Capstone Project-5

Deep learning and MLE Project:

Live Class Monitoring System(Face Emotion Recognition)

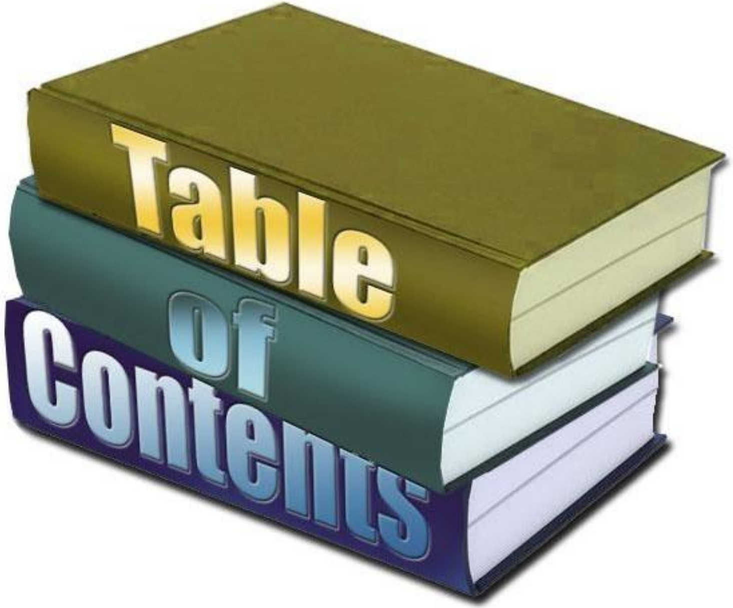
Team Observant Force Group

i) Shubham Deshmukh

ii) Saurabh Yadav

Content

- Introduction
- Problem Statement
- Data Summary
- Model Creation
    1) Using DeepFace
    2) Using Resnet50 (Residual Neural Network) model
    3) Using Custom CNN (Convolutional Neural Network) model
    4) Using Xception
- Loss & Accuracy Plot
- Real-time Local Video Face Detection
- Deployment of Streamlite Web App in Heroku and Streamlit
- Various prediction Images from the WebApp
- Challenges
- Conclusions
- Some Real Life Learning from this Project



Introduction

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.

Global E-learning is estimated to witness an 8X over the next 5 years to reach USD 2B in 2021.

India is expected to grow with a CAGR of 44% crossing the 10M users mark in 2021. Although the market is growing on a rapid scale, there are major challenges associated with digital learning when compared with brick and mortar classrooms. One of many challenges is how to ensure quality learning for students. Digital platforms might overpower physical classrooms in terms of content quality but when it comes to understanding whether students are able to grasp the content in a live class scenario is yet an open-end challenge. In a physical classroom during a lecturing 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 video telephony software program (ex-Zoom) where it's not possible for medium scale class (25-50) to see all students and access the mood. Because of this drawback, students are not focusing on content due to lack of surveillance.

While digital platforms have limitations in terms of physical surveillance but it comes with the power of data and machines which can work for you. It provides data in the form of video, audio, and texts which can be analyzed using deep learning algorithms. Deep learning backed system not only solves the surveillance issue, but it also removes the human bias from the system, and all information is no longer in the teacher's brain rather translated in numbers that can be analyzed and tracked.

Problem Statements

We will solve the above-mentioned challenge by applying deep learning algorithms to live video data.The solution to this problem is by recognizing facial emotions.

What is Face Emotion Recognition?

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.

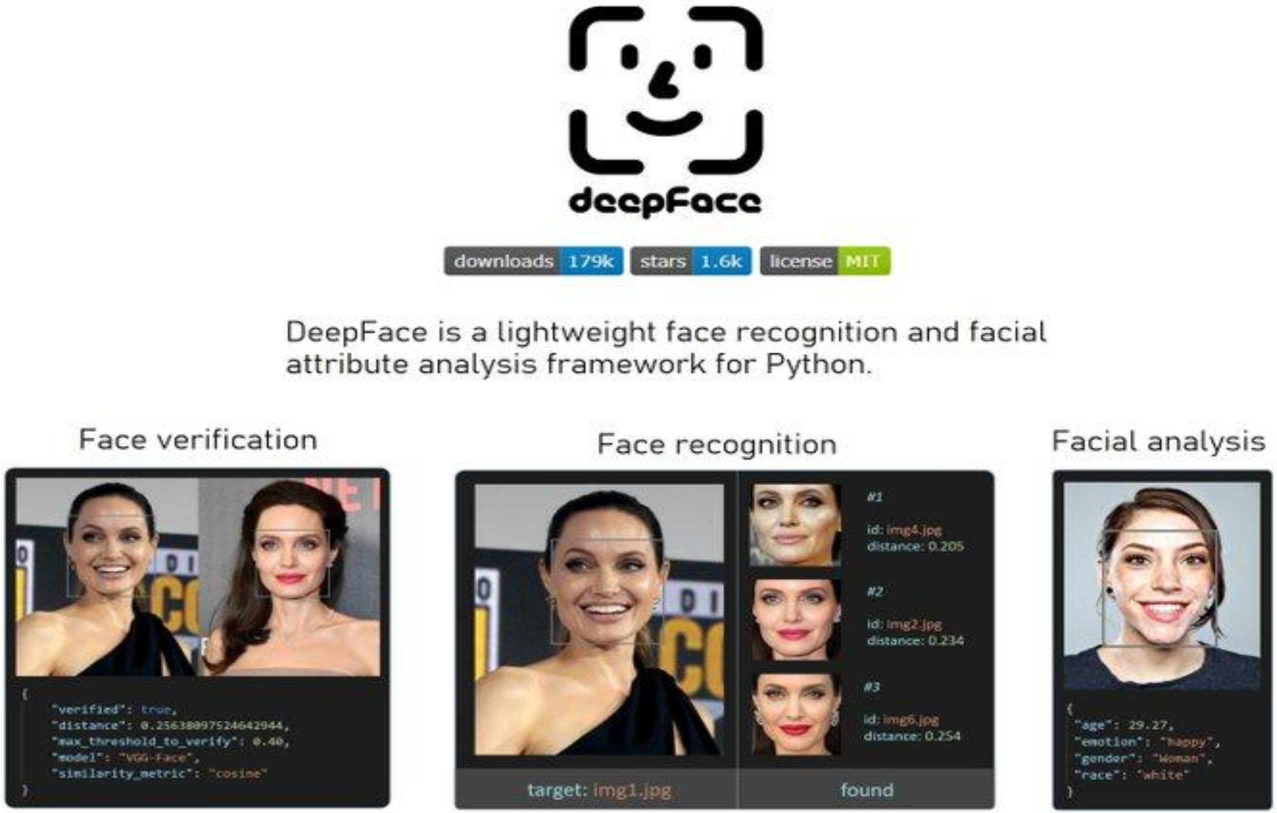
Data Summary:

- I have built a deep learning model which detects the real time emotions of students through a webcam so that teachers can understand if students are able to grasp the topic according to students' expressions or emotions and then deploy the model. The model is trained on the FER-2013 dataset .
- This dataset consists of 35887 grayscale, 48x48 sized face images with seven emotions - angry, disgusted, fearful, happy, neutral, sad and surprised.
- Here is the dataset link:- https://www.kaggle.com/msambare/fer2013

Label	Emotion	Number of images for Training	Number 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

Model Creation

1)Using DeepFace



DeepFace is a lightweight face recognition and facial attribute analysis framework for Python.

