# Recognition of Face Emotion using Convolutional Neural Network

**Panyam Narahari Sastry, Mohammed Sameer Syed**

***Abstract*: Recognition of face emotion has been a challenging task for many years. This work uses machine learning algorithms for both, a real-time image or a stored database image in the area of facial emotion recognition system. So it is very clear that, deep learning technology becomes important for Human-computer interaction (HCI) applications. The proposed system has two parts, real-time based facial emotion recognition system and also the image based facial emotion recognition system. A Convolutional Neural Network (CNN) model is used to train and test different facial emotion images in this research work. This work was executed successfully using Python 3.7.6 platform. The input Face image of a person was taken using the webcam video stream or from the standard database available for research. The five different facial emotions considered in this work are happy, surprise, angry, sad and neutral. The best recognition accuracy with the proposed system for the webcam video stream is found to be 91.2%, whereas for the input database images is found to be 90.08%.***

***Keywords*: Convolutional Neural Network, Deep Learning, Human Computer Interaction, Machine Learning, Python.**

## I. INTRODUCTION

Deep learning is one of the booming and most electrifying areas in machine learning. With recent advancements in graphics processing unit, it is possible to use Deep Learning for real-time applications. Emotions are an incredibly important aspect of human life, and play an important role in human interaction. Facial expressions represent the emotion of a person and it can give an indication of the emotional response of a person to the interaction with a computer. So detecting facial expressions can help create a better user experience. As computers become increasingly worldwide and their relationship with user changes, they need new tools to obtain feedback from their interactions with those users, and respond accordingly. Nowadays, there are different alternatives to extract feedback from users such as heart rate, tone of voice, body movement, body language, etc. However, some of those alternatives are obtrusive to users; or do not provide enough or accurate feedback in order for a system to be reliable.

**Revised Manuscript Received on July 30, 2020.**
***** Correspondence Author

**Dr. Panyam Narahari Sastry***, Professor, Department of ECE, Chaitanya Bharathi Institute of Technology (A), Hyderabad, India. E-mail: naraharisastry_ece@cbit.ac.in

**Mohammed Sameer Syed,** PG Scholar, Department of ECE, Chaitanya Bharathi Institute of Technology (A), Hyderabad, India. E-mail: mail.sdsameer@gmail.com



Feedback, in the form of user's emotions, offers valuable information that could have a positive impact in different areas such as e-marketing, robotics, smart products, etc. For instance, developers could create a music application that moderately adjusted the kind of music that is being played accordingly to the emotions detected in order that people feeling a negative mood like sad or anger could change those emotions and feel them better.

One unobtrusive alternative that gives a sensible amount of feedback is that the face and particularly the facial expressions. Facial expressions have been considered a good source of information to determine the true emotions of an individual [1]. Even before Charles Darwin conducted "studies on how people recognize emotion in faces" [2], ancient thinkers, such as Aristoteles, already knew the importance of facial expressions. However, it was until Paul Ekman conducted cross cultural experiments around the world that a set of universal emotions, namely surprise, happiness, sadness, fear, anger, and disgust; were finally accepted [4, 5].

In the past, automation of face emotion recognition accurately was unimaginable not only because the computational power was short and costly, but also because of the techniques used executed poorly on image recognition from raw pixels [6]. Although, with the advances in faster Graphic Processing Units (GPUs) and parallelization, the growth of special-purpose machine learning models, and therefore the availability of significant amounts of information, the unimaginable has become possible. To a certain extent, faster GPUs and parallelization are helpful, but getting to a specific answer faster does not guarantee that it is the correct response. Finding the answer which is closer to the correct one or, in other words, learning from data is the purpose of machine learning models. Recently, deep learning that is a part of machine learning is has become popular. Deep learning models have achieved better accuracy than traditional approaches such as SVM or kNN. In the 21st century, HCI products, such as Siri from Apple, Echo from Amazon and Cortana from Windows, became more and more popular in the world. The recent successes of AlphaGo brought machine learning to the world. AlphaGo uses a Monte Carlo tree search algorithm to find its moves based on the knowledge gathering from a pre-train data, which trained by artificial neural network (ANN) [7]. The successful use of machine learning in Go (game) encourages us to sketch a facial emotion recognition system that could be used for HCI and solve facial emotion recognition problem with machine learning. This project is concerned with developing a facial emotion recognition system using a deep learning model, i.e. Convolutional Neural Networks (CNN) in the facial expression domain.

The prototype should analyze images captured in real time via webcam and images from the extended Cohn-Kanade database and display the results.

## II. RELATED WORK

The researchers started doing research in facial emotion recognition area in the 1970s and they had tremendous advances since then. There are many techniques often used for facial expression recognition, such as deep learning [8][9][10], Facial Action Unit Coding System(FACS) [11][12], linear discriminant analysis [13], Local Parametric model, Gaussian Process Classification etc. FACS, which was developed by Ekman and Friesen, has been widely used to describe facial behaviors and help clinical psychologists to categorize the physical expression of emotions. It has also been widely used for facial expression recognition or related fields. Deep Learning is a subfield of machine learning concerned with algorithms that are inspired by an ANN.

CNN is a kind of feed-forward ANN, where the connectivity pattern between its neurons is encouraged by the organization of the animal visual cortex. It is attractive for many deep learning tasks like image classification, scene recognition, and natural language processing. In 1962, Hubel and Wiesel [14] showed some individual neuronal cells in the brain responded only in the presence of edges of certain orientation in an experiment.

They found that a columnar architecture was organizing those neurons and those neurons were able to produce visual perception together. CNN accepts input data, such as audio, video and image, with an optional dimension as the input layer and generates an output layer that has a vector of highly distinguishable features related to the pre-classified category. As a type of supervised learning, CNN used labeled training data which consist a set of training examples [15]. Like the many other supervised learning algorithms, CNN scans the training data and concluded a method that can be used for mapping new examples.

CNN normally requires a large amount of training data and the inference accuracy can be improved by adopting more data. The inference accuracy in CNN also can be improved by adopting deeper and wider network. However, larger training or/and more complicated models will result in longer training time. There are a few distinct types of layers used commonly in CNNs [16], such as convolutional layer, pooling layer, ReLu (Rectified Linear Units) layer, fully connected layer and loss layer. Convolutional layer is the fundamental building block of a CNN based algorithm and it can be applied a varied number of times based on the given methodology. The primary purpose of a convolution layer is to extract features from the input image. Pooling layer is a form of non-linear down-sampling.

