Indian Institute of Technology,
Bombay

Foundations of Machine Learning CS725

PROJECT REPORT

Facial Emotion Recognition

Made by :
Mohit Agarwala (19307r004)
Navneet Ranjan (203050106)
Nisha Kumari (193300013)
Shivaji Chirumamilla (203050056)
Vivek Kumar 203050013

December 8, 2020

Abstract

In this project, we develop a ***facial emotion recognition*** *model using Convolutional Neural Network (CNN) and deploy the trained model to a cam interface that enable the users to detect facial expression in real-time.*

Our main focus was on CNN models to predict facial expressions. We built several models capable of recognizing seven basic emotions (happy, sad, angry, fear, surprise, disgust and neutral). Using the FER-13 dataset, we achieve 55% accuracy on both AlexNet and Inception models. Although, Inception model was ran for less epochs than AlexNet, because it was found to be overfitting after some point. Then we used the VGG 16 model, which gave the best result by far with an accuracy of 61%. Moving forward, we have plotted confusion matrix for all the models, and it was found that fear and disgust emotions were the least predicted, so we removed those two from training and tested our best model VGG-16 on this and achieved an accuracy of 67%. Also, we used Data Augmentation to somehow mitigate the imbalance among the data of vaious categories, this gave a boost of 3% increased accuracy on the VGG 16 model.

Flow of the report is as follows:

- Dataset and Features
- CNN Architecture
- Different CNN Models and their features
- Results and Summary
- Real time Face Detection demo
- Discussion

1 Dataset and features

We used FER-13 dataset for our project. It comprises a total of 35887 pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes:

- Angry
- Disgust
- Fear
- Happy
- Sad
- Surprise
- Neutral

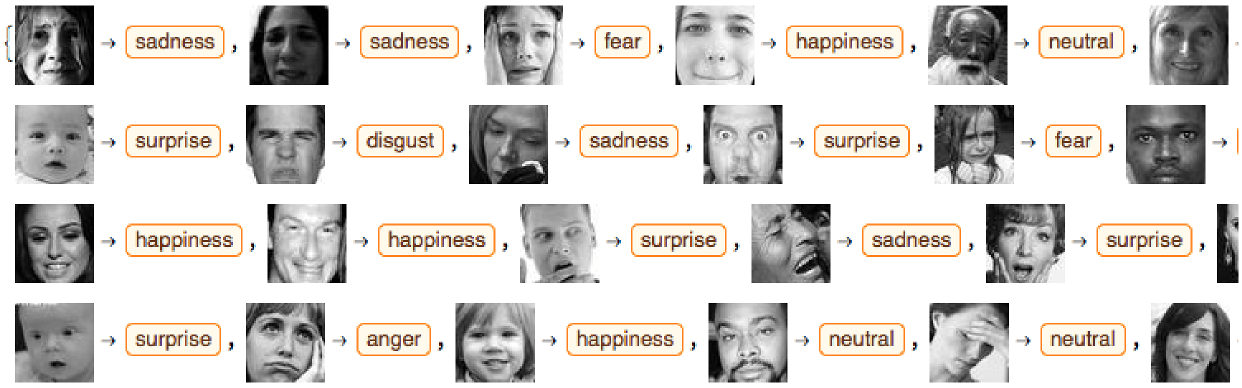

Figure 1: Dataset

Data Distribution

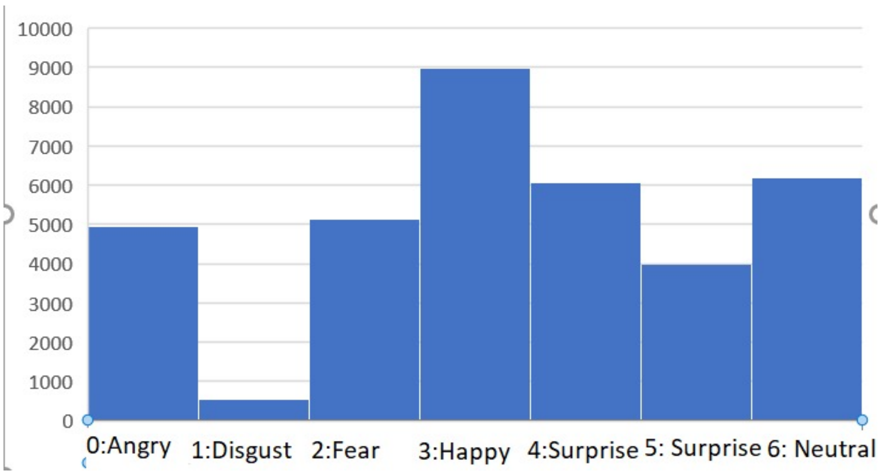

Figure 2: Distribution of various data

Shortcomings of the dataset :