Face verification

```
{
    "verified": true,
    "distance": 0.25638097524642944,
    "max_threshold_to_verify": 0.40,
    "model": "VGG-Face",
    "similarity_metric": "cosine"
}
```

Face recognition

```
#1
id: img4.jpg
distance: 0.205

#2
id: img2.jpg
distance: 0.234

#3
id: img6.jpg
distance: 0.254
```

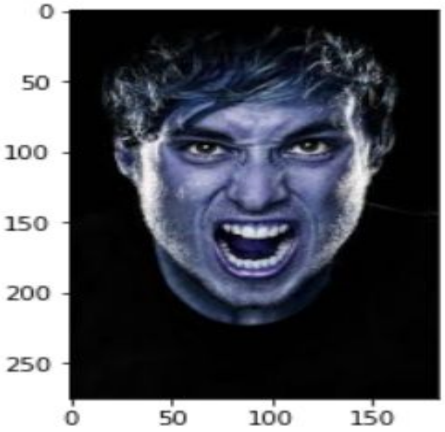
target: img1.jpg

found

Facial analysis

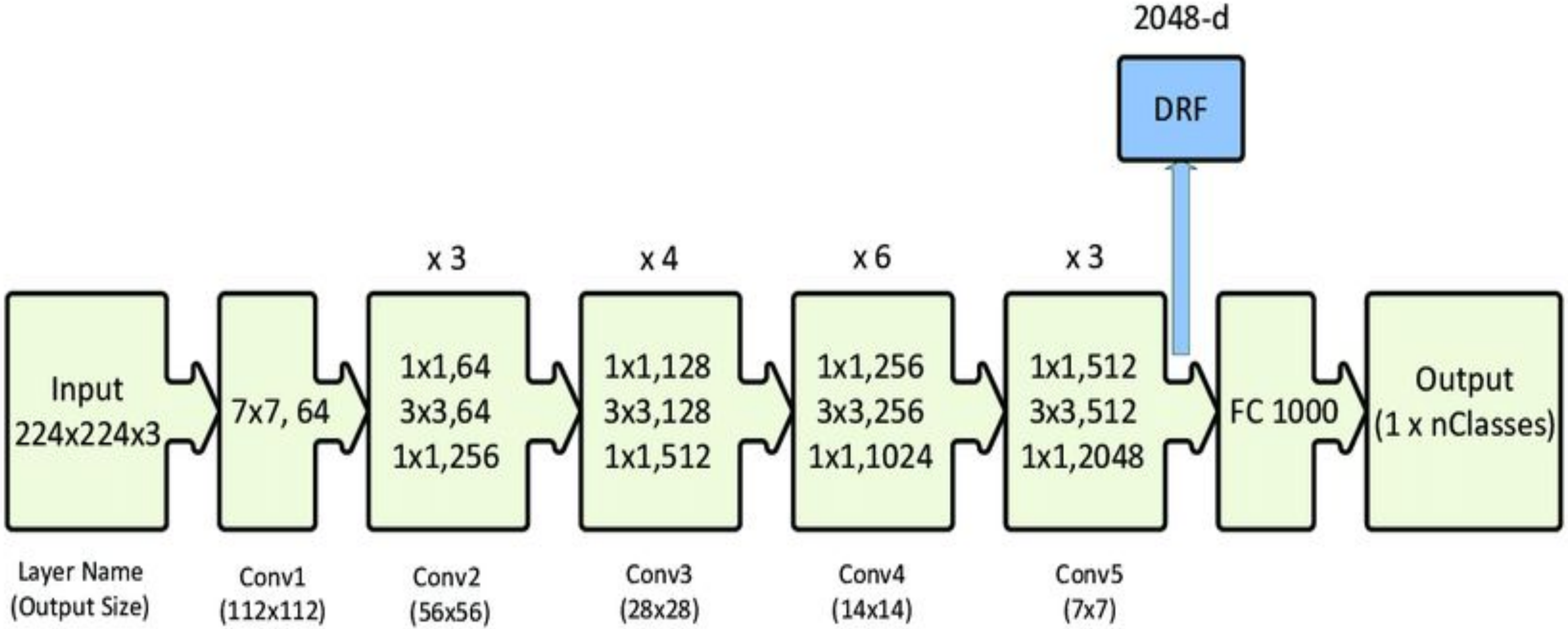
```
{
 "age": 29.27,
 "emotion": "happy",
 "gender": "Woman",
 "race": "white"
}
```

DeepFace Prediction:

