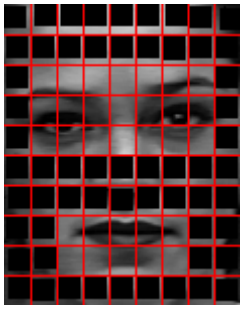

Figure 20 Windows in experiment 3

Accuracy using LBP features – 70.98%

Table 8 Results using LBP in experiment 3

Predicted → Actual |	Anger	Happiness	Neutral	Surprise
Anger	31(64.58%)	0	13 (27.08%)	4 (8.33%)
Happiness	5 (8.33%)	52 (86.67%)	3 (5%)	0
Neutral	18 (27.69%)	2 (3.08%)	42 (64.15%)	3 (4.62%)
Surprise	9 (17.64%)	0	8 (15.68%)	34 (66.67%)

Accuracy using LDP features – 47.76%

Table 9 Results using LDP in experiment 3

Predicted → Actual |	Anger	Happiness	Neutral	Surprise
Anger	11 (22.91%)	3 (6.25%)	31 (64.58%)	3 (6.25%)
Happiness	9 (15%)	24 (40%)	27 (45%)	0
Neutral	13 (20%)	1 (1.53%)	51 (78.46%)	0
Surprise	11 (21.57%)	0	19 (37.25%)	21 (41.17%)

Results summary:

Table 10 Comparison of LBP and LDP

Experiment	Features	LBP accuracy	LDP accuracy
1	**Window size of 18X21 (6X7 = 42 windows)**	**70.54%**	**46.87%**
2	**Window size of 15X13 (Windows containing eyes and mouth, 21+10 = 31 windows)**	**70.98%**	**47.76%**
3	**Window size of 22X25 (Windows containing eyes and mouth, 8 + 3 = 11 windows)**	**67.41%**	**38.39%**

LBP features perform much better than LDP features in all our experiments. LBP features are better for facial expression recognition.

4.3 Geometric normalization of detected faces

Till previous experiments, we were using Viola Jones face detector for detecting faces in images. These faces are not geometrically normalized. We have used geometrically normalized faces all the experiments performed after this point.

For geometrical normalization of faces, we used a shape model to detect eyes in an image. The face is then normalized with respect to distance between centers of eyes. Suppose the horizontal distance between the centers of two eyes is 'D'. In the figure below, the method of cropping the face after detecting eye centers in described.

Figure 21 Geometric normalization of face

The aspect ratio of the geometrically normalized face is $2: 2.3$ i.e. $1: 1.35$

4.4 Weber Normalization

Weber normalization is used to remove the lightning effect in images. Weber normalization algorithm is described below:

Algorithm : Weber Normalization

Data: Face image
Output: Weber face image

1 Smoothen the image using a gaussian filter
$F = F * G(x, y, \sigma)$
2 **foreach** *pixel in the image* **do**
3 $Value = \Sigma(pixelIntensity - neighborvalues)$;
4 $Value = \arctan(Value/pixelIntensity)$;
5 Assign value to the pixel in Weber Face Image
6 **end**

