Capstone project

on

Face Emotion Recognition

By

Shubhangi Dharmik

(Individual)

Procedure

1. Problem statement
2. Introduction
3. Data summary
4. Method used
5. Analysis
6. Evaluation of Model
7. Results
8. Deployment of model
9. Conclusions

Problem Statement

- Digital classrooms are conducted via video telephony software program (exZoom). We have to detect emotion of students.
- We will solve this challenge by applying deep learning algorithms to live video data. The solution to this problem is by recognizing facial emotions.
- The main objective is to build a model which can recognize the facial emotion of students through web cam in real time.

Data Summary

Dataset taken from kaggle FER2013

- **Emotion** - The dataset consist of emotions that is happy, sad, angry, neutral, surprise, disgust and fear.
- **Pixels** –
- **Usage** – usage are divided into three test that is training test, private test and public test.

Method

- Clean data
- Data Preprocessing
- Split into train test and validation
- Visualize image through emotion
- Initialize the CNN
- Train the model
- Testing

Method

Convolutional neural network

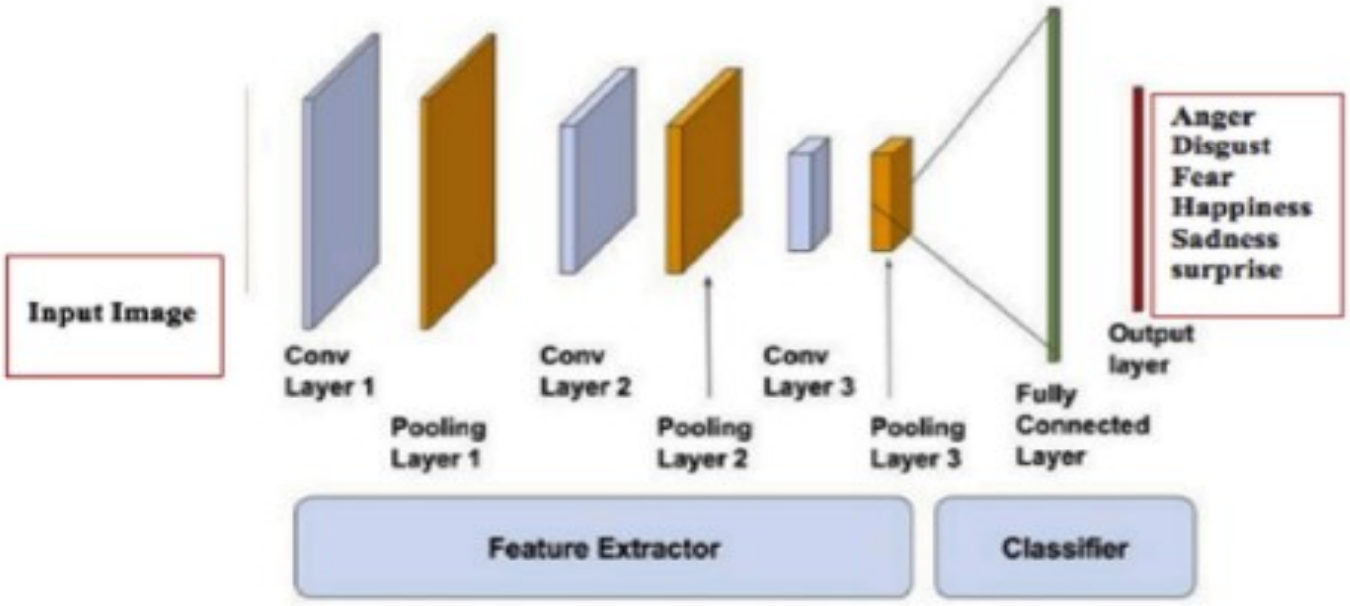


Fig. 6. Facial landmarks system structure.

