A PROJECT REPORT

On

Classifying Human Facial Expressions and mapping them to emoji.

Submitted By

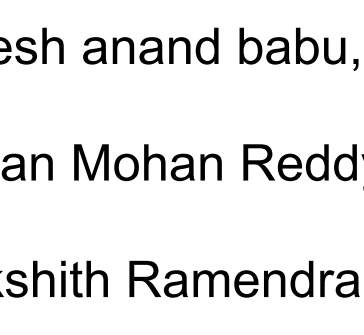
M.Venkata kalyan babu, 18BCS049

T. Umesh anand babu, 18BCS105

G. Jagan Mohan Reddy, 18BCS029

Y. Mokshith Ramendra, 18BCS112

Under the Guidance of
Dr. B. Jayalakshmi



INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DHARWAD

Indian Institute of Information Technology Dharwad

Computer Science and Engineering

Acknowledgement

We hereby certify that the work which is being presented in the Mini Project 1 report entitled “Classifying the Human Facial Expressions and mapping them to emoji” in partial fulfillment of the requirements for the award of B.Tech degree and submitted to the Department of Computer Science and Engineering of Indian Institute of Information Technology Dharwad , Karnataka is an authentic record of our own work carried out during the period from January 2021 to May 2021 under the supervision of Dr. B.Jayalakshmi , Asst.Professor, IIIT Dharwad.

The matter proposed in this report has not been published earlier and has never been submitted by us for the award of any other degree elsewhere.

Name of the Students :

M.Venkata kalyan babu	18BCS049
T. Umesh anand babu	18BCS105
G. Jagan Mohan Reddy	18BCS029
Y. Mokshith Ramendra,	18BCS112

DR.B.Jayalakshmi	DR.B. Jayalakshmi	DR. Uma Sheshadri
Project Supervisor	Project Coordinator	Head of the Department

ABSTRACT

Human facial expressions convey a lot of information visually rather than articulately. Facial expression recognition plays a crucial role in the area of human-machine interaction. Facial expression recognition system have many applications including, but not limited to, human behavior understanding, detection of mental disorders, and synthetic human expressions. Recognition of facial expression by computer with high recognition rate is still a challenging task.

Two popular methods utilized mostly in the literature for the automatic FER systems are based on geometry and appearance. Facial Expression Recognition is usually performed in four-stages consisting of pre-processing, face detection, feature extraction, and expression classification.

In this project we applied various deep learning methods (convolutional neural networks) to identify the key six human emotions: anger, fear, happiness, sadness, surprise and neutrality.

Table Of Contents

1.Introduction:

Image Processing is a vast area of research in the present day world and its applications are very widespread. One of the most important applications of Image processing is Facial expression recognition. Facial Expressions play an important role in interpersonal communication. Facial expression is a non verbal scientific gesture which gets expressed in our face as per our emotions.

Recognition of facial expression plays an important role in artificial intelligence and robotics Some applications related to this include Personal identification and Access control, Videophone and Teleconferencing, Forensic application, Human-Computer Interaction, Automated Surveillance.

The objective of this project is to develop Facial Expression Recognition System which can take human facial images containing some expression as input and recognize and classify it into six different expression class such as :

I. Neutral
II. Angry
III. Fear
IV. Happy
V. Sadness
VI. Surprise

Several Projects have already been done in this field and our goal will not only be to develop a Facial Expression Recognition model and mapping them to emoji but also improving the accuracy of this system compared to the other available models.

Motivation:

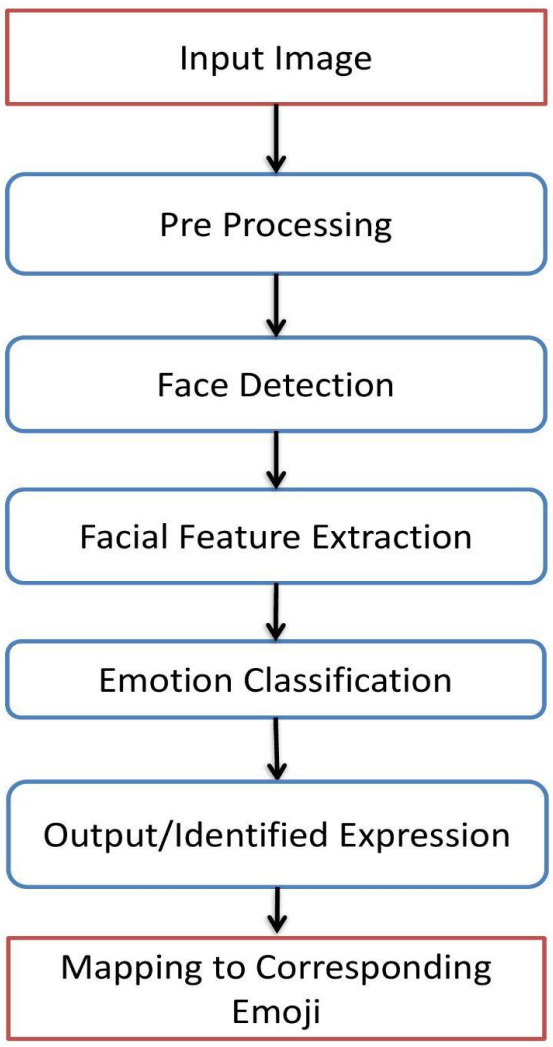
We have been motivated observing the benefits of physically handicapped people like deaf and dumb. But if any normal human being or an automated system can understand their needs by observing their facial expression then it becomes a lot easier for them to make the fellow human or automated system understand their needs.

