Facial Expression Recognition System

Himani Singh
Dept of Computer Science
San Diego State University
California, USA
RED ID: 821691995
singh.himani0406@gmail.com

Pritesh Amrelia
Dept of Computer Science
San Diego State University
California, USA
RED ID: 823720567
priteshamrelia@gmail.com

Yuan Wang
Dept of Computer Science
San Diego State University
California, USA
RED ID: 815261129
wangyuan880216@gmail.com

***Abstract*—Facial appearance Recognition (FER) can be broadly applied to different research fields, for example, mental infection detection and human social/physiological collaboration detection. 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 accomplishes moderately high precision, around 65%, the specialized moving from the lab to true applications faces an extraordinary hindrance of exceptionally low exactness, roughly half.**
This research model will use 28,709 images to categorize each face based on the emotion in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral) categories. This research will make use of Convolutional Neural Network and "Support Vector Machine" models as they are designed specifically for handling image data and will cover the importance of using CNN model vs SVM in such classification problems.

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 up that can perceive feelings also. 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. Artificial intelligence can identify feelings by realizing what every facial appearance implies and applying that information to the new data displayed to it. Enthusiastic man-made reasoning, or feeling AI, is an innovation that is fit for perusing, impersonating, translating, and reacting to human facial expressions and feelings.
Understanding relevant emotion has far reaching ramifications for society and business. In the open circle, administrative organizations could utilize the capacity to distinguish emotions like guilt, fear, and uncertainty. It's not hard to imagine the TSA auto-checking aircraft passengers for indications of terrorism, and in the process making the world a more secure place. Organizations have additionally been exploiting emotion recognition to drive business results. For the up and coming arrival of Toy Story 5, Disney intends to utilize facial recognition to pass judgment on the passionate reactions of the group of spectators. Apple even released a new feature on the iPhone X called Animoji, where you can get a machine simulated emoji to mimic your facial expressions. It's not so far away to accept they'll utilize those abilities in different applications soon.
In this project, we tried to solve the same problem using the CNN and SVM model and Fer2013 dataset which contains 28,709 images of 7 different emotions.

II. Task Description

A. Support Vector Machines

A Support Vector Machine (SVM) is a model which is defined by a separating hyperplane. Here we take the support of support vectors in order to classify the data Given labeled training data the algorithm outputs an optimal hyperplane which categorizes new data. In two-dimensional space this hyperplane is a line dividing a plane in two parts.
If a data-set is linearly separable it means that we could use a Hard margin approach, or rather find the two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible and so we are being able to identify to which class belongs each point of the data set
The file convert_fer2013_to_images_and_landmarks.py is used to convert the dataset to extract Face Landmarks and HOG features. The file train.py is used to train the model and save the trained model as well as generating the testing results. The code also tests the model for various kernel types and C value and generates the best model. The file optimize_parameters.py is used to generate the minimized results for provided combinations of the hyper parameters while the file parameter.py is used to set default parameters.

B. Convolution Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which takes in an input image, assign weights to various features(pixels) in the images and have the ability to separate one from the other. The pre-processing required in a ConvNet is much lower when contrasted with other order algorithms. While in primitive strategies filters are manually designed by providing enough training, ConvNets can gain proficiency with these channels/attributes.
The data folder contains folders for fer2013 which contains the dataset file and the emojis folder which contains the 7 emojis for all the 7 emotions recognized by the model. The

script main.py has to be specified the task to be performed like providing the input as "train" while running the python file trains the model. Similarly, various other input arguments like "test" or "validate" can be provided for the respective tasks.
The file model.py loads the pre-trained module and trains a new model on top of expressions data.The script will write out the new model trained on the facial expressions categories in model/emotion_model.pb. Convolutional Neural Network uses two hidden layer to train data to classify them to appropriate facial expression category.

III. Major Challenges and Solutions

A. Challenges

The different difficulties that we understood with the Emotion arrangement dataset are: 1) The dataset for order is enormous and contains around 28,709 examples for different classes of feelings. The size of the real dataset is around 2Gb. However, we have to decrease the size of the examples for quicker handling and model preparing. 2) The greatest downsides of the enormous dataset is time taken for processing and considering irrelevant data and in this way wasting the assets. 3) It is precarious to pre-process the tests of differing estimate and diminish the size of each example to 48*48.

