Capstone Project

Live Class Monitoring Face Emotion Recognition

Sharath Diwakar

Contents:

Introduction

Facial emotion recognition is the process of detecting human emotions from facial expressions. The human brain recognizes emotions automatically, and software has now been developed that can recognize emotions as well. This technology is becoming more accurate all the time, and will eventually be able to read emotions as well as our brains do.

Applications:

- Product Development
- Video Game

Problem Statement

The Indian education landscape has been undergoing rapid changes for the past 10 years owing to the advancement of web-based learning services, specifically, eLearning platforms.

In a physical classroom during a lecture, the teacher can see the faces and assess the emotion of the class and tune their lecture accordingly, whether he is going fast or slow. He can identify students who need special attention. Digital classrooms are conducted via a video telephony software program (ex-Zoom) where it's not possible to see all students and access the mood. Because of this drawback, students are not focusing on content due to a lack of surveillance.

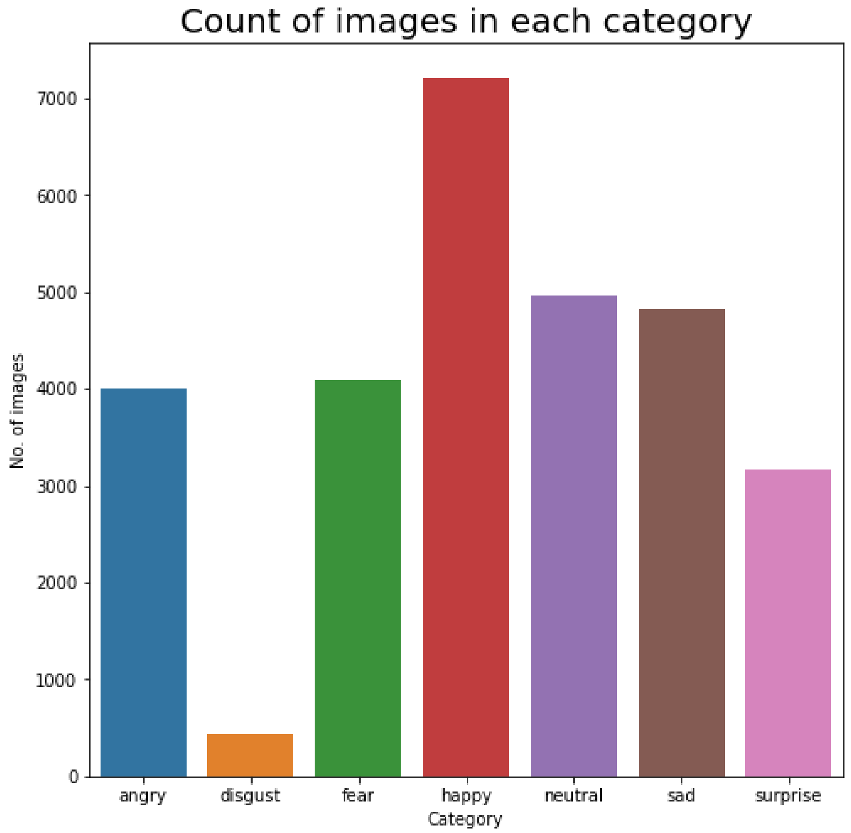
Digital platforms have limitations in terms of physical surveillance but it comes with the power of data and machines which can work for you. Its data can be analysed using deep learning algorithms which not only solves the surveillance issue but also removes the human bias from the system.

Data Summary

The data comes from the past Kaggle competition “Challenges in Representation Learning: Facial Expression Recognition Challenge”: we have defined the image size to 48 so each image will be reduced to a size of 48x48.

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. Each image corresponds to a facial expression in one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The dataset contains approximately 36K images.

