Summer Project 2020
Science and Technology Council
IIT Kanpur

Facial Expression Recognition

Documentation

Date of Submission:

July 14, 2020

Contents

1 Acknowledgements

We are immensely grateful to Brain and Cognitive society and Science & Technology Council, IIT Kanpur for providing us with the great opportunity and motivation for this Summer Project. We thank our mentor, Avisha Gaur , the Coordinators of BCS : Avisha , Ishika and Yatin for their guidance and support. We would not be able to complete this project without the help of these people.

2 Introduction

2.1 Aim

The goal of this project is to build a machine learning based model which will be able to classify seven basic emotions using facial expressions of humans. In simple words , the input will be a image of a human face and the model will classify that image having one of the seven basic emotions by extracting the facial expressions from that human face. The detailed article about this problem statement can be found here.
The seven emotions that can be classified using this model are :

- Happy
- Sad
- Anger
- Disgust
- Fear
- Surprise
- Contempt

2.2 Dataset

We used the extended cohn-kanade dataset which is known as CK+ . It contains 981 images in 7 different folders labelled accordingly as the 7 basic emotions .Of which 135 images are of anger , 177 of disgust , 207 of happy , 249 of surprise , 84 of sadness , 75 of fear and 54 of contempt .

2.3 Research Paper

The research paper that we followed for this project is Li2020_Facial_Expression_Recognition which was published on January 2019 as a part of Springer nature 2019.

3 Ideation

We divided the whole project basically into four parts :

- Pre-processing
- CNN model
- Combination
- Evaluation

3.1 Pre-processing

The very first part is pre-processing of the images of the dataset so as to make the images much more susceptible for extracting features from them.
The pre-processing part is further divided into four sub-parts which are:

- Face Alignment
- Face Cropping
- Data Normalization
- Data Augmentation

3.2 CNN model

The Convolutional neural networks model was applied to extract features from the face images and to categorize their expressions. This model takes as an input the pre-processed face images and give as output the probabilities of seven emotions.

3.3 Combination

Following the pre-processing and CNN model , we had to combine them both and finally train our model on the dataset we had chpsen.

3.4 Evaluation

Finally , as per the paper we need to evaluate our model on the basis of :

- Neuron number
- Cropping Methods

Also we had to perform ten-fold **cross validation** on our model.

4 Implementation

4.1 Pre-processing

The implementational details of pre-processing are as follows :

4.1.1 Face Alignment

The images in the dataset show various postures. These variations affect the model performance. Face alignment was performed to address this problem. The algorithm used for face alignment was based on the position of the eyes. The Dlib toolkit was used to obtain the face landmarks. A total of 68 sequential points, each of which could be represented by a coordinate, were identified. The centres of the left and right eyes were computed based on twelve points (No. 36 to 47). The first six points encircle the left eye centre, and the other six points encircle the right eye centre. The rotation angle is calculated on the basis of these twelve points using Eq. (1).

$$angle = \tan^{-1} \frac{(\sum\limits_{n=42}^{47} x_n - \sum\limits_{n=36}^{41} x_n)}{(\sum\limits_{n=42}^{47} x_n - \sum\limits_{n=36}^{41} x_n)} \tag{1}$$

where x_n is the x-coordinate of the nth point and y_n is the y-coordinate of the nth point.