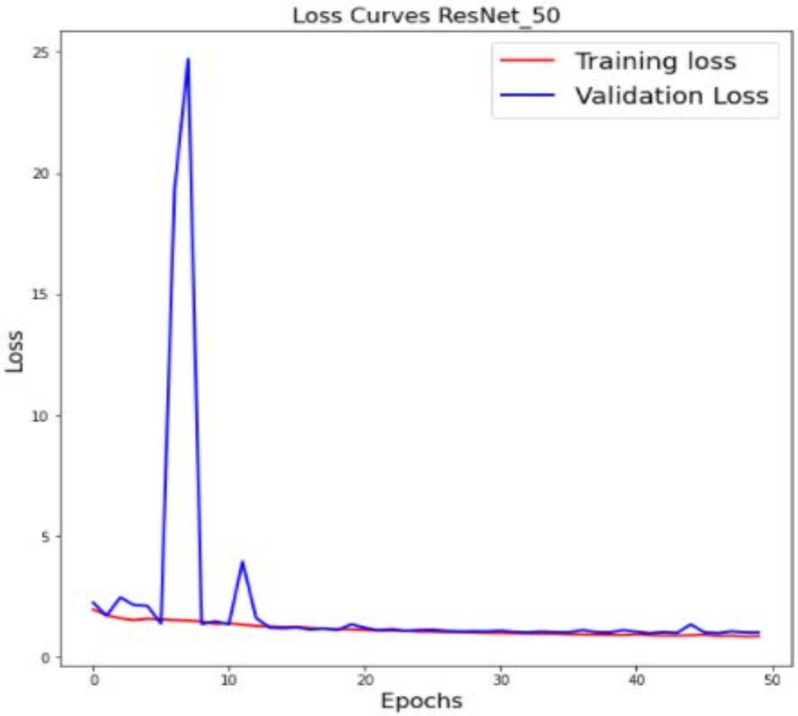


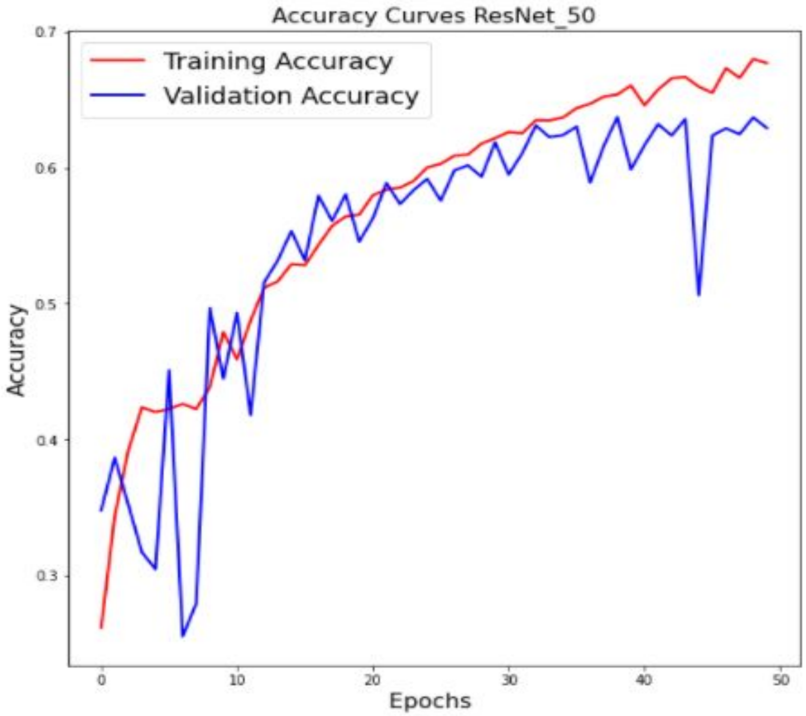
We imported above image which looks angry but our model give us “27 years old white **fear** Man” this result. To get better results we decided to train our own model.

2) Resnet50

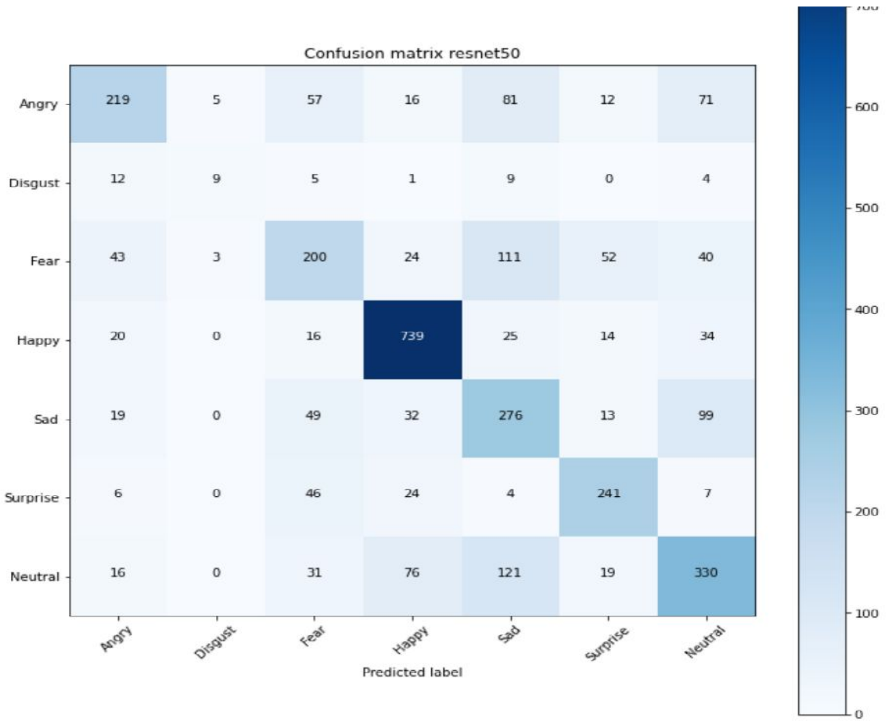

2048-d
DRF
x 3
x 4
x 6
x 3
Input
224x224x3
7x7, 64
1x1,64
3x3,64
1x1,256
1x1,128
3x3,128
1x1,512
1x1,256
3x3,256
1x1,1024
1x1,512
3x3,512
1x1,2048
FC 1000
Output
(1 x nClasses)
Layer Name
(Output Size)
Conv1
(112x112)
Conv2
(56x56)
Conv3
(28x28)
Conv4
(14x14)
Conv5
(7x7)

Accuracy and loss plot:





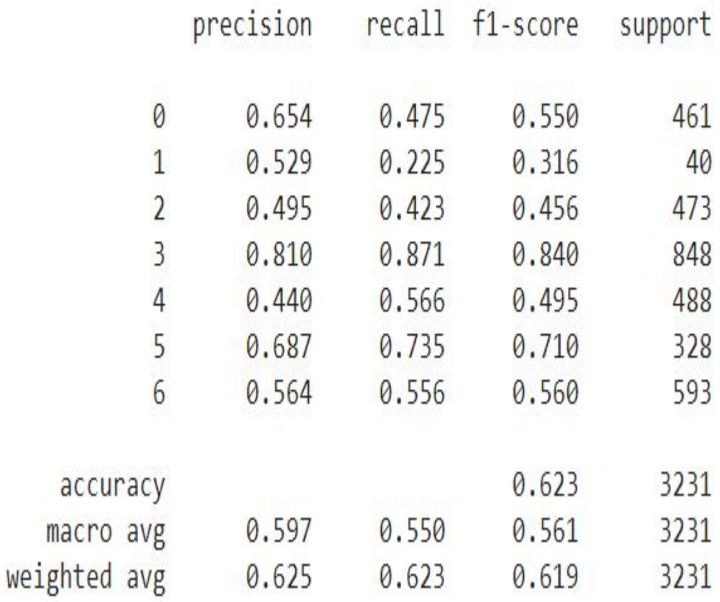
Confusion Matrix



Confusion matrix resnet50

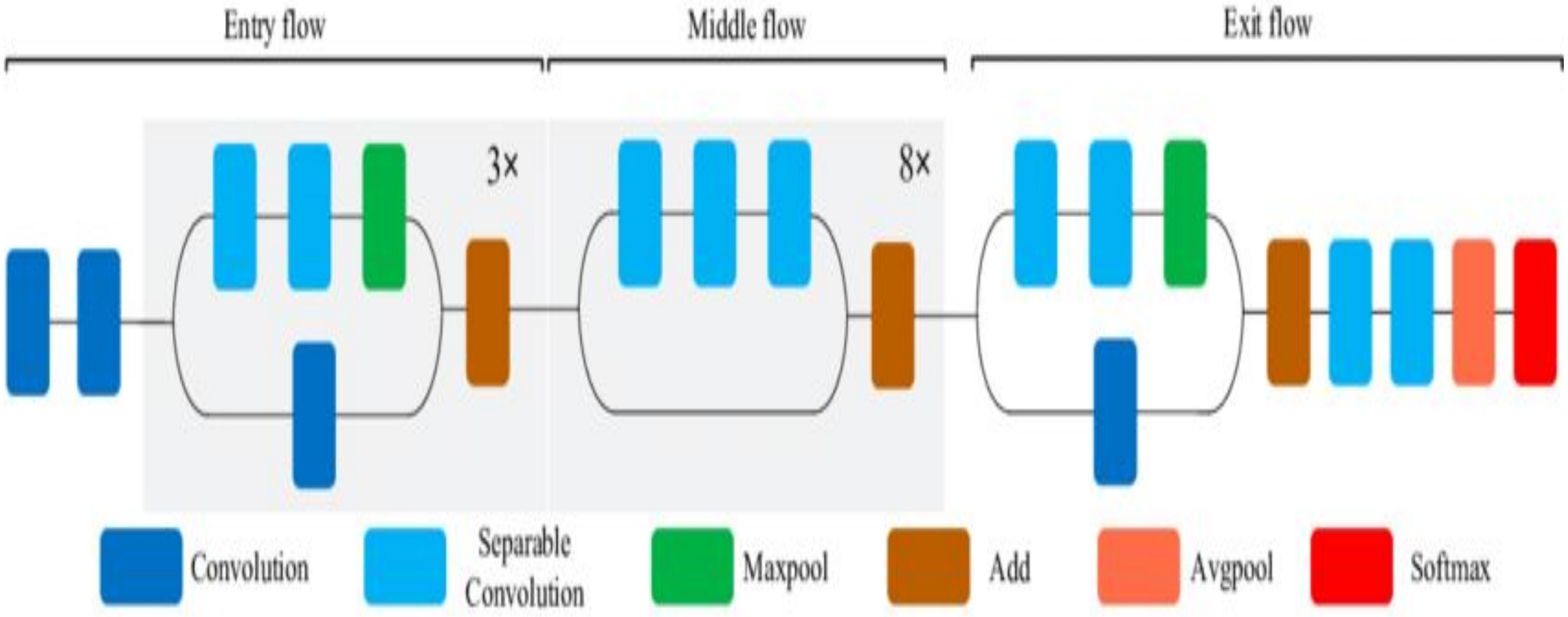
	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	219	5	57	16	81	12	71
Disgust	12	9	5	1	9	0	4
Fear	43	3	200	24	111	52	40
Happy	20	0	16	739	25	14	34
Sad	19	0	49	32	276	13	99
Surprise	6	0	46	24	4	241	7
Neutral	16	0	31	76	121	19	330

Classification Report

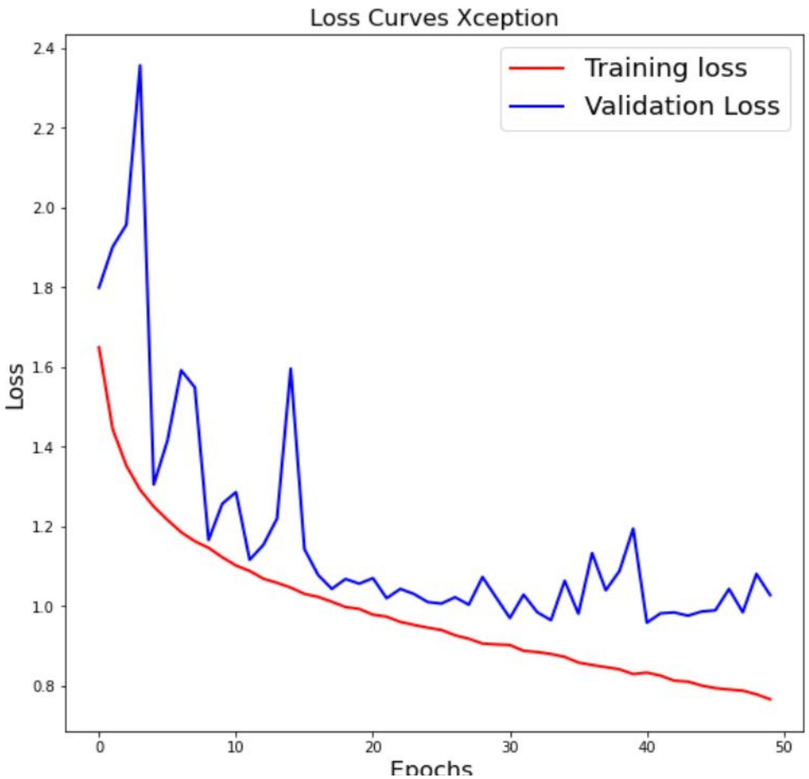


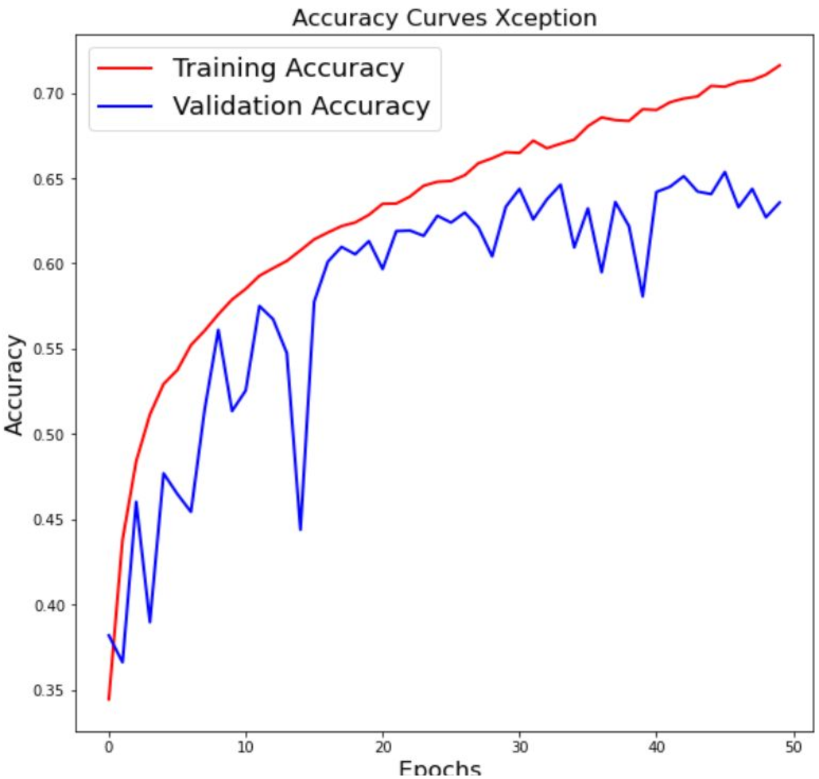
```
              precision    recall  f1-score   support

           0      0.654     0.475     0.550       461
           1      0.529     0.225     0.316        40
           2      0.495     0.423     0.456       473
           3      0.810     0.871     0.840       848
           4      0.440     0.566     0.495       488
           5      0.687     0.735     0.710       328
           6      0.564     0.556     0.560       593

    accuracy                          0.623      3231
   macro avg      0.597     0.550     0.561      3231
weighted avg      0.625     0.623     0.619      3231
```