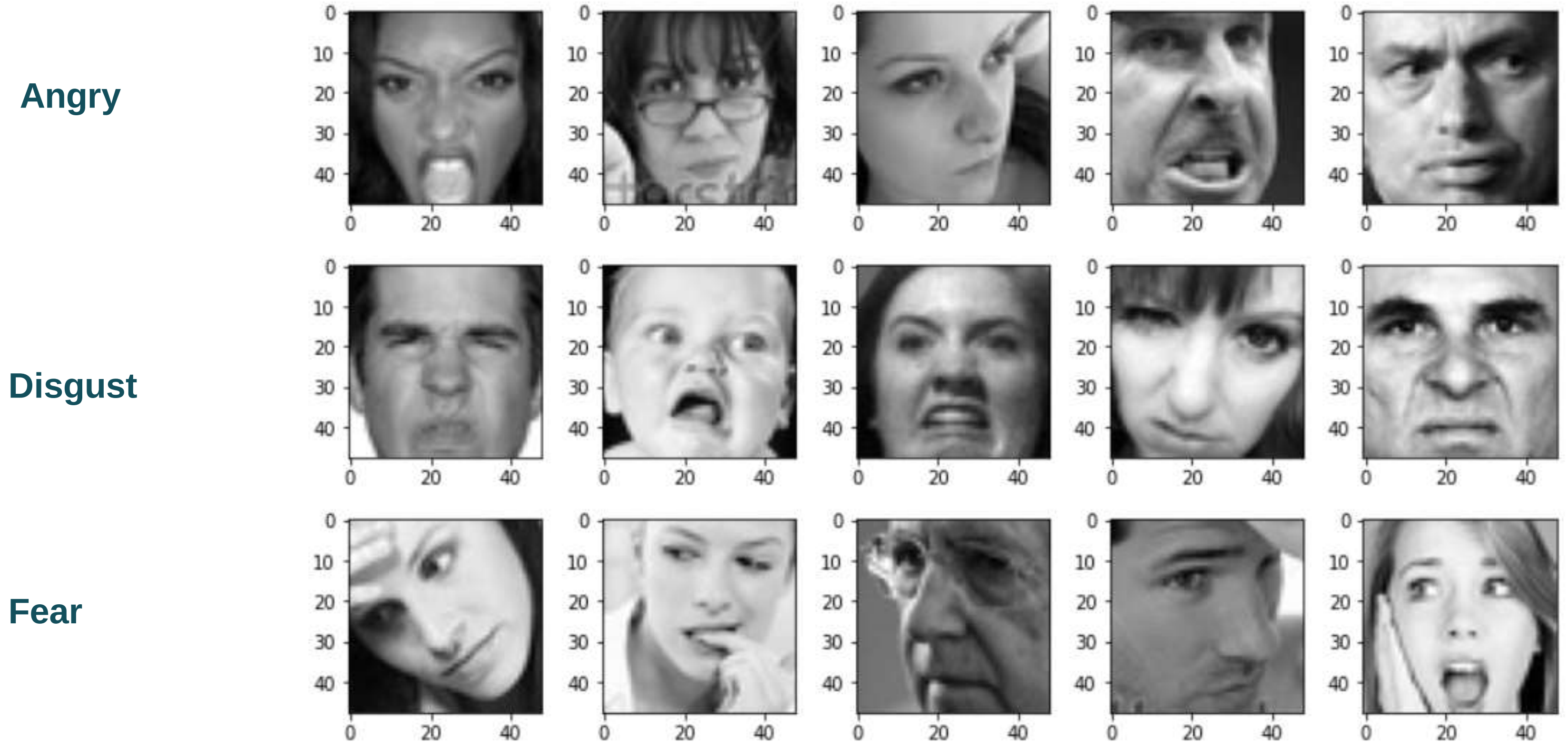
Data Summary



Data Summary

Label	Emotion	No. of images for Training	No. of images for Testing
0	Angry	3995	958
1	Disgust	436	111
2	Fear	4097	1024
3	Happy	7215	1774
4	Sad	4830	1247
5	Surprised	3171	831
6	Neutral	4965	1233