There is two common type of pooling layers: max-pooling layer and average-pooling layer. It partitions the input images into a set of non-overlapping N by N rectangles, N can be any divisor of the image size and outputs the maximum/average value from each sub-region. Max-pooling works better on highlighting images. ReLU is one of the most popular types of nonlinearity to use in neural networks that is applied after the convolutional layer and before max pooling. It replaces all negative pixel values in the feature map by zero. It normally used after convolutional layer. The fully connected layers that is only applied at the end of the network, take an input volume which is output from the previous process and outputs an N dimensional vector where N is the number of classes that the program had to select. It is one of the economical ways of learning non-linear combinations of these features. OpenCV is an open source computer vision library that is designed to obtain computational efficiency and has a strong focus on real-time applications [17].

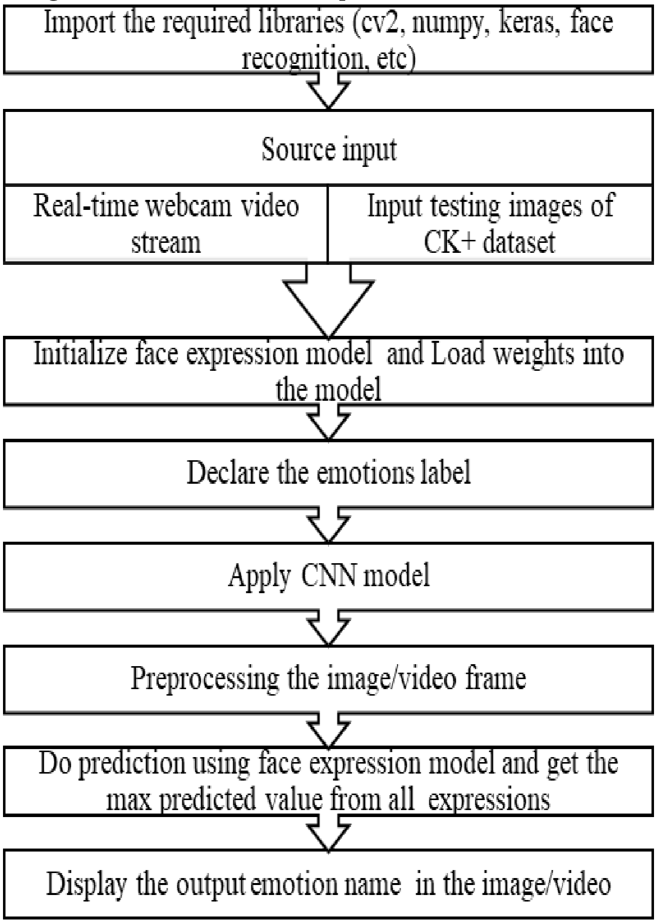
OpenCV has been used in a lot of computer vision application, such as monitoring system [18], object tracking [19], security detection system, medical image noise reduction and manufacturing inspection system. It was also used to design a facial detector. OpenCV contains a cascade classifier function. A cascade classifier is an object detection framework to issue competitive object detection rates in real-time that was initiated in 2001 by Paul Viola and Michael Jones [20] and improved by Rainer Lienhart [21]. It can be used for different types of object classification, but it was motivated primarily by the problem of face detection. Cascade classifier has a very high detection rate (true-positive rate) and a very low false-positive rate. In our system, we needed facial detection and image cropping process on frame every 0.5 seconds. OpenCV cascade classifier can process instantly to meet our requirement. TensorFlow [22] is an open source machine learning library that allows user to deploy computation to signed processing units with a single API. The TensorFlow provides a high-level API for different types of layers in neural network, such as convolutional layer, pooling layer and fully connected layer. It also provides methods that adding activation function and applying dropout regularization. Tong Zhang, et al. proposed a novel deep neural network (DNN) driven feature learning method for the multi-view facial expression recognition (FER) in 2016 [8]. DNN is an ANN with the multiple hidden layers of units between the input and the output layers. In their method, the scale invariant feature transform (SIFT) features corresponding to a set of landmark points are first extracted from each facial image. The feature matrix which is consisting of the extracted SIFT feature vectors is used as the data input for the DNN model to learning optimal discriminative features for expression classification. Their DNN model engages several layers to characterize the corresponding relationship between the SIFT feature vectors and their corresponding high-level semantic information. They used two non-frontal facial expression databases, namely BU-3DFE and Multi-PIE to evaluate the effectiveness of their method. The experimental results show that their algorithm outperforms the state-of-the-art methods. Their method achieves 80.1%. Yanan Guo, et al. proposed a scheme termed Deep Neural Networks with Relativity Learning (DNNRL) for Facial Expression Recognition in 2016 [9]. According to sample importance, DNNRL treats samples differently. DNNRL consists of three convolutional layers, four pooling layers, three Inception layers, and one fully connected layer. By using the exponential triplet loss, DNNRL can directly learn a mapping from original images to a Euclidean space, where relative distances correspond to a measure of facial expression similarity. In their experiment, they used the FER2013 dataset and the SFEW2.0 [23] dataset to demonstrate the effectiveness of their proposed method.

They had overall 69.85% accuracy on FER2013 test set which is only trained on FER2013 training set and 73.71% accuracy on FER2013 test set which is trained on both FER2013 training set and SFEW2.0.

Peter Burkert, et al. proposed a convolutional neural network (CNN) architecture for facial expression recognition in 2015 [10].The proposed architecture is independent of any hand-crafted feature extraction and performs better than the earlier proposed convolutional neural network based approaches. Visualizing the automatically extracted features which have been learned by the network to provide a better understanding of those features. They used Extended CK and MMI Facial Expression Database for the quantitative evaluation. On the CK set, the current state of the art approach, using CNN achieves an accuracy of 99.2%. For the MMI dataset, currently the best accuracy for emotion recognition is 93.33%. The proposed architecture achieves 99.6% for CK and 98.63% for MMI.

## III. METHODOLOGY

The proposed method for recognition of facial emotion using CNN includes various steps:




### A. *Software Requirement*

This project was executed successfully through Python 3.7.6 version in the spyder Integrated Development Environment (IDE) of Anaconda Navigator. Python comes with a high amount of inbuilt libraries. Many of the libraries are for Artificial Intelligence and Machine Learning. Python is widely used in scientific and research communities because it's easy to experiment with new ideas and code prototypes quickly in a language with minimal syntax.

The open source Anaconda Distribution is the easiest way to perform Python Data Science and Machine Learning on Linux, Windows and Mac OSX.

*Installing CMake in Anaconda*: CMake means Cross Platform Make. It is a free and open source software tool for building software using a compiler-independent method. It is used to compile and install Deep learning library (Dlib). CMake version 3.16.3 is used in this work. Open Anaconda prompt and issue command "pip install cmake" to install CMake library.

*Installing OpenCV in Anaconda*: OpenCV is open source computer vision library which can be used for real time computer vision or image processing related activities. OpenCV version 4.2.0 is used in this work. Open Anaconda prompt to issue command "pip install opencv-python" to install OpenCV library.