Emojis or avatars are ways to indicate nonverbal cues. These cues have become an essential part of online chatting, product review, brand emotion.

2.Problem Definition:

Human facial expressions can be easily classified into 6 basic emotions: happy, sad, surprise, fear, anger and neutral. Our facial emotions are expressed through activation of specific sets of facial muscles. Through facial emotion recognition, we are able to measure the effects that content and services have on the audience/users through an easy and low-cost procedure. We designed a deep learning neural network that gives machines the ability to make inferences about our emotional states. In other words, we give them eyes to see what we can see.

Problem formulation of our project:



Preprocessing:-

Preprocessing is a common name for operations with images at the lowest level of abstraction both input and output are intensity images. Most preprocessing steps that are implemented are –

a. Reduce the noise

b. Convert The Image To Binary/Grayscale.

c. Pixel Brightness Transformation.

d. Geometric Transformation.

Face registration:-

Face Registration is a computer technology being used in a variety of applications that identifies human faces in digital images. In this face registration step, faces are first located in the image using some set of landmark points called “face localization” or “face detection”. These detected faces are then geometrically normalized to match some template image in a process called “face registration”.

Facial Feature Extraction:-

Facial Features extraction is an important step in face recognition and is defined as the process of locating specific regions, points, landmarks, or curves/contours in a given 2-D image or a 3D range image. In this feature extraction step, a numerical feature vector is generated from the resulting registered image. Common features that can be extracted are-

a.Lips
b. Eyes
c. Eyebrows
d. Nose tip

Emotion Classification:

In the third step, of classification, the algorithm attempts to classify the given faces portraying one of the seven basic emotions.

3. Algorithm:-

Step 1 :Collection of a data set of images. (In this case we are using the FER2013 database of 28273 pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes: anger, fear, happiness, sadness, surprise, and neutral.

Step 2 :Pre-processing of images.

Step 3 :Detection of a face from each image.

Step 4 :The cropped face is converted into grayscale images.

Step 5 : The pipeline ensures every image can be fed into the input layer as a (1, 48, 48) numpy array.

Step 6 :The numpy array gets passed into the Convolution2D layer.

Step 7 :Convolution generates feature maps.

Step 8 :Pooling method called MaxPooling2D that uses (2, 2) windows across the feature map only keeping the maximum pixel value.

Step 9 :During training, Neural network Forward propagation and Backward propagation performed on the pixel values.

Step 10:The Softmax function presents itself as a probability for each emotion class.

The model is able to show the detailed probability composition of the emotions in the face.

Step 11: Mapping the Facial expression to corresponding emoji.

4. Methodology: 5

Various facial datasets

1. Japanese Female Facial Expression (JAFFE)
2. FER
3. CMU MultiPIE
4. Lifespan
5. MMI
6. FEED
7. CK

But, among all those Datasets, only FER2013 is available online. So we opted for it.

4.1 Dataset Description:

(FER2013). The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression into one of six categories (0=Angry, 1=Fear, 2=Happy, 3=Sad, 4=Surprise, 5=Neutral). The training set consists of 28,273 examples. The public test set consists of 7067 examples.