Data Summary



Data Summary

Happy

Sad

Neutral

Surprised

Dependencies

1. Python 3
2. OpenCv
3. DeepFace
4. Streamlit
5. Streamlit-Webrtc
6. Tensorflow 2.0
7. Heroku

Pipeline

Data Exploration

Understanding the Data

- Types of emotions
- Images in each category
- It’s properties

Modeling

Modeling structures

- Deep Face
- Transfer Learning
- CNN

Model Evaluation & Deployment

Graphs and applications

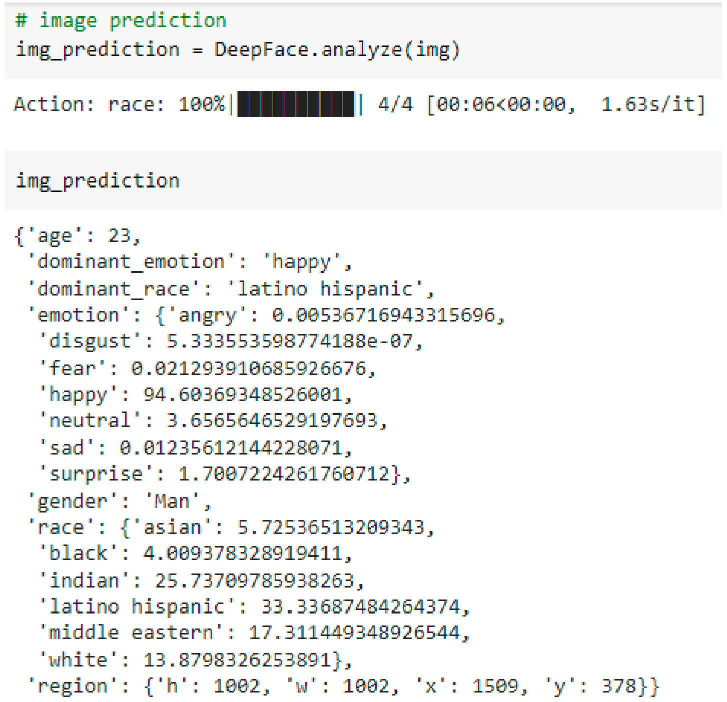
- Loss & accuracy plots
- Confusion matrix (Heatmap)
- Streamlit Share
- Heroku

DeepFace

DeepFace is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for python.

Test Image

Result:



```
# image prediction
img_prediction = DeepFace.analyze(img)
```

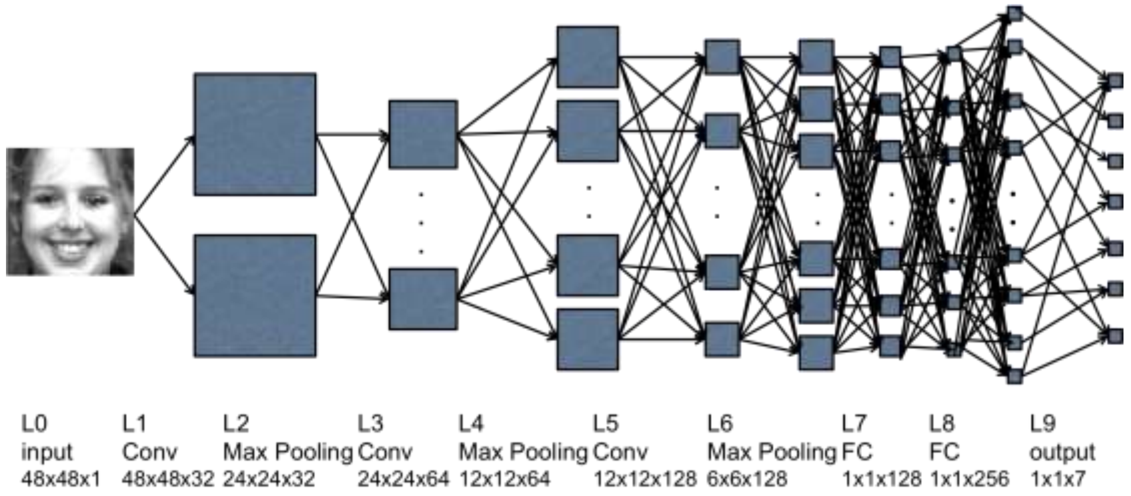
```
Action: race: 100%|██████████| 4/4 [00:06<00:00,  1.63s/it]
```

```
img_prediction
```

```
{'age': 23,
 'dominant_emotion': 'happy',
 'dominant_race': 'latino hispanic',
 'emotion': {'angry': 0.005367169433l5696,
  'disgust': 5.333553598774188e-07,
  'fear': 0.021293910685926676,
  'happy': 94.60369348526001,
  'neutral': 3.6565646529197693,
  'sad': 0.01235612144228071,
  'surprise': 1.7007224261760712},
 'gender': 'Man',
 'race': {'asian': 5.72536513209343,
  'black': 4.009378328919411,
  'indian': 25.73709785938263,
  'latino hispanic': 33.33687484264374,
  'middle eastern': 17.311449348926544,
  'white': 13.8798326253891},
 'region': {'h': 1002, 'w': 1002, 'x': 1509, 'y': 378}}
```

