A Case Study on the Efficacy of Machine Learning Models in Recognizing Facial Expressions

Dhaval Harish Sharma, Harsh NareshKumar Javia, Siddhant Vijay Kajale

***Abstract*—This research project applies various Machine Learning models for recognizing the facial expressions from the dataset "Fer2013". There are many applications for this project in different research fields, for example, mental infection detection and human social/physiological collaboration detection. This research model uses 35,887 images to categorize each face based on the emotion in to one of the seven categories (0 - Angry, 1 - Disgust, 2 - Fear, 3 - Happy, 4 - Sad, 5 - Surprise, 6 - Neutral). It makes the use of Convolution Neural Network, Decision Tree, Random Forest, Naïve Bayes, K-Nearest Neighbors, Logistic Regression and Support Vector Machine models as they are considered good for handling image data.**

I. INTRODUCTION

Facial emotion recognition is the way of recognizing human feelings from facial expressions. The human mind perceives feelings naturally, and technology has now been built that can perceive feelings too. This innovation is turning out to be progressively precise constantly and will in the end have the option to peruse feelings just as our brains do. In this project, we tried to categorize the human expressions into one of the seven categories (0 – Angry, 1 – Disgust, 2 – Fear, 3 – Happy, 4 – Sad, 5 – Surprise, 6 - Neutral) using the Convolution Neural Network, Support Vector Machine, Decision Tree, Random Forest, Logistic Regression, K-Nearest Neighbors and Naïve Bayes models on the Fer2013 dataset containing 35,887 images of 7 different emotions. Of these models, we found that Convolution Neural Network and Naïve Bayes were the most stable and accurate models. This is largely due to their ability to accurately predict the image data.

II. TASK DESCRIPTION

Our task was to build and train several different Machine Learning models on the dataset. The first step was to preprocess the data. The models were then given a validation test to tune their hyper-parameters. Thereafter, the models were tested on a test set and tasked with predicting the facial expressions for individuals in the test set. The results were analyzed to determine a good model for this classification problem.

Once we were able to determine the most accurate model, we set out to determine the most accurate subset of data. To do this, we shuffled the dataset into three bins and ran the model on each subset.

III. MAJOR CHALLENGES AND SOLUTIONS

A. Challenges

The different difficulties that we faced with the facial expression dataset are as follows: The dataset was enormous and contained around 35,887 examples for different classes of human facial feelings. The size of the real dataset was around 2 GB. However, we needed to decrease the size of the examples for quicker handling and model preparing. The greatest downsides of an enormous dataset are the time taken for processing, considering irrelevant data and in this way wasting the assets. It was precarious to pre-process the tests of differing estimate and diminish the size of each example to 48*48. Along with the various emotions for human expressions, there were a lot of invalid images and images that didn't belong to any category. Thus, data needed to be cleaned.

B. Solution

Since the classification dataset was humongous, a rational approach was inevitable. Firstly, we studied the various classes present in the dataset. Secondly, we defined our objective to classify the emotions as per the category, pre-processed the dataset by removing unwanted images, reducing the size of the images to 48x48 using Image Resizer for Windows (Link in references) so that every image was of the same size. Therefore, the resources consumed during the processing would be less and training on the model would be faster. After all the pre-processing on the data the dataset got reduced to 294 MB and the processing time reduced which helped us to train the model faster. Now our dataset was ready with images of resolution size 48x48 on which our models can be trained.