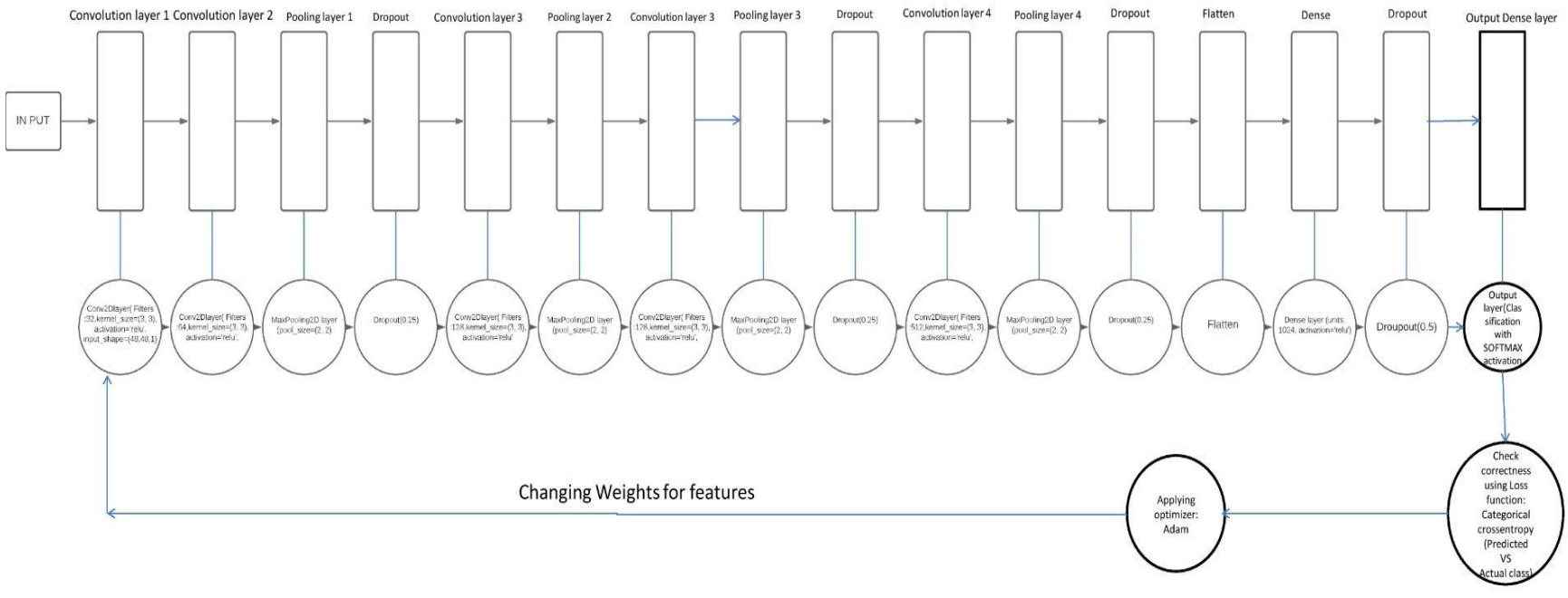
Emotion labels in the dataset:
0: -4593 images- Angry
1: -5121 images- Fear
2: -8989 images- Happy
3: -6077 images- Sad
4: -4002 images- Surprise
5: -6198 images- Neutral

As we were exploring the dataset, we discovered an imbalance of the “disgust” class compared to many samples of other classes. We decided to merge disgust into anger given that they both represent similar sentiment. We used 28273 labeled faces as the training set and held out the remaining two test sets (3533/set) for after-training validation. The resulting is a 6-class, balanced dataset, that contains angry, fear, happy, sad, surprise, and neutral. Now we‟re ready to train.

4.2 THE MODEL

4.2.1 Convolutional Neural Network

Deep learning is a popular technique used in computer vision. We chose Convolutional Neural Network (CNN) layers as building blocks to create our model architecture. CNNs are known to imitate how the human brain works when analyzing visuals.The architecture of a convolutional neural network contains an input layer, some convolutional layers, some dense layers (fully-connected layers), and an output layer . These are linearly stacked layers ordered in sequence. In Keras, the model is created as Sequential() and more layers are added to build architecture.



Input Layer:-

The input layer has predetermined, fixed dimensions, so the image must be pre-processed before it can be fed into the layer. We used OpenCV, a computer vision library, for face detection in the image.

The haarcascade_frontalface_default.xml in OpenCV contains pre-trained filters and uses The cropped face is then converted into grayscale using cv2.cvtColor and resized to 48-by-48 pixels with cv2.resize. This step greatly reduces the dimensions compared to the original RGB format with three color dimensions (3, 48, 48). The pipeline ensures every image can be fed into the input layer as a (1, 48, 48) numpy array.

Convolutional Layers:

The numpy array gets passed into the Convolution2D layer where we specify the number of filters as one of the hyperparameters. The set of filters(kernel) 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.

Convolution generates feature maps that represent how pixel values are enhanced, for example, edge and pattern detection. A feature map is created by applying filter 1 across the entire image. Other filters are applied one after another creating a set of feature maps.