4) Xception model

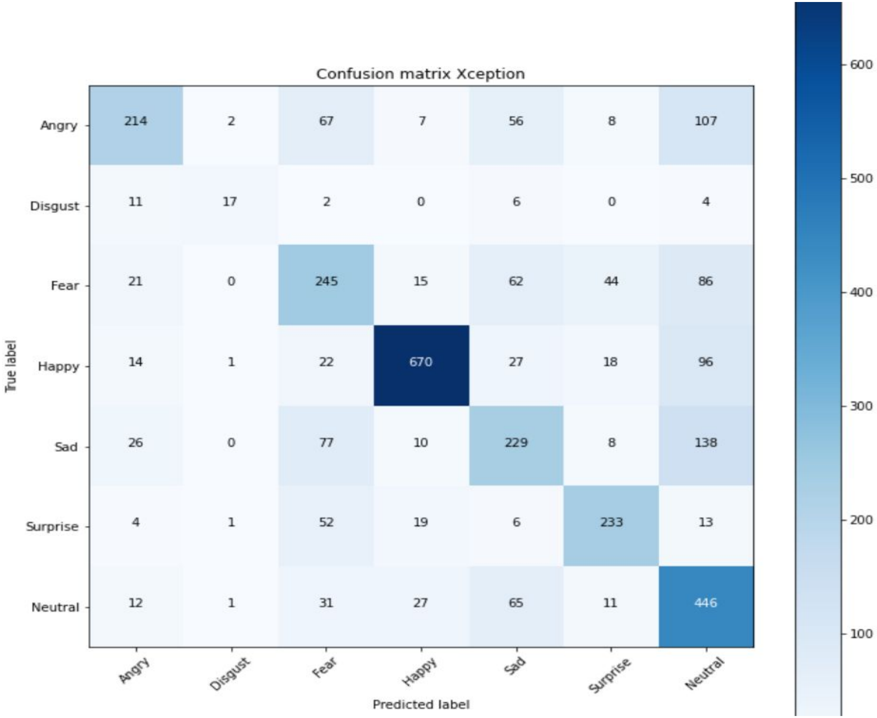


Accuracy and loss plots



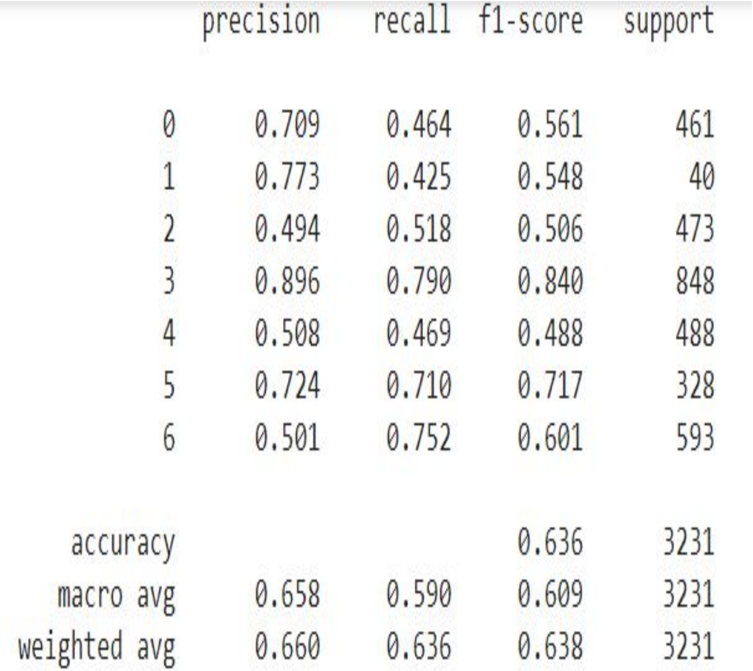


Confusion Matrix



True label \ Predicted label	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	214	2	67	7	56	8	107
Disgust	11	17	2	0	6	0	4
Fear	21	0	245	15	62	44	86
Happy	14	1	22	670	27	18	96
Sad	26	0	77	10	229	8	138
Surprise	4	1	52	19	6	233	13
Neutral	12	1	31	27	65	11	446