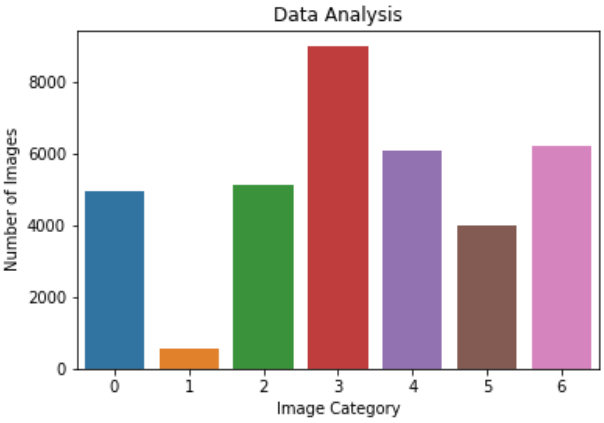
IV. EXPERIMENTS

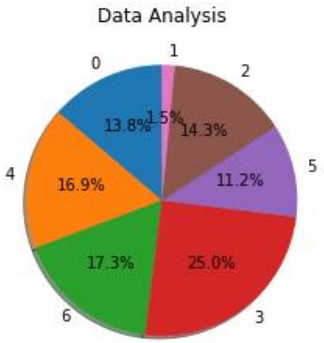
A. Dataset Description

The data consists of images of faces each having 48x48 pixels. The faces have been naturally enlisted with the goal of being pretty much focused and possesses about a similar measure of room in each image. The task is to categorize each face based on the feeling appeared through the facial expression in to one of seven classes (0 - Angry, 1 - Disgust, 2 - Fear, 3 - Happy, 4 - Sad, 5 - Surprise, 6 - Neutral). The "fer2013.csv" contains three columns, "emotion", "pixels" and "usage". The "emotion" segment contains a numeric code extending from 0 to 6, comprehension, for the emotion that is available in the image. The "pixels" segment contains a string encompassed in quotes for each image. The contents of this string are a space-isolated pixel values in row major order. The "usage" column indicates the category of the images. There are three categories in the dataset namely "Training", "PublicTest" and "PrivateTest". The Training set is used for the training the models, PublicTest is used for validation and tuning of hyper-parameters and PrivateTest is used for testing the models.

The Training set comprises of 25,840 different input images. The validation set utilized for the model comprises of 2,870 images. The test set, which is utilized to decide the outward appearances comprises of another 7,176 images.

Here are some of the visual representations of the dataset as produced by the "data_analysis.py" file,





This dataset was set up by Pierre-Luc Carrier and Aaron Courville, as a component of a continuous research venture. They have benevolently given the workshop coordinators a primer adaptation of their dataset to use for any articulation acknowledgment model.

B. Evaluation Metrics

The models were trained on the training set and their performance on the test set was evaluated using a confusion matrix and its corresponding metrics.

Confusion Matrix – A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).

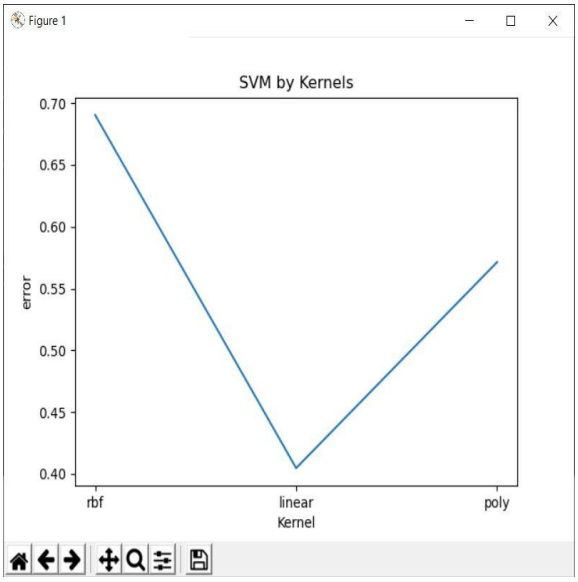
Accuracy - The sum of the correct predictions for

each class divided by the total number of samples in the test set.