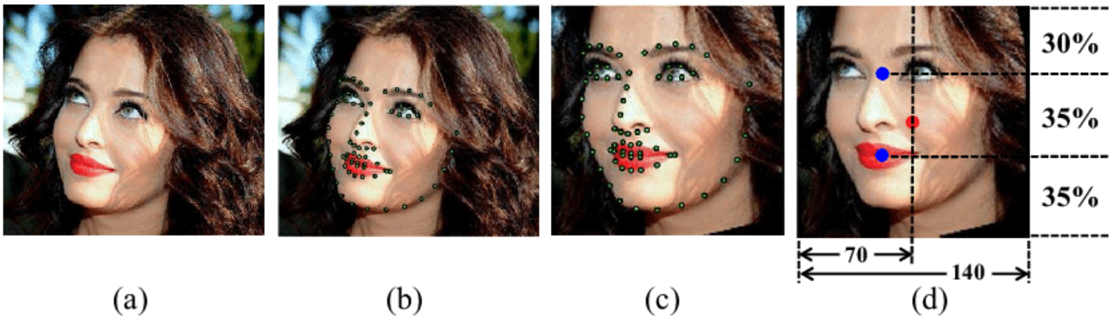


Figure 1: Face Alignment

4.1.2 Face Cropping

Since almost all images have background along with the face image , which can cause certain problems while extracting features from faces , so face cropping was performed on all the images. The openCV toolkit was used for face cropping and above mentioned three methods of cropping were compared. The same 68 points landmarks were obtained and the forehead region was then removed in such a way that perpendicular distance from the top side of the cropped image to the horizontal line connecting the eye centres is 0.6 d (d is the distance between eye centres) , where d was found to be :

$$d = \frac{(\sum_{n=42}^{47} x_n - \sum_{n=36}^{41} x_n)}{6} \tag{2}$$

Figure 2: Face Cropping

4.1.3 Data Normalization

Brightness and contrast can differ even between images of the same person with the same expression. So, Histogram Equalization was applied to reduce this variation and further Z-score normalization was also done using Eq.(3) to enhance contrast.

$$x' = \frac{x - \mu}{\sigma} \tag{3}$$

where x' is the value of the new pixel, x is the value of the original pixel, is the mean pixel value of all sample images, and is the standard deviation of the pixel values of all sample images. Finally, the image is down-sampled to 32x32 pixels.

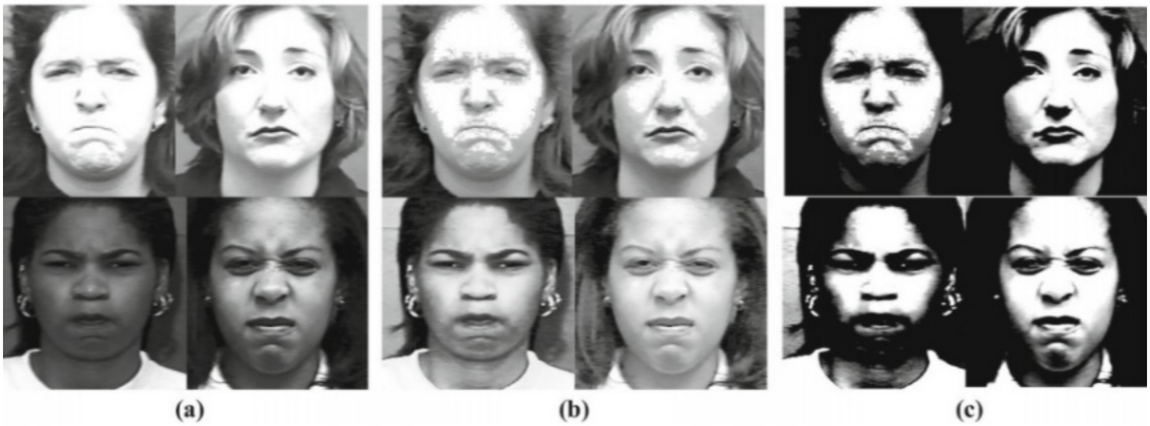


Figure 3: Data Normalization

where (a) shows Original image , (b) shows Histogram equalized and (c) Z-score Normalized images.

4.1.4 Data Augmentation

Having done the previous pre-processing steps , there was a slight tilt in the images and also the number of images on which we're training our model was not abundant. Hence, to ensure adaptability and abundancy of the data , we adopted random flipping and random rotation of all the images. In random horizontal flipping, an image can be flipped before training to address the problem of uneven cropping. Random rotation expands the original data by rotating an image by a random angle within an interval. To achieve this , we set horizontal flip to 'True' and rotation angle to '2' in the in-built function **ImageDataGenerator**.

