1. INTRODUCTION

Micro-Expression are brief facial Expression that reveals the feeling of person. A lot of work has been done on these expressions. They exist for short period of time like ½ of second so they are negligible by naked human eyes. Study of micro-expression will help in find out what a person is feeling even if he wants to hide his expressions. In 1969, Ekman analyzed a person's video of psychological treatment who was depressed and tried to commit suicide and found micro-Expressions on his face. A micro-expression is a brief, involuntary facial expression that is shown on the face of humans according to the emotions that are being experienced. Unlike regular pro-longed facial expressions, it is difficult to fake a micro-expression. First we should recognize, as David Matsumoto has pointed out, that there are behaviors, gestures, or expressions of the face that do occur without conscious prompting which leak or reveal our true feelings or sentiments. Some of these behaviors or expressions flash before us very quickly and others loiter there seemingly too long. Also there are behaviors that are difficult to observe because they are so tiny (twitching muscles just under the eye for example) while others are quite large or as "large" as they can be given the size of some small facial muscles. Micro expressions happen when people have hidden their feelings from themselves (repression) or when they deliberately try to conceal their feelings from others. Importantly, both instances look the same; you cannot tell from the expression itself whether it is the product of suppression (deliberate concealment) or repression (unconscious concealment). They can help to find if a person is lying that's why they have such applications like national security, psychiatric treatment, lie detector, human robotics.

- In Lie detector, when a person is lying some micro-expressions appears on his face due to emotions like fear then this system can be used to recognize whether a person is lying.
- In Human Robotics, as robots begin to interact more and more with humans and start becoming a part of our living spaces and work spaces, they need to become more intelligent in terms of understanding the human's moods and emotions. Expression recognition systems will help in creating this intelligent visual interface between the man and the machine.
- In National security, these systems can be used for detecting for suspicious activities.
- In Psychological treatment, this system can be used to understand emotions of mental patient so that they can be treated.

These Expressions are different from normal expressions because these are uncontrollable. These expression occurs on human face when a person is trying to hide his feeling then some facial movements occurs on his face that reveals its true emotions. There are basically 6 type of expressions of person's face-

1) Happiness 2) Sadness 3) Fear 4) Surprise 5) Disgust 6) Repression

There are different facial expression databases but micro-Expression databases are rare. They can be used for designing algorithms for micro-Expression recognition system.

- SMIC (Spontaneous Micro-expression Database) [1] contains only 3 classes of expression.

  1) Positive 2) Negative 3) Surprise

  In this database 16 subjects were there with 164 samples and frame rate of 100 frame per second.

  For better expressions such type of environment is created like there is punishment threat and highly emotional clips were chosen to create extreme conditions in which participants going under high emotional feeling has to suppress their emotions.

- CASME [2] Database (Chinese Academy of Sciences Micro-expression) contains 195 micro-Expression filmed with 60 frame per second camera. There were 35 participants (22 males and 13 females) with mean age of 22.03 Years. Micro-Expression with the duration less than 500ms were selected for database but some expression that last more than 500ms with their onset less than 250ms were also selected because fast onset facial expression can also be categorized under micro-Expressions. Facial expression for this database were recorded in different environment with two different type of cameras having different resolutions. Frame rate of both the camera were same (60 fps). They can be divided into 2 classes-

a) Class A

   The samples in this class were recorded in natural light with BenQ M31 camera with resolution 1280×720 pixels.

b) Class B

   The samples were taken in a room with presence of two LED lights with Point Grey GRAS-03K2C camera with resolution of 640×480 pixels.

- CASME II [3] Database comprises of 246 micro-expressions from 26 subjects with a mean age of 22.03 years. The videos were collected using Point Grey GRAS-03K2C camera at a resolution of 640 × 480 pixels and a frame rate of 200fps. The average frame length is 244 frames (~1.22s), with the longest being 1,024 frames (~5.12s) and the shortest being 51 frames (~0.26s). There are five main categories of micro-expressions, in the following distribution: 25 surprise videos, 27 repression videos, 32 happiness videos, 63 disgust videos and 99 other videos. The micro-expressions are elicited from the subjects by showing them some video clips and asking them to keep a poker face when watching the videos.

2. LITERATURE SURVEY

Table 1

Summary of work based on Micro-Expressions

Authors	Year	Database	Methodology	Classifier	Accuracy
Xiaobai Li et al. [1]	2013	SMIC-HS SMIC-VIS SMIC-NIR	LBP-TOP And TIM	SVM	Detection - 65.55% Recognition - 48.78%
Wen-Jing Yan et al. [2]	2013	CASME	Action Units	SVM	75.66%
Wen-Jing Yan et al. [3]	2014	CASME II	LBP-TOP Action Units	SVM	63.41%
Yong-Jin Liu et al. [4]	2016	SMIC CASME CASME II	LBP-TOP HOOF MDMO	SVM	71.40% LBP-TOP on SMIC 55.69% HOOF on CASME 67.37% MDMO on CASMEII
Xiaohua Huang et al. [5]	2015	CASME II SMIC	LBP-TOP	SVM	59.51%
Iyanu Pelumi Adegun et al. [6]	2016	CASME II	LBP-TOP	ELM	96.12%
Dae Hoe Kim et al. [7]	2015	CASME II	LSTM	CNN	60.98%
Xiaobai Li et al. [8]	2017	SMIC-VIS SMIC-HS SMIC-NIR CASME II	LBP HOOF	SVM	81.7% on SMIC-VIS 68.29% on SMIC-HS 67.61% on SMIC-NIR 67.21% on CASMEII
Yandan Wang et al. [9]	2016	CASME II	LBP	SVM	75.30%
Sze-Teng Liong et al. [10]	2016	SMIC CASME II	LBP-TOP	SVM	74.68% on CASME II 74.52% on SMIC
Pavithra M et al. [11]	2016	N/A	LBP-TOP HOG	N/A	89%