- Less number of pixels
- Imbalance Problem
  - 'Disgust' Images was far less compared to others.
- Intraclass Variation
  - Over-fitting must be avoided.
- Occlusion
  - Face covered with hands.
- Contrast Variation
  - Some images are very dark and some are very light.
- Eyeglasses
- Outliers

2 CNN Architecture

Deep learning is a popular technique used in computer vision. We chose convolutional neural network (CNN) layers as building blocks to create my model architecture. CNNs are known to imitate how the human brain works when analyzing visuals. A typical architecture of a convolutional neural network will contain an input layer, some convolutional layers, some dense layers (aka. fully-connected layers), and an output layer (Figure 1). These are linearly stacked layers ordered in sequence. In Keras, the model is created as **Sequential()** and more layers are added to build architecture.

CNNs are preferred for computer vision because :

- Ability to capture spatial and temporal dependencies.
- It reduces images into forms which are easier to process without losing critical features.
- Less pre-processing is required.

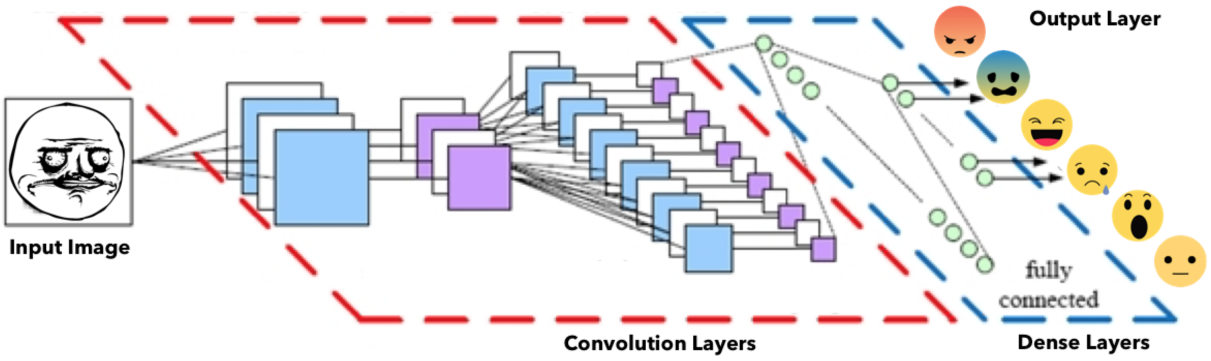

Figure 3: CNN Architecture

Convolution layers

A convolution layer is a fundamental component of the CNN architecture that performs feature extraction, which typically consists of a combination of linear and nonlinear operations, i.e., convolution operation and activation function.

Non-linear activation function

The outputs of a linear operation such as convolution are then passed through a nonlinear activation function. Although smooth nonlinear functions, such as sigmoid or hyperbolic tangent (tanh) function, were used previously because they are mathematical representations of a biological neuron behavior, the most common nonlinear activation function used presently is the rectified linear unit (ReLU), which simply computes the function:f(x)=max(0,x)

Max pooling

The most popular form of pooling operation is max pooling, which extracts patches from the input feature maps, outputs the maximum value in each patch, and discards all the other values. A max pooling with a filter of size 2*2 with a stride of 2 is commonly used in practice. This downsamples the in-plane dimension of feature maps by a factor of 2. Unlike height and width, the depth dimension of feature maps remains unchanged.

Padding

Padding is a term relevant 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.