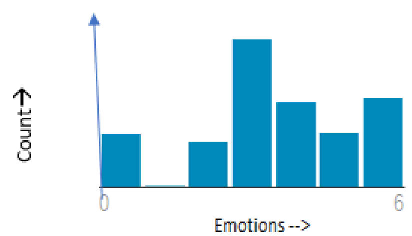
B. Solution

Along with the various emotions for human expressions, there were a lot of images that didnt belong to any category and had invalid images. Further data needs to be cleaned. All these steps will utilize less resources also reducing the time taken. Since the classification dataset is humongous, a rational approach is 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 48*48 using Image Resizer for Windows(Link in references) so that every image is of same size, resources consumed during the processing will be less and training on the model will faster as the size of the dataset will be reduced tremendously. After all the pre-processing on the data the dataset got reduced to 294Mb and processing time reduced which helped us to train the model faster. Now our dataset is 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 pixel grayscale. The faces have been naturally enlisted with the goal that the face is 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 in the facial expression in to one of seven classes (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). train.csv contains two columns, "emotion" and "pixels". The "emotion" segment contains a numeric code extending from 0 to 6, comprehensive, 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 a space-isolated pixel values in row major order. test.csv contains just the "pixels" column and your task is to predict the emotion column.
The preparation set comprises of 28,709 models. The test set utilized for the model comprises of 3,589 models. The last test set, which was utilized to decide the outward appearances comprises of another 3,589 models.
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.



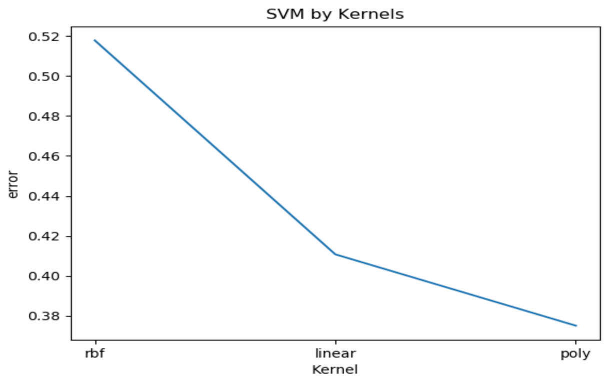
B. Evaluation Metrics

Emotion Classes: Happy, Sad, Anger, Disgust, Neutral, Surprise, fear

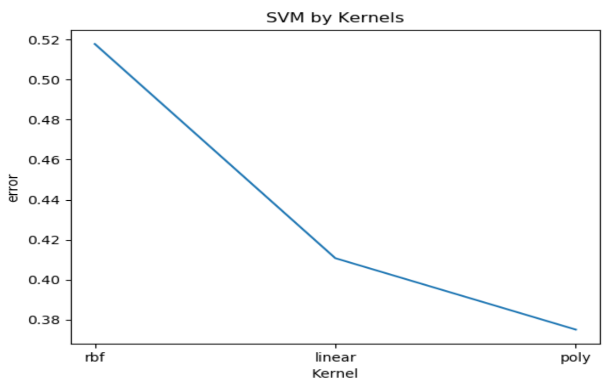
1) Support Vector Machine: Hyperparameters required for the model