Amr Suleiman et al. [12]	2014	N/A	HOG	N/A	82.56%

Yong-Jin Liu et al. [4] proposed a simple yet effective MDMO feature for micro-expression recognition. The MDMO is a ROI-based optical flow feature, which makes use of both local statistic motion information (i.e., the mean of all optical flow vectors in a ROI falling into a bin with the maximum count) and its spatial location (i.e., the ROI to which it belongs). The feature dimension of MDMO is small, i.e., 362 ¼ 72, where36 is the number of ROIs. To obtain reliable optical flow vectors, they proposed an alignment method in the optical flow domain to remove noise induced by small head movements in micro-expression video clips. Experimental results on three spontaneous micro-expression databases CASME, CASME II and SMIC showed that compared to two baseline features, i.e., LBP-TOP and HOOF, MDMO consistently has the best performance in both subject-independent and subject-dependent evaluations.

Xiaohua Huang et al. [5] proposed a novel frame work based on a new spatio temporal facial representation to analyze micro-expressions with subtle facial movement. Firstly, an integral projection method based on difference images is utilized for obtaining horizontal and vertical projection, which can preserve the shape attributes of facial images and increase the discrimination for micro-expressions. Furthermore, they employ the local binary pattern operators to extract the appearance and motion features on horizontal and vertical projections. Intensive experiments are conducted on available published micro-expression databases for evaluating the performance of the method. Experimental results demonstrate that the new spatio temporal descript or can achieve promising performance in micro-expression recognition.

Iyanu Pelumi Adegun et al. [6] proposed a framework for detecting the presence of micro-expressions using local binary patterns on three orthogonal planes (LBP-TOP) because of its ability to extract temporal features and extreme learning machine (ELM) because of its fast learning speed . To evaluate the performance of the algorithm, CASME II micro-expression database was used for the experiment. they obtained an accuracy of 96.12% which is a significant improvement when compared with the state-of-the-art methods.

Dae Hoe Kim et al. [7] proposed a learning method that consists of two parts. First, the spatial features of micro-expressions at different expression-states (i.e., onset, onset to apex transition, apex, apex to offset transition and offset) are encoded using convolutional neural networks (CNN). The expression-states are taken into account in the objective functions, to improve the expression class separability of the learned feature representation. Next, the learned spatial features with expression-state constraints are transferred to learn temporal features of micro-expression. The temporal feature learning encodes the temporal characteristics of the different states of the micro-expression using long short-term memory (LSTM) recurrent neural networks. Extensive and comprehensive experiments have been conducted on the publically available CASME II micro-expression dataset. The experimental results showed that the proposed method outperformed state-of-the-art micro-expression recognition methods in terms of recognition accuracy.

Xiaobai Li et al. [8] proposed novel methods for both ME spotting and ME recognition. For ME spotting, they are the first to proposed a method able to spot MEs from spontaneous long videos. The method is based on feature difference (FD) comparison. Two features (LBP and HOOF) are employed, and LBP is shown to outperform HOOF on two databases. For ME recognition, they proposed a new framework where motion magnification is employed to counter the low intensity of MEs. they validated the new framework on SMIC and CASMEII databases, and showed that our method outperforms the state of the art on both databases. they also drew many interesting conclusions about the respective benefits of their method's components. Finally, they proposed the first automatic ME analysis system which first spots and then recognizes MEs. It outperforms humans at ME recognition by a significant margin, and performs comparably to humans at the combined ME spotting & recognition task. This method has many potential applications such as in lie detection, law enforcement and psychotherapy.

Sze-Teng Liong et al. [10] proposed a novel feature extraction approach for the detection and recognition of facial micro-expressions in video clips. The proposed method describes the fine subtle movements on the face using optical strain technique in two different ways. The first, a direct usage of optical strain information as a feature histogram, and second, the usage of strain information as weighted coefficients to LBP-TOP features. The concatenation of the two feature histograms enable us to achieve promising results in both detection and recognition tasks. Experiments were performed on two recent state-of-the-art databases – SMIC and CASME II.

Pavithra M et al. [11] proposed a system that introduced a new detection mechanism for detect and localize the anomalies in Crowded Environments. Here the histogram of Local Binary Patterns from Three Orthogonal Planes (LBP-TOP) is used for represent the dynamic texture. In a time window of each frame average triplets of HOG, HOS and LBP-TOP are consecutively computed. Then, these features are passed as an input to classifier. Here proximal support machine is used for anomaly classification.