3 Different models and their features

Various CNNs and Comparison					
Network	**Year**	**Salient Feature**	**Top5 Accuracy (ImageNet Data)**	**Parameters**	**FLOP**
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-Size kernels	92.30%	138M	19.6B
Inception	2014	Wider-parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

AlexNet :

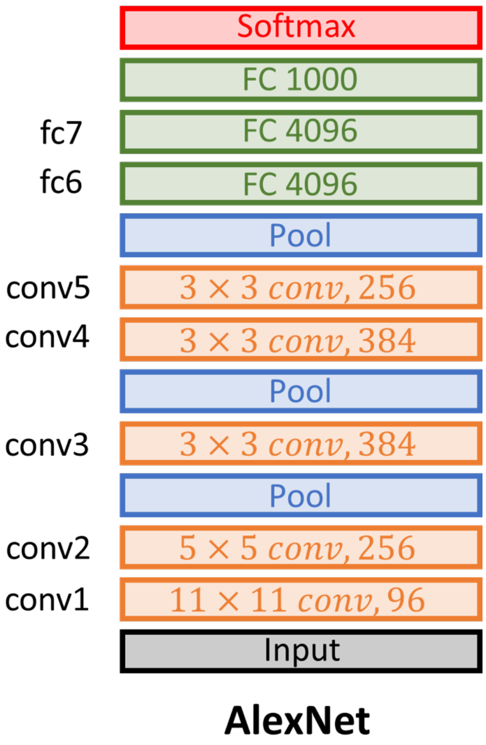

Figure 4: AlexNet

AlexNet is considered one of the most influential papers published in computer vision, having spurred many more papers published employing CNNs and GPUs to accelerate deep learning.

AlexNet contained eight layers; the first five were convolutional layers, some of them followed by max-pooling layers, and the last three were fully connected layers.[2] It used the non-saturating ReLU activation function, which showed improved training performance over tanh and sigmoid.

AlexNet won the 2012 ImageNet competition with a top-5 error rate of 15.3% compared to second place top-5 error rate of 26.2%.

Inception :

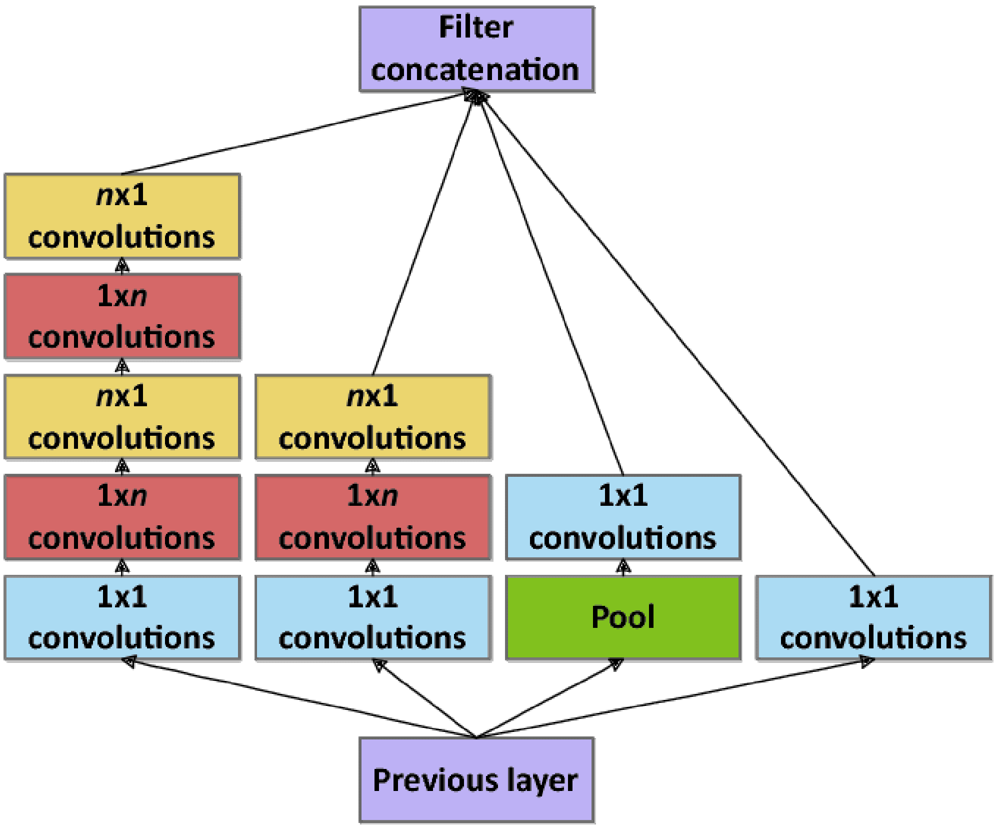

Figure 5: Inception

It uses wider and parallel kernels. Moreover, the trainable parameters are 10 times less than VGGNet and AlexNet. Hence, also the computational complexity is greatly reduced.