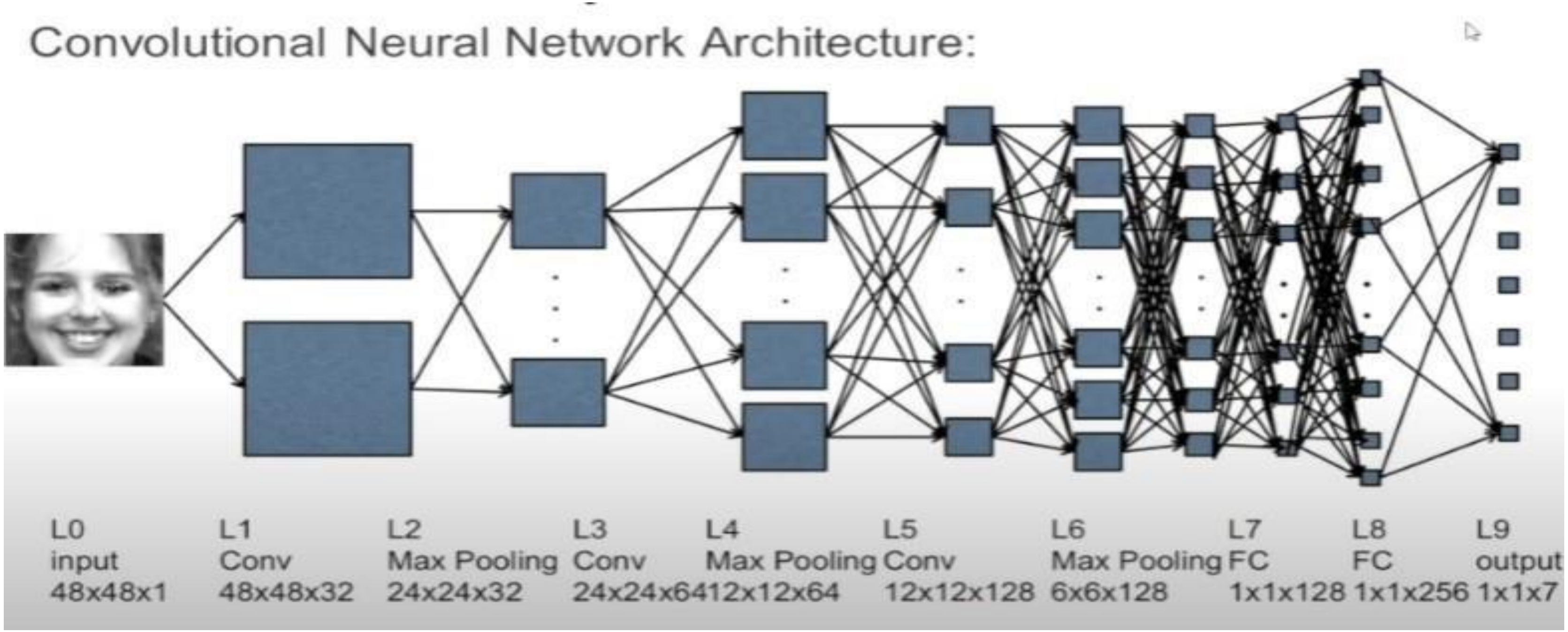
Classification Report



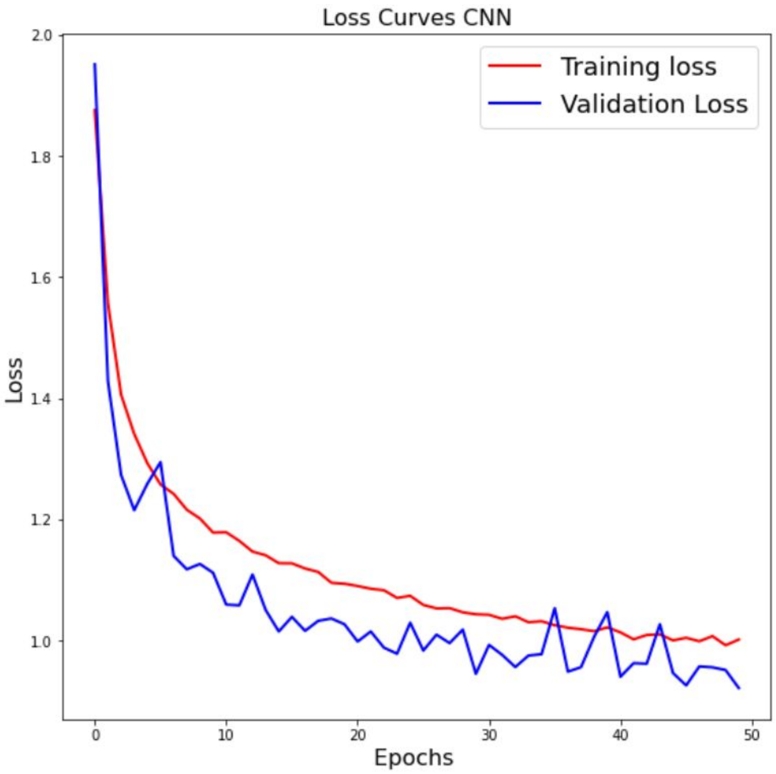
```
              precision    recall  f1-score   support

           0      0.709     0.464     0.561       461
           1      0.773     0.425     0.548        40
           2      0.494     0.518     0.506       473
           3      0.896     0.790     0.840       848
           4      0.508     0.469     0.488       488
           5      0.724     0.710     0.717       328
           6      0.501     0.752     0.601       593

    accuracy                          0.636      3231
   macro avg      0.658     0.590     0.609      3231
weighted avg      0.660     0.636     0.638      3231
```

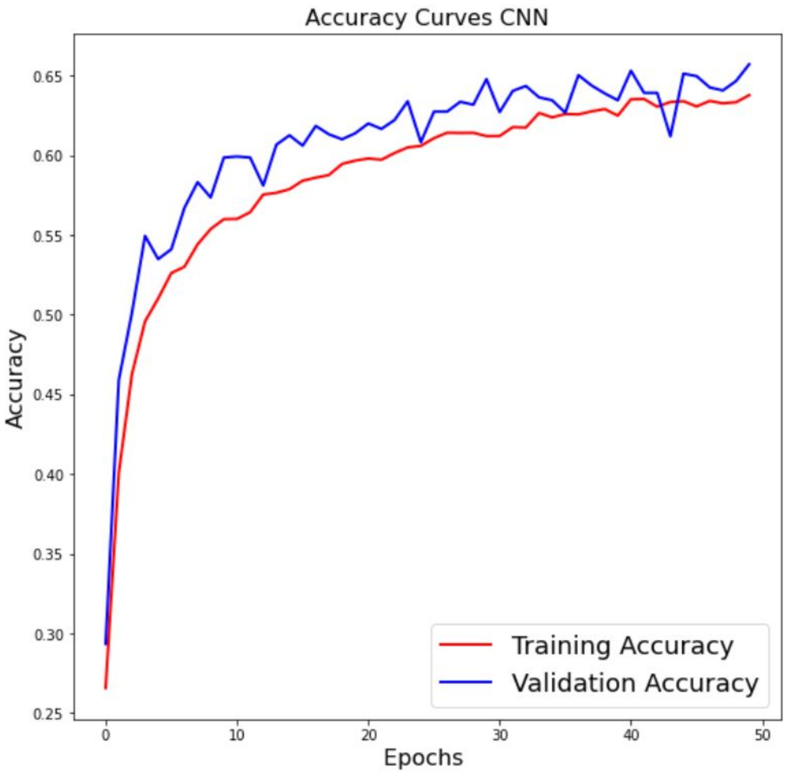
3) Using CNN layers

What is CNN ?