Figure 22 Weber normalization algorithm

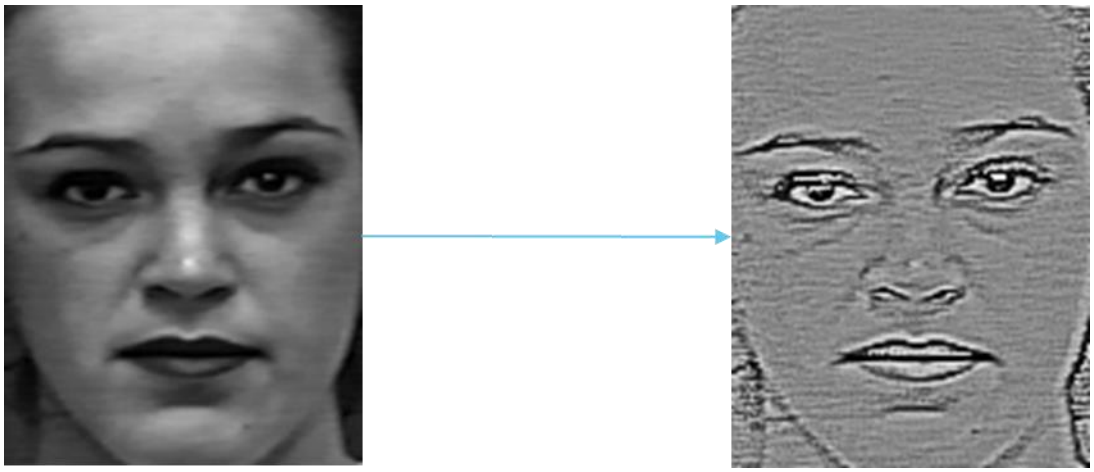

Figure 23 Weber normalization example

This technique is used in face recognition. We tried this technique in facial expression recognition.

The following steps were followed:

- Given an image, geometrically normalized face with resolution 135X100 was cropped from it
- The face was converted to weber normalized face
- This face was divided into 7X6 windows, LBP histogram for each window was computed and the histograms from all windows were concatenated to form a feature vector
- A dataset from Cohn-Kanade dataset was created for training and testing
- SVM with linear kernel was used for training

In all our experiments, we have not smoothed the image using any filter. The first step in Weber normalization is smoothing the face with Gaussian filter. So we created two weber normalized datasets, one where image was smoothed in weber normalization and one where smoothing step was skipped.

Dataset:

A dataset was selected from Cohn-Kanade dataset. CK database consists of 486 image sequence from 97 posers. From the image sequences of Anger, Happiness and Surprise expressions, a dataset was created. In an image sequence, the first image is a neutral image and the last image contains the peak of an expression. For creating a still image dataset, first frame from every image sequence was taken and placed in “Neutral” class. The last three images of the image sequence were placed in expression class i.e. if the

image sequence was labelled Anger, the last three images of the image sequence were placed in “Anger” class.

From the total 97 posers, images from 75 posers were used for training and the images from 22 posers were used for testing.

Table 11 Dataset details for Weber normalization

Expression	Number of training images	Number of testing images
Anger	159	48
Happiness	204	60
Surprise	171	48
Neutral	178	52
Total	712	208

Results:

Experiment 1: Image not smoothed during Weber normalization

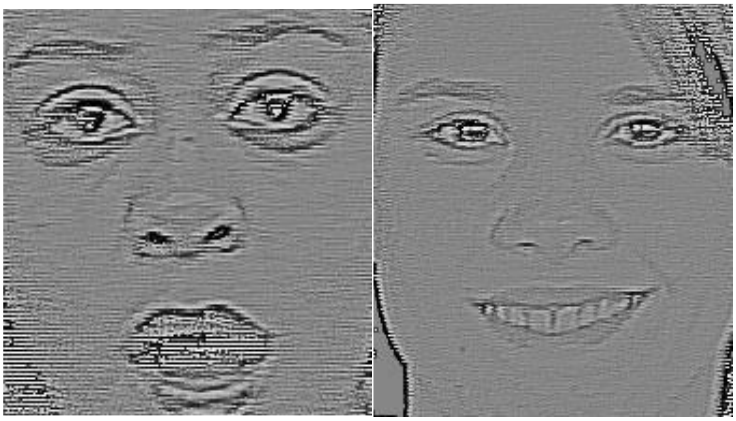

Figure 24 Sample weber faces when smoothing is skipped

Accuracy using SVM – 39.9%

Table 12 Results of Weber normalization when smoothing is skipped

Predicted → Actual |	Anger	Happiness	Surprise	Neutral
Anger	22 (45.83%)	15 (31.25%)	0	11 (22.91%)
Happiness	20 (33.33%)	18 (30%)	0	22 (36.67%)
Surprise	6 (12.5%)	6 (12.5%)	32 (66.67%)	4 (8.33%)
Neutral	24 (46.15%)	15 (28.84%)	2 (3.84%)	11 (21.15%)