VGG-16 :

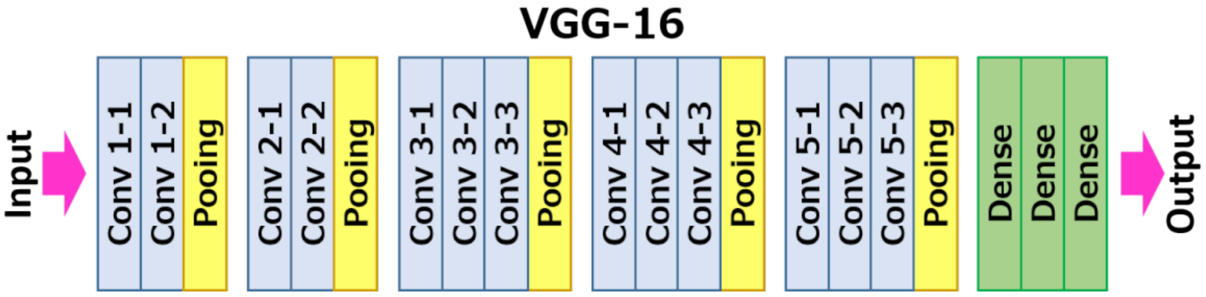

Figure 6: VGG-16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another.

4 Results and Summary

- We started by implementing AlexNet architecture, in which the maximum accuracy achieved was **55%** after 30 epochs. Also, we plotted confusion matrix for the same.

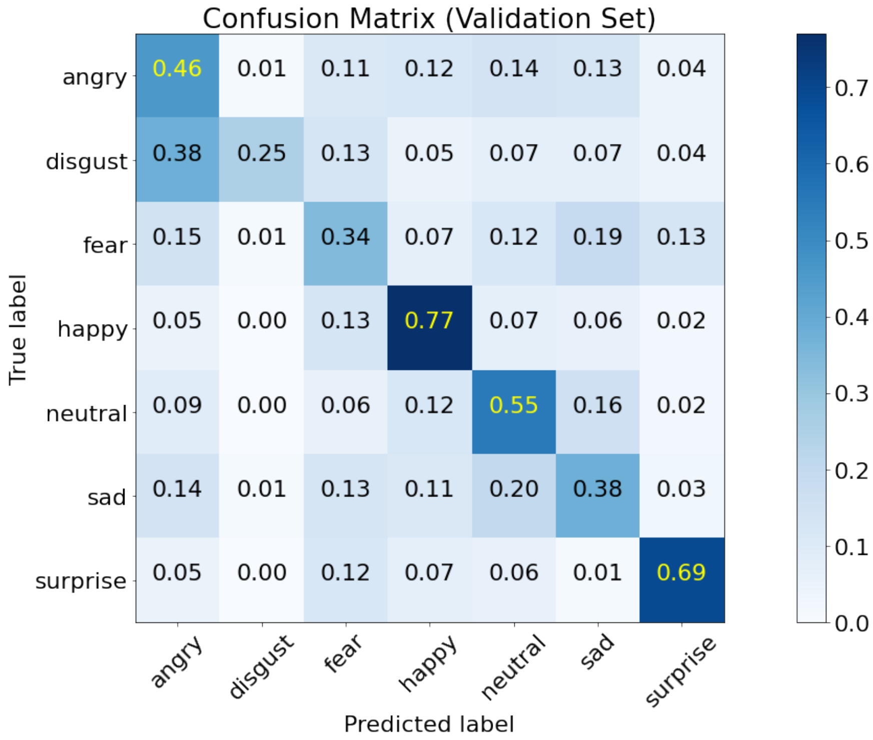

Figure 7: AlexNet confusion matrix

- Then, we implemented Inception model which gave the same accuracy i.e **55%** as before but this model suffered from **over-fitting** after some epoch as can be seen from the graph. Also, confusion matrix is similar to AlexNet.