Model summary

```
: model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 46, 46, 64)        640
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 44, 44, 64)        36928
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 22, 22, 64)        0
_________________________________________________________________
dropout_4 (Dropout)          (None, 22, 22, 64)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 20, 20, 64)        36928
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 18, 18, 64)        36928
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 9, 9, 64)          0
_________________________________________________________________
dropout_5 (Dropout)          (None, 9, 9, 64)          0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 7, 7, 128)         73856
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 5, 5, 128)         147584
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1024)              525312
_________________________________________________________________
dropout_6 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dropout_7 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_5 (Dense)              (None, 7)                 7175
=================================================================
Total params: 1,914,951
```

Evaluation of model

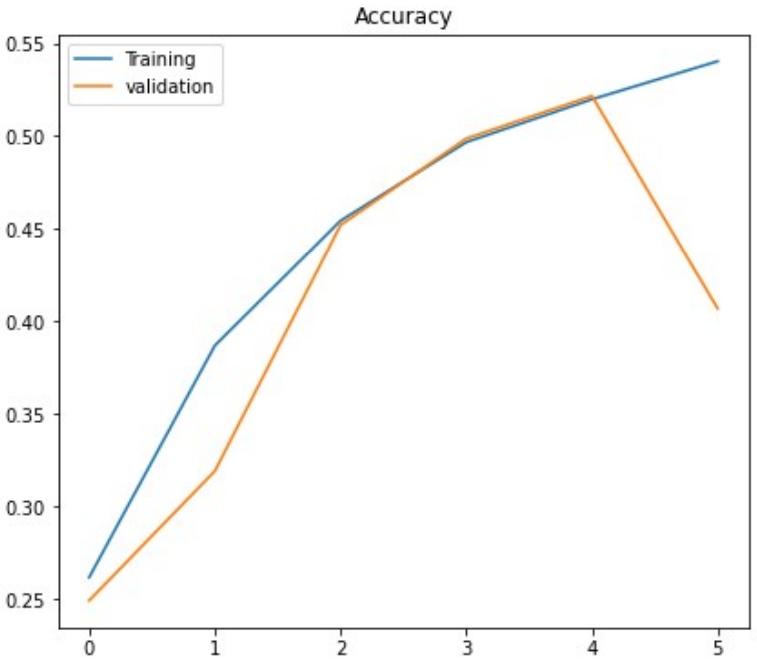
For face detection OpenCV has been used. Given a real-time live stream, the face detector examines each location of the image and classifies it as a face or no face region.

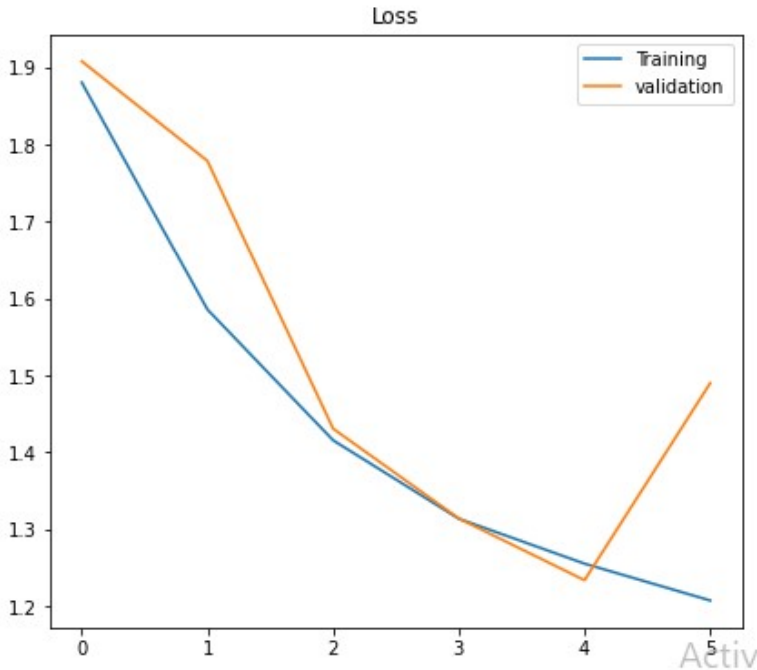
The largest face region is extracted as the region of interest, converted to a grayscale image and resized pixels

Result

- The emotion of the face is predicted and the label with the largest associated probability is taken. The live stream is displayed in a window with a bounding box around the largest face and the emotion label is shown above the bounding box in real-time
- We get the accuracy of training data is 55% and for test data it is approximately 40%, the accuracy will be better if number of epoch is increased

Accuracy and Loss





Confusion matrix

```
[44]: plt.figure(figsize=[15,5])
      sns.heatmap(confusion_matrix(y_test,pred),annot = True, fmt= "d", cmap = "Greens");
```

Deployment of model

Saving file fer.json and fer2013.h5 model

For the deployment of model we use google cloud platform.