- 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	rbf

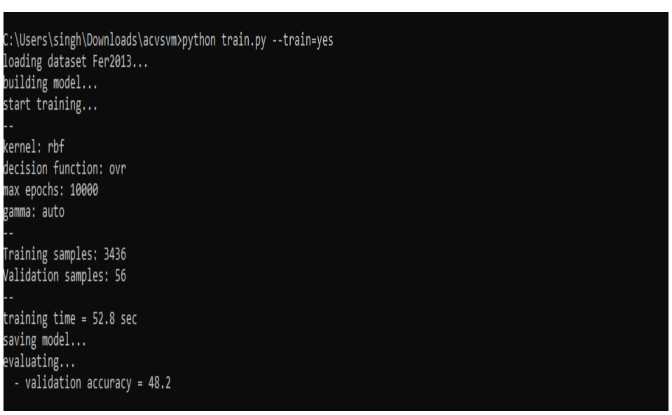


Comparison graph for kernel types

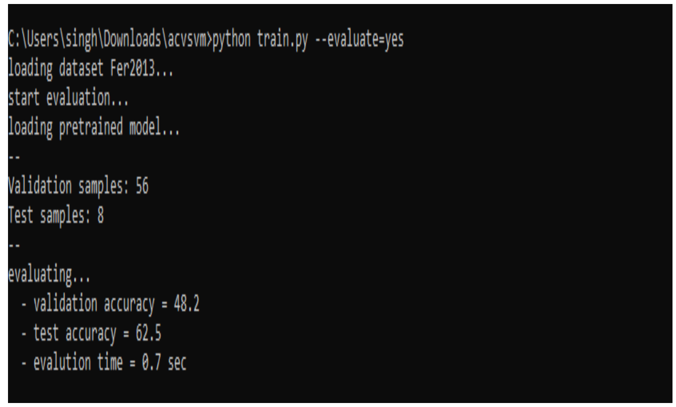


Comparison graph for C values

The svm model is tested for different kernel types and C values and the best of them is picked as the final model and the results are generated using the same.



```
C:\Users\singh\Downloads\acvsvm>python train.py --train=yes
loading dataset Fer2013...
building model...
start training...
--
kernel: rbf
decision function: ovr
max epochs: 10000
gamma: auto
--
Training samples: 3436
Validation samples: 56
--
training time = 52.8 sec
saving model...
evaluating...
  - validation accuracy = 48.2
```

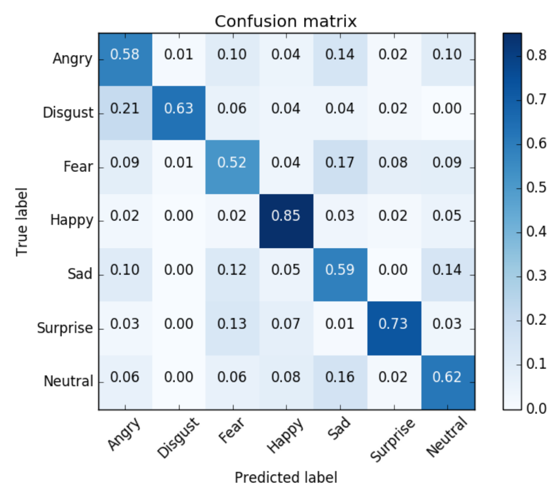


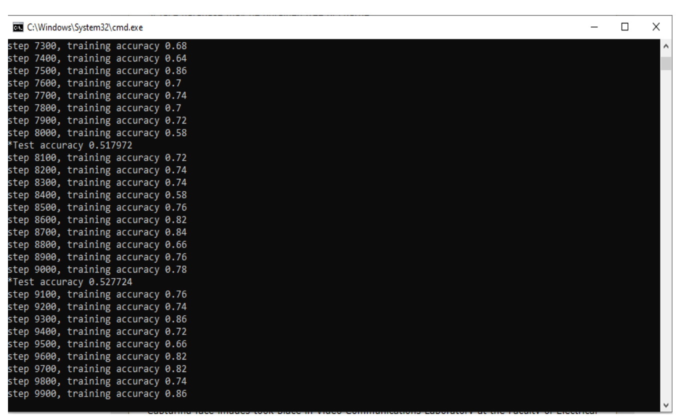
```
C:\Users\singh\Downloads\acvsvm>python train.py --evaluate=yes
loading dataset Fer2013...
start evaluation...
loading pretrained model...
--
Validation samples: 56
Test samples: 8
--
evaluating...
  - validation accuracy = 48.2
  - test accuracy = 62.5
  - evalution time = 0.7 sec
```