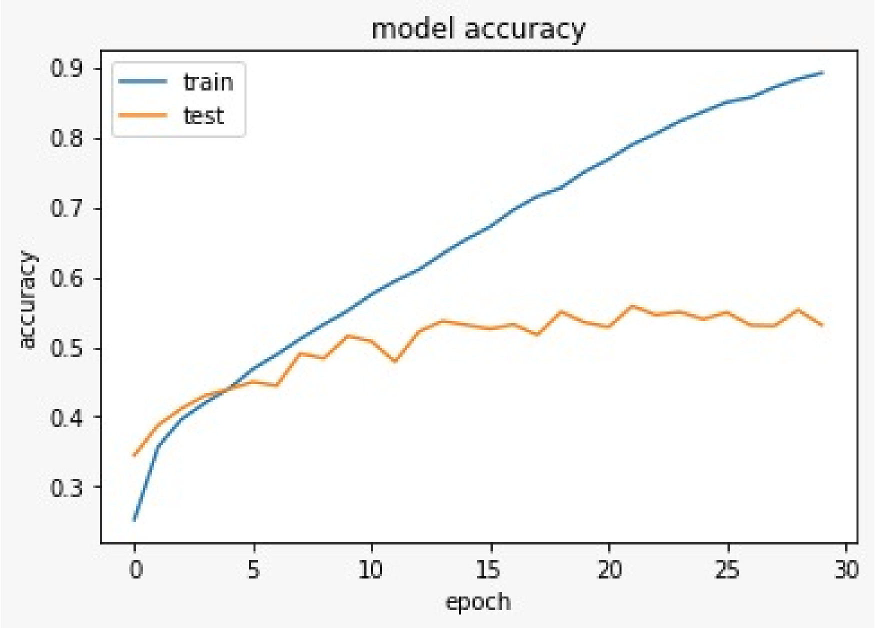

Figure 8: Accuracy Vs Epoch

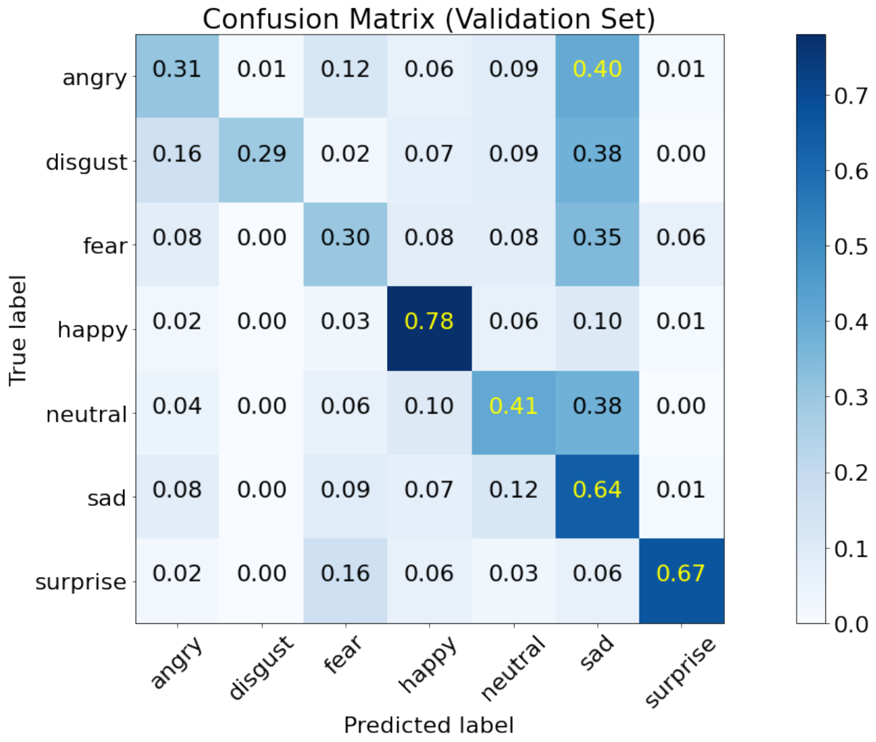

Figure 9: Inception confusion matrix

- Third, we implemented VGG-16 which gave **61%** prediction, an improvement over previous models. But this time the 'Disgust' label was not predicted at all, as can be seen from the confusion matrix.