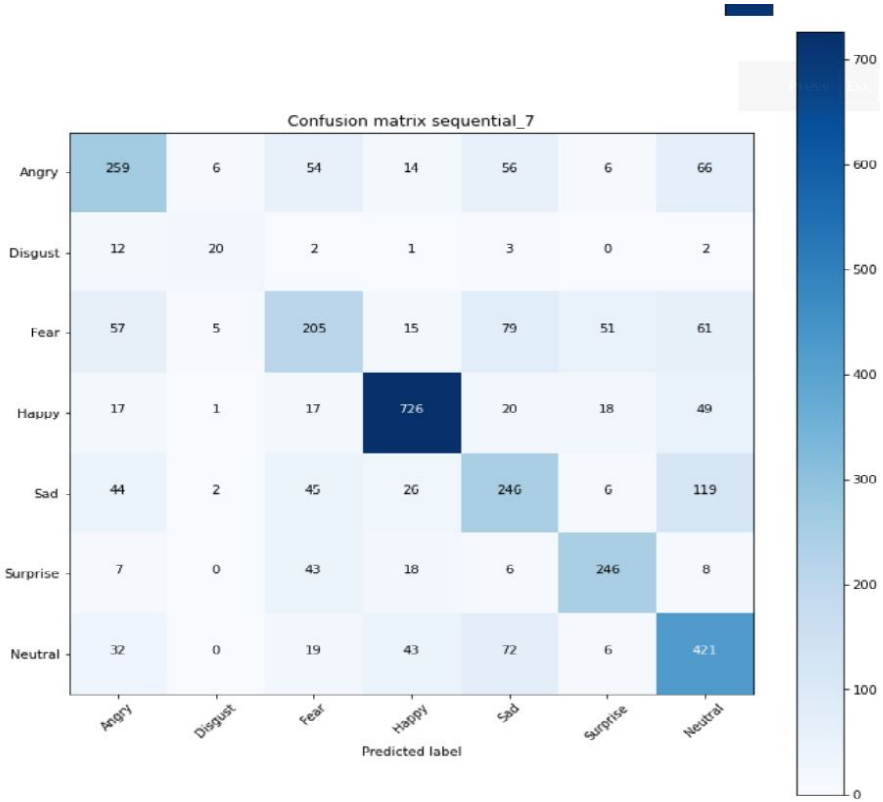


Accuracy and loss plots


Loss Curves CNN

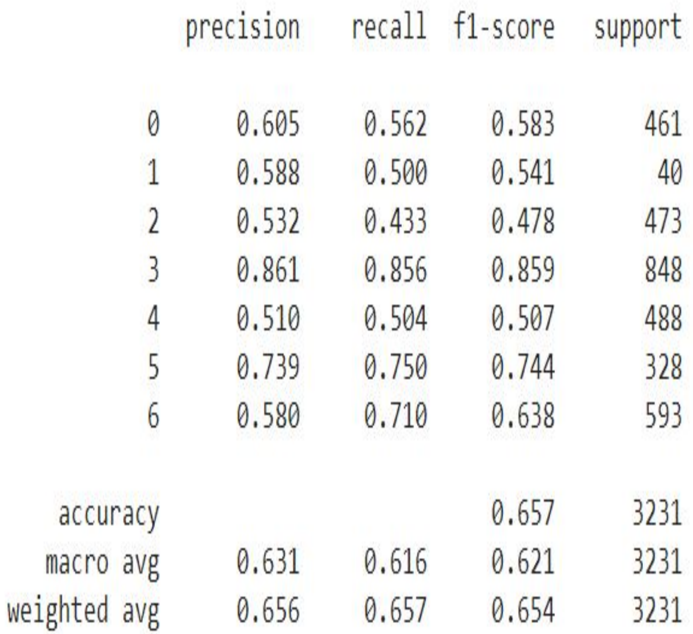

Accuracy Curves CNN

Confusion Matrix



	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	259	6	54	14	56	6	66
Disgust	12	20	2	1	3	0	2
Fear	57	5	205	15	79	51	61
Happy	17	1	17	726	20	18	49
Sad	44	2	45	26	246	6	119
Surprise	7	0	43	18	6	246	8
Neutral	32	0	19	43	72	6	421

Classification Report



```
              precision    recall  f1-score   support

           0      0.605     0.562     0.583       461
           1      0.588     0.500     0.541        40
           2      0.532     0.433     0.478       473
           3      0.861     0.856     0.859       848
           4      0.510     0.504     0.507       488
           5      0.739     0.750     0.744       328
           6      0.580     0.710     0.638       593

    accuracy                          0.657      3231
   macro avg      0.631     0.616     0.621      3231
weighted avg      0.656     0.657     0.654      3231
```

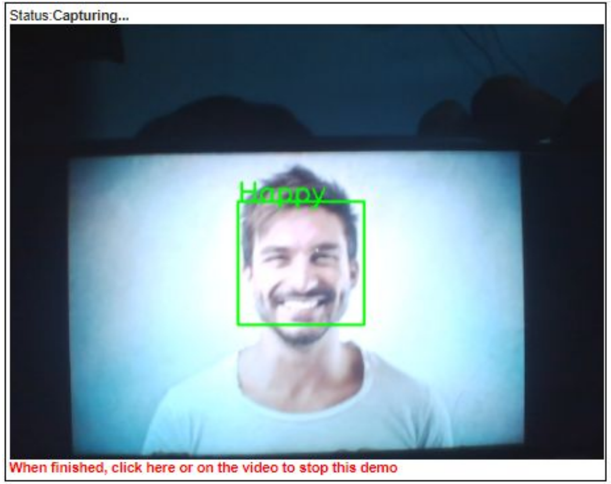
Real-Time Local Video Face Emotion Detection

- We created patterns for detecting and predicting single faces and as well as multiple faces using OpenCV videocapture in local webcam.

&

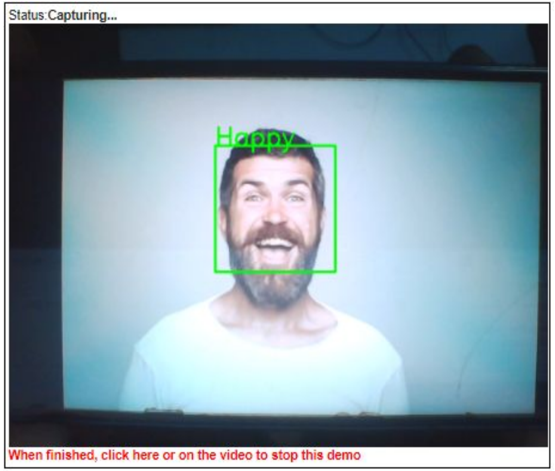
- For Web-app , OpenCV can’t be used. Thus, using Streamlit-Webrtc for front-end application.