*Installing Dlib in Anaconda*: Dlib means Deep Learning library. It is a C++ tool kit containing machine learning algorithms and tools. It is used for face mapping, face detection and face recognition. The frontal face detector in Dlib is simple, fast and powerful. Dlib version 19.19.0 is used in this work. Open Anaconda prompt and issue command "pip install dlib" to install Dlib library.

*Installing Face Recognition in Anaconda*: It is created by Adam Geitgey. It recognizes faces from python or from command line. It is built using dlib's face recognition feature. It is built with deep learning model with accuracy of 99.38%. Face recognition version 1.2.3 is used in this work. Open Anaconda prompt and issue command "pip install face-recognition" to install face recognition library.

*Installing numpy in Anaconda:* It is used to deal with number based multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Numpy version 1.18.1 is used in this work. Open Anaconda prompt and issue command "pip install numpy" to install numpy library.

*Installing TensorFlow in Anaconda*: TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. TensorFlow version 1.14.0 is used in this work. Open Anaconda prompt and issue command "conda install tensorflow" to install TensorFlow library.

*Installing Keras in Anaconda*: Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Keras version 2.3.1 is used in this work. Open Anaconda prompt and issue command "conda install keras" to install Keras library.

Face detection is the process of automatically locating human faces in virtual media (digital images or video). There are two types of face detector image-recognition libraries that is HoG (Histogram of oriented Gradients) face detector and CNN (Convolutional Neural Network) based face detector which is included in Dlib.

*CNN face detector* is used in the proposed method.

It uses object detector with CNN based features. The training process is simple and no need to give large amount of training data. It can detect multiple face orientations and works with medium occlusion. It is fast on GPU but very slow on CPU.

### B. *Data Collection*

In the proposed method, the training dataset consists of 28,709 facial images taken from FER2013 database and converted to csv file having the pixel values. This training dataset is comprised of 48x48 pixel grayscale images of faces. The testing dataset of 625 images is considered from the Extended Cohn-Kanade (CK+) database. The 5 facial emotions namely happy, neutral, surprise, angry and sad are to be recognized in the proposed method. Each emotions dataset consists of 125 images .This testing dataset is comprised of 640x490 pixel grayscale images of faces.

### C. *Implementation*

Import all the required libraries such as numpy, cv2, face recognition and keras. Load the image to detect. Load the training dataset that is the facial expression model structure which is loaded along with the weights model. Declare the emotions label in an array, that is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). Detect all the faces in the image, up sample and then apply the CNN model to get the all face locations. Print the number of faces in the input image. Looping through the face locations and the four position values of each face are top, right, bottom and left positions. Print locations of each faces inside the loop. Extract the face image from the main image and a rectangular frame is drawn along the face area to show that the face is detected. Preprocessing main input image(s) like training dataset. Then convert to grayscale (BGR2GRAY), resized (48x48 pixels). Then convert this PIL image into 3d numpy array so that the numpy array will be in the pixels format. Later, expand that shape of an array into single row multiple columns. Now the pixels are in range of [0, 255]. So divide all the pixel values by 255 to be normalized and get all pixels in scale of [0, 1]. Do the prediction using facial expression prediction and get the prediction values for all the emotions. Find the maximum indexed prediction value and equalize it to the corresponding emotion label. Finally, the output is displayed as text in the bottom left corner of the rectangular face detection frame. The algorithm for the input webcam video stream is same as mentioned above but with a few changes.

### D. *CNN Model*

In this model, there are three sequential convolutional layers followed by a maxpooling layer, rectified activation function for convolutional layer "relu" is used, and the same padding pattern. The basic structure of this model is followed as: Input >> Conv2D >> MaxPooling2D >> Conv2D >> Conv2D >> AveragePooling2D >> Conv2D >> Conv2D >> AveragePooling2D >> Flatten >> Dense >> Dropout >> Dense >> Dropout >> Dense >> Output.

So, from the above basic structure of CNN model, we added more and more convolutional layers with different features captured. The features to be captured from convolutional layer increased from 32 to 128, it is recommended that such hierarchical structure that is with the increase in layer nodes, performs better for deep neural network. Finally, the convolved layer is first flattened and then goes through two more dense layers to reach the output layer in which softmax activation function is used for multiclass classification. This CNN model written in code is as follows:

```
model = Sequential()
#1st convolution layer
model.add(Conv2D(32, (5, 5), activation='relu',
input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))
#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2,
2)))
#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2,
2)))
model.add(Flatten())
#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
```

A pooling layer is a building block of CNN and that operates on each feature map independently. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Max pooling is a sample-based discretization process. Average pooling involves calculating the average for each patch of the feature map which is down sampled to the average value in the square. Rectified linear unit (ReLu), is a type of activation function most commonly used in CNNs. ReLU overcomes the vanishing gradient problem, allowing models to learn faster and perform better. Padding is a term appropriate to convolutional neural networks as it refers to the amount of pixels added to an image when it is being processed by the kernel of a CNN. For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero. Softmax layer is generally the final output layer in a neural network that executes multi-class classification. This comes from the softmax function that takes as input a number of scores values (), and compresses them into values in the range between 0, and 1 whose sum is 1.

## IV. RESULTS AND DISCUSSIONS

All the experiments are performed over a system having a 64 bit operating system, and Intel (R) Core(TM) i7-8750H CPU @ 4.41 GHz and were done through Python 3.7.6. Anaconda Navigator is installed which is a desktop graphical user interface that allows to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Spyder, the Scientific Python Development Environment, is a free Integrated Development Environment (IDE) that is preinstalled in the Anaconda Navigator. Spyder includes editing, interactive testing, debugging and introspection features. The facial emotion recognition code is written and executed in Spyder.

***Results I – for the input images of CK+ dataset***

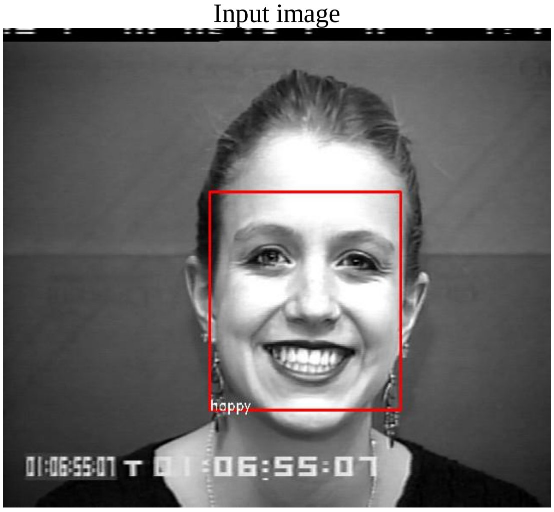
The table 1 shows the results obtained for the recognition of happy emotion of 125 testing images from CK+ database. The testing samples were varied and different recognition accuracies (R.A) were obtained. The R.A obtained for all the given 125 testing images was 96.8%. The average recognition accuracy of the varied testing samples for happy emotion had obtained 98.69%.