Pooling is a dimension reduction technique usually applied after one or several convolutional layers. It is an important step when building CNNs as adding more convolutional layers can greatly affect computational time. We used a popular pooling method called MaxPooling2D that uses (2, 2) windows across the feature map only keeping the maximum pixel value. The pooled pixels form an image with dimensions reduced by 4.

Dense Layers:

The dense layer (fully connected layers), is same as the way neurons transmit signals through the brain. It takes a large number of input features and transforms features through layers connected with trainable weights.

These weights are trained by forward propagation of training data then backward propagation of its errors. Back propagation starts from evaluating the difference between prediction and true value, and back calculates the weight adjustment needed to every layer before. We can control the training speed and the complexity of the architecture by tuning the hyper-parameters, such as learning rate and network density. As we feed in more data, the network is able to gradually make adjustments until errors are minimized.Essentially, the more layers/nodes we add to the network the better it can pick up signals. The model also becomes increasingly prone to overfitting the training data. One method to prevent overfitting and generalize on unseen data is to apply dropout.

Dropout randomly selects a portion (usually less than 50%) of nodes to set their weights to zero during training. This method can effectively control the model's sensitivity to noise during training while maintaining the necessary complexity of the architecture.

Strides:

This parameter is an integer or tuple/list of 2 integers, specifying the “step” of the convolution along with the height and width of the input volume.

Its default value is always set to (1, 1) which means that the given Conv2D filter is applied to the current location of the input volume and the given Filter takes a 1-pixel step to the right and again the filter is applied to the input volume and it is performed until we reach the far right and bottom border for rows and columns of the volume in which we are moving our filter.

Activation function RELU:

ReLu is a non-linear activation function that is used in multi-layer neural networks or deep neural networks. This function can be represented as:

:f(x) = max(0,x) where x = an input value ----(1)

According to equation 1, the output of ReLu is the maximum value between zero and the input value. An output is equal to zero when the input value is negative and the input value when the input is positive. Thus, we can rewrite equation 1 as follows:

$$f(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases} \quad (2)$$

where x = an input value .

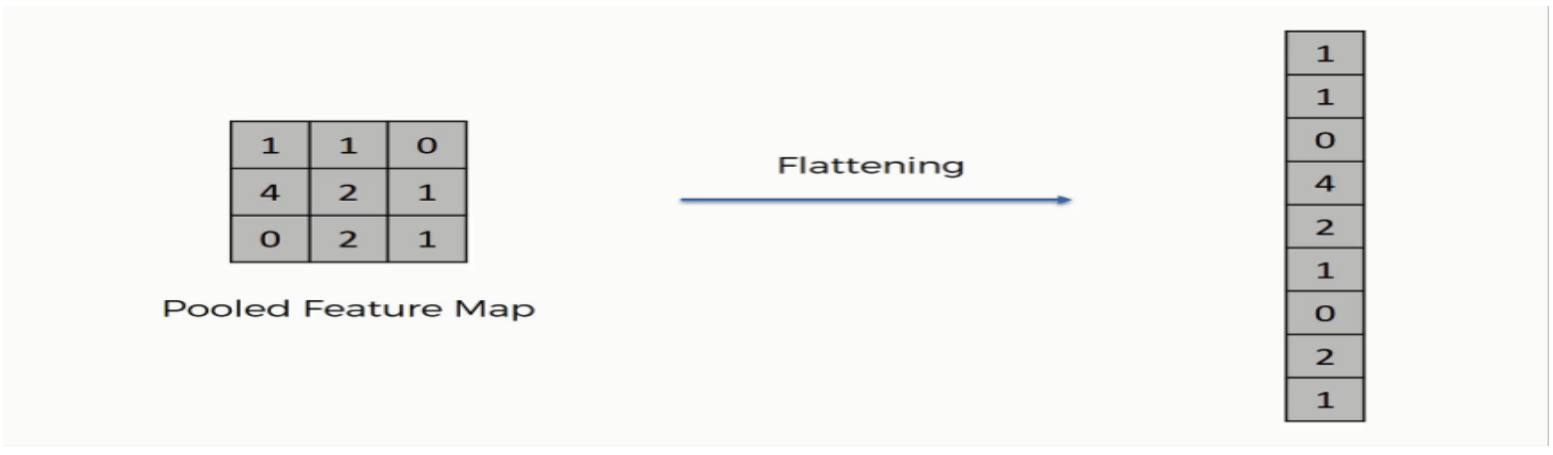
The ReLu function is able to accelerate the training speed of deep neural networks

MAX POOLING:

It is a pooling operation that selects the maximum element from the region of the feature map covered by the filter.Thus,the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.In below example stride=2.