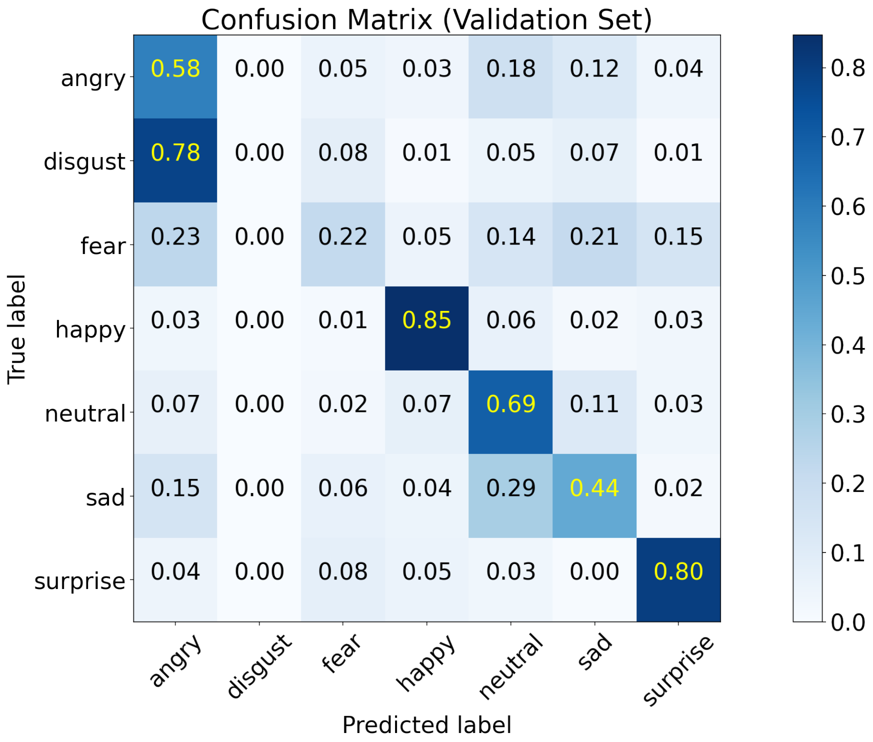

Figure 10: VGG-16 7 emotions

So, again we implemented the same model but this time with dominant 5 emotions, thereby removing 'disgust' and 'fear' we got **67%** accuracy.