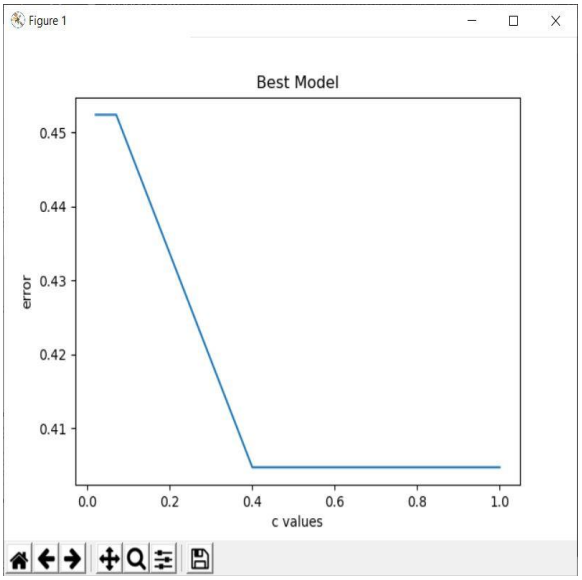
Some of the examples from the models are as follows:

1. Support Vector Machine:
   Hyper-parameters required for the model:
   • C: It is the regularization parameter, C, of the error term.
   • Kernel: It specifies the kernel type to be used in the algorithm. It can be linear, poly, rbf, sigmoid, precomputed, or a callable. The default value is rbf.
   • Degree: It is the degree of the polynomial kernel function (poly) and is ignored by all other kernels. The default value is 3.
   • Gamma: It is the kernel coefficient for rbf, poly, and sigmoid. If gamma is auto, then 1/n features will be used instead.

Hyper Parameters	Values
C	0.001
Number of iterations	10000
Decision Function	ovr
Number of classes	7
kernel	linear



Comparison Graph for Kernel Types



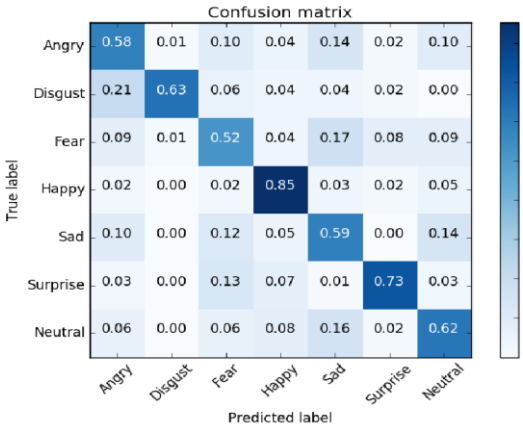
Comparison Graph for C Values

The SVM model is tested on different kernel types and C values, then the best of them is picked as the final model. Using that final model, the results of the evaluation are generated.

2. Convolution Neural Networks:
   • Learning Rate: Quantifies the learning progress of a model in a way that can be used to optimize its capacity.
   • Number of Hidden Units: Key to regulate the representational capacity of a model.
   • Convolution Kernel Width: it influences the number of parameters in a model which, in turns, influences its capacity.

Hyper Parameters	Values
Learning Rate	0.001
Number of iterations	30001
Batch Size	50
Number of classes	7
Kernel Radius	4

An example of a confusion matrix for the Convolution Neural Network is as follows:



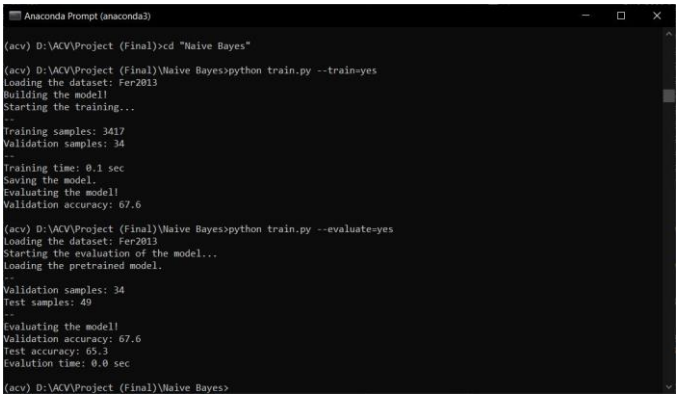
Following the evaluation of the models, the most accurate models were used to measure the accuracy of the four subsets of parameters in predicting the facial expressions.