Flatten:

Flatten is the function that converts the pooled feature map to a single column that is passed to the fully convolutional layer.



Output Layer :
Instead of using sigmoid activation function, we used softmax at the output layer.

Softmax:

Softmax function calculates the probability distribution of the event over „n" different events. In general, this function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.

The main advantage of using Softmax is the output probabilities range. The range will be 0 to 1, and the sum of all the probabilities will be equal to one. If the softmax function is used for a multi-classification model it returns the probabilities of each class and the target class will have the high probability.

Mathematically,softmax is defined as,

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)}$$

Where Y is an input vector to a softmax function,S.It consists of ‘n’ elements for ‘n’ classes(possible outcomes).The Y_i is the Ith element of the input vector.It can take any value between -inf to +inf.

exp(Y_i) is a standard exponential function applied to Y_i.

Sum of exponential values of all values in the inputs is a normalization term it ensures that the values of output vector S(Y)_i sums to 1 for ith class and each of them is in the range 0 and 1 which makes up a valid probability distribution.

Where, N is the number of classes(possible outcomes).This output presents itself as a probability for each emotion class.Therefore, the model is able to show the detailed probability composition of the emotions in the face.

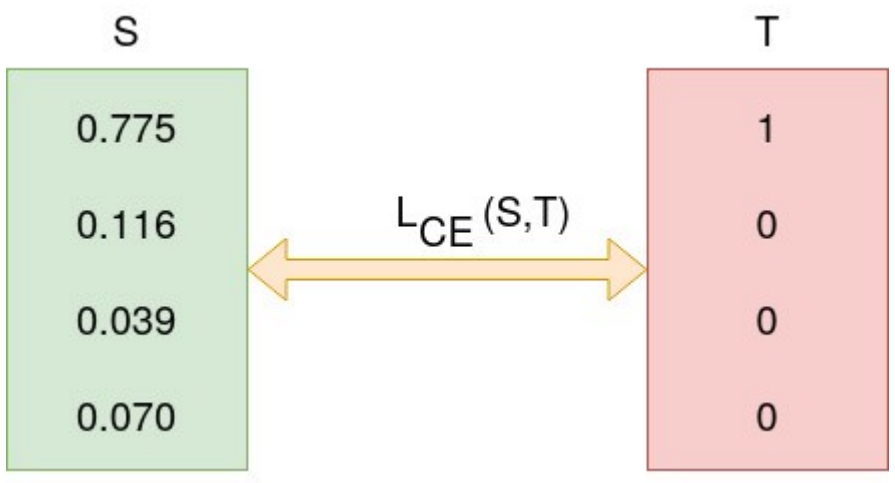
4.3 Loss Function:

4.3.1 categorical cross-entropy :

Also called Softmax Loss. It is a Softmax activation plus a Cross-Entropy loss. If we use this loss, we will train a CNN to output a probability over the N classes for each image. It is used for multi-class classification.In the specific (and usual) case of Multi-Class classification the labels are one-hot, so only the positive class Np keeps its term in the loss. There is only one element of the Target vector t which is not zero .

$$L_{CE} = -\sum_{i=1} T_i \log(S_i)$$

For Example



L_CE= -[1log_2(0.775)+0log_2(0.116)+0log_2(0.039)+0log_2(0.070)

=-log_2(0.775)

=0.3677 loss for given probability given above

Softmax is a continuously differentiable function. This makes it possible to calculate the derivative of the loss function with respect to every weight in the neural network. This property allows the model to adjust the weights accordingly to minimize the loss function (model output close to the true values).

4.4 Learning Rate:

The amount that the weights are updated during training is referred to as the “learning rate.” Specifically, the learning rate is a configurable hyperparameter used in the training of neural networks that has a small positive value, often in the range between 0.0 and 1.0.

The learning rate controls how quickly the model is adapted to the problem. Smaller learning rates require more training epochs given the smaller changes made to the weights each update, whereas larger learning rates result in rapid changes and require fewer training epochs.

How Decay Works:

Adam uses mini batches to optimize. During optimization, you need the cost function to be less, so quickly using a high learning rate. we have to reduce the learning rate in order not to miss the optimal point. Basically we have to decrease the learning rate to have more accurate steps by reducing the learning rate For each epoch, TensorFlow uses the same learning rate and after finishing each epoch, the next epoch will be started using the current learning rate divided by the decay parameter.

5. Mapping To Emoji

We’ve taken the weights of the trained model and included them in the mapping function to get the corresponding image.

6.Results and Discussion:

The model performs really well on classifying positive emotions resulting in relatively high precision scores for happy and surprised. Happy has high precision which could be explained by having the most examples (~7000) in the training set. surprise has less precision than happy having the least examples in the training set.