2) Convolution Neural Network: Hyperparameters required for the model

- 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

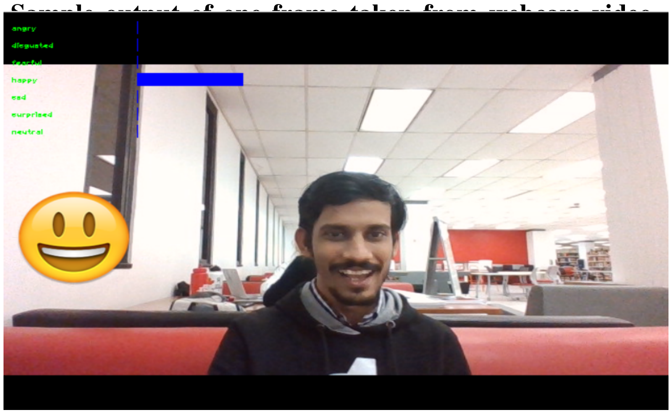




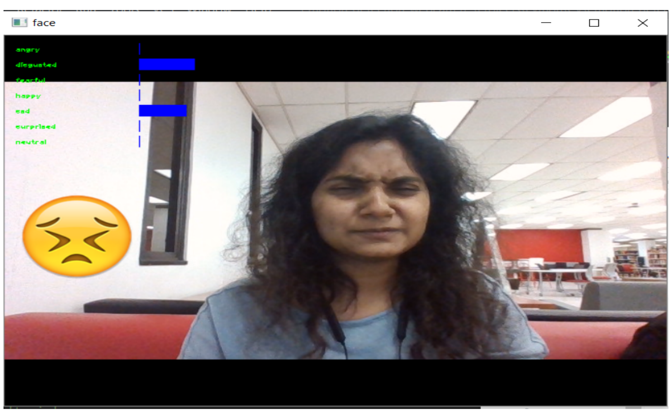
```
C:\Windows\System32\cmd.exe
step 7300, training accuracy 0.68
step 7400, training accuracy 0.64
step 7500, training accuracy 0.86
step 7600, training accuracy 0.7
step 7700, training accuracy 0.74
step 7800, training accuracy 0.7
step 7900, training accuracy 0.72
step 8000, training accuracy 0.58
*Test accuracy 0.517972
step 8100, training accuracy 0.72
step 8200, training accuracy 0.74
step 8300, training accuracy 0.74
step 8400, training accuracy 0.58
step 8500, training accuracy 0.76
step 8600, training accuracy 0.82
step 8700, training accuracy 0.84
step 8800, training accuracy 0.66
step 8900, training accuracy 0.76
step 9000, training accuracy 0.78
*Test accuracy 0.527724
step 9100, training accuracy 0.76
step 9200, training accuracy 0.74
step 9300, training accuracy 0.86
step 9400, training accuracy 0.72
step 9500, training accuracy 0.66
step 9600, training accuracy 0.82
step 9700, training accuracy 0.82
step 9800, training accuracy 0.74
step 9900, training accuracy 0.86
```

C. Major Results

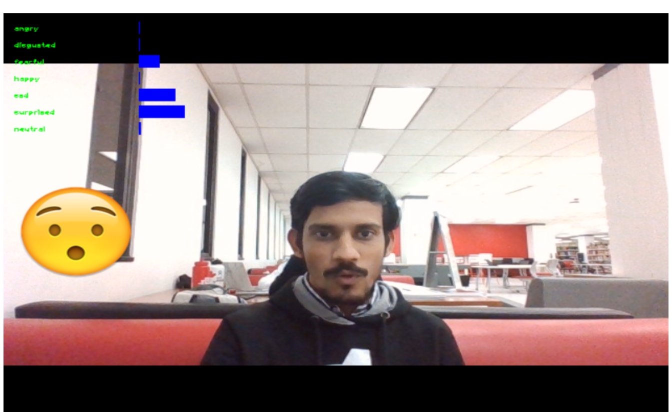
Sample output of one frame taken from webcam video



