Capstone Project - 5

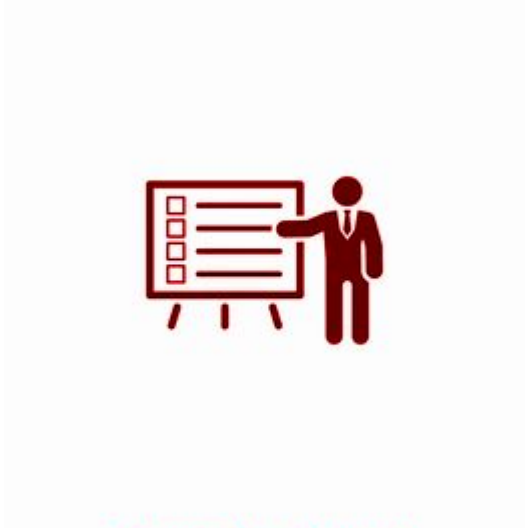
Live Class Monitoring System

(Face Emotion Recognition)

Deep Learning & MLE

Swapnil Patil

Problem Statement



There are major challenges associated with digital learning when compared with physical classrooms.

One of many challenges is how to ensure quality learning for students.

Understanding whether students are able to grasp the content in a live class scenario is yet an open-end challenge.

In this project, our task is to build a real time emotion recognition system for live class monitoring and deploy the model.

Introduction

- The Indian Education system has been undergoing rapid changes owing to the advancement of web-based learning services, specifically, eLearning platforms.
- In a physical classroom a lecturing teacher can see the faces and assess the emotion of the class and tune their lecture accordingly.
- The teacher should be able to be constantly aware of the fluctuation of their student's mood by using facial emotion recognition systems.
- Human emotions and intentions are expressed through facial expressions.
- Facial emotion recognition is the process of detecting human emotions from facial expressions.
- In this project, we have tried to build a real time emotion recognition system to monitor student's mood through class activity.

Data Overview

- The model is trained on FER 2013 dataset.
- This dataset contains 35k+ face images of different expressions.
- These images are categorized into 7 emotions - Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise.
- Dataset link: https://www.kaggle.com/datasets/msambare/fer2013