Experiment 2: Image is smoothed during Weber normalization

Figure 25 Sample Weber faces

Accuracy using SVM – 42.78%

Table 13 Results using Weber normalization

Predicted → Actual |	**Anger**	**Happiness**	**Surprise**	**Neutral**
Anger	20 (41.67%)	16 (33.33%)	2(4.17%)	10 (20.83%)
Happiness	15 (25%)	26 (43.33%)	0	19 (31.67%)
Surprise	8 (16.67%)	2 (4.17%)	34 (70.83%)	4 (8.33%)
Neutral	22 (42.31%)	21 (40.38%)	0	9 (17.31%)

Even after smoothing the image in Weber normalization, the accuracy does not improve much. After removing the lightning effect, local information around the pixels is not a good feature for facial expression recognition.

Chapter 5

5. Facial Expression Recognition in videos

Until now, we were working with still images. We were not making use of the information present in a continuous video to predict the expressions. In this section, we will discuss LBP features in three orthogonal planes (LBP-TOP) and Geometric features.

5.1 LBP features in three orthogonal planes (LBP-TOP)

Problem Statement: Predict the labels of image sequences of Cohn-Kanade dataset. The image sequences start from neutral frame and the last frame is the peak of expression.

Solution:

From the input image sequence, faces are detected to form face image sequence. LBP-TOP features are extracted from the face image sequence to form a feature vector. This feature vector is used by machine learning algorithm to predict the expression in the image sequence.

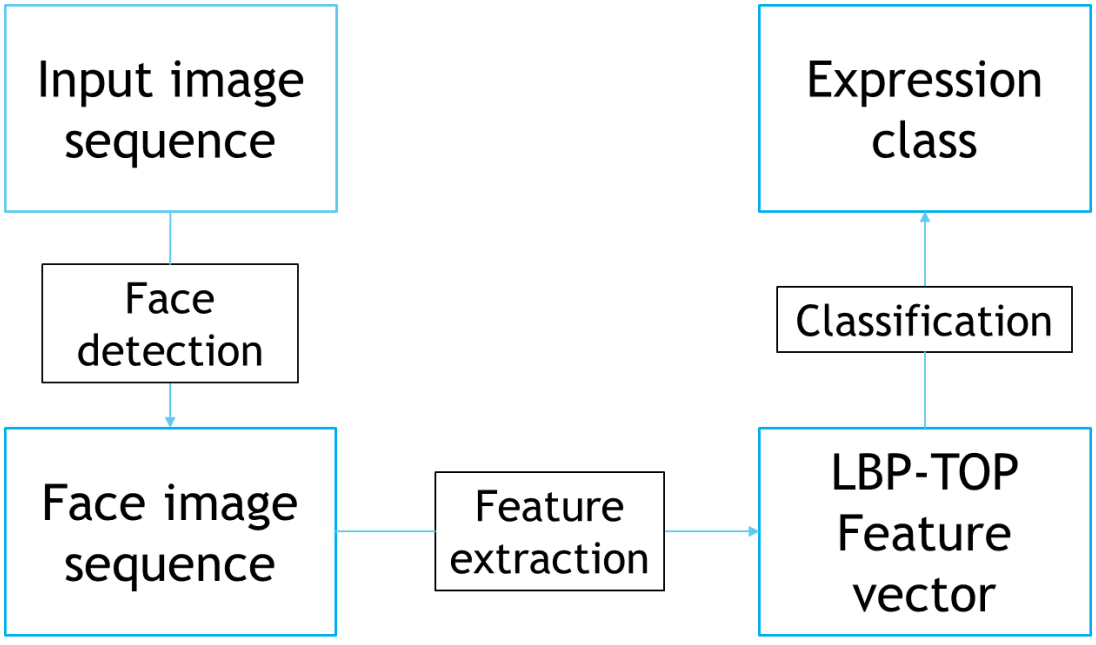

Figure 26 Method for expression recognition in image sequence

Face detection:

In the *first frame* of image sequence, the co-ordinates of rectangle which contains geometrically normalized face are computed. In all the frames of an image sequence, this rectangular portion is cropped to form a face image sequence. In the image sequences of Cohn-Kanade dataset, the motion of posers is minimal. Hence it is justified to detect the face window in first fame and track this window in subsequent frames. All the images in a particular face image sequence are of same resolution. Size of faces in different image sequence can vary.