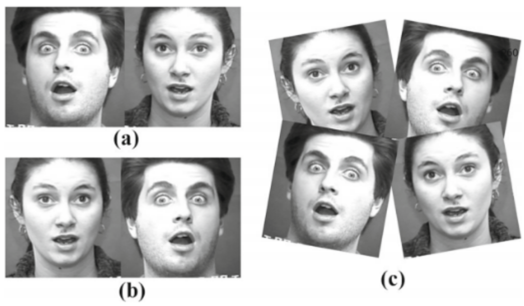


Figure 4: Data Augmentation

where (a) shows Original images , (b) shows Horizontally flipped images and (c) shows Randomly rotated images.

4.2 CNN model

The Convolutional neural networks model was applied to extract features from the face images and to categorize their expressions. This model takes as input the pre-processed face images and give as output the probabilities of all the seven emotions. We used two convolutional layers , two sub-sampling layers and one output dense layer with seven units and a softmax activation function to output the probabilities of seven emotions.6. The first and third layers are convolution layers (Conv2D) with 32 and 64 kernels, respectively, which have the size of 55 and a relu activation function. The second and fourth layers are Subsampling layers (MaxPooling2D) with a kernel size of 2x2. **For regularization purpose , as a fifth layer we also used a Dropout layer of 0.5 drop in addition to what given in the research paper**. Then , we flattened the input using Flatten layer and finally added the Dense output layer at the end. Also , we used Adam optimizer , Cross-entropy loss and Xavier initialization in the CNN model.

4.3 Combination

As such , there's nothing much to describe in this section except the efforts needed to combine all things together to come up with an assembled model that works as per our expectation , which makes it worth writing. We have to set priorities of things to make it work , make functions for convenience ,detect bugs and make modifications accordingly until we achieve what's expected from our model.

4.4 Evaluation

In this section we have three sub-parts worth describing :

4.4.1 Neuron number

For this part we had to add another Dense layer prior to the output layer in our CNN model and then vary the neuron number of that hidden dense layer as :

- 0
- 256
- 512
- 1024

4.4.2 Cropping Methods

We evaluated results on three different cropping methods which are :

- **Cropping with background**: In this method cropping of human face was done but leaving behind a little background also along with the face.
- **Cropping without background**: As the name suggests , in this method cropping is done without leaving background i.e. only the cropped face remains after application of this method.

- **Cropping without forehead**: This method suggests that face cropping should be done in a manner that the forehead from the face is also cropped away along with the background.

4.4.3 Cross-Validation

As given in the paper , we followed a ten-fold cross validation on our dataset. We performed ten fold cross validation four times each time varying the neuron number of the hidden Dense layer but keeping the cropping method fixed i.e. the cropping without background method. As in a general k-fold cross validation method , we divide the dataset into k folds and then train our model on k-1 folds while evaluating on the remaining fold , k times and shuffling the data every single time to get a good generalization. Finally , we take the average of the validation accuracy on all the k-folds.

5 Results

The results derived from the project work are as follows :