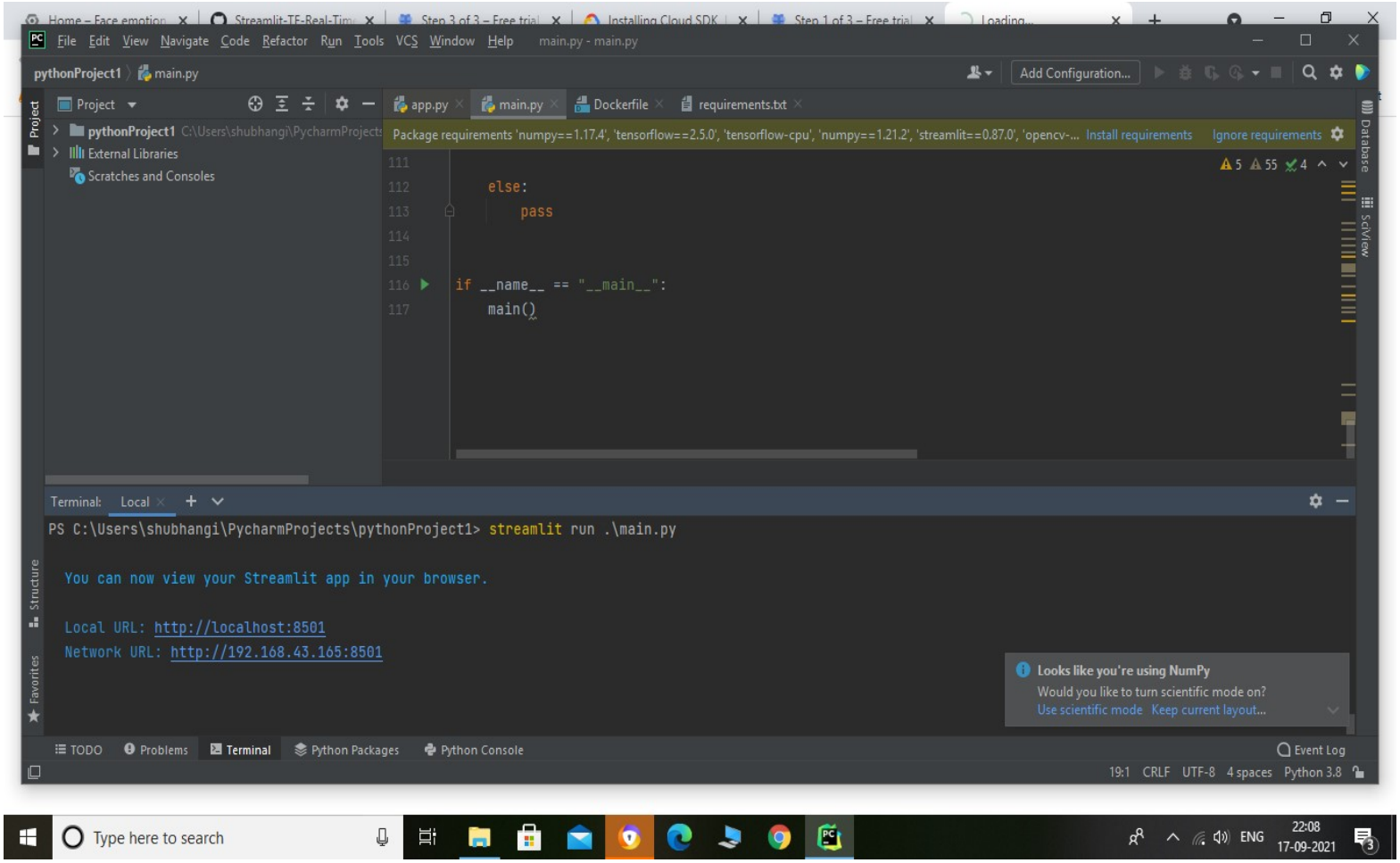
For that we need a docker file for the deployment of the model

deployment link :