LBP-TOP features extraction:

TOP – Three Orthogonal Planes

The images in an image sequence are stacked on time axis. This arrangement of the images stacked along time axis is known as volume. There are three orthogonal planes in the cuboidal volume, XY, XT and YT plane.

Method for LBP-TOP feature extraction from a volume:

- The volume is divided into cuboidal sub-volumes
- In each sub-volume, LBP histograms are computed in XY, XT and YT planes
- Since number of images in each sequence is not fixed, each of the LBP histograms is normalized so that all values in a histogram lie in $[0, 1]$
- The normalized LBP histograms in XY, XT, YT planes are concatenated to form a feature vector of the sub-volume
- Feature vectors from all the sub-volumes are concatenated to form LBP-TOP feature vector

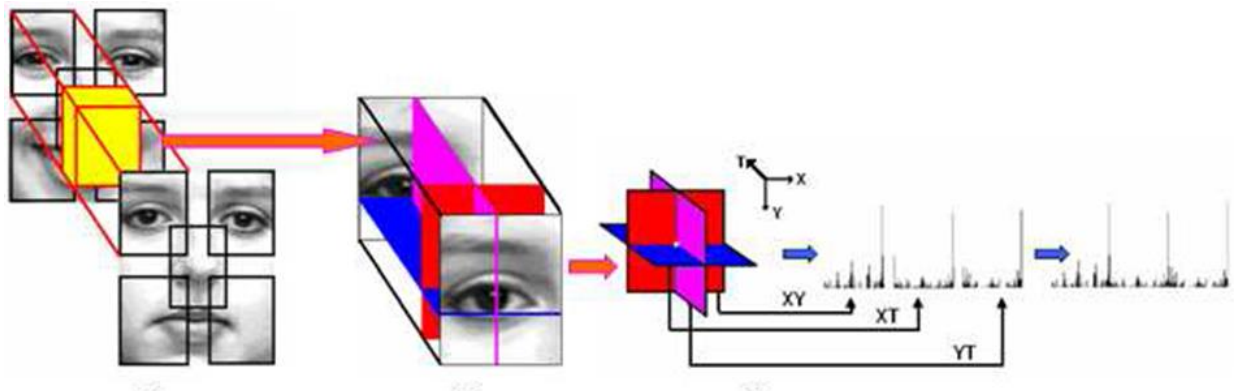

Figure 27 LBP-TOP features. Image source - Zhao and Pietik [7]

Dataset:

Cohn-Kanade dataset consists of image sequences from 97 people. There are a total of 230 image sequences of Anger, Happiness and Surprise expressions. Image sequences from 75 persons were used for training and 22 people were used for testing.

Table 14 Details of dataset for LBP-TOP features

Expression	Number of training sequences	Number of testing sequences
Anger	50	19
Happiness	65	23
Surprise	54	19
Total	169	61

Classification:

After extracting LBP-TOP features from image sequences, a machine learning algorithm is required to label these image sequences. Nearest neighbor, Naïve Bayes and SVM classifiers were trained and the results were obtained on test data.

Nearest Neighbor classifier:

Using the labels of 10 nearest neighbors for classification, 80.32% accuracy on test data is observed.

Table 15 Results using nearest neighbor classifier

Predicted → Actual |	Anger	Happiness	Surprise
Anger	15 (78.95%)	4 (21.05%)	0
Happiness	3 (13.04%)	20 (86.96%)	0
Surprise	5 (26.31%)	0	14 (73.69%)

Using 5 nearest neighbors, 77.07% accuracy is observed.

Naïve Bayes classifier:

Using Naïve Bayes classifier, 86.88% accuracy on test data is observed.