C. Major Results:

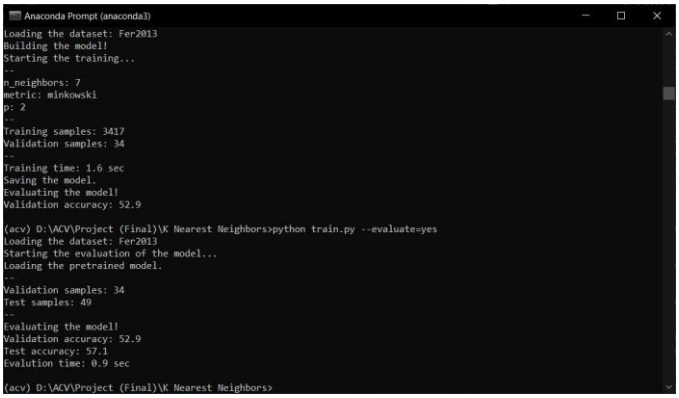
In the model evaluation phase, we found that all the models struggled to accurately predict the correct facial expressions. This is evidenced by shuffling the dataset and then running the different models on the dataset several times. The Convolutional Neural Networks and Naïve Bayes Classifier were the most accurate models, and although they did not have the highest scores in all the measured metrics, they did exhibit some stable and good overall scores. One more thing to note is that, the K-Nearest Neighbors also did a pretty decent job due to its nature of finding the nearest image from the training samples.

Model	**Accuracy**
Convolution Neural Network	65.4
Random Forest	51.0
Support Vector Machine	48.5
Logistic Regression	55.1
Naïve Bayes	65.3
Decision Tree	46.9
K-Nearest Neighbors	57.1

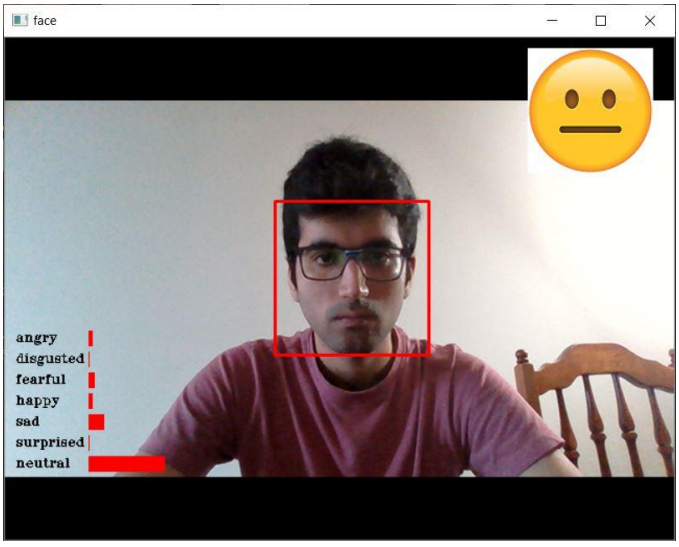
Here are some of the examples of the results,