**Table- 1: Experiment 1 for Happy Emotion Recognition**

| S.no. | No. of testing samples | Recognition Accuracy (%) |
|---|---|---|
| 1 | 25 | 100 |
| 2 | 50 | 100 |
| 3 | 75 | 98.66 |
| 4 | 100 | 98 |
| 5 | 125 | 96.8 |

The Fig.1 and Fig.2 shows the correctly identified and wrongly identified happy emotion images using the proposed algorithm. From Fig.1, it was observed that in the output image the face was detected and the happy emotion had correctly identified. From Fig.2, it was observed that the recognized emotion had wrongly identified as neutral instead of happy.

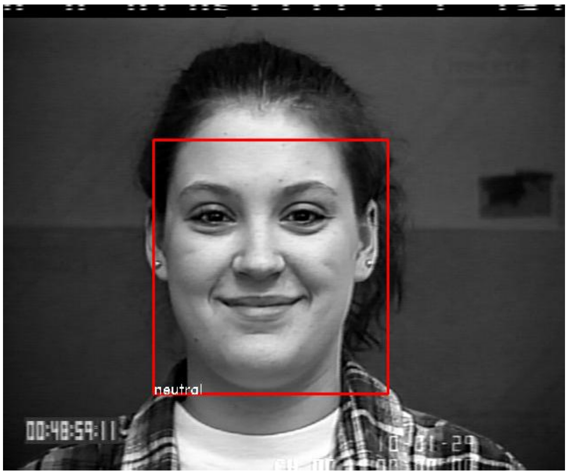
Input image




Output image

**Fig.1. Correctly identified happy emotion**

Input image




Output image

**Fig.2. Wrongly identified happy emotion**

The recognition accuracies from table 1 was represented graphically by a bar graph in Fig.3, we can observe that the R.A was highest for both the data of 25 and 50 number of testing samples.

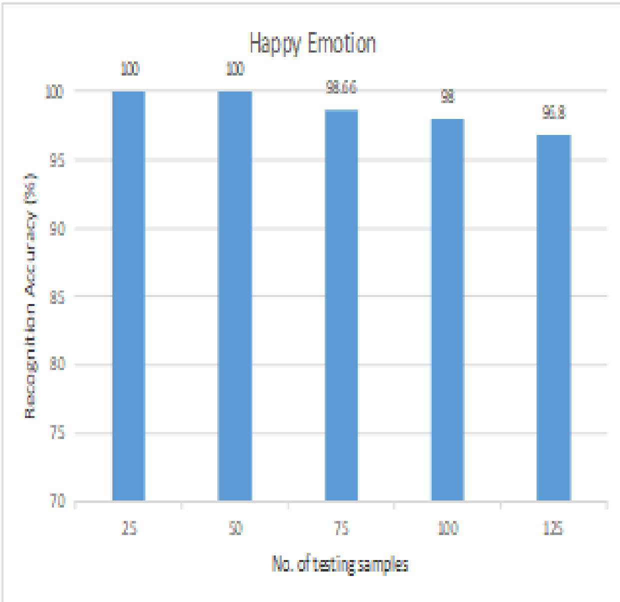



**Fig. 3. Graphical representation of experiment 1 for Happy Emotion Recognition**

The table 2 shows the results obtained for the recognition of neutral emotion of 125 testing images from CK+ database. The testing samples were varied and different recognition accuracies (R.A) were obtained. The R.A obtained for all the given 125 testing images was 92%. The average recognition accuracy of the varied testing samples for neutral emotion had obtained 91.93%.

**Table- 2: Experiment 2 for Neutral Emotion Recognition**

| S.no. | No. of testing samples | Recognition Accuracy (%) |
|---|---|---|
| 1 | 25 | 88 |
| 2 | 50 | 92 |
| 3 | 75 | 94.66 |
| 4 | 100 | 93 |
| 5 | 125 | 92 |

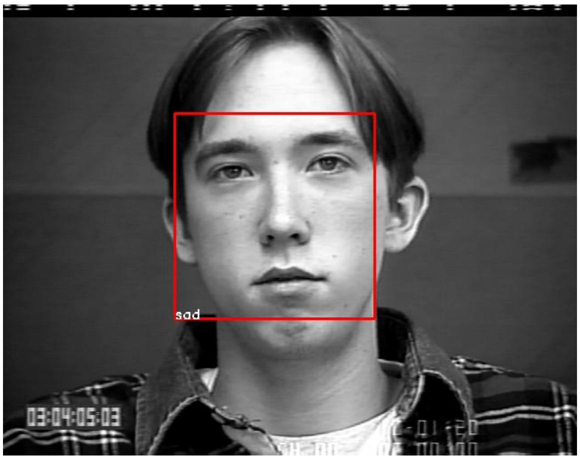
The Fig.4 and Fig.5 shows the correctly identified and wrongly identified neutral emotion images using the proposed algorithm. From Fig.4, it was observed that in the output image the face was detected and the neutral emotion had correctly identified. From Fig.5, it was observed that the recognized emotion had wrongly identified as sad instead of neutral.


Input image


Output image

**Fig.4. Correctly identified neutral emotion**


Input image




Output image

**Fig.5. Wrongly identified neutral emotion**

The recognition accuracies from table 2 was represented graphically by a bar graph in Fig.6, we can observe that the R.A was highest for the data of 75 number of testing samples.