3. CONTRIBUTION

We propose a modified LBP feature extractor method for 8 neighbour pixel. Existing LBP feature extractor method include 16 neighbour pixel. In our method we are using only those neighbour pixels that are responsible for changes in region of interest. This modified method gives better accuracy in recognizing micro-expressions.

3.1. METHODOLOGY

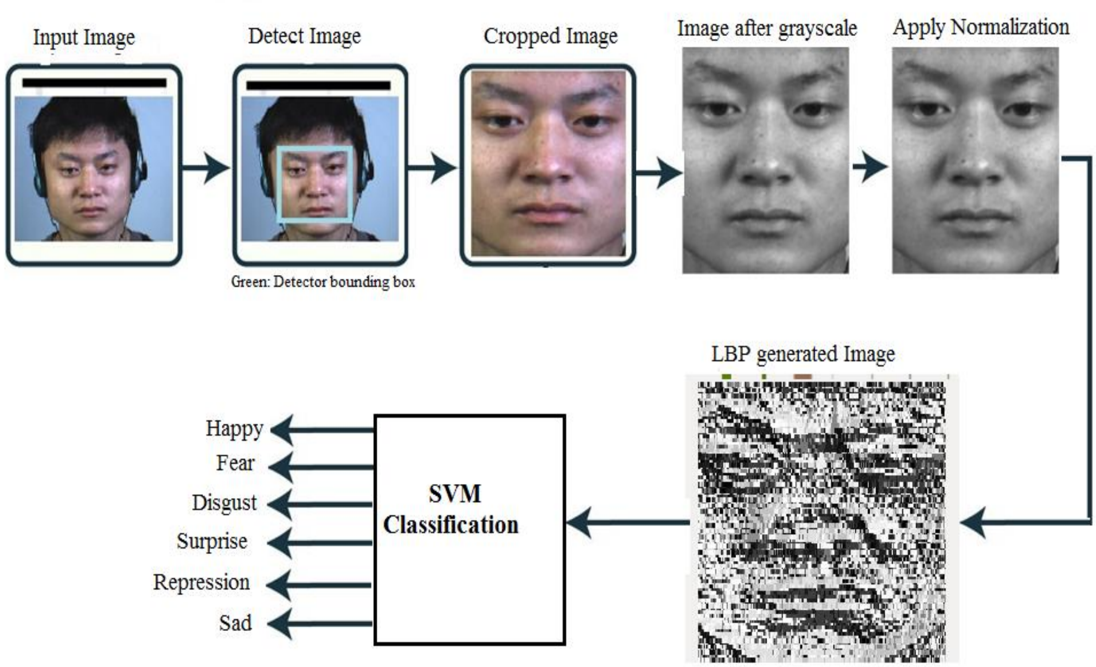

Fig 1 - Overall Frame Work

Our proposed method comprises following steps:

3.1.1 Data acquisition

CASME II Database has been collected from Chinese Academy of Science. at http://fu.psych.ac.cn/CASME/casme2-en.php.CASME II Database comprises of 246 micro-expressions from 26 subjects with a mean age of 22.03 years. The videos were collected using Point Grey GRAS-03K2C camera at a resolution of 640 × 480 pixels and a frame rate of 200fps. Micro-expressions has been classified into six categories- (Happy, Fear, Disgust, Surprise, Sad, Repression).

3.1.2 Data Normalization

Noise have been removed from collected data by -

1. Detect face in raw image by using Viola-Jones object detection framework.

   Viola-Jones object detection framework widely used method for real-time object detection because, detection is very fast. This algorithm is implemented in OpenCV as cvHaarDetectionObjects( ).

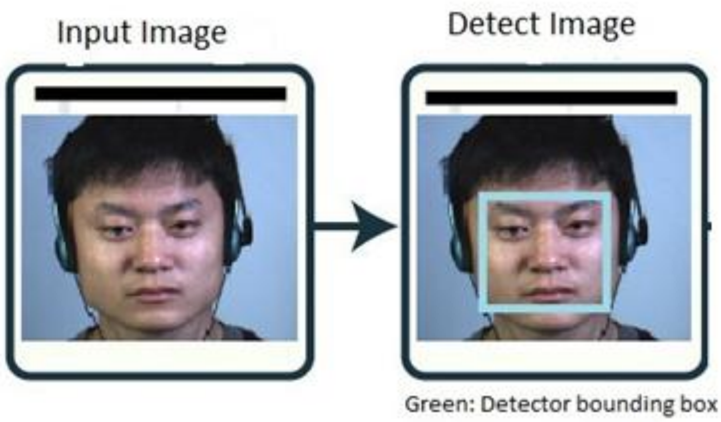

Fig 2- Face Detection in raw image

2. Crop desired area (face) and remove unwanted part from image.

Fig 3 - Face Cropped image

3. Converting image into grayscale

If each color pixel is described by a triple (R, G, B) of intensities for red, green, and blue - By luminosity method, It averages the values, but it forms a weighted average to account for human perception. We're more sensitive to green than other colors, so green is weighted most heavily. T

Grayscaled image=0.21 R + 0.72 G + 0.07 B.