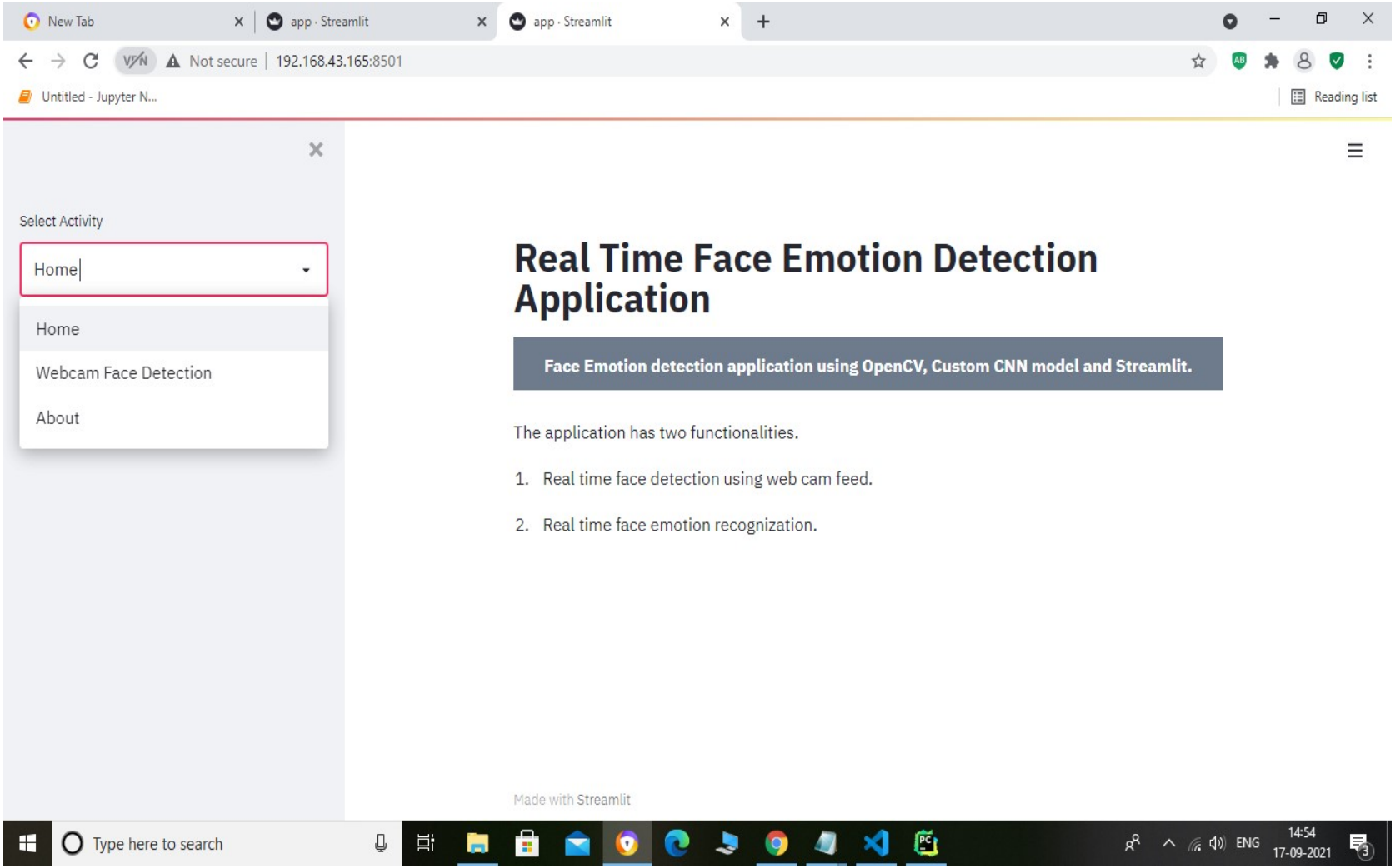
[https://face-emotion-detection-326315.as.r.appspot.com]