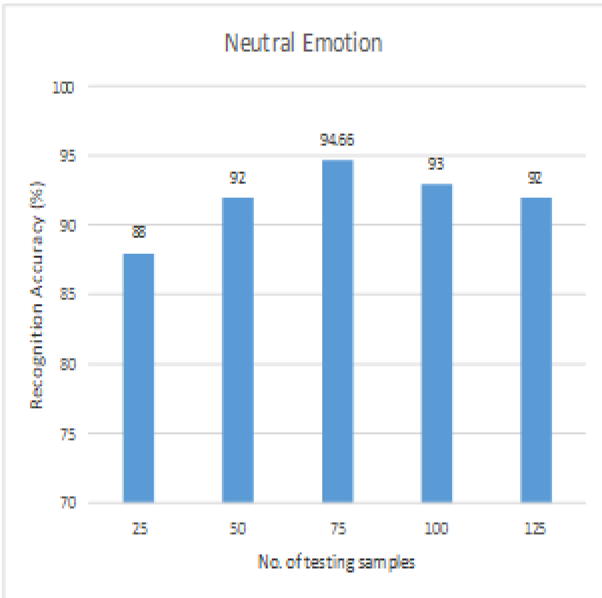



**Fig. 6. Graphical representation of experiment 2 for Neutral Emotion Recognition**

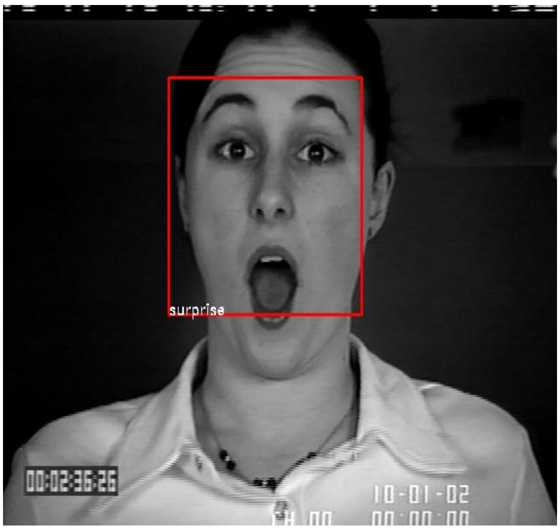
The table 3 shows the results obtained for the recognition of surprise emotion of 125 testing images from CK+ database. The testing samples were varied and different recognition accuracies (R.A) were obtained. The R.A obtained for all the given 125 testing images was 90.4%. The average recognition accuracy of the varied testing samples for surprise emotion had obtained 92.61%.

**Table- 3: Experiment 3 for Surprise Emotion Recognition**

| S.no. | No. of testing samples | Recognition Accuracy (%) |
|---|---|---|
| 1 | 25 | 96 |
| 2 | 50 | 94 |
| 3 | 75 | 90.66 |
| 4 | 100 | 92 |
| 5 | 125 | 90.4 |

The Fig.7 and Fig.8 shows the correctly identified and wrongly identified surprise emotion images using the proposed algorithm. From Fig.7, it was observed that in the output image the face was detected and the surprise emotion had correctly identified. From Fig.8, it was observed that the recognized emotion had wrongly identified as neutral instead of surprise.


Input image



Output image

**Fig.7. Correctly identified surprise emotion**


Input image

Output image

**Fig.8. Wrongly identified surprise emotion**

The recognition accuracies from table 3 was represented graphically by a bar graph in Fig.9, we can observe that the R.A was highest for the data of 25 number of testing samples.

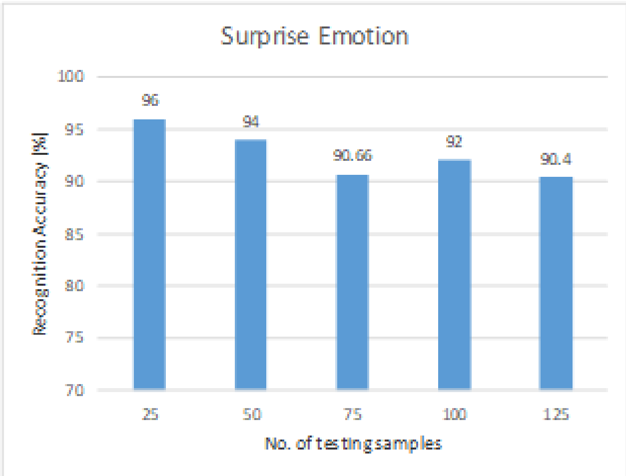



**Fig. 9. Graphical representation of experiment 3 for Surprise Emotion Recognition**

The table 4 shows the results obtained for the recognition of angry emotion of 125 testing images from CK+ database. The testing samples were varied and different recognition accuracies (R.A) were obtained. The R.A obtained for all the given 125 testing images was 87.2%. The average recognition accuracy of the varied testing samples for angry emotion had obtained 86.5%.

**Table- 4: Experiment 4 for Angry Emotion Recognition**

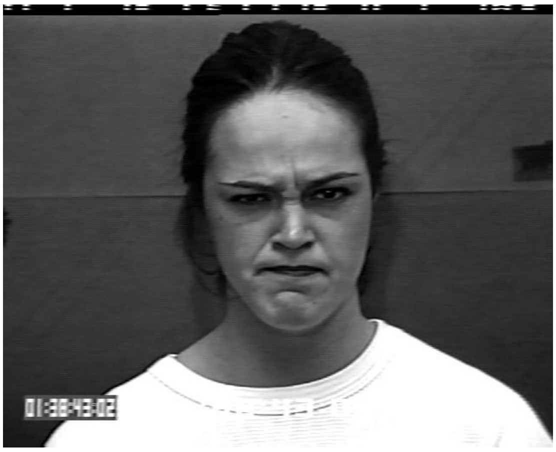
| S.no. | No. of testing samples | Recognition Accuracy (%) |
|---|---|---|
| 1 | 25 | 84 |
| 2 | 50 | 86 |
| 3 | 75 | 89.33 |
| 4 | 100 | 86 |
| 5 | 125 | 87.2 |

The Fig.10 and Fig.11 shows the correctly identified and wrongly identified angry emotion images using the proposed algorithm. From Fig.10, it was observed that in the output image the face was detected and the angry emotion had correctly identified. From Fig.11, it was observed that the recognized emotion had wrongly identified as sad instead of angry.

Input image


Output image

**Fig.10. Correctly identified angry emotion**



Input image


Output image

**Fig.11. Wrongly identified angry emotion**

The recognition accuracies from table 4 was represented graphically by a bar graph in Fig.12, we can observe that the R.A was highest for the data of 75 number of testing samples.

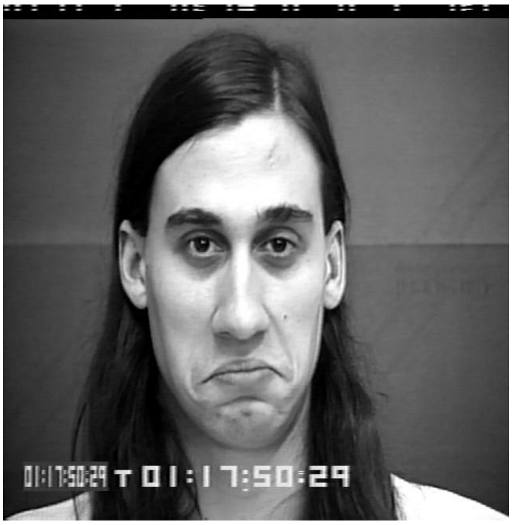

**Fig. 12. Graphical representation of experiment 4 for Angry Emotion Recognition**

The table 5 shows the results obtained for the recognition of sad emotion of 125 testing images from CK+ database. The testing samples were varied and different recognition accuracies (R.A) were obtained. The R.A obtained for all the given 125 testing images was 84%. The average recognition accuracy of the varied testing samples for sad emotion had obtained 87.8%.