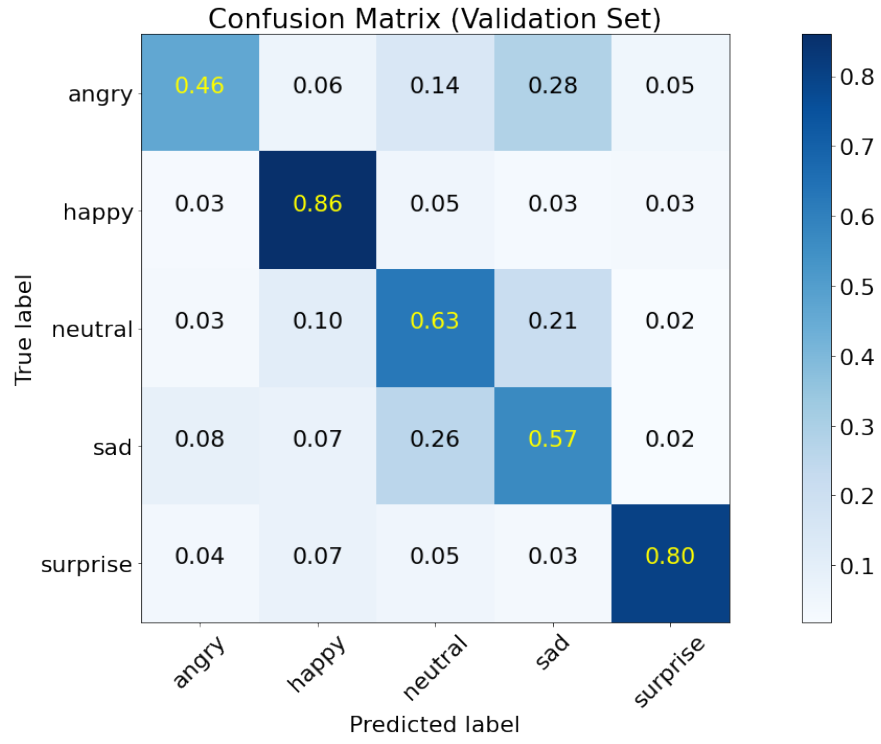

Figure 11: VGG-16 5 emotions

This gave the best performance on FER-13 dataset.

5 Real time Face Detection demo

We deployed our best model i.e, VGG-16 with 5 emotions for real time prediction. We used cv2 cascade classifier to locate faces in real time through face-cam, and a code was written to capture the images and validate using the model.

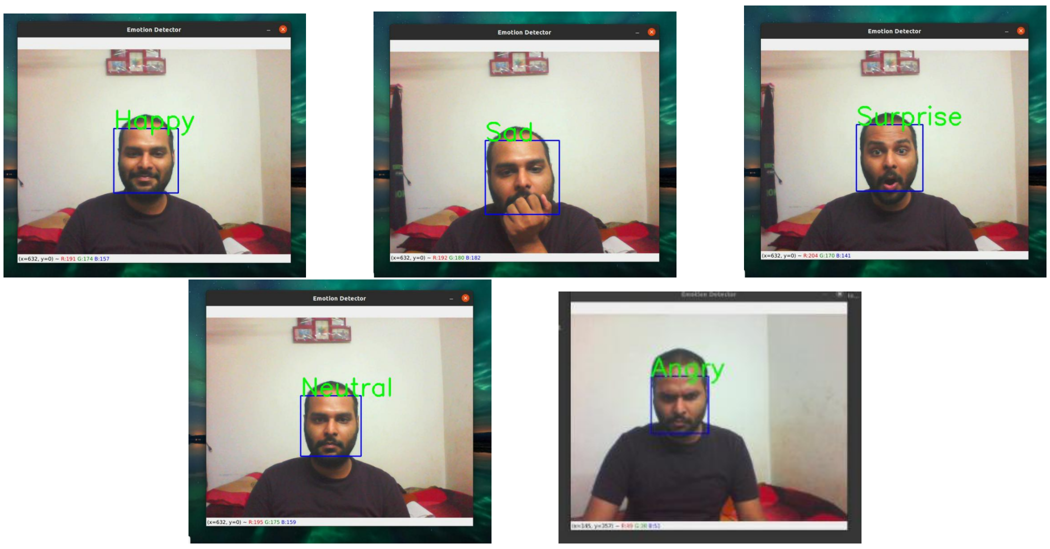

Figure 12: Screenshots from face-cam

6 Discussion

On the FER-2013 dataset, we achieve a test accuracy of 67% with our best CNN for 5 emotions and using data augmentation. Around 3-4% accuracy was alone achieved by augmenting the data in each model. The top score on the Kaggle competition using this dataset achieved an accuracy of 71%. Furthermore, human scores on the FER-2013 are accuracies of 65% +/- 5% further showing that our model is very accurate on static images.

Due to the nature of real-time classification, it is hard to get a definitive metric of our real-time system's accuracy but its worth mentioning that the model performed very accurate in good lightning conditions.

In future work, we may try to predict multiple labels e.g, gender, emotion and age on a mixture of test set.

7 References

[1] ALEXNET: *https://analyticsindiamag.com/hands-on-guide-to-implementing-alexnet-with-keras-for-multi-class-image-classification/*

[2] VGG 16: *Tensorflow and keras structure*

[3] Inception model: *https://www.youtube.com/watch?v=fq2srEX7VV0*

[4] *https://github.com/JostineHo/mememoji1-motivation*

Link to Git hub Repository: https://github.com/mohit-iitb/CS-725-Project-2020.git