Fig 4 - Image into grayscale

4. Apply Median Filtering on grayscale image.

Fig 5 - Image after applying median filter

The median filter is an effective method that can, to some extent, distinguish out-of-range isolated noise from legitimate image features such as edges and lines. Specifically, the median filter replaces a pixel by the median, instead of the average, of all pixels in a neighborhood w

$$y[m,n] = median\{x[i,j],(i,j) \in w\} \quad \text{.........................}(1)$$

where w represents a neighborhood defined by the user, centered around location $[m,n]$ in the image.

3.1.3 Feature extraction

We use the Local Binary Pattern in to extract the features from images. The LBP feature vector, in its simplest form, is created in the following manner:

- Divide the examined window into cells (e.g. 16x16 pixels for each cell).

- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.
- Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1". This gives an 8-digit binary number (which is usually converted to decimal for convenience).
- Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center). This histogram can be seen as a 256-dimensional feature vector.
- Optionally normalize the histogram.
- Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window.
- Generate First order feature of histogram. These features are mean, variance, skewness, energy, entropy, kurtosis .
- Generate Feature Vector with these features.

Fig 6- LBP generated Image

We have extracted 24 features of images by dividing each image into 4 planes and apply LBP on each plane and then generate histogram and calculate six first order feature (Mean, Variance, Skewness, Energy, Entropy, Kurtosis) of that histogram.

- Energy: The energy indicates the strength of the signal as it gives the area under the curve of power at any interval of time.

  $\text{Energy}(E_i) = \sum_{j=1}^{N} \left| {D_{ij}}^2 \right|$; (i=1,2,3........) ....(2)

- Entropy: Entropy is numerical measure of uncertainty of outcome where signal contained thousands of bits of information.

Entropy(EN) $= \sum_{j=1}^{N} {D_{ij}}^2 log({D_{ij}}^2)$; (i=1,2,3........) ....(3)

- Variance : The variance of a data set is the arithmetic average of the squared differences between the values and the mean.

Variance $(\sigma^2) = \sqrt{\frac{1}{N}\sum(X_i - \emptyset_i)^2}$; (i=1,2,3........) ....(4)

- Mean: X is random variable with mean μ.

Mean $(\mu) = \frac{1}{N}\sum_{j=1}^{N} D_{ij}$; (i=1,2,3........) ....(5)

- Skewness: It is measure of symmetry, or more precisely, the lack of symmetry. A distribution is symmetric if it looks the same to the left and right of the center point.

Skew (X) = $E[(\frac{X-\mu}{\sigma})^3]$ ....(6)

- Kurtosis: It is the measure of whether the data are heavily-tailed or light-tailed relative to normal distribution.

Kurt (X) = $E[(\frac{X-\mu}{\sigma})^4]$ ....(7)