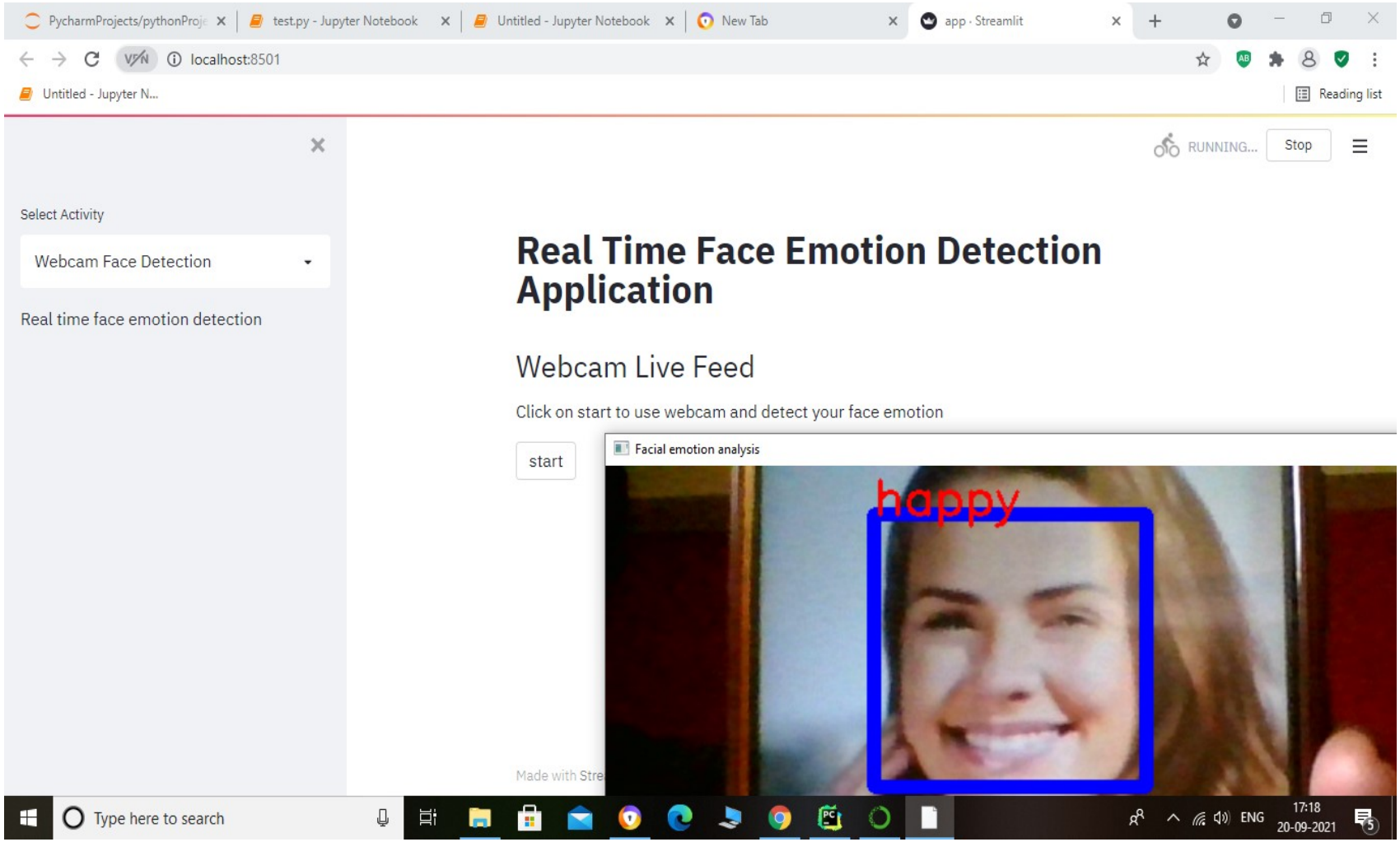
Streamlit

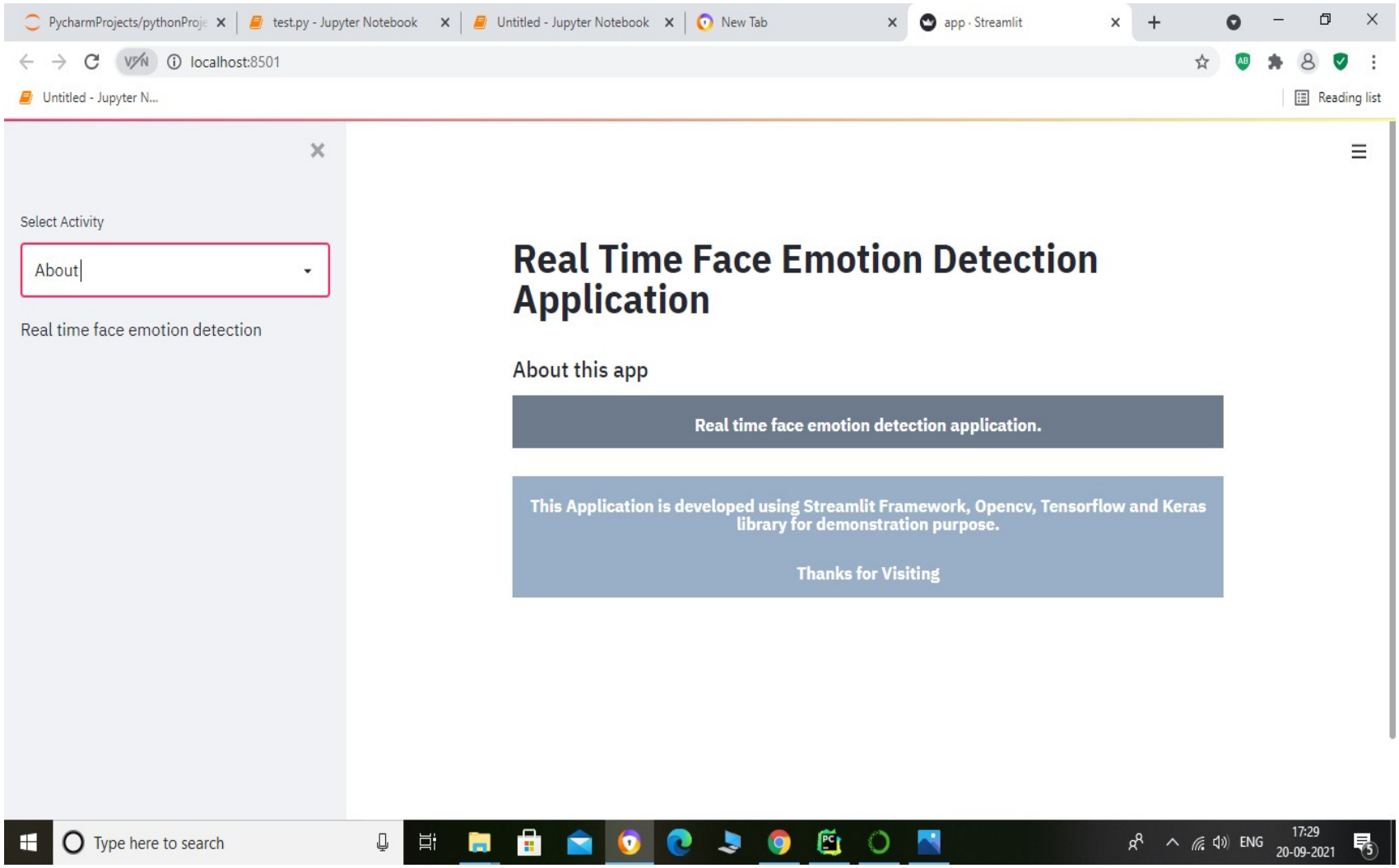
Demo



Streamlit app




PycharmProjects/pythonProje
test.py - Jupyter Notebook
Untitled - Jupyter Notebook
New Tab
app · Streamlit
localhost:8501
Untitled - Jupyter N...
Reading list
RUNNING...
Stop
Select Activity
Webcam Face Detection
Real time face emotion detection
Real Time Face Emotion Detection Application
Webcam Live Feed
Click on start to use webcam and detect your face emotion
start
Facial emotion analysis
happy
Made with Stre
Type here to search
ENG
17:18
20-09-2021


PycharmProjects/pythonProje
test.py - Jupyter Notebook
Untitled - Jupyter Notebook
New Tab
app · Streamlit
localhost:8501
Untitled - Jupyter N...
Reading list
Select Activity
About
Real time face emotion detection
Real Time Face Emotion Detection Application
About this app
Real time face emotion detection application.
This Application is developed using Streamlit Framework, Opencv, Tensorflow and Keras library for demonstration purpose.
Thanks for Visiting
Type here to search
ENG
17:29
20-09-2021

Conclusions

- The goal of this work has been to design a deep neural network for real time facial emotion recognition.
- We get accuracy in model 1 is 72% for training and 40% in testing at 30^{th} epoch.
- We get accuracy in model 2 is 55% for training and 40% in testing at 6^{th} epoch.
- By using more types of data augmentation, by increasing number of epoch, increasing the number of layers, deepening the network and adding regularization further better accuracy can likely be obtained.

Thank You