Table 16 Results using Naive Bayes classifier

Predicted → Actual |	Anger	Happiness	Surprise
Anger	15 (78.95%)	4 (21.05%)	0
Happiness	0	23 (100%)	0
Surprise	3 (15.79%)	1 (5.26%)	15 (78.94%)

When SVM classifier was used, all the test instances were classified as “Surprise”.

5.2 Geometric features

Till now, we have been using appearance features like LBP, LDP and LBP-TOP in our experiments. Geometric features use the location of points on eyes and mouth and also the distance between various points.

In this experiment, geometric features were used to predict the expressions in frames of image sequences of Cohn-Kanade dataset.

Feature extraction:

- Using a shape model, location (X and Y co-ordinates) of the shown 49 points were detected in all the frames of an image sequence
- In Cohn-Kanade dataset, any image sequence starts from “Neutral” pose and the final frame is the peak of expression. First frame is taken as a reference frame. Let X0 stores the 98 co-ordinates of the 49 points in the first frame of an image sequence
- In the subsequent frames, 98 co-ordinates are detected using the shape model. These co-ordinates are subtracted from X0 and the resultant vector is used as feature vector
- Second, third and fourth frame in image sequences still look like neutral. Hence the feature vectors corresponding to these frames are labelled “Neutral”
- Feature vectors corresponding to all the other frames in the image sequence are labelled with an expression. This expression is same as the expression label of the image sequence.
- SVM classifier with linear kernel was used for training the model

Dataset:

Out of 97 posers in Cohn-Kanade dataset, images from 75 posers were used for training the model and images from 22 posers were used for testing.

Table 17 Details of dataset for geometric features

Expression	Number of training images	Number of testing images
Anger	776	242
Happiness	1065	348
Surprise	652	195
Sad	655	193
Neutral	643	228
Total	3791	1206

Results:

At first, we only considered the expressions of Anger, Happiness and Surprise.

Accuracy using SVM – 86.54%

Table 18 Resuts using geometric features

Predicted → Actual |	Anger	Happiness	Surprise
Anger	190 (78.51%)	21 (8.67%)	31 (12.81%)
Happiness	13 (3.74%)	335 (96.26%)	0
Surprise	27 (13.85%)	14 (7.18%)	157 (80.5%)

PCA was performed on training data and the training examples were projected on first two principal components.

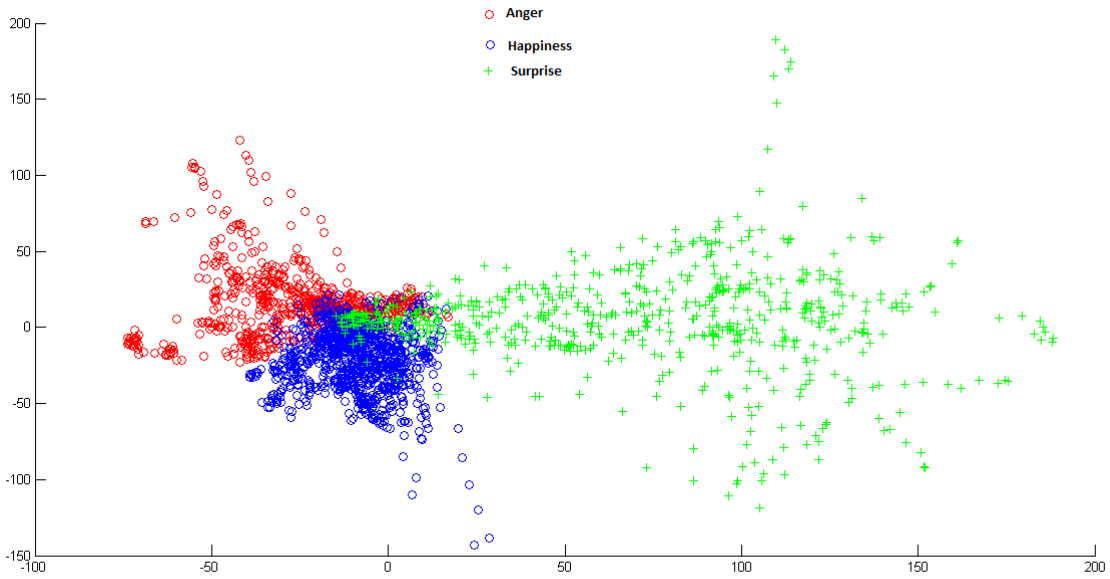

Figure 28 Projection on first two principal components

It is observed that the training data is getting separated reasonably well in two principal components and when more components are used, good accuracy during classification is observed.