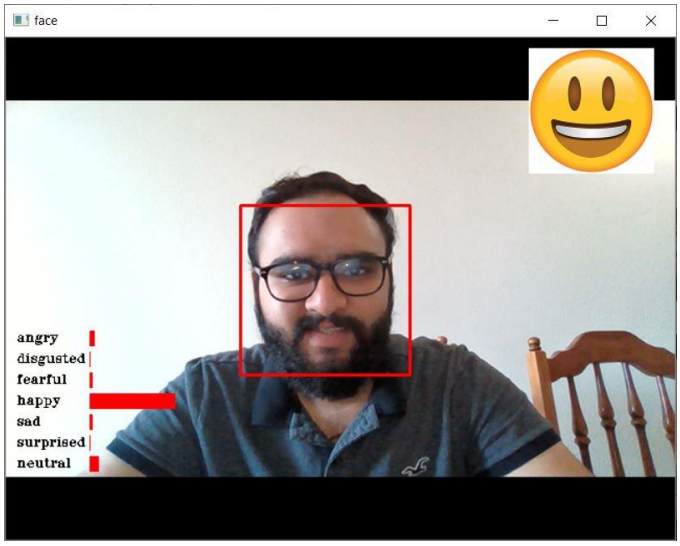
Naïve Bayes



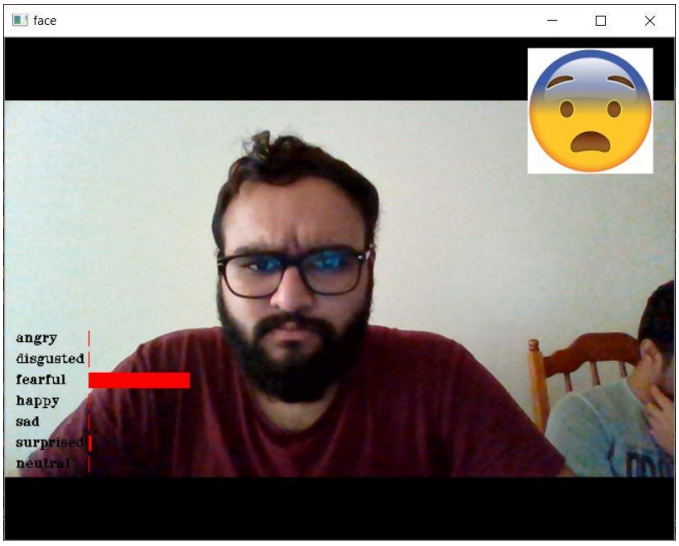
K-Nearest Neighbors



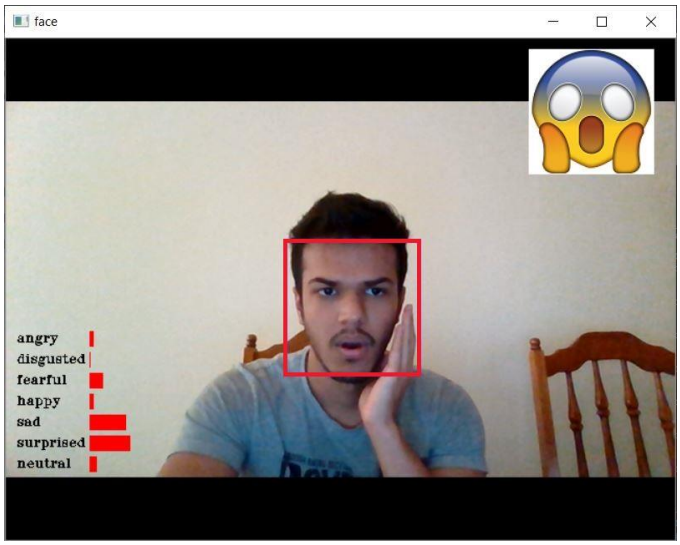
Output: Neutral Face



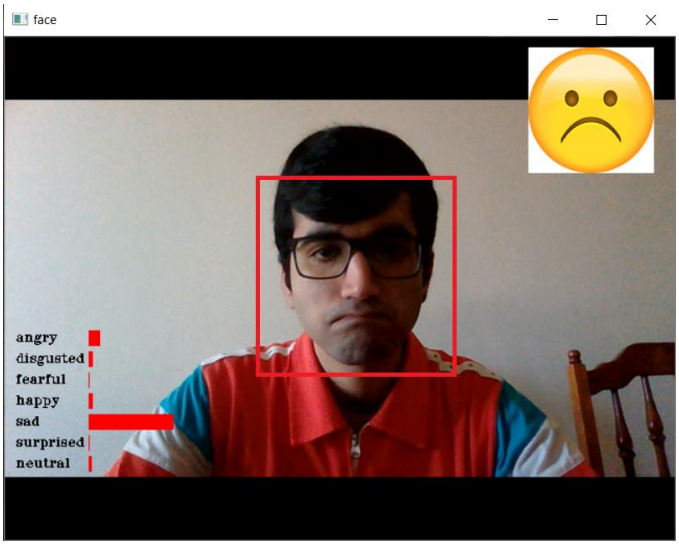
Output: Happy Face



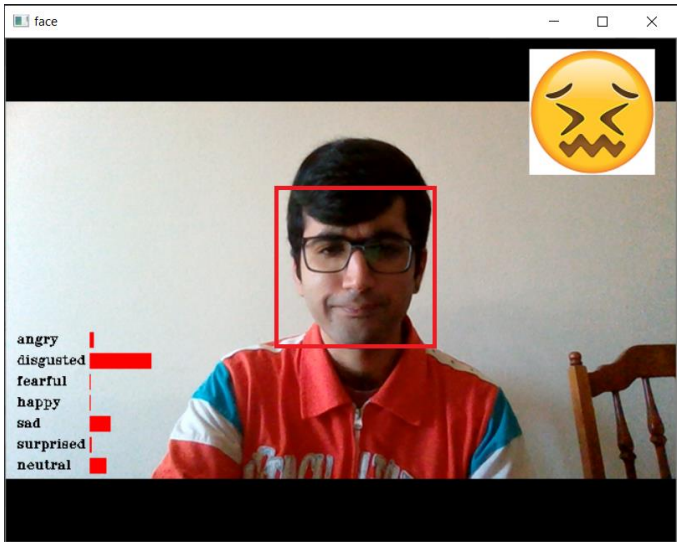
Output: Fearful Face



Output: Surprised Face



Output: Sad Face



Output: Disgusted Face

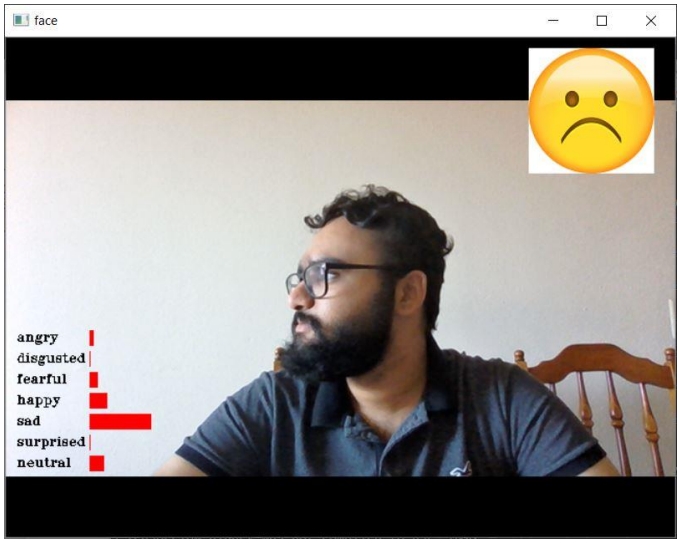
D. Analysis

The task was mainly to compare the results of the different models and conclude the good models for the problem statement. As it can be seen from the results in the above table, the accuracy received from using CNN, Naïve Bayes and KNN were good as compared to the other models.

As mentioned in the results section, all the models struggled to predict the facial expressions. Their average accuracy for prediction among various data shuffles were around 55%. The uneven distribution of the images in the different classes caused the models to exhibit a bias towards certain classes. Also, the dataset contained images of the people not facing in the direction of the camera which resulted in the failure of detection of their faces. This caused