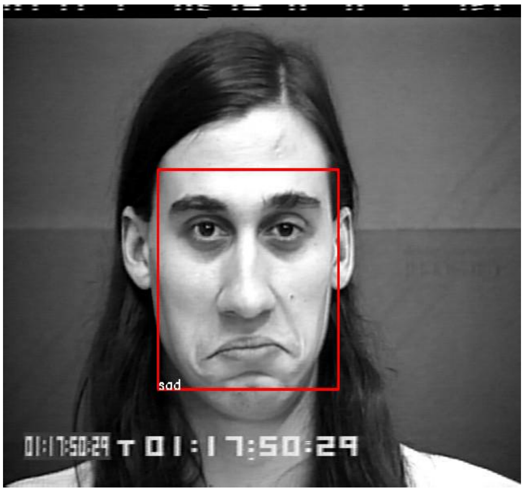
**Table- 5: Experiment 5 for Sad Emotion Recognition**

| S.no. | No. of testing samples | Recognition Accuracy (%) |
|---|---|---|
| 1 | 25 | 96 |
| 2 | 50 | 90 |
| 3 | 75 | 84 |
| 4 | 100 | 85 |
| 5 | 125 | 84 |

The Fig.13 and Fig.14 shows the correctly identified and wrongly identified sad emotion images using the proposed algorithm. From Fig.13, it was observed that in the output image the face was detected and the sad emotion had correctly identified. From Fig.14, it was observed that the recognized emotion had wrongly identified as neutral instead of sad.

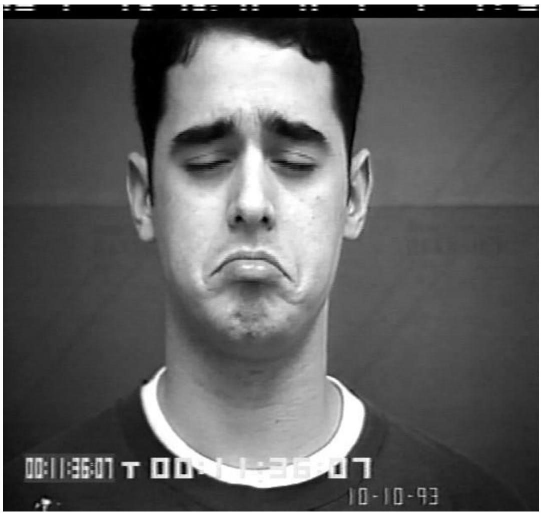



Input image




Output image

**Fig.13. Correctly identified sad emotion**




Input image


Output image

**Fig.14. Wrongly identified sad emotion**

The recognition accuracies from table 5 was represented graphically by a bar graph in Fig.15, we can observe that the R.A was highest for the data of 25 number of testing samples.


**Fig. 15. Graphical representation of experiment 5 for Sad Recognition**

The table 6 shows the results obtained for the average recognition accuracy of each emotion from CK+ database. Since there are 5 emotions that are recognized with each of 125 testing dataset. So, the total recognition accuracy of all the 5 facial emotions of the 625 testing images (correctly recognized testing images = 563) from CK+ database obtained was 90.08%. Also, the overall average recognition accuracy from table 6 obtained was 91.5%.

**Table- 6: Average Recognition Accuracy of Each Emotion**

| S.no. | Name of the Emotion | Average Recognition Accuracy (%) |
|---|---|---|
| 1 | Happy | 98.69 |
| 2 | Surprise | 92.61 |
| 3 | Neutral | 91.93 |
| 4 | Sad | 87.8 |
| 5 | Angry | 86.5 |

The average recognition accuracies from table 6 was represented graphically by a bar graph in Fig.16, it is observed that the average R.A was highest for happy emotion and the lowest for angry emotion. Also, the Fig.16 represents a decreasing graph that is gradually decreasing from happy and followed by surprise, neutral, sad and angry respectively.

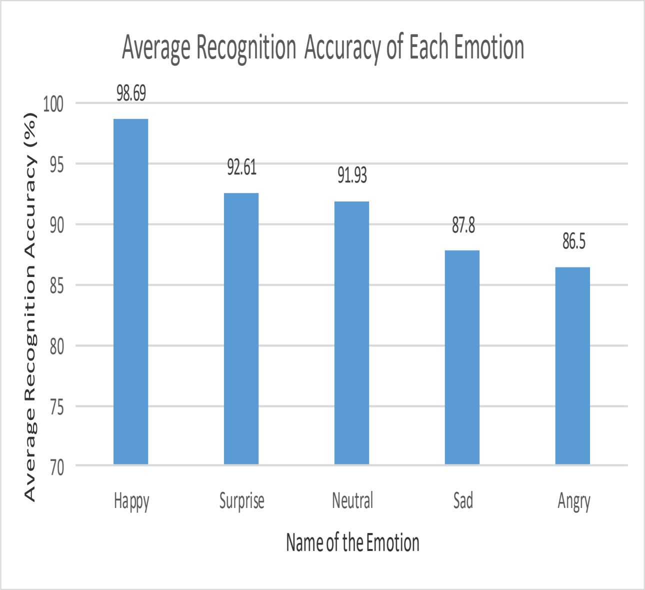



**Fig. 16. Graphical representation of Average Recognition Accuracy of Each Emotion**

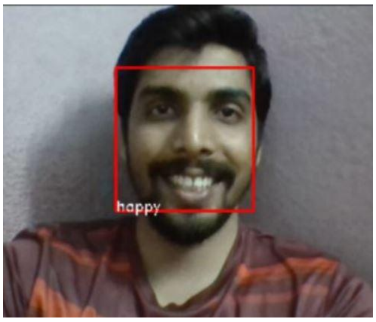
From the table 7, it is observed that the database used by the proposed method is more and comparatively greater number of training samples as well as in the testing samples. The average recognition accuracy for the published method was observed to be 85% and 90% for the SVM and Softmax classifier respectively. On the other hand, the average recognition accuracy about 91.5% has been observed for the proposed method.

