Mood Detection based on Facial Expressions

Mrs. Ashwini Pansare[#1], Mrs. Monali Shetty[#2]

[1] *Assistant Professor, Computer Department, Fr. CRCE, Mumbai University, Maharashtra, India*

[2] *Assistant Professor , Computer Department, Fr. CRCE, Mumbai University, Maharashtra, India*

Abstract— *Emotions have an extremely important role in human lives. They determine how humans think, behave and communicate with others and are the only thing which separates us from machines. tone of voice, Gestures, body posture, etc. all express some kind of information about human emotions but the facial features and expressions are one which expresses human emotions clearly and accurately during daily communication. There are many situations in real world where human and computer needs to interact with each other. The interaction between humans and computers will become more natural if computers can perceive and respond to non verbal communication of humans. Therefore there exist need of machines which are able to identify human mood so that a communication bridge can be established between humans and machines and a better interaction will be facilitated.*

This paper proposes a system using Pulse Coupled Neural Network (PCNN) for detecting facial features which are responsible for portraying the facial expression. This information is then passed to a trained Convolution Neural Network (CNN) which is responsible for the classification of expressions in six categories as happy, sad, neutral, fear, angry and surprised.

Keywords — *Human machine interaction, Emotions, PCNN, Facial expression*

I. INTRODUCTION

The human face is an elastic object that consists of organs, numerous muscles, skins and bones. When a muscle contracts, the transformation of the corresponding skin areas attached to the muscle results in facial expressions [1]. These facial expressions are examined for identifying the basic human mood like anger, fear, disgust, surprised, happiness, sadness. The proposed work has many practical application in the field of security, education, medical, game, monitoring, law, marketing, entertainment etc by identifying users response to video games, commercials, or newly launched products, to identify struggling students in a classroom environment, or help autistics better interact with others, to better measure TV ratings, adding another security layer to security at malls, airports, sports arenas, and other public venues to detect malicious intent.[2]

Mood detection is a challenging problem as it involves 3 sub problems 1) face detection 2) facial expression feature extraction and 3) expression classification to identify mood. Each sub problem has difficulties such as background details of input, illumination changes, and variable size of an input and poses variations. The approaches to facial expression recognition can be divided into two classes 1) static image based approaches and image sequence based approaches. Static image based approaches classifies facial expressions based on a single image and the image sequence based approaches use the motion information in an image sequence. In another way, they can be classified into geometric feature based approach[3][4][5] and appearance based approach. The geometric feature based approach relies on the geometric facial features such as locations and contours of eyebrows, nose, mouth etc.Appearance based approach uses whole face or specific region in a face image for a feature extraction via some kind of filters or transformation [1].

A. Approach

The approach used in proposed work is

1) Capture a frame showing the subject’s face from a pre-recorded video or a webcam feed.
2) Process the image frame to standardize it by adjusting the colour-scale, brightness and resolution.
3) Pass the image through the Pulse Coupled Neural Network (PCNN) to extract features of facial expressions.
4) Pass the positional information of the features to a convoluted neural network (CNN) for classification of human mood as happy, sad, surprised, angry ,neutral.

II. METHODOLOGY

In this section, block diagram of a system is discussed. Figure-1 gives the block diagram of proposed mood detection system. The design is divided into two stages:

1) Training stage

2) Testing stage

A. Algorithm

1) Create database by storing images of person showing varied facial expressions.
2) Training stage:
   i) Pre-process the image

ii) Create Pulse Coupled Neural Network (PCNN)
iii) Apply the pre processed image to constructed Pulse Coupled Neural Network (PCNN).
iv) Apply the output of PCNN to Convolution Neural Network (CNN).
v) Continue training the CNN till desired accuracy is achieved

3) Testing stage:
i) Capture user input through camera or retrieve input to be tested from database.
ii) Pre process the test image.
iii) Apply the test image to PCNN used in training stage.
iv) Apply the output of the PCNN to the CNN for predicting the mood.
v) Observe and note down CNN output which is a recognized mood of a person.