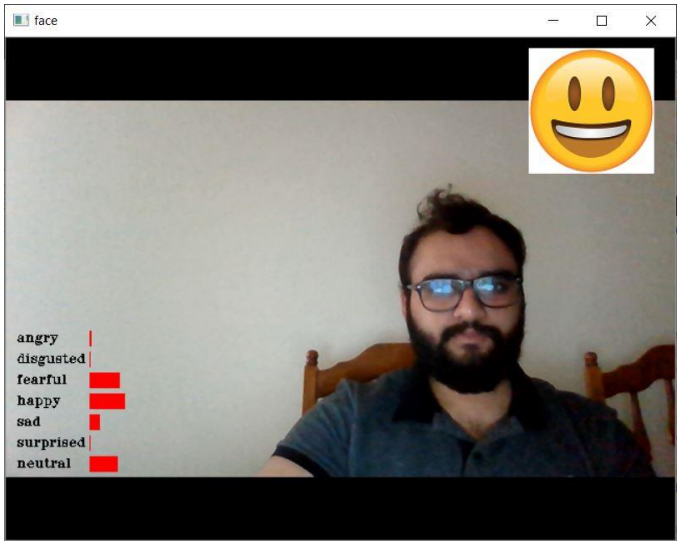
the samples in the processing to be lower than actual.

Failure Cases: Even though CNN is the best model, the accuracy attained is only 65%. Few of the failure cases for it are explained here.
• The model does not recognize the correct expressions in case the image is not front facing. If the image is sideways, the expression detected is generally wrong.



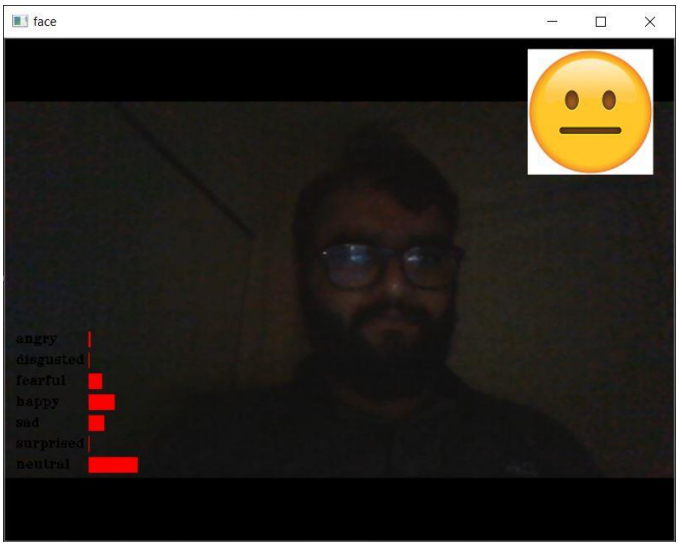
Expected Output: Neutral Face
Actual Output: Sad Face

• The second case can be defined as the case when the face is present at some specific angle and position which is not recognized by the algorithm.



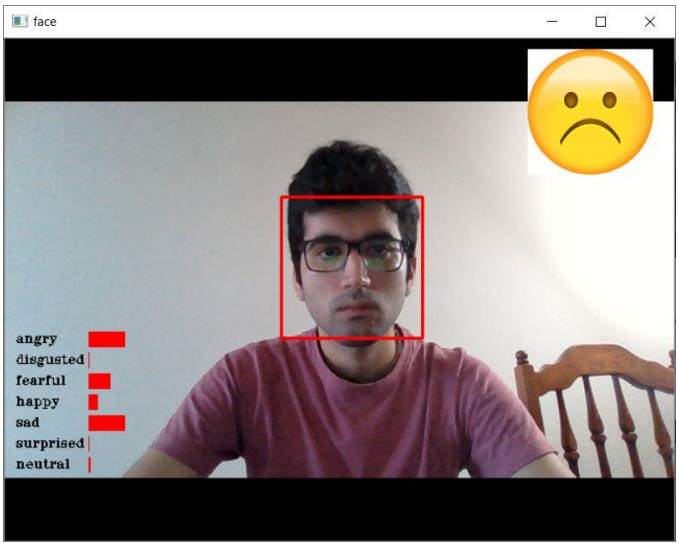
Expected Output: Neutral Face
Actual Output: Happy Face

• Another of the failure cases can be considered in the case of lighting issues. There are significant errors in finding the correct expressions when the image present for the testing does not have the correct lighting.