Model performance seems weaker across negative emotions on average. In particularly, the emotion sad has a low precision. The model frequently misclassified fear and a sad. In addition, it is most confused when predicting sad and fearful faces because these two emotions are probably the least expressive (excluding crying faces).

Performance As it turns out, the final CNN had a validation loss of 1.0564 validation accuracy of 62.79%(for 35 EPOCS).

When we trained the model for 50 EPOCS per iteration we got validation loss 1.2106, accuracy of 62.36%.

By adding padding technique, the model is taking more time to train and getting large difference between train_loss and validation_loss which indicates that the model is not is giving optimal solution so,In comparison of models with padding and without padding technique, the model without padding technique shown to achieve a good accuracy, and optimal solution.

Future Work:

1.we need to improve in specific areas like -

1.Number and configuration of convolutional layers
2. Number and configuration of dense layers
3. Dropout percentage in dense layers

But due to the lack of a highly configured system we could not go deeper into dense neural networks as the system gets very slow and we will try to improve in these areas in future. We would also like to train more databases into the system to make the model more and more accurate.

2. Applying the model to real time video streams and image data.

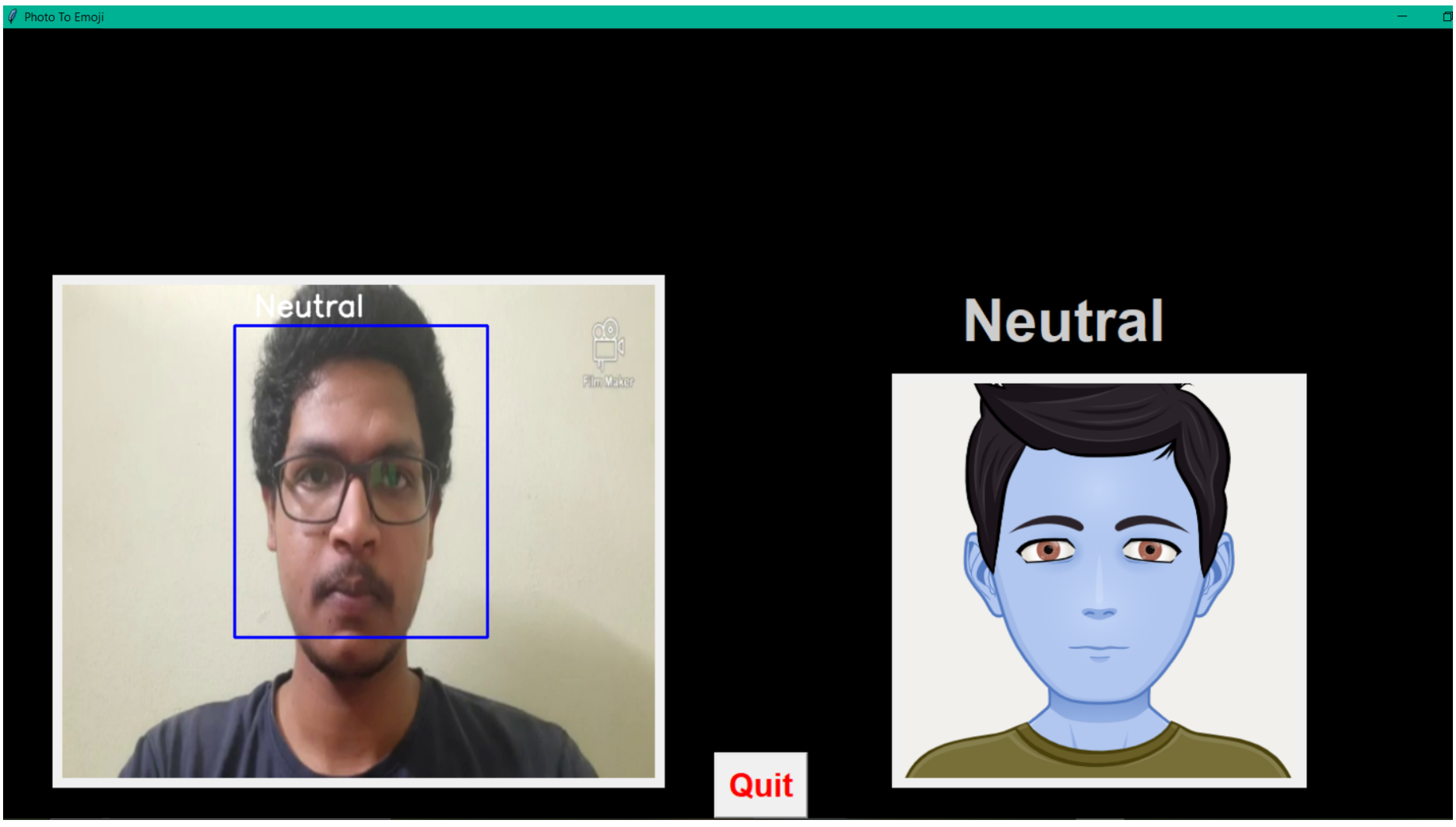
3. We're also thinking of making a Music player , like it will play the song Based upon our Facial Expression.