**Table- 7: Comparison between published and proposed method** [24]

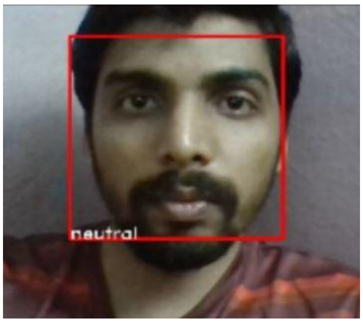
| s.no | Description | Published method | Proposed method |
|---|---|---|---|
| 1 | No. of emotions | 6 | 5 |
| 2 | Name of the database | CK+ | CK+ |
| 3 | Feature extraction | Local Binary Pattern(LBP) | 68 landmark feature predictor |
| 4 | No. of training samples | 600 | 28,709 |
| 5 | No. of testing samples | 300 | 625 |
| 6 | Classifier | SVM and Softmax | CNN |
| 7 | Recognition Accuracy | 85% & 90% | 90.08% |

***Results II– for the input webcam video stream***

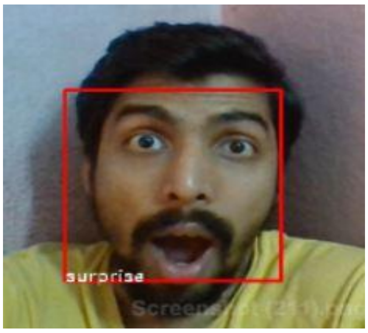
In the previous results, the input source was an image but here the input source is the webcam video stream. The whole procedure is same as mentioned in the previous results except the input source is the webcam instead of an image. The output recognizes the emotions of human face dynamically. The emotions considered in this work are happy, surprise, neutral, sad and angry. For each emotion 25 tests have been done, that is 125 tests for all five emotions. The following are some of the sample outputs that have recognized the emotions of the human face correctly.




(a)

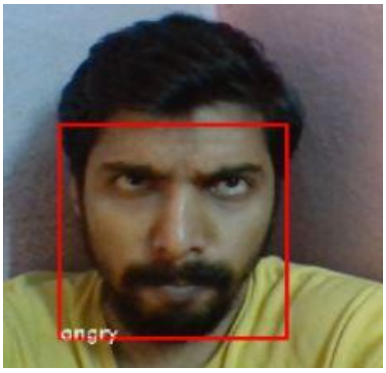



(b)



(c)

(d)



(e)

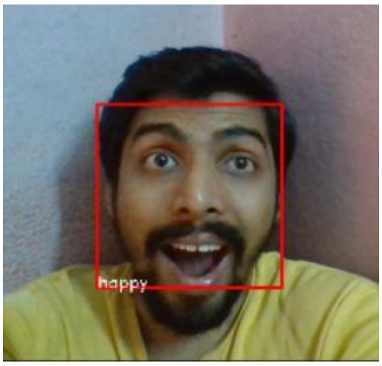
**Fig.17. Correctly recognized emotions from the webcam that represents (a) happy, (b) neutral, (c) surprise, (d) sad and (e) angry**

The following are some of the sample outputs that have recognized the emotions of the human face incorrectly:

(f)

(g)

(h)



(i)

(j)

Fig.18. Incorrectly recognized emotions from the webcam that the image (f) is actually happy emotion but is recognized as neutral, the image (g) is actually neutral emotion but is recognized as sad, the image (h) is actually sad emotion but is recognized as neutral, the image (i) is actually surprise emotion but is recognized as happy, the image (j) is actually angry emotion but is recognized as sad.

The table 8 shows the results obtained for the real-time recognition accuracy of each emotion from webcam video stream. This work included 5 emotions with each of 25 tests. So, the total recognition accuracy of all the 5 facial emotions of the 125 tests (correctly recognized emotions = 114) obtained was **91.2%**.

**Table-8: Real-time Recognition Accuracy of Each Emotion**

| S.no. | Name of the Emotion | No. of testing samples | Recognition Accuracy (%) |
|---|---|---|---|
| 1 | Happy | 25 | 96 |
| 2 | Neutral | 25 | 96 |
| 3 | Surprise | 25 | 92 |
| 4 | Sad | 25 | 88 |
| 5 | Angry | 25 | 84 |

The real-time recognition accuracies of each emotion from table 8 was represented graphically by a bar graph in Fig.19, it is observed that the real-time recognition accuracy was highest for both happy and neutral emotions and the lowest for angry emotion.

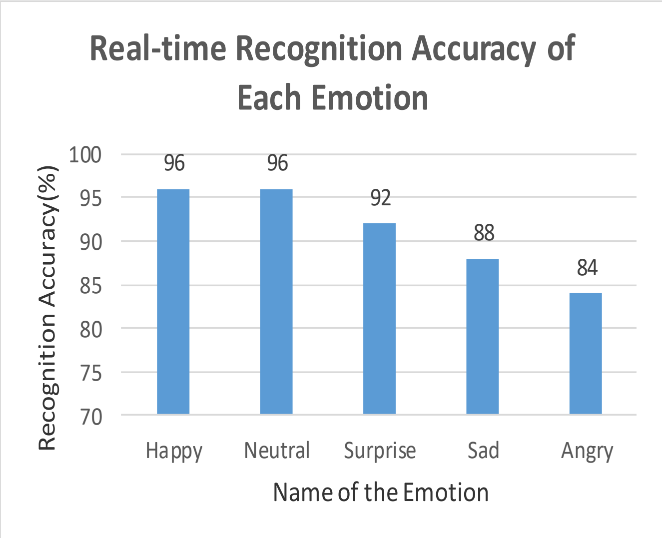



**Fig. 19. Graphical representation of Real-time Recognition Accuracy of Each Emotion**

## V. CONCLUSIONS

In this work, the training dataset consisted of 28,709 face images (emotions) considered from a standard data base namely FER2013. The testing dataset of 625 images were created from the Extended Cohn-Kanade (CK+) database successfully. This testing data set is composed of 5 emotions namely Happy, Neutral, Surprise, Sad and Angry, each of 125 samples. The average recognition accuracy (RA) obtained for **"Happy"** emotion of the 125 testing samples is **98.69%,** where as for **"Neutral"** emotion is **91.93%,** using python 3.7.6 of Anaconda Navigator. Similarly, the average RA for **"Surprise"** emotion is **92.61%,** for **"Sad"** emotion is **87.8%** and for the **"Angry"** emotion it is found to be **86.5%.** The Happy emotion gave the highest RA whereas the Angry emotion gave the least RA. The **overall average RA** of all the **5 emotions** i.e. 625 test images (125 images of each emotion) considered together was found to be **91.5%**. These results are very encouraging for various applications.

The **real-time** (dynamic type) human face emotion recognition results **using a Webcam** are also a part of the present work. Even in this case the same 5 emotions namely Happy, Neutral, Surprise, Sad and Angry are considered. The RA for **Happy** and **Neutral** emotion is found to be **96%,** where as for **Surprise** emotion it is **92%**. The RA for **Sad** emotion is **88%** and for **Angry** emotion it is obtained as **84%**.

The **Overall RA** when all these emotions were considered together, it is obtained as **91.2%**.

Finally it is concluded that, these proposed methods of emotion recognition gave RA **greater than 90%** which is suitable to many applications. In the future work, these algorithms can be used to make a product for different applications. Researchers can try and develop the model so that many standard face emotion recognition systems can be proposed. In addition, a mass view of the face emotion by gathering different emotions as well as compound emotions could be determined by using advancing technology.