B. Block Diagram of System

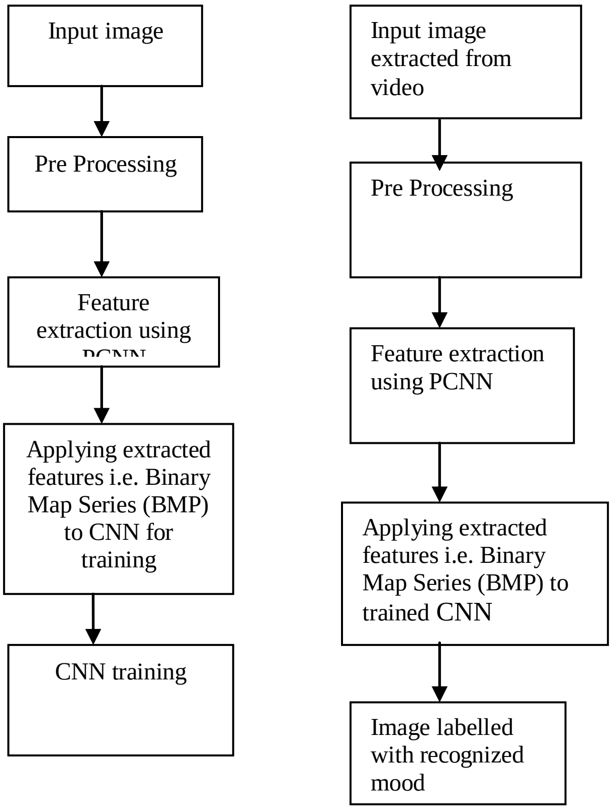

Fig-1 Block diagram of system

1) **Pre processing of input image:**
After extracting selected frames from the video, face of a person is cropped, image is resized and converted to gray scale.

2) **Construction of PCNN:**
PCNN used for image processing is a single layer two dimensional array of laterally linked pulse coupled neurons where all neurons are identical. The number of neurons in the network is equal to the number of pixels in the input image; there exists a one to one relation. The PCNN outputs different Binary maps for every point of time.
The PCNN neuron consists of three parts as shown in Figure 2 the receptive field, the modulation field and the pulse generator.

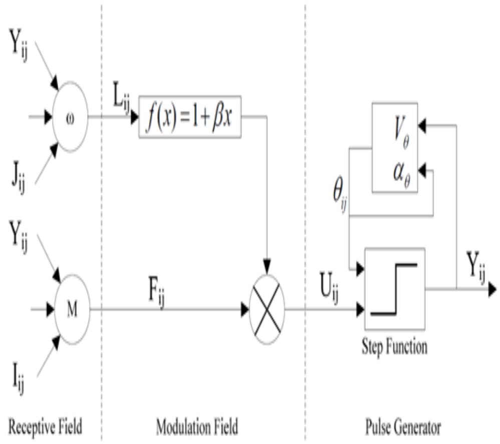

Fig-2 PCNN Neuromime

A PCNN doesn't require any user input. Unlike any other neural network, it does not use stored weights and hence must be compiled every time input is provided to it. The number of neurons in the PCNN has 1:1 proportion with the number of pixels in the image. In PCNN, the similar group of neurons will issues synchronous pulses under effect of mutual coupling pulses. These pulses constitute a 3Dbinary map series(BMS)which effectively describes the information of the edge and regional distribution of the image. But the BMS cannot be directly used because of its large size and hence must be converted to lower dimension object.

PCNN used in proposed work has 100x100 neurons. There is only one layer in PCNN. Output of PCNN is binary map series (BMP) which is then applied to Convolution Neural Network (CNN).

3) **Construction of convolution Neural Network (CNN)**
Convolution Neural Networks and ordinary neural networks are very similar in the sense that both are made up of neurons which have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. Convolution neural network differs from a regular Neural Network as the layers of a CNN have neurons arranged in 3dimensions: width, height, depth (the word depth here refers to the third dimension of an activation volume, not to the depth of a full Neural Network).

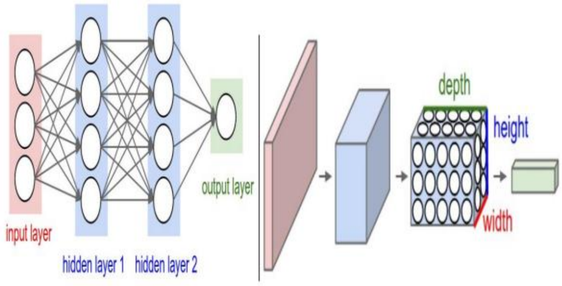

Fig-3 General structure of CNN

*1) **The structure of CNN***

The structure of the CNN can be inferred from Figure 4 .After the input layer, there are nine convolution layers with two pooling layers between two pairs of three convolution layers followed by a fully connected layer connected to the output.

```
# model architecture:
model = Sequential()
model.add(Convolution2D(32, 3, 3, border_mode='same', activation='relu',
                        input_shape=(1, X_train.shape[2], X_train.shape[3])))
model.add(Convolution2D(32, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(32, 3, 3, border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Convolution2D(64, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(64, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(64, 3, 3, border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Convolution2D(128, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(128, 3, 3, border_mode='same', activation='relu'))
model.add(Convolution2D(128, 3, 3, border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(2, activation='softmax'))
```

Fig 4 CNN Model

2) **Input Layer:**

The input layer has pre determined fixed dimensions, proposed work needs image of 48x48 dimensions. So the image must be pre processed before it can be fed into the layer. PCNN quickly finds and crops the face. After that the cropped face is resized to a size of 48x48 using openCV library and is also converted into grey scale. This greatly reduces the dimension compared to the original color dimensions. The pipeline ensures that every image can be fed into the input layer as a(1,48,48) numpy array.