7. Summary and Conclusions

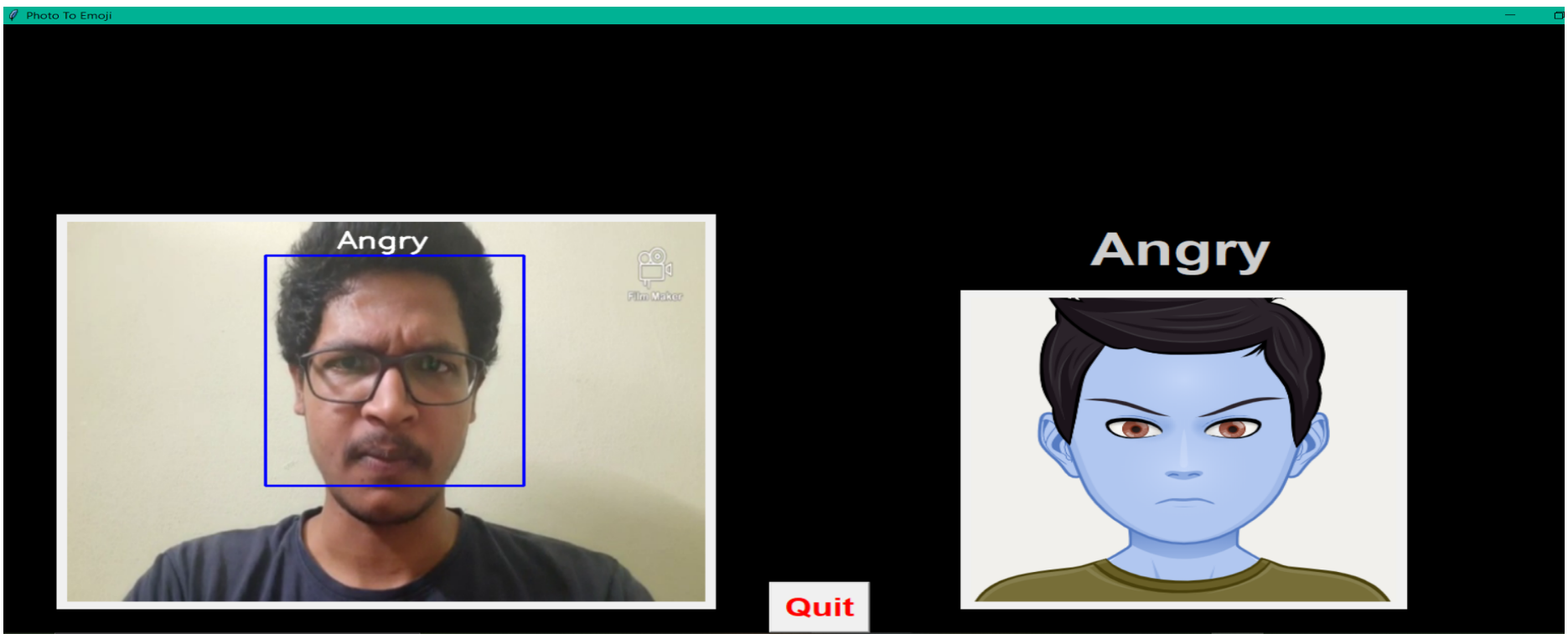
The facial expression recognition system presented in this project work contributes a resilient face recognition model based on the mapping of behavioral characteristics with the physiological biometric characteristics. The physiological characteristics of the human face with relevance to various expressions such as happiness, sadness, fear, anger, surprise are associated with geometrical structures which are restored as base matching template for the recognition system. The training set evaluates the expressional uniqueness of individual faces and provides a resilient expressional recognition model in the field of biometric security