Then we also considered "Sad" expression.

Accuracy using SVM – 86.3%

Table 19 Results using geometric features

Predicted → Actual |	**Anger**	**Happiness**	**Surprise**	**Sad**
Anger	185 (76.44%)	12 (4.95%)	19 (7.85%)	26 (10.74%)
Happiness	15 (4.31%)	332 (95.4%)	0	1 (0.28%)
Surprise	26 (13.13%)	0	153 (77.27%)	19 (9.6%)
Sad	10 (5.26%)	2 (1.05%)	4 (2.1%)	174 (91.58%)

PCA was performed on training data and the training examples were projected on first two principal components.

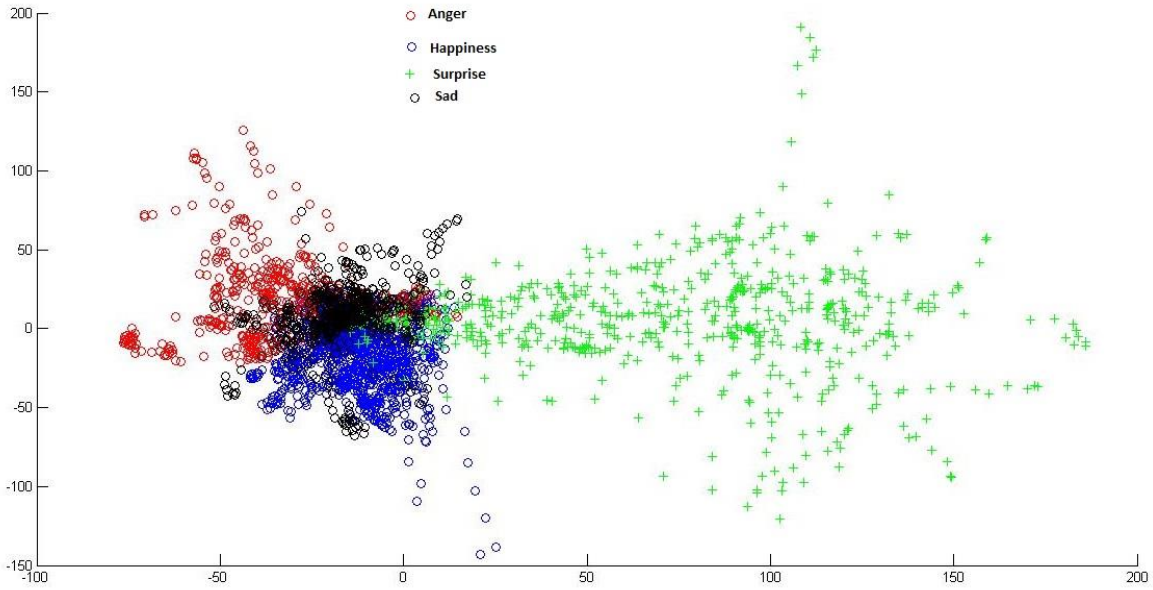

Figure 29 Projection on first two principal components

It is observed that "Surprise" is getting separated from all the other expressions. "Sad" is getting mixed with "Anger" in first two principal components space. When more principal components are considered, the data is getting well separated and good accuracy on test data is observed.