Expected Output: Happy Face
Actual Output: Neutral Face

• There is also significant confusion between a few expressions, and they are generally wrongly detected. Example: there's always a significant confusion between expressions for Sad and Neutral faces as well as between Angry and Disgust faces.



Expected Output: Neutral Face
Actual Output: Sad Face

V. CONCLUSION AND FUTURE WORKS

With the rising cutting-edge innovations in equipment and sensors, FER frameworks have been created to help true application scenes, rather than research facility situations. In spite of the fact, the research facility-controlled FER frameworks accomplish moderately high precision, around 65%, the specialized moving from the lab to true applications faces an extraordinary hindrance of exceptionally low exactness, roughly half.

From the above accuracies, it is evident that the Convolution Neural network and Naïve Bayes are better in performance than the other models. However, the dataset was highly unbalanced, which caused random results during each shuffle of the dataset. Due to this fact, we cannot conclusively state that Convolution Neural Network and Naïve Bayes are the optimal model for this type of classification problem. It is possible that with a larger, more balanced dataset, another model would have been more accurate. But for this dataset, the optimal model were Convolution Neural Network and Naïve Bayes. Also, even though the CNN is the best model, the accuracy received from it is still 65% which is not high enough. Therefore, there's a huge scope improvement in the same.

There are several ways in which the performance can be improved for the models -

• Increasing the number of features.

• Tuning Parameters - To improve CNN model performance, we can tune parameters like epochs, learning rate etc. We need to do certain experimentation for deciding epochs, learning rate. We can see after certain epochs there is not any reduction is training loss and improvement in training accuracy. Accordingly, we can decide number of epochs. This way, the performance can be improved.

• Training a more complex/deeper model. Currently, only 2 hidden layers are being used. We can increase the layers and check the difference in accuracies received.

• Decrease Regularization, over-regularization can make the model underfit thus, we can reduce regularization in our model to improve performance.

• Data Augmentation - Image augmentation parameters that are generally used to increase the data sample count are zoom, shear, rotation, preprocessing function and so on. Usage of these parameters results in generation of images having these attributes during training of model. Image samples generated using image augmentation, in general existing data samples increases by the rate of nearly 3x to 4x times.

• Progressive Resizing is another popular method to improve the CNN model performance using transfer learning in which we do re-using of layers and weights from previous models and build new ones.

References

[1] Cowie R., Douglas-Cowie E., Tsapatsoulis N., Votsis G., Kollias S., Fellenz W., Taylor J.G. Emotional expression recognition in human- computer interaction. IEEE Signal Process. Mag. 2001;18:3280. doi: 10.1109/79.911197 [CrossRef].

[2] Liu M., Li S., Shan S., Wang R., Chen X. Deeply learning deformable facial action parts model for dynamic expression analysis; Proceedings of the Asian Conference on Computer Vision; Singapore. 15 November 2014; pp. 143157. [Google Scholar]

[3] Gan Q., Wu C., Wang S., Ji Q. Posed and spontaneous facial expression differentiation using deep Boltzmann machines; Proceedings of the 2015 International Conference on Affective Computing and Intelligent Interaction (ACII); Xian, China. 2124 September 2015; pp. 643648. [Google Scholar]

[4] Revina I.M., Emmanuel W.R.S. A Survey on Human Face Expression Recognition Techniques. J. King Saud Univ. Comput. Inf. Sci. 2018 doi: 10.1016/j.jksuci.2018.09.002. [CrossRef]

[5] Kumar Y., Sharma S. A systematic survey of facial expression recog- nition techniques; Proceedings of the 2017 International Conference on Computing Methodologies and Communication (ICCMC); Erode, India. 1819 July 2017; pp. 10741079. [Google Scholar]

[6] Ko B. A Brief Review of Facial Emotional expression recognition Based on Visual Information. Sensors. 2018;18:401. doi: 10.3390/s18020401. [CrossRef]