10.Appendix

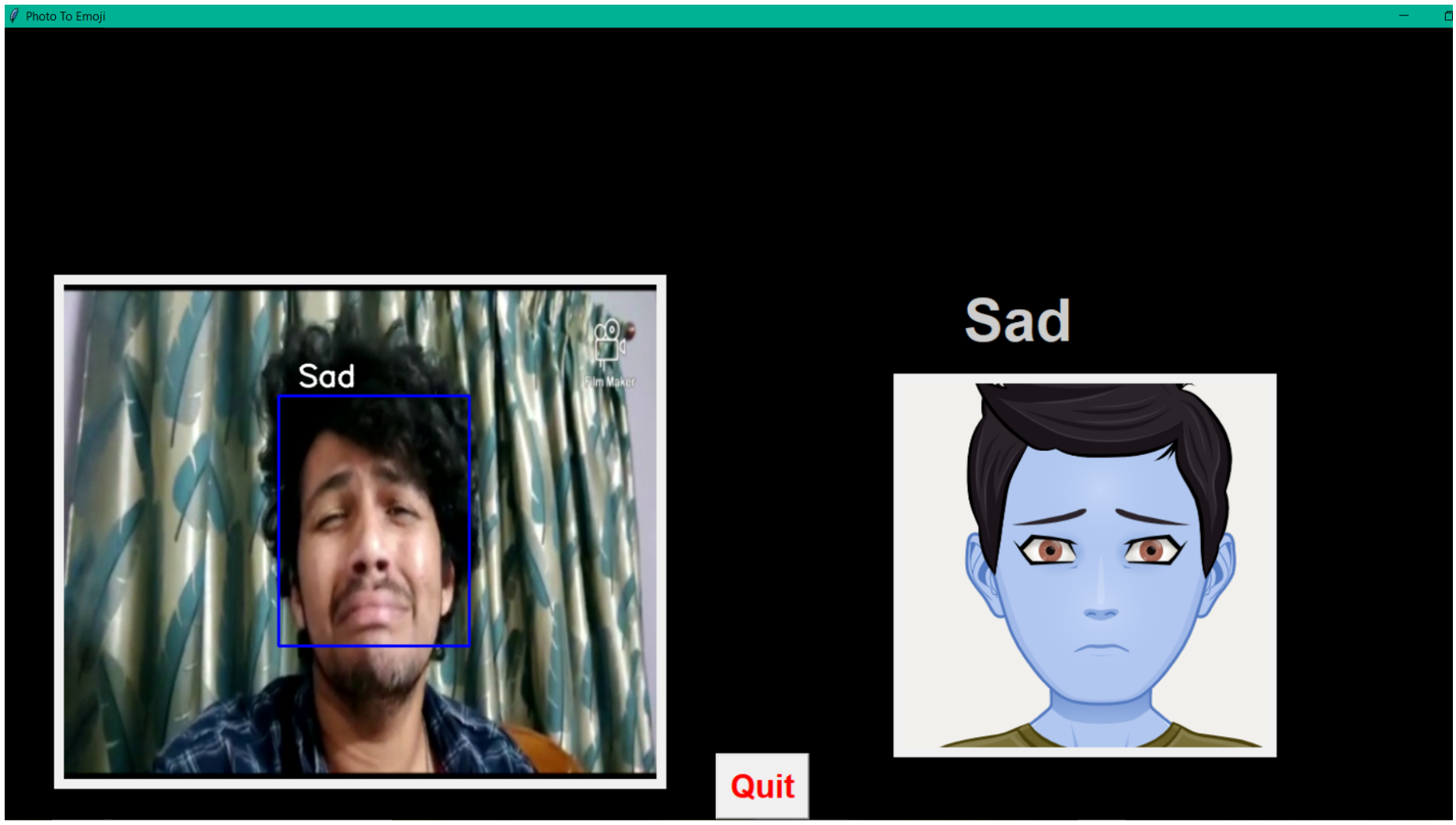
OUTPUT SAMPLE 1



2.



3.



9. References

1. A literature survey on Facial Expression Recognition using Global Features by Vaibhav Kumar J. Mistry and Mahesh M. Goyani,International Journal of Engineering and Advanced Technology (IJEAT),April,2013

2.Convolutional Neural Networks (CNN) With TensorFlow by Sourav from Edureka.

3. Recognizing Facial Expressions Using Deep Learning by Alexandru Savoiu Stanford University and James Wong Stanford University.

4. “Robust Real-Time Face Detection”, International Journal of Computer Vision 57(2), 137–154, 2004.

5. Going Deeper in Facial Expression Recognition using Deep Neural Networks, by Ali Mollahosseini1, David Chan2, and Mohammad H. Mahoor1 Department of Electrical and Computer Engineering, Department of Computer Science, University of Denver, Denver, CO.

6.Facial Expression Detection Techniques: Based on Viola and Jones algorithm and Principal Component Analysis by Samiksha Agrawal and Pallavi Khatri, ITM UniversityGwalior(M.P.), India, 2014.