CNN

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The network generally consists of alternate convolution and max-pooling operations. The output obtained after applying convolution operation is shrunk using max-pooling operation which is then used as an input for the next layer.



CNN Modeling Steps

Layers

- Layer 1- 3*3,Conv,64
- Layer 2- 3*3,Conv,128
- Layer 3- 3*3,Conv,254
- Layer 4- 3*3,Conv,512
- Flatten layer
- FCL- 512 units
- FCL- 256 units
- FCL- 7 units

Also we use common techniques for each layer

- Batch Normalization
- Dropout

Parameters

- Activation Function- ReLu, Softmax
- Epoch -50
- Optimizer -Adam
- Batch size-32
- Callbacks- EarlyStopping, ReduceLROnPlateau

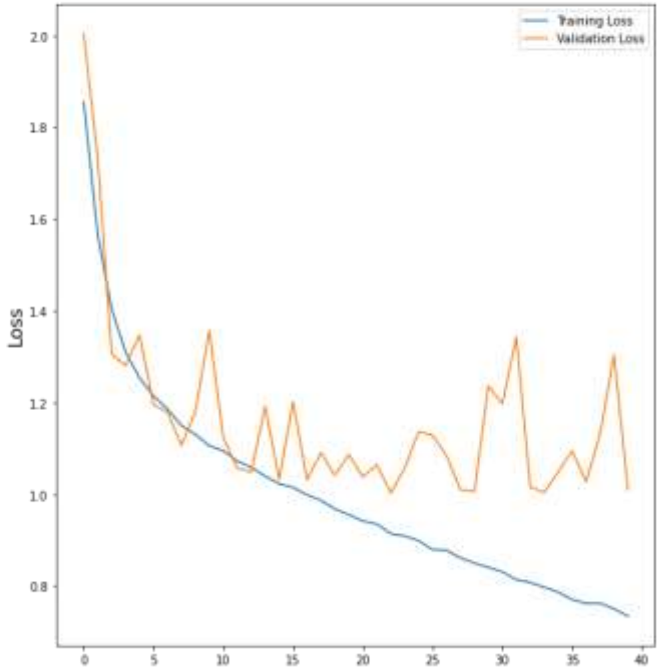
Evaluation

- Loss & accuracy plots
- Confusion matrix (Heatmap)