Finally, we also included "Neutral" expression

Accuracy using SVM – 77.22%

Table 20 Accuracy using geometric features

Predicted → Actual |	**Anger**	**Happiness**	**Surprise**	**Neutral**	**Sad**
Anger	148	7	0	69	18
Happiness	1	306	0	41	0
Surprise	37	0	133	10	18
Neutral	19	10	5	177	14
Sad	1	0	3	21	165

Table 21 Percentage accuracy using geometric features

Predicted → Actual |	Anger	Happiness	Surprise	Neutral	Sad
Anger	61.157	2.8926	0	28.5124	7.438
Happiness	0.2874	87.931	0	11.7816	0
Surprise	18.6869	0	67.1717	5.0505	9.0909
Neutral	8.4444	4.4444	2.2222	78.6667	6.2222
Sad	0.5263	0	1.5789	11.0526	86.8421

When "Neutral" expression is considered, the accuracy is decreased. Confusion arises between anger and neutral, anger and surprise. The cause for confusion between anger and neutral is that during anger, the movement of points around eyes and mouth is not that much. The texture change is much better feature for discriminating between anger and neutral.

PCA was performed on training data and the training examples were projected on first two principal components.

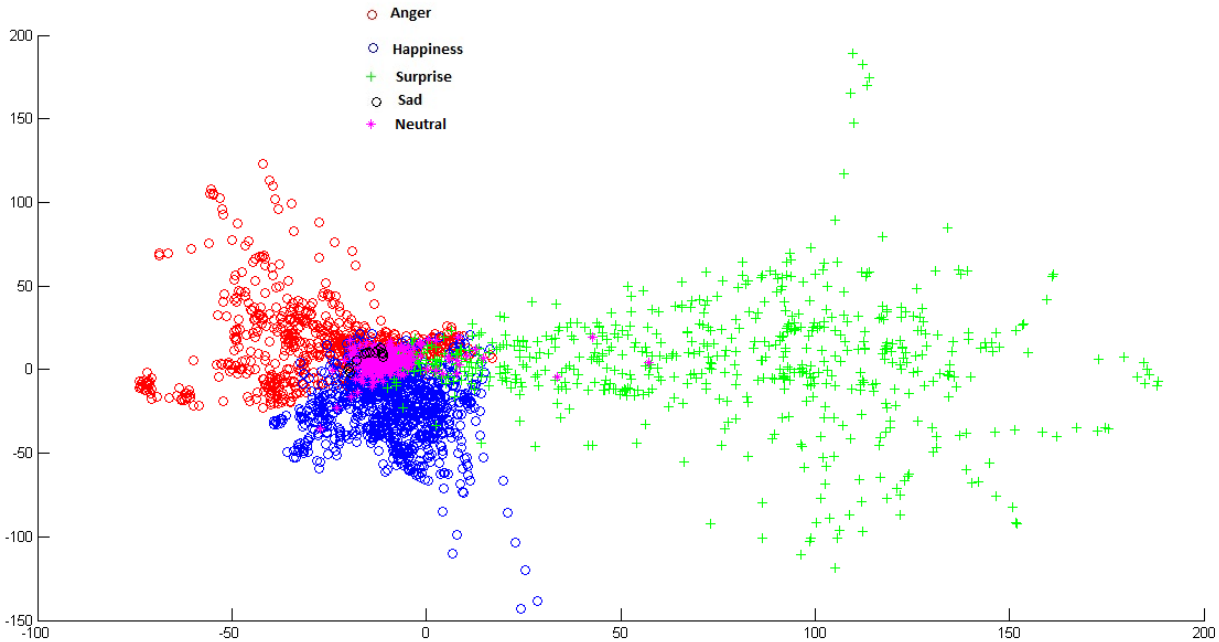

Figure 30 Projection on first two principal components

It is observed that “Surprise” is getting well separated from all the other expressions. “Anger” and “Happiness” are also getting separated from each other. Neutral and Sad are not getting separated. When more principal components are considered, these expressions are getting moderately well separated and satisfactory accuracy on test data is observed.

References

[1] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, 2001.

[2] P. Viola and M. Jones. Robust Real-Time Face Detection, 2003.

[3] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. Proceedings of the 2nd European Conference on Computational Learning Theory (EuroCOLT '95), pp. 23–37, Barcelona, Spain, March 1995.

[4] Zhan, C., Li, W., Ogunbona, P. & Safaei, F. (2008). A real-time facial expression recognition system for online games. International Journal of Computer Games Technology, 2008 (Article No. 10), 1-7.