Some Predicatied Images



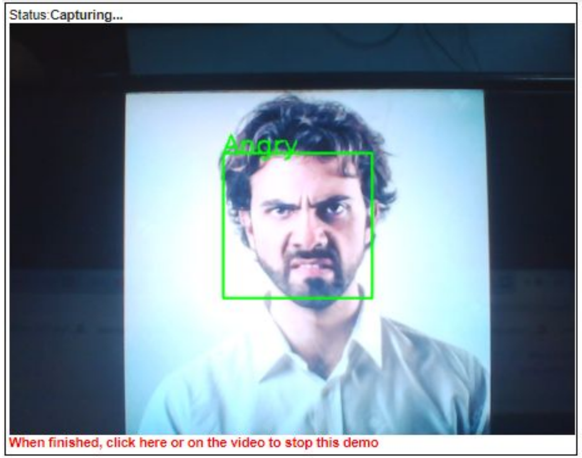
Status:Capturing...

When finished, click here or on the video to stop this demo



Status:Capturing...

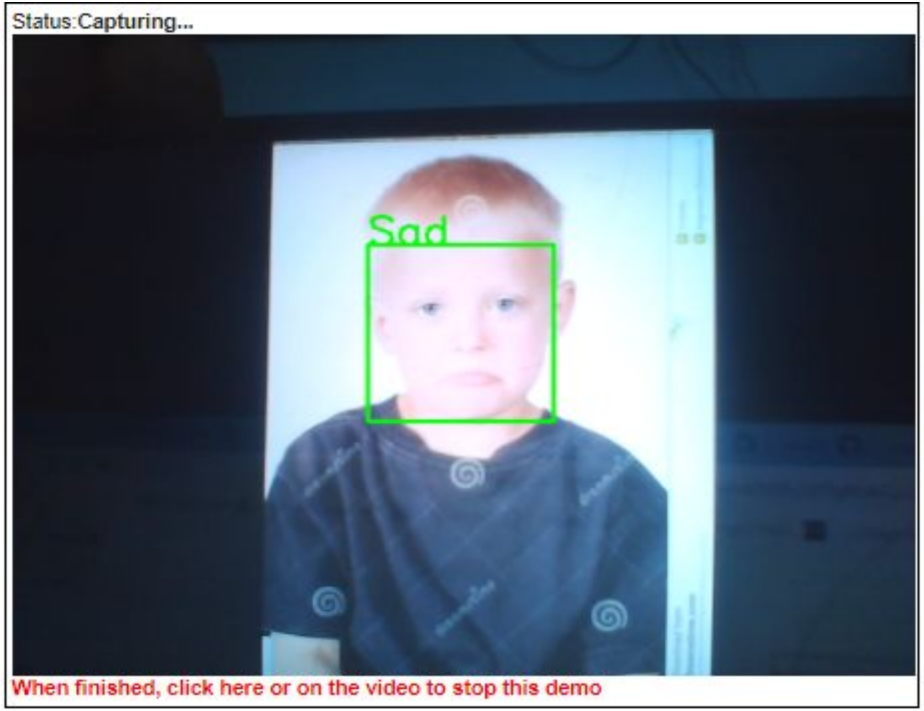
When finished, click here or on the video to stop this demo



Status:Capturing...

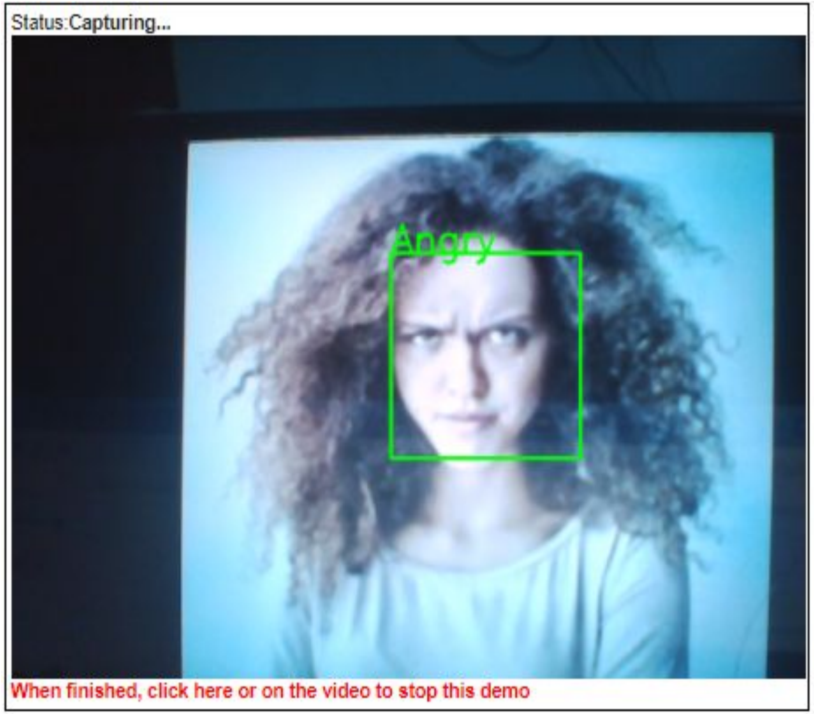
When finished, click here or on the video to stop this demo

Contd..



Status:Capturing...

When finished, click here or on the video to stop this demo



Status:Capturing...

When finished, click here or on the video to stop this demo

Deployment of Streamlite Web App in Heroku and Streamlit:

- We deployed the app in Heroku if you saw in the starting section of github repo you see the all the requirement files are there for creating an app on Heroku of name “face-emotion-recognition-ofg”.

- But due to high slug size the buffering takes time so we have ran our app working on local and it ran properly and app is also fine also we’ve included video on github repo.

Heroku Link: **https://face-emotion-recognition-ofg.herokuapp.com/**

Challenges

- Large Image Dataset to handle.
- Couldn’t able to connect GPU with Jupyter Notebook.
- Tried creating lot of models till find the best one.
- Continuous Runtime and RAM Crash due to large dataset.
- Carefully tuned Hyper parameters .

Conclusion

- We build the WebApp using streamlit and deployed in Heroku and Streamlit Sharing.
- The model which was created by custom CNN model gave training accuracy of 77% and test accuracy of 69% .
- I have also included the video of my WebApp working in Local.
- Codes which are deployed are in Github Repository.
- It was such an amazing and interesting project. We learnt a lot from this.

Some real life experience form project

- **Understand the deep concept of project**
- **Don't afraid to failure**
- **Never give up**
- **Have some patience good things happen**
- **Try new things and execute your idea**

Contributors Role:

Shubham Deshmukh :

- Build Deep-Face model and Resnet_50 (Residual Neural Network) model.
- Ran the local webcam for emotion detection on single as well as multiple face.

Saurabh Yadav:

- Build Xception model and Custom CNN (Convolutional Neural Network) model.
- Created streamlit webapp and deployed on Heroku.

Thank You!