5.1 Cropping without background

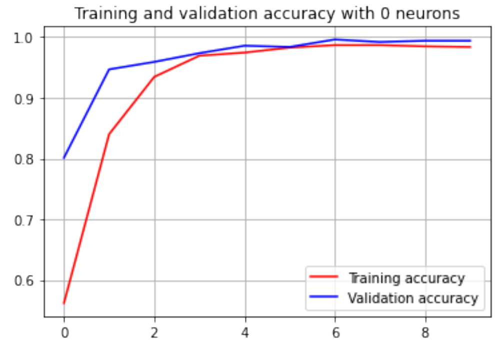
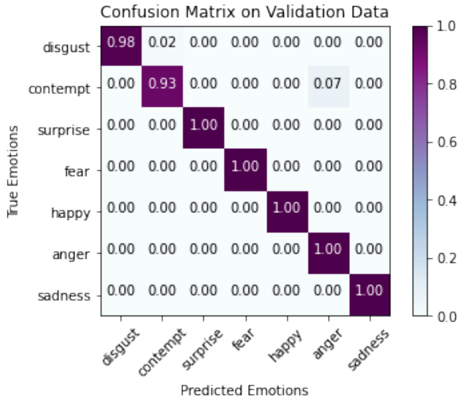
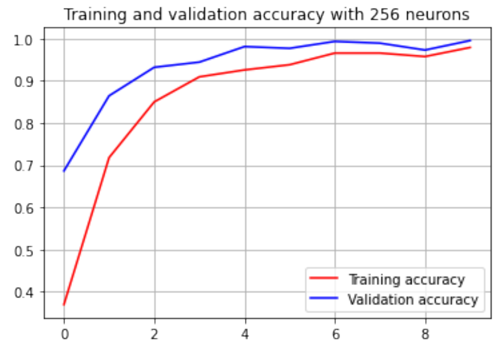
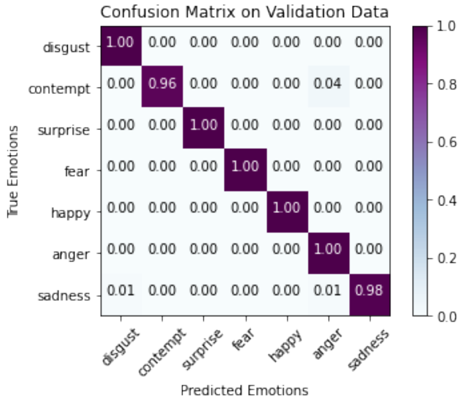
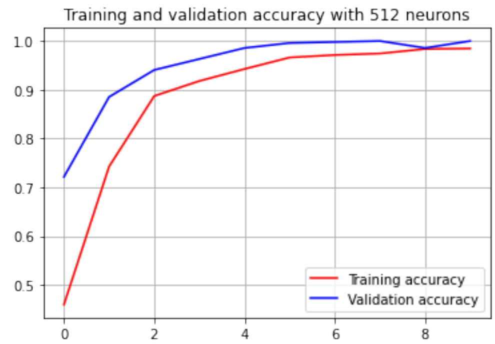
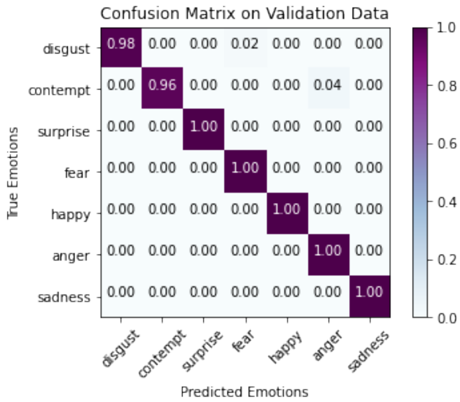
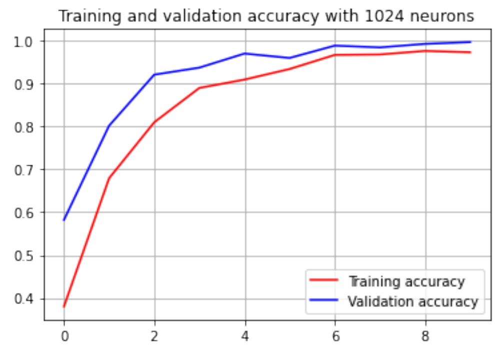
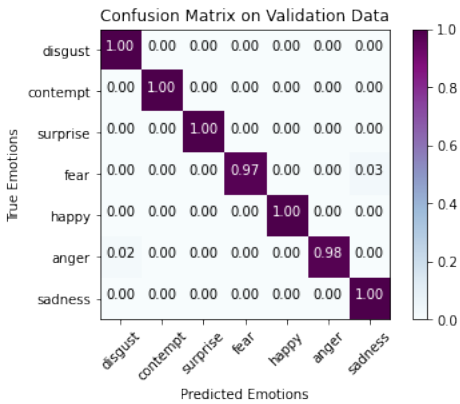
<table>
<tr><th>No. of Neurons</th><th>Accuracy graph</th><th>Confusion matrix</th></tr>
<tr><td>0</td><td>
</td><td>
</td></tr>
<tr><td>256</td><td>
</td><td>
</td></tr>
<tr><td>512</td><td>
</td><td>
</td></tr>
<tr><td>1024</td><td>
</td><td>
</td></tr>
</table>