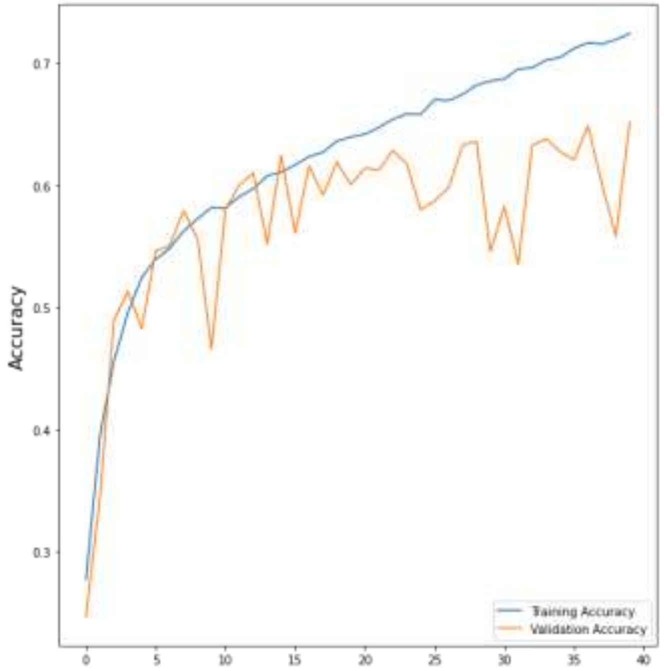
Model Evaluation

Optimizer : Adam

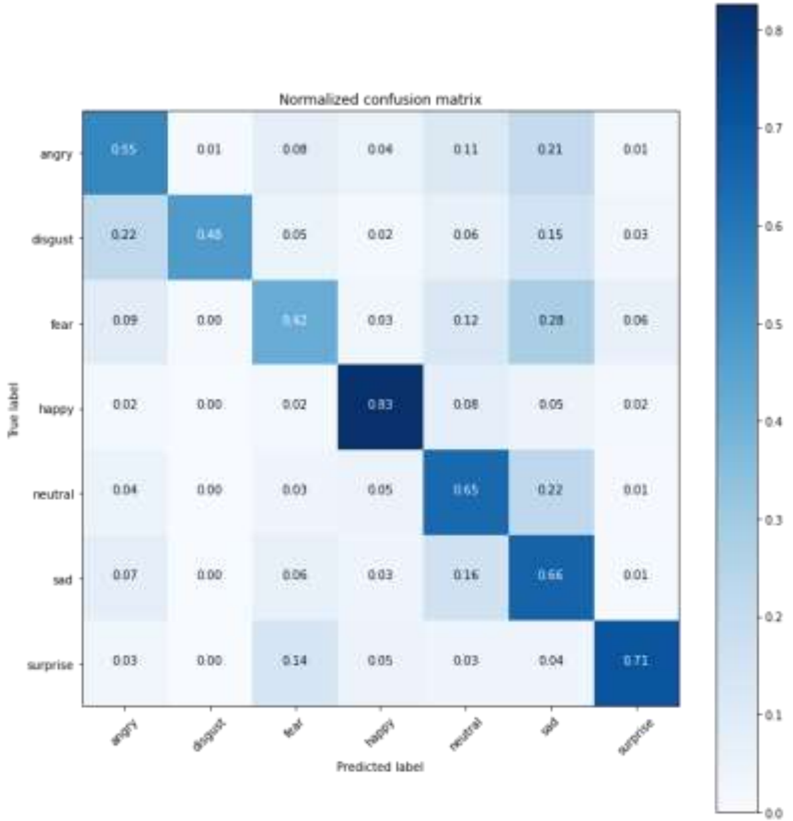
Loss



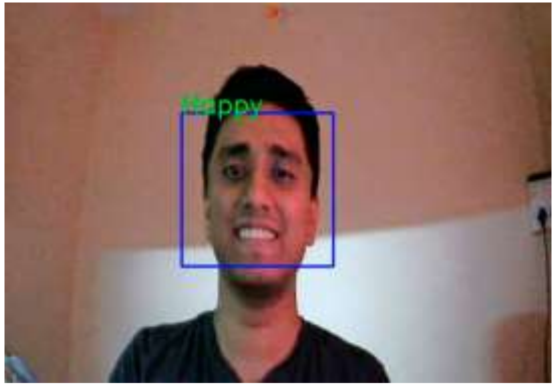
Accuracy

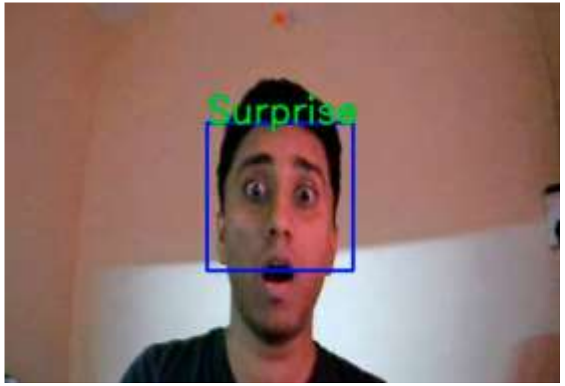


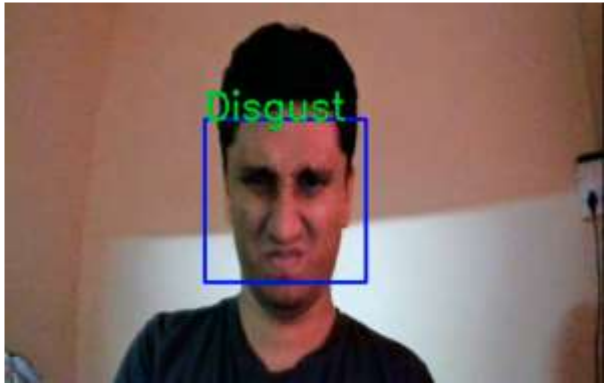
Confusion matrix(Heatmap)