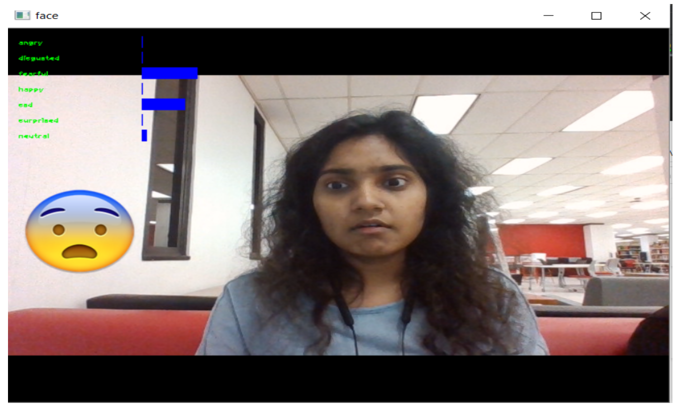
Expected Output: Happy Face



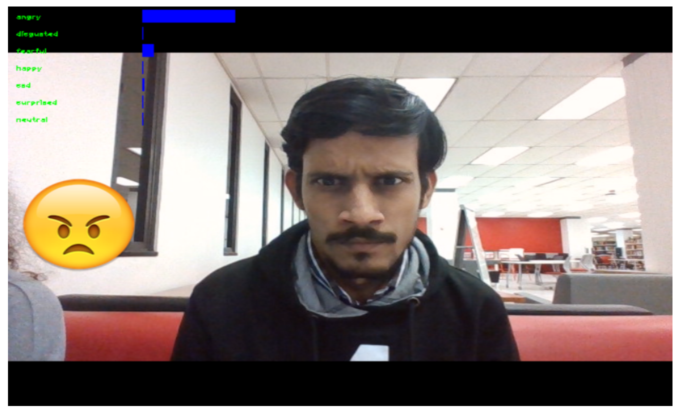
Expected Output: Disgusted Face



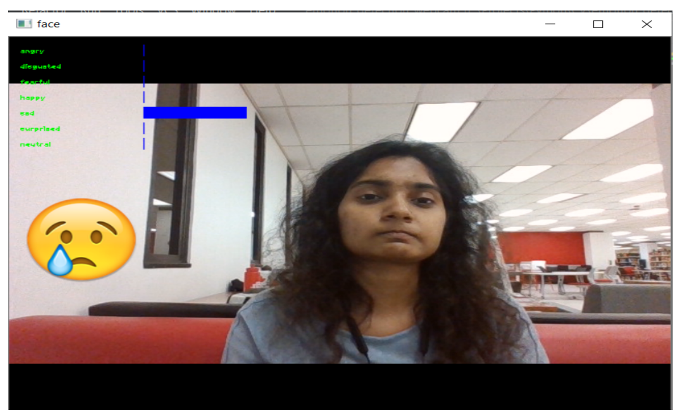
Expected Output: Surprised Face



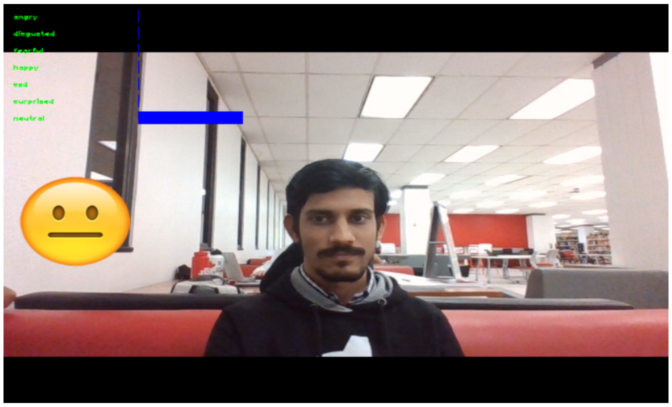
Expected Output: Fearful Face



Expected Output: Angry Face



Expected Output: Sad Face



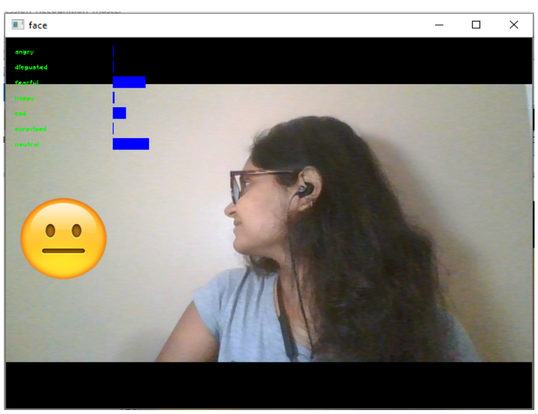
Expected Output: Neutral Face

D. Analysis

The task was mainly to compare the results of models cnn and svm and conclude the best model for the problem statement. As it can be seen from the results above, the accuracy received from using the CNN is better than that of SVM.
The CNN model generates an accuracy of approx. 65% while the accuracy of SVM is approx 58-60%.