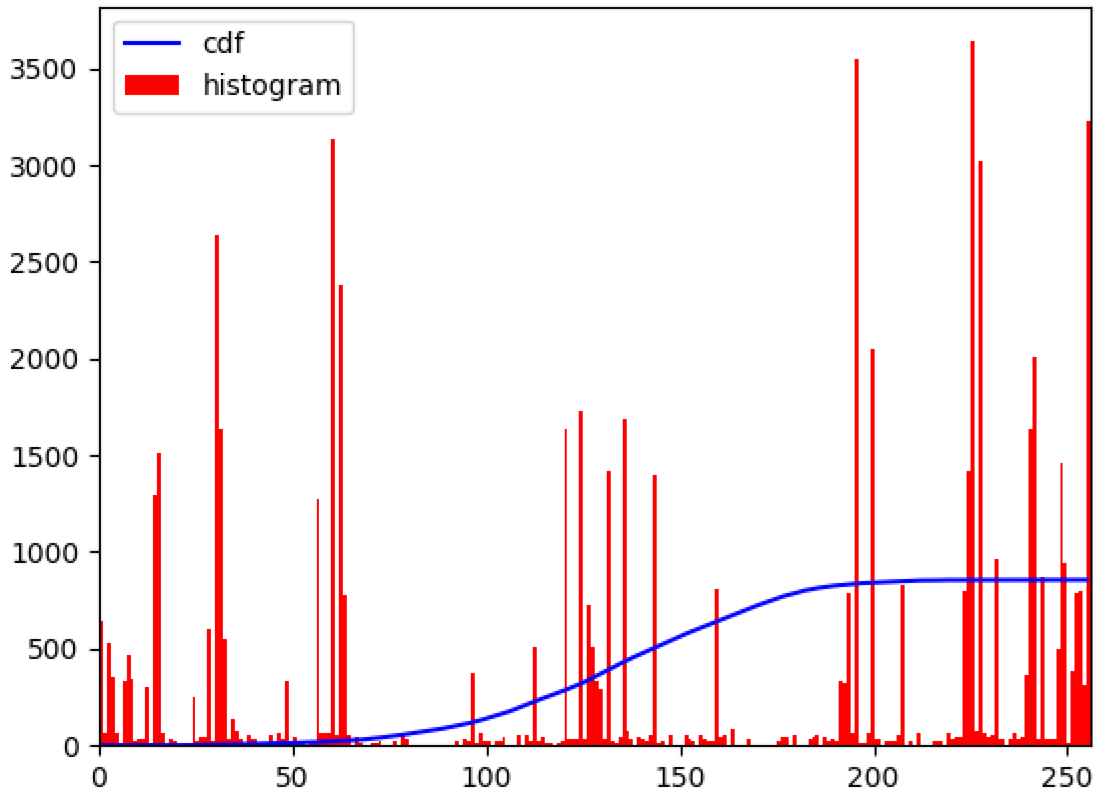

Fig 7 - LBP generated Histogram

3.1.4 Feature Classification

SVM (Support Vector Machine) This method constructs a set of hyper planes in a high dimensional space, which is use for classification or regression. The good separation achieved by the hyper plane. SVM uses non-parametric with binary classifier approach and handles more input data efficiently. Accuracy depends on hyper plane selection. The Structure of the SVM algorithm is more complicated than other methods. This gives the low result transparency.

For classification of emotions 488 images were taken as training data (100 for Disgust, 38 for Fear, 50 for Sad, 100 for Happy, 100 for Repression and 100 for Surprise) and 123 images were taken as test data (25 for Disgust, 8 for Fear, 15 for Sad, 25 for Happy, 25 for Repression and 25 for Surprise).

4. RESULTS

Table 2

Result Analysis of Micro-expressions Using Non-linear SVM

S. N.	Micro-Expressions	Training Samples	Testing Samples	Accuracy in Percentage
1	Happy	100	25	92%
2	Sad	50	15	80%
3	Repression	100	25	88%
4	Fear	38	8	75%
5	Surprise	100	25	96%
6	Disgust	100	25	92%
Total	6s MEs	488	123	**89.43%**

When the feature matrices were passed into Non-linear SVM, 89.43% accuracy was achieved as shown in table 2.

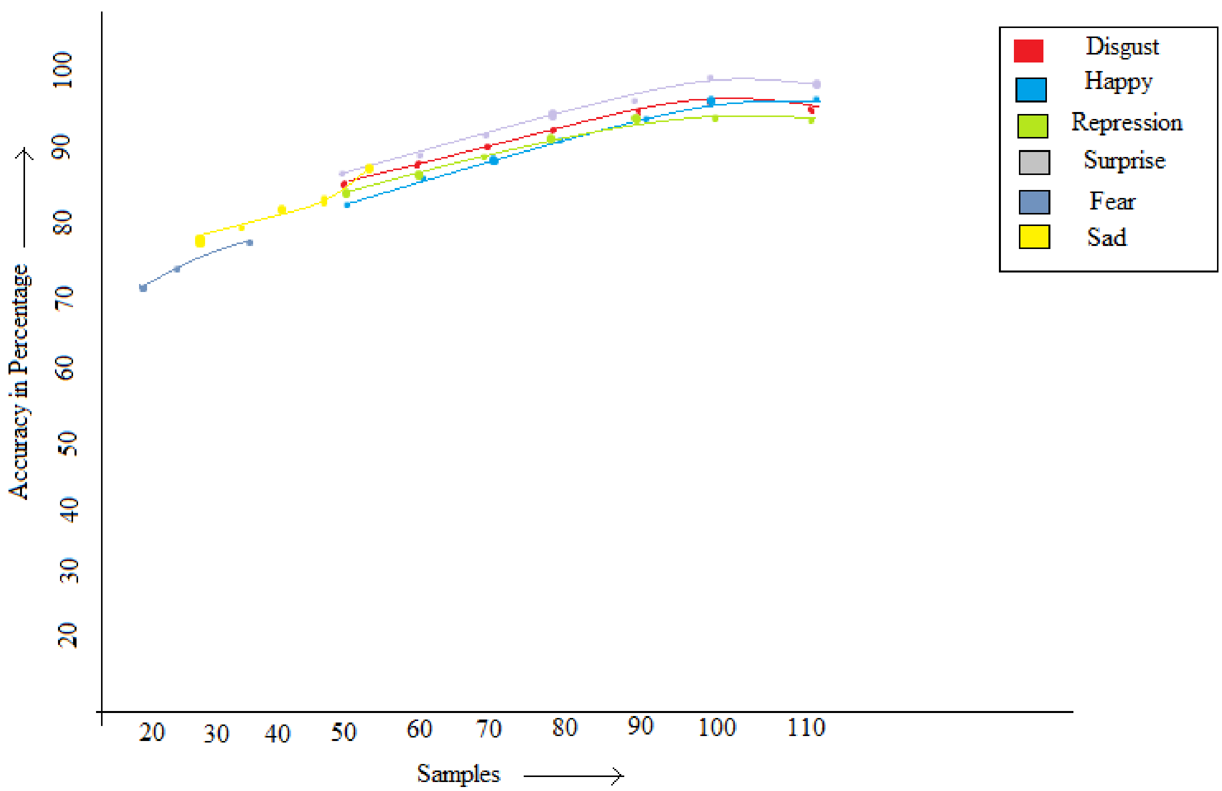

Fig - 8 Micro-expressions accuracy using Non-linear SVM for gamma=0.001 and C=100