2) **Convolution Layer:**

The number of filters to be used is specified as one of the parameters. The set of filters are unique with randomly generated weights. Each filter,(3,3) receptive field ,slides across the original image with shared weights to create a feature map. The number of filters for the three sets of convolution layers are 32,64,128 respectively. Convolution generates feature maps that represent how pixel values are enhanced.

3) **Pooling Layer:**

Pooling is a technique used for dimension reduction. It is usually applied after one or several convolution layers (proposed work uses three convolution layers).It is an important step while building CNN as adding more convolution layers can greatly affect computation time. Proposed work uses MaxPooling2D technique that uses (2,2) window across the feature map only keeping the maximum pixel value. The pooled pixels form an image with dimensions reduced by 2 across each dimension.

4) **Dense Layer:**

The dense layer receives a number of input features and transformed features through layers which are connected with trainable weights. These weights are trained by forward propagation of data and then backward propagation of their errors. The training speed and the complexity of the architecture can be controlled by adjusting the parameters such as learning rate and network density. Ideally, it is considered to be better to have as many layers as possible but that can lead to over fitting of training data. To prevent this drop out mechanism is used. Drop out randomly selects a portion (here 20%) of the nodes to set their weights to 0 while training. This effectively controls the model's sensitivity to noise during training while maintaining the complexity of the architecture.

5) **Output Layer:**

The CNN outputs the probabilities of the various moods instead of arbitrarily classifying one mood. Instead of using sigmoid activation function proposed work uses softmax activation function.

III. RESULTS

In order to provide input to PCNN, the user has a choice of using webcam feed as input. In this case, the webcam stream gets recorded when user presses key 'C 'and stops when user presses 'Q'. In case the webcam feed is not open, the user can choose to load a stored video file. If even that input is not provided, the system uses the default video. The frames are stored and manipulated using openCV library. Frames from the stream are stored in'./data/stream_images/' as shown in figure 6 and 100x100 cropped versions of the same image are stored in'./data/images' as shown in figure 7. Once the input is loaded ,the method for PCNN is called which in turn calls the CNN method which stores its output as mood labeled images in'./data/output.' as shown in figure 8.For a single image extracted from input video output of PCNN is as shown in figure 5.

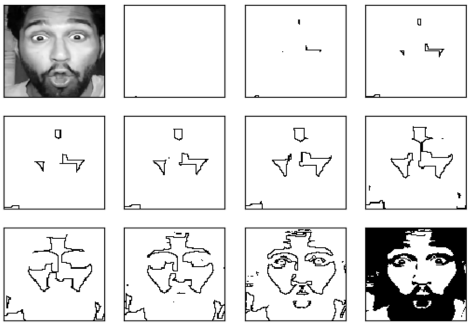

Fig 6- Frames from the video stream stored in /data/stream_images

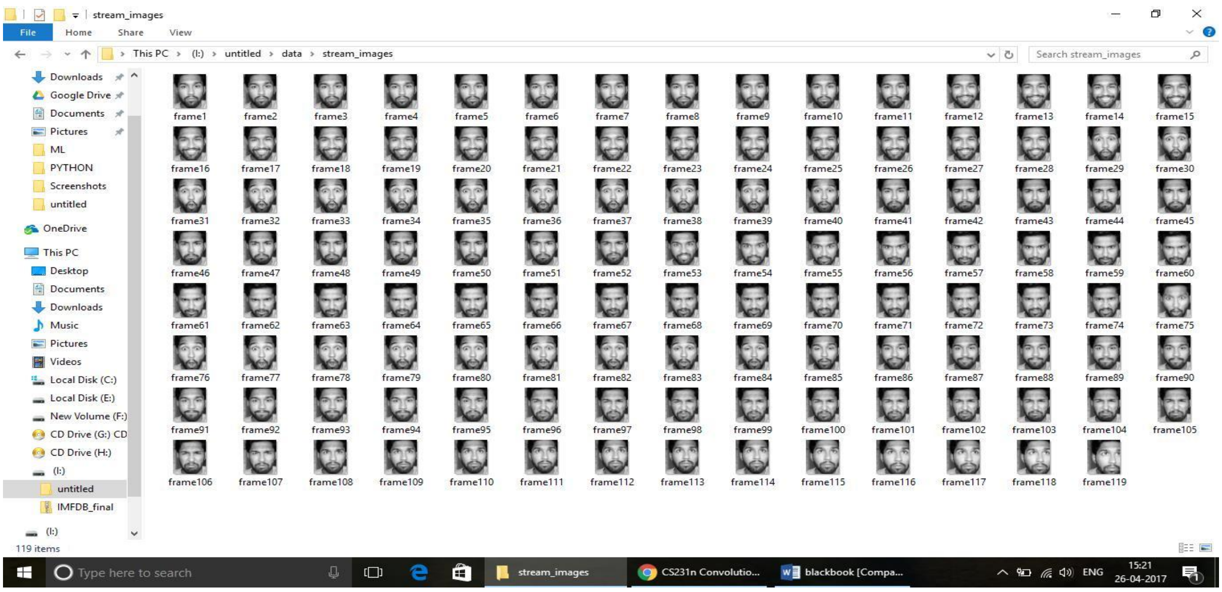

Fig- 7 100x100 cropped versions of the same images stored in'./data/images'