Table 1: Accuracy and Confusion matrices of cropping without background

5.2 Cropping with background

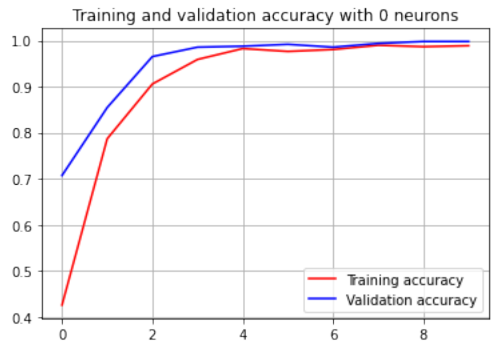
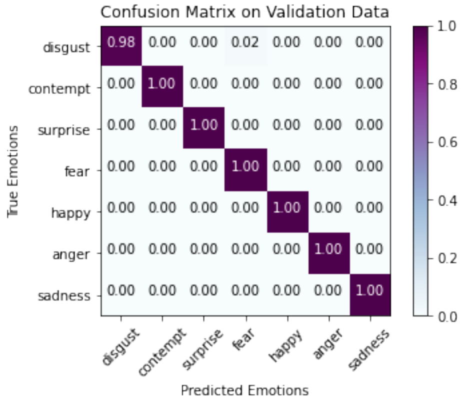
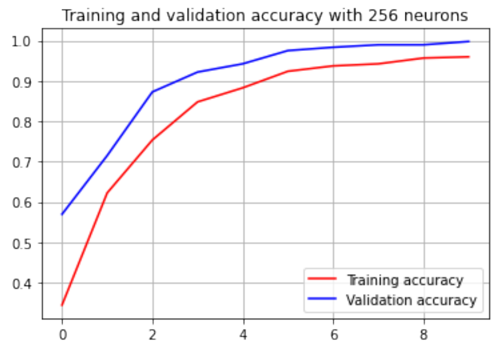
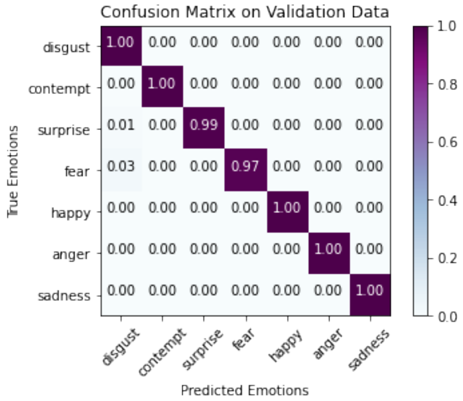
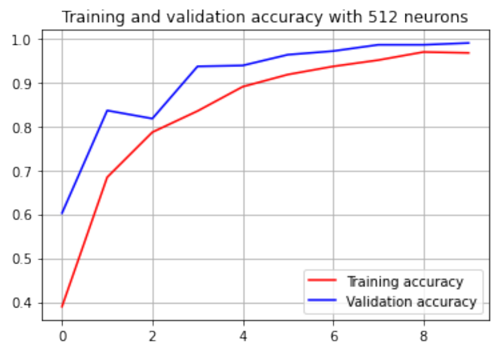
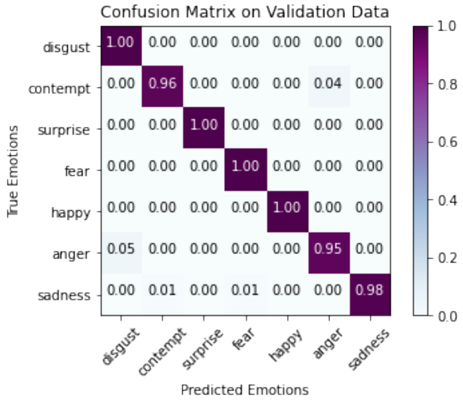
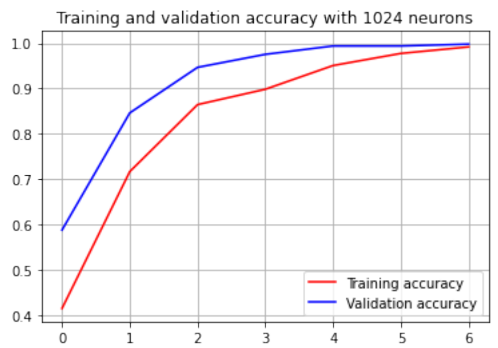
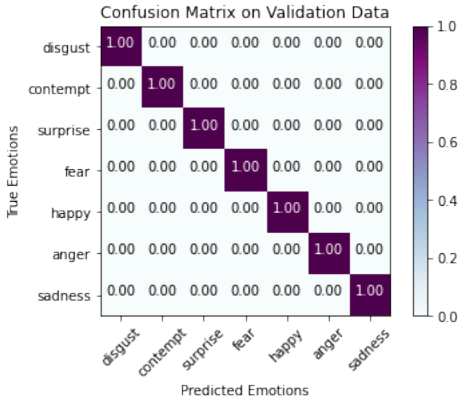
<table>
<tr><th>No. of Neurons</th><th>Accuracy graph</th><th>Confusion matrix</th></tr>
<tr><td>0</td><td>
</td><td>
</td></tr>
<tr><td>256</td><td>
</td><td>
</td></tr>
<tr><td>512</td><td>
</td><td>
</td></tr>
<tr><td>1024</td><td>
</td><td>
</td></tr>
</table>