## REFERENCES

1. Ekman, P. and Rosenberg, E. (1997). What the face reveals. New York: Oxford University Press.
2. Jabr, Ferris. (2010) "The Evolution of Emotion: Charles Darwin's Little-Known Psychology Experiment". Scientific American Blog Network, 2 Mar. 2016.
3. Anima Majumder, Laxmidhar Behera, Venkatesh K. Subramanian, "Automatic Facial Expression Recognition System Using Deep Network-Based Data Fusion", IEEE TRANSACTIONS ON CYBERNETICS, VOL. 48, NO. 1, 2018.
4. Bettadapura, Vinay. (n.d.) "Face Expression Recognition and Analysis: The State Of The Art". arXiv: 1203.6722 [cs.CV]
5. Siegman, Aron Wolfe, and Stanley Feldstein. Nonverbal Behavior and Communication. Chapter 4. Facial Expression. Hillsdale, N.J.: L. Erlbaum Associates, 1978.
6. LeCun, Y. and Huang, F. (n.d.). Large-scale Learning with SVM and Convolutional Nets for Generic Object Categorization.
7. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. Nature 529, 7587 (2016), 484–489.
8. Zhang, T., Zheng, W., Cui, Z., Zong, Y., Yan, J., and Yan, K. A deep neural network driven feature learning method for multi-view facial expression recognition. IEEE Trans. Multimed 99 (2016).
9. Guo, Y., Tao, D., Yu, J., Xiong, H., Li, Y., and Tao, D. Deep neural networks with relativity learning for facial expression recognition. In Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on (2016), IEEE, pp. 1–6.
10. Burkert, P., Trier, F., Afzal, M. Z., Dengel, A., and Liwicki, M. Dexpression: Deep convolutional neural network for expression recognition. arXiv preprint arXiv:1509.05371 (2015).
11. Gavrilescu, M. Proposed architecture of a fully integrated modular neural network-based automatic facial emotion recognition system based on facial action coding system. In Communications (COMM), 2014 10th International Conference on (2014), IEEE.
12. Lien, J. J., Kanade, T., Cohn, J. F., and Li, C.-C. Automated facial expression recognition based on facs action units. In Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on (1998), IEEE, pp. 390–395.
13. Lyons, M. J., Budynek, J., and Akamatsu, S. Automatic classification of single facial images. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 12 (1999), 1357–1362.
14. Hubel, D. H., and Wiesel, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology 160, 1 (1962), 106–154.
15. Mohri, M., Rostamizadeh, A., and Talwalkar, A. Foundations of machine learning. MIT press, 2012.

16. Liu, T., Fang, S., Zhao, Y., Wang, P., and Zhang, J. Implementation of training convolutional neural networks. arXiv preprint arXiv:1506.01195 (2015).
17. Bradski, G., et al. The opencv library. Doctor Dobbs Journal 25, 11 (2000), 120–126.
18. Manoharan, R., and Chandrakala, S. Android opencv based effective driver fatigue and distraction monitoring system. In Computing and Communications Technologies (ICCCT), 2015 International Conference on (2015), IEEE, pp. 262–266.
19. Coelho, G., Kougianos, E., Mohanty, S. P., Sundaravadivel, P., and Albalawi, U. An iot-enabled modular quadrotor architecture for realtime aerial object tracking. In Nanoelectronic and Information Systems(iNIS),2015 IEEE International Symposium on (2015), IEEE, pp. 197–202.
20. Viola, P., and Jones, M. Robust real-time object detection. International Journal of Computer Vision 4 (2001).
21. Lienhart, R., and Maydt, J. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on (2002), vol. 1, IEEE, pp. I–I.
22. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016).
23. A. Dhall, O. V. R Murthy, R. G. J., and T.Gedeon. Video and image based emotion recognition challenges in the wild: Emotiw 2015. In the 2015 ACM on international Conference on Multimodal interaction (2015), ACM, pp. 423–426.
24. Chao Qi, Min Li, Qiushi Wang, Huiquan Zhang, Jinling Xing, Zhifan Gao and Huailing Zhang. Facial Expressions Recognition Based on Cognition and Mapped Binary Patterns. IEEE Access, 2018.
25. Hemanth kumar meena, Kamalesh kumar sharma, S D Joshi, "Facial Expression Recognition with Enhanced Feature Extraction using Graph Fourier Transform", IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, 2017.
26. M.Murugappan, Subbulakshmi Murugappan, "Human Emotion Recognition Through Short Time Electroencephalogram (EEG) Signals Using Fast FourierTransform (FFT)", IEEE 9th International Colloquium on Signal Processing and its Applications, 8 - 10 Mac. 2013, Kuala Lumpur, Malaysia, 2013.
27. Seyad Mehdi Lajevardi, Katrina L.Neville, Zahir M. Hussain, "Facial Expression Recognition Over FFT-OFDM", International Conference on Advanced Technologies for Communications, 2009.
28. Kaihao Zhang, Yongzhen Huang, Yong Du, Liang Wang, "Facial Expression Recognition Based on Deep Evolutional Spatial-Temporal Networks", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 26, NO. 9, 2017.
29. Kasi Prasad. Mannepalli, Dr. P.Narahari Sastry, Dr.V.Rajesh, "Analysis and Design of Speaker Identification System using Neural Networks" National Conference on Advanced Signal Processing Applications (NCASPA'13), 2013.
30. Veena Mayya, Radhika M. Pai, Manohara Pai M. M, "Automatic Facial Expression Recognition using DCNN", 6th International Conference On Advances In Computing & Communications, 2016.

## AUTHORS PROFILE

**Dr. P. Narahari Sastry** is working in CBIT **since 2003**, and is involved in Teaching and Research. He has recognized as Supervisor for JNTU Hyderabad, Osmania University Hyderabad and K.L. University Vijayawada. He has guided three **(03) PhD students** and also guiding eight (**8**) **Research scholars** from Osmania University Hyderabad presently. He has published **6 SCI Journals** in the last **5 years** and 2 of them are published in the reputed journals which are "Electrical **and Computer Engineering**" and **IEEE Communication letters**. Also he has applied and published his work in Indian Journal of patents, Chennai, India. He was the **PI** for an **AICTE project** titled "Design and Development of Palm Leaf Character Recognition System" granted under RPS scheme, which was successfully completed during the period **2013-16.**

**Mohammed Sameer Syed** is currently pursuing Post-Graduation (PG) i.e. M.E final year in the Electronics and Communication Engineering (ECE) department at Chaitanya Bharathi Institute of Technology (CBIT-A), Hyderabad, India. He has keen interest in the field of image processing, machine learning and artificial intelligence. He worked on projects related to communication, antennas, IoT, neural networks and pattern recognition. He was an intern in BSNL, Tirupati. He is also doing a research project in the field of machine learning.