Table 3

Result Analysis of Micro-expressions Using Linear SVM

S. N.	**Micro-Expressions**	**Training Samples**	**Testing Samples**	**Accuracy in Percentage**
1	Happy	100	25	80%
2	Sad	50	15	66.66%
3	Repression	100	25	80%
4	Fear	38	8	73.68%
5	Surprise	100	25	84%
6	Disgust	100	25	76%
Total	6s MEs	488	123	**76.72%**

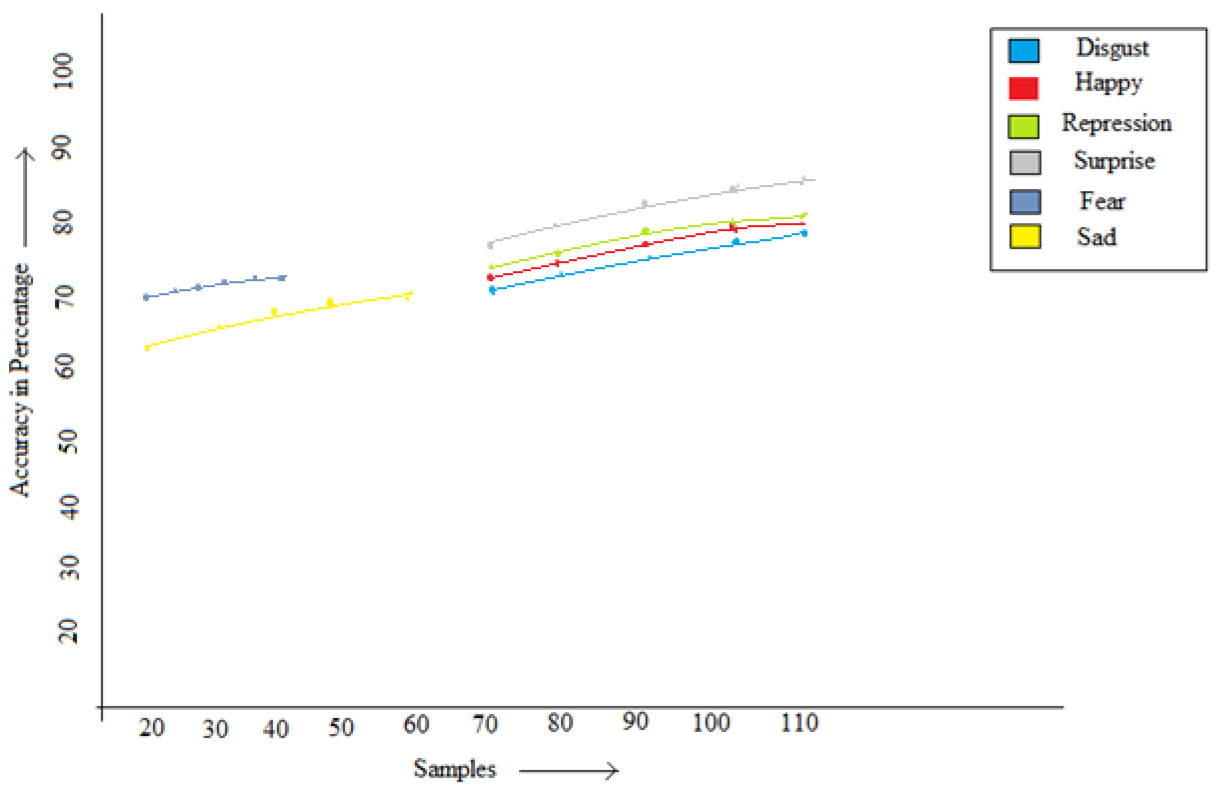

Fig - 9 Micro-expressions accuracy using Linear SVM

When the feature matrices were passed into Linear SVM, 76.72% accuracy was achieved as shown in table 3.