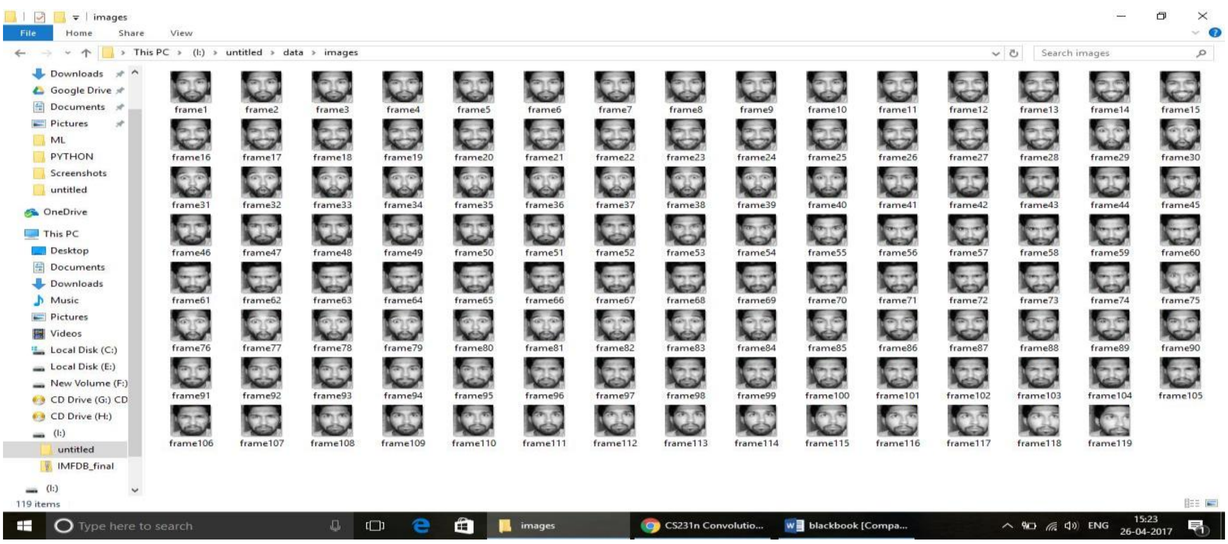

Fig-5 Binary map series (output of PCNN)

IV. CONCLUSIONS

The system was developed in Python and its functionality is based on recognition of facial lines and features gained by the PCNN and classified by neural networks. Eight steps are required in system operation, which does not seem prompt enough and further optimization may need to be taken if it were to be used for video processing. The conducted experiments have shown a 46% to 80% rate of successful recognition that comes down to the average precision of 70%.The efficiency could be improved by increasing the number of samples for individual emotional types and increasing the precision of the facial region detection stage. The issue of universal emotion recognition causes difficulties due to ambivalent psychological and physical characteristics of emotions that are linked to the traits of each person individually. Therefore, this field of research will remain under continuous study for many years to come because many problems remain to be solved in order to create an ideal user interface or at least improve recognition of complex emotional states.

REFERENCES

[1] Mu-chun-su,Chun-kai Yang,shih-Chieh Lin,De- Yuan Huang, Yi-Zeng Hsieh and Pa- Chun Wang,"An SOM-based Automatic facial expression recognition system",Intrernational Journal on Soft computing and Artificial Intelligence applications(IJSCAI), Vol 2, No.4,August 2013

[2] Pranti Dutta, Nachmai M," A Contemplation of facial expression recognition system in real world applications", International Journal of Advances in Electronics and computer science, Vol.2,Issue 9,September 2015.

[3]G.-C.Luh and C.-Y.Lin,"PCA based immune networks for human face recognition," Applied Soft Computing,vol.11,no.2,pp.1743–1752,2011.

[4]G.F.Lu,J.Zou, and Y.Wang,"Incremental complete LDA for face recognition," Pattern Recognition,vol.45,no.7,pp.2510–2521,2012.

[5]I.Marques and M.Gra˜na,"Face recognition with lattice independent component analysis and extreme learningmachines,"Soft Computing,vol.16,no.9,pp.1525–1537,2012.