[5] M. S. Bartlett, G. Littlewort, I. Fasel, et al., Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction, in IEEE CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction, Wisconsin, USA, 2003.

[6] Z. Zhang, M. J. Lyons, M. Schuster, S. Akamatsu, Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron, in IEEE International Conference on Automatic Face & Gesture Recognition (FG), 1998.

[7] Guoying Zhao and Matti Pietik ainen. Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2007.

[8] Caifeng Shan, Shaogang Gong, Peter W. McOwan. Facial expression recognition based on Local Binary Patterns: A comprehensive study. Image and Vision Computing 27 (2009) 803–816.

[9] T. Ojala, M. Pietikäinen, D. Harwood. A comparative study of texture measures with classification based on featured distribution. Pattern Recognition 29 (1) 1996 51–59.

[10] Z. Zhang, M. Lyons. M. Schuster, and S. Akamatsu. Comparison between geometry based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron. Proc. IEEE 3rd Int'l Conf. on Automatic Face and Gesture Recognition, Nara, Japan, April 1998.

[11] Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel, Javier R. Movellan. Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction. Conference: Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on, Volume: 5.

[12] Y. Tian. Evaluation of face resolution for expression analysis. IEEE Workshop on Face Processing in Video, 2004.

[13] I. Cohen, N. Sebe, Garg A., L. Chen, and T. Huang. Facial expression recognition from video sequences: Temporal and static modeling. CVIU, vol. 91, pp. 160–187, 2003.

[14] T. Ahonen, A. Hadid, and M. Pietikinen. "Face recognition with local binary patterns. ECCV, 2004, pp. 469–481.

[15] Frank Wallhoff. Facial Expressions and Emotion Database. http://cotesys.mmk.e-technik.tu-muenchen.de/waf/fgnet/feedtum.html. Technische Universität München 2006-2015.

[16] Michael J. Lyons, Shigeru Akemastu, Miyuki Kamachi, Jiro Gyoba. Coding Facial Expressions with Gabor Wavelets. 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200-205 (1998).

[17] Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, 46-53.

[18] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101.

[19] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] MATLAB and Statistics and Machine Learning Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.

[21] Paul Ekman. Strong Evidence for Universals in Facial Expressions: A reply to Russell's mistaken critique. 1994

[22] P. Ekman and W.V. Friesen. Manual for the Facial Action Coding System. 1977.

[23] P. Ekman, W.V. Friesen. Facial Action Coding System: Investigator's Guide. 1978.

[24] Caifeng Shan, Shaogang Gong and Peter W. McOwan. Robust Facial Expression Recognition Using Local Binary Patterns. 2005.

[25] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sebastien Jean, Pierre Froumenty, Aaron Courville, Pascal

Vincent, Roland Memisevic, Christopher Pal, Yoshua Bengio. EmoNets: Multimodal Deep Learning Approaches for Emotion Recognition in Video. 2013

[26] Yaseer Yacoob, Larry S. Davis. Recognizing Human Facial Expressions from Long Image Sequences Using Optical Flow. 1996

[27] Swapna Agarwal, Sanjoy Mazumder, Dipti Prasad Mukherjee. Recognizing Facial Expressions in the Orthogonal Complement of Principal Subspace. 2014

[28] Timur R. Almaev, Michel F. Valstar. Local Gabor Binary Patterns from Three Orthogonal Planes for Automatic Facial Expression Recognition. 2013

[29] Jagdish Lal Raheja, Umesh Kumar. Human Facial Expression Detection from Detected in Captured Image using Back Propagation Neural Network. 2010

[30] Chuan Wan, Yantao Tian, Shuaishi Liu. Facial Expression Recognition in Video Sequences. 2012

[31] Philipp Michel, Rana El Kaliouby. Real Time Facial Expression Recognition in Video using Support Vector Machines. 2003

[32] Taskeed Jabid, Md. Hasanul Kabir, and Oksam Chae. Local Directional Pattern (LDP) for Face Recognition. 2010