5. SOURCE CODE

Source code:

```
import cv2
import numpy as np
import random
from matplotlib import pyplot as plt
from sklearn import svm
import os.path

abc="Test_Images/Happy/2"
source_image = cv2.imread(abc+".jpg")

def main():

    source_width, source_height = source_image.size
    print 'Image is {}x{}'.format(source_width, source_height)

    target_width = 1000
    target_height = 200


    source_image.thumbnail((target_width, source_height), Image.ANTIALIAS)


    faces = faces_from_pil_image(source_image)
    faces_found_image = draw_faces(source_image, faces)
    faces_found_image.show()


    top_of_faces = top_face_top(faces)
    bottom_of_faces = bottom_face_bottom(faces)

    all_faces_height = bottom_of_faces - top_of_faces
    print 'Faces are {} pixels high'.format(all_faces_height)

    if all_faces_height >= target_width:
        print 'Faces take up more than the final image, you need better logic'
        exit_code = 1
    else:

        face_buffer = 0.5 * (target_height - all_faces_height)
        top_of_crop = int(top_of_faces - face_buffer)
        coords = (0, top_of_crop, target_width, top_of_crop + target_height)
        print 'Cropping to', coords
        final_image = source_image.crop(coords)
        final_image.show()
        exit_code = 0

    return exit_code


def faces_from_pil_image(pil_image):
    "Return a list of (x,y,h,w) tuples for faces detected in the PIL image"
    storage = cv.CreateMemStorage(0)
    facial_features = cv.Load('haarcascade_frontalface_alt.xml', storage=storage)
    cv_im = cv.CreateImageHeader(pil_image.size, cv.IPL_DEPTH_8U, 3)
    cv.SetData(cv_im, pil_image.tostring())
```

```
    faces = cv.HaarDetectObjects(cv_im, facial_features, storage)

    return [f[0] for f in faces]


def top_face_top(faces):
    coords = [f[1] for f in faces]

    return min(coords)


def bottom_face_bottom(faces):

    coords = [f[1] + f[3] for f in faces]
    return max(coords)


def draw_faces(image_, faces):
    "Draw a rectangle around each face discovered"
    image = image_.copy()
    drawable = ImageDraw.Draw(image)

    for x, y, w, h in faces:
        absolute_coords = (x, y, x + w, y + h)

        drawable.rectangle(absolute_coords)
    return image

# load the image and show it



cropped = source_image[160:410, 200:480]
cv2.imwrite(abc+"crop.jpg", cropped)
#cv2.imshow(abc+"crop.jpg", cropped)
median = cv2.medianBlur(cropped,5)
cv2.imwrite(abc+"median.jpg", median)
#cv2.imshow(abc+"median.jpg", median)


#LBP
def thresholded(center, pixels):
    out = []
    for a in pixels:
        if a >= center:
            out.append(1)
        else:
            out.append(0)
    return out

def get_pixel_else_0(l, idx, idy, default=0):
    try:
        return l[idx,idy]
    except IndexError:
        return default

img = cv2.imread(abc+"median.jpg", 0)
transformed_img = cv2.imread(abc+"median.jpg", 0)
```

```
for x in range(0, len(img)):
    for y in range(0, len(img[0])):
        center        = img[x,y]
        top_left      = get_pixel_else_0(img, x-1, y-1)
        top_up        = get_pixel_else_0(img, x, y-1)
        top_right     = get_pixel_else_0(img, x+1, y-1)
        right         = get_pixel_else_0(img, x+1, y )
        left          = get_pixel_else_0(img, x-1, y )
        bottom_left   = get_pixel_else_0(img, x-1, y+1)
        bottom_right  = get_pixel_else_0(img, x+1, y+1)
        bottom_down   = get_pixel_else_0(img, x,   y+1 )

        values = thresholded(center, [top_left, top_up, top_right, right, bottom_right,
                                      bottom_down, bottom_left, left])

        weights = [1, 2, 4, 8, 16, 32, 64, 128]
        res = 0
        for a in range(0, len(values)):
            res += weights[a] * values[a]

        transformed_img.itemset((x,y), res)

   # print x

#cv2.imshow('image', img)
#cv2.imshow('thresholded image', transformed_img)
cv2.imwrite(abc+"lbp.jpg", transformed_img)

hist,bins = np.histogram(img.flatten(),256,[0,256])

cdf = hist.cumsum()
cdf_normalized = cdf * hist.max()/ cdf.max()

plt.plot(cdf_normalized, color = 'b')
plt.hist(transformed_img.flatten(),256,[0,256], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
#plt.show()
plt.savefig(abc+'hist')

ii=-1
ss=""
r=os.path.abspath(os.path.join(abc+'.jpg', os.pardir))
for i in range(0,len(r)):
        if(r[i]== '/'):
                ii=i

for i in range(ii+1,len(r)):
        ss+=r[i]
def ab():
        coeffs = pywt.wavedec2(img,np.ones((4,4)), 'db10')
        cD1 = detcoef(C,S,1)
        cD2 = detcoef(C,S,2)
        cD3 = detcoef(C,S,3)
        cD4 = detcoef(C,S,4)

        mean1 = cD1.mean()
        mean2 = cD2.mean()
        mean3 = cD3.mean()
```

```
        mean4 = cD4.mean()

        variance1=cD1.std()
        variance1=variance1*variance1
        variance2=cD2.std()
        variance2=variance2*variance2
        variance3=cD3.std()
        variance3=variance3*variance3
        variance4=cD4.std()
        variance4=variance4*variance4

        Skew1=cD1.skew()
        Skew2=cD2.skew()
        Skew3=cD3.skew()
        Skew4=cD4.skew()

        Energy1=energy(cD1)
        Energy2=energy(cD2)
        Energy3=energy(cD3)
        Energy4=energy(cD4)

        k1=kurtosis(cD1)
        k2=kurtosis(cD2)
        k3=kurtosis(cD3)
        k4=kurtosis(cD4)

        entr1=entropy(cD1)
        entr2=entropy(cD2)
        entr3=entropy(cD3)
        entr4=entropy(cD4)

text_file = open("In.txt", "w")

if(ss=="Happy"):
        File = open("ha.txt", 'r', 0)
        line = File.readline()[:]
        text_file.write("%s" % line)

elif(ss=="Sadness"):
        File = open("sa.txt", 'r', 0)
        line = File.readline()[:]
        text_file.write("%s" % line)
elif(ss=="Surprise"):
        File = open("su.txt", 'r', 0)
        line = File.readline()[:]
        text_file.write("%s" % line)

elif(ss=="Disgust"):
        File = open("di.txt", 'r', 0)
        line = File.readline()[:]
        text_file.write("%s" % line)

elif(ss=="Repression"):
        File = open("re.txt", 'r', 0)
        line = File.readline()[:]
        text_file.write("%s" % line)

elif(ss=="Fear"):
        File = open("fe.txt", 'r', 0)
```

```
        line = File.readline()[:]
        text_file.write("%s" % line)
text_file.close()



#SVM IMPLEMENTATION

ei=np.loadtxt('In.txt', delimiter=",")

tri=np.loadtxt('input.txt', delimiter=",")

tro=np.loadtxt('output.txt', delimiter=",")

clf = svm.SVC(gamma=0.001, C=100)

d=clf.fit(tri,tro)

c=clf.predict(ei[:])

#print ss
if (c==3):
        print "Happy"

elif (c==1):
        print "Disgust"

elif (c==2):
        print "Fear"

elif (c==4):
        print "Repression"

elif (c==6):
        print "Sadness"

elif (c==5):
        print "Surprise"
```

6. CONCLUSION

Research on the Micro-expression Recognition has witnessed a good deal of progress when compared to that described in the previous survey papers. At those times, a few small-sized datasets of micro-expression existed, And available data was not shared among researchers, micro expression data was rare, and the micro-expression analysis of spontaneous displays of affective behavior seemed to be in a distant future. Today, several large collections of acted affective displays are shared by the researchers in the field, and some data sets of spontaneously displayed expressions have been recently made available. A number of feature methods for image descriptors have been proposed. This project aims to detect the emotions on person`s face who is trying to suppress his emotions. We have used is CASME II dataset. The approach we have used for feature extraction is LBP feature extractor and SVM is used to classify the emotions into 6 classes. Input image is given and lbp has been used to extract his feature and SVM is used to classify and compare results with the actual emotions. The accuracy of our system is 89.43% for CASME II database and SVM classifier.

REFERENCES

[1] X. P. T. H. X. Z. G. &. P. M. Li, " A spontaneous micro-expression database: Inducement, collection and baseline," *In Automatic Face and Gesture Recognition (FG),* vol. 12, no. 3, pp. 1-6, 2013, April.

[2] W. J. W. Q. L. Y. J. W. S. J. &. F. X. Yan, "CASME database: A dataset of spontaneous micro-expressions collected from neutralized faces," *In Automatic Face,* vol. 6, no. 2, pp. 55-76, 2013.

[3] W. J. L. X. W. S. J. Z. G. L. Y. J. C. Y. H. &. F. X. Yan, "CASME II: An improved spontaneous micro-expression database and the baseline evaluation," *ACM on Multimedia Conference,* vol. 9, no. 1, pp. 22-31, 2014.

[4] Y. J. Z. J. K. Y. W. J. W. S. J. Z. G. &. F. X. Liu, " A main directional mean optical flow feature for spontaneous micro-expression recognition," *IEEE Transactions on Affective Computing,* vol. 4, no. 7, pp. 299-310, 2016.

[5] X. W. S. J. Z. G. &. P. M. Huang, "Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection," *IEEE International Conference on Computer Vision Workshops* , vol. 5, no. 2, pp. 1-9, 2015.

[6] I. P. &. V. H. B. Adegun, "Automatic recognition of micro-expressions using local binary patterns on three orthogonal planes and extreme learning machine," *In Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech),* vol. 4, no. 3, pp. 1-5, 2016, November.

[7] D. H. B. W. J. &. R. Y. M. Kim, "Micro-Expression Recognition with Expression-State Constrained Spatio-Temporal Feature Representations," *2016 ACM on Multimedia Conference,* vol. 12, no. 2, pp. 382-386, 2015, October.

[8] X. X. H. O. N. G. M. A. H. X. P. T. Z. G. &. P. M. Li, "Towards Reading Hidden Emotions: A Comparative Study of Spontaneous Micro-expression Spotting and Recognition Methods," *IEEE Transactions on Affective Computing,* vol. 2, no. 1, pp. 23-56, 2017.

[9] Y. S. J. O. Y. H. P. R. C. W. R. Y. L. H. C. .. &. L. X. Wang, "Effective recognition of facial micro-expressions with video motion magnification," *Multimedia Tools and Applications,* vol. 6, no. 2, pp. 1-26, 2016.

[10] S. T. S. J. P. R. C. W. O. Y. H. L. N. A. C. W. K. &. T. S. W. Liong, "Spontaneous subtle expression detection and recognition based on facial strain. Signal Processing," *Image Communication*, vol. 47, no. 1, pp. 170-182, 2016.

[11] Y. &. C. P. Ma, "Event detection using local binary pattern based dynamic textures," *IEEE Computer Society Conference* , vol. 5, no. 2, pp. 38-44, 2016.

[12] A. &. S. V. Suleiman, "Energy-efficient HOG-based object detection at 1080HD 60 fps with multi-scale support," *Signal Processing Systems (SiPS),* vol. 3, no. 1, pp. 1-6, 2014.