Table 2: Accuracy and Confusion matrices of cropping with background

5.3 Cropping without forehead

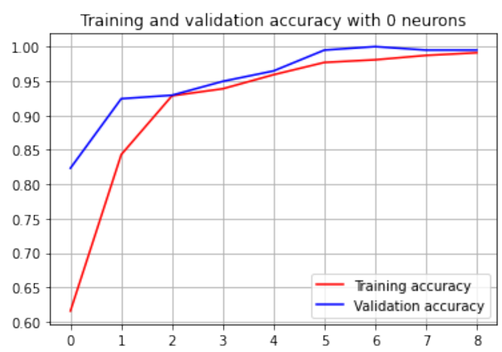
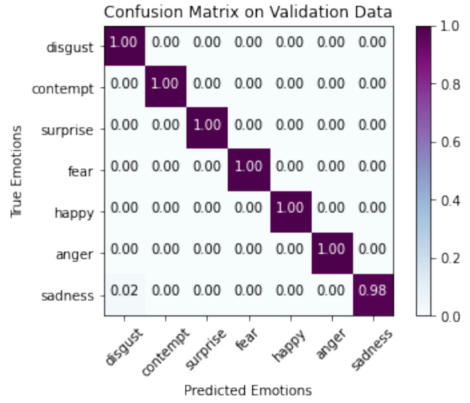
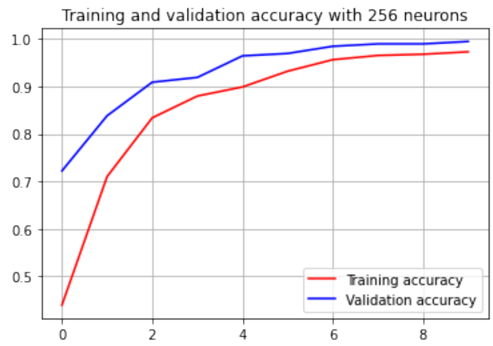
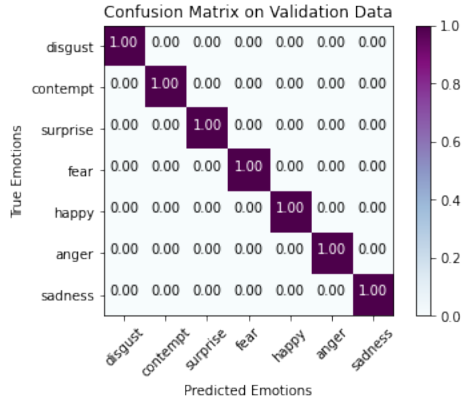
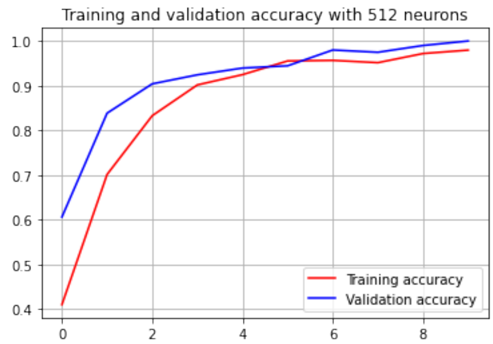
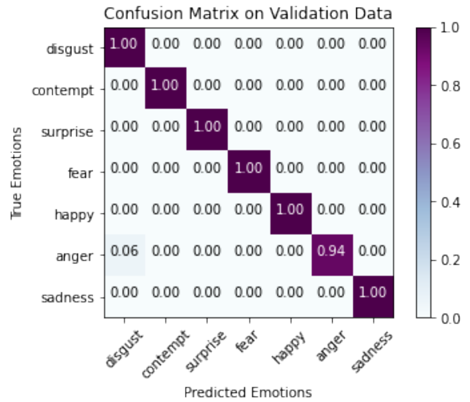
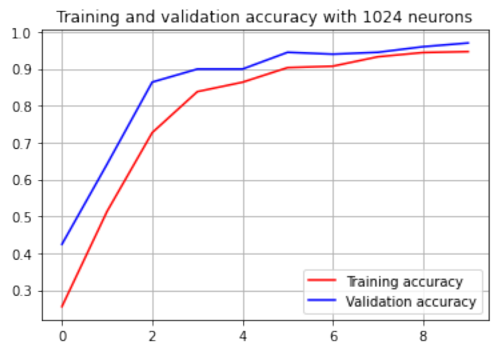
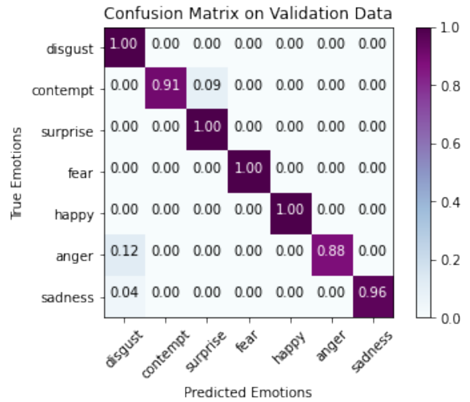
No. of Neurons	Accuracy graph	Confusion matrix
0		
256		
512		
1024		

Table 3: Accuracy and Confusion matrices of cropping without forehead

5.4 Cross Validation

No. of Neurons	Average Accuracy
0	97.96
256	98.27
512	96.62
1024	97.14

Table 4: Cross Validation results

5.5 Comparision with paper

On the Paper	On our model
97.38	97.49

Table 5: Comparision with paper

6 Github link

All the work we have done , well-assembled code and other details can be found here.

7 Other references

We used the following links of some articles or courses as references for our overall work :

- https://link.springer.com/article/10.1186/s13640-018-0324-4
- https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/
- https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
- https://coursera.org/share/111ff958aae8ede07800d98664152420
- https://www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/
- https://machinelearningmastery.com/k-fold-cross-validation/