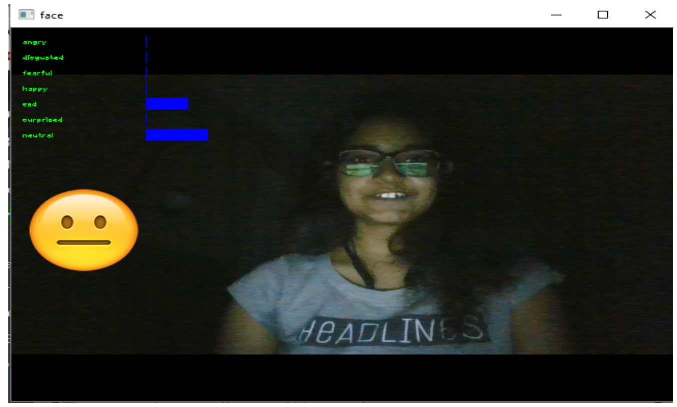
1) Failure Cases: Even though CNN is the best model, the accuracy attained is only 65%. Few of the failure cases for it are explained here.

- The model does not recognize the correct expressions in case the image is not front facing. If the image is side-ways, the expression detected is generally wrong.



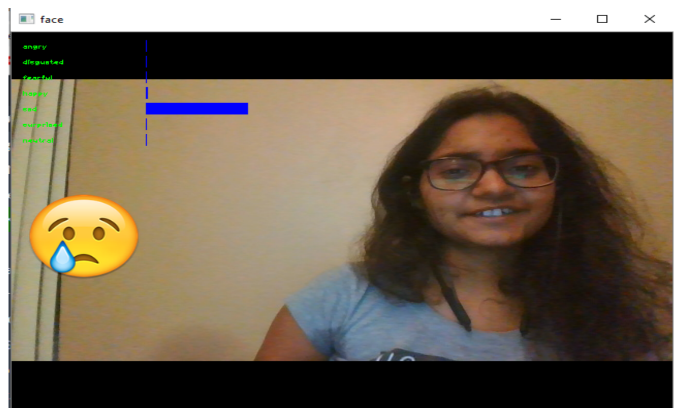
Expected Output: Happy face
Output received: Neutral face

- 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.



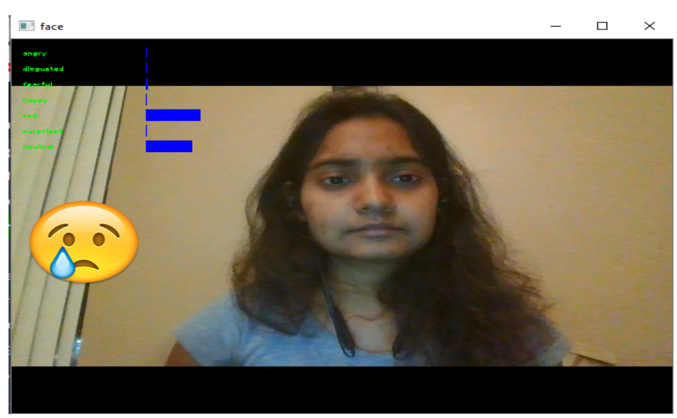
Expected Output: Happy face
Output received: Neutral face

- The third case can be defined as the case when the face is not present at the center of the screen.



Expected Output: Happy face
Output received: Sad 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
Output received: Sad face

V. Conclusion and Future Works

Model	Accuracy
Support Vector machines	48.5
Convolution Neural Network	65.4

From the above accuracies, it is evident that the Convolution Neural network is way better in performance than the Support Vector Machine. 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 CNN model -

- 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 recognition 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]