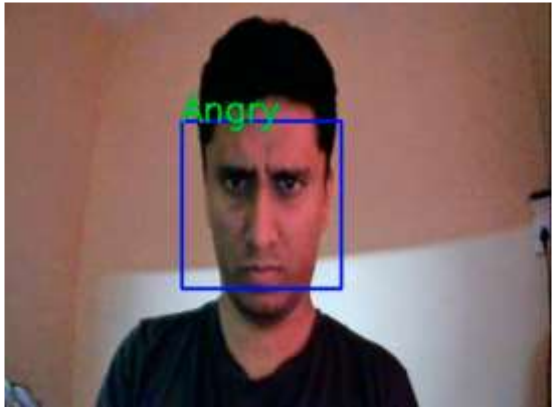


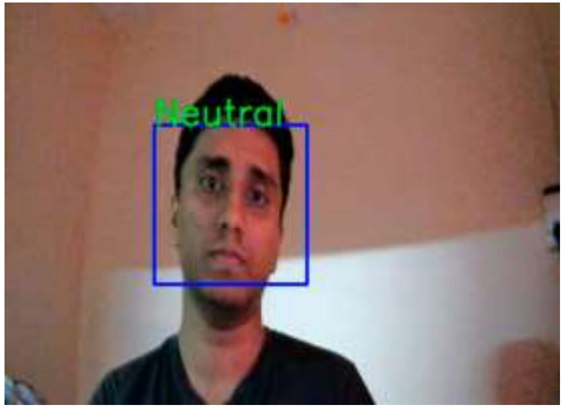
Real Time Face Emotion Detection

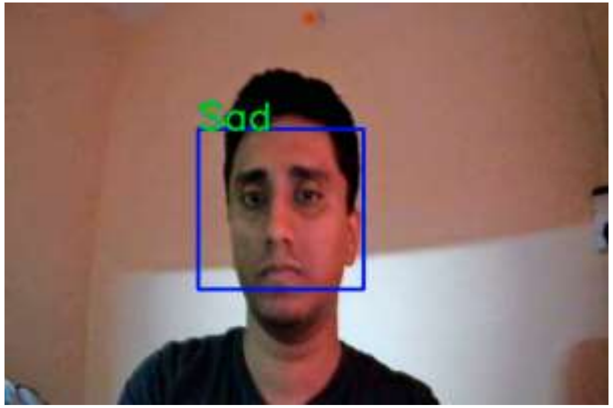












Deployment

Creating Web App Using Streamlit

Streamlit is an open-source python framework for building web apps for Machine Learning and Data Science. We can instantly develop web apps and deploy them easily using Streamlit. Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

Streamlit Link- https://share.streamlit.io/sharath2021/-live-class-monitoring-system-face-emotion-recognition-/main/app.py

Deployment

Deployment in Heroku cloud platform

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. It’s platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

Heroku Link- https://livefaceemotiondetection.herokuapp.com/

Challenges

- Large image dataset to handle
- Connecting GPU to Jupyter notebook
- Selecting number of filters and neurons
- Selecting batch size to avoid crashing of the system
- Deployment

Conclusion

- Trained the neural network and we achieved the highest training accuracy of 75.05% and validation accuracy of 65.19%.
- Pre Trained Model didn't gave appropriate result.
- The application is able to detect face location and predict the right expression while checking it on a local webcam.
- The front-end of the model was made using Streamlit for webapp and running well on local webapp link.
- Finally, we successfully deployed the Streamlit WebApp on Heroku and Streamlit share that runs on a web server.
- Our Model can successfully detect face and predict emotion on live video feed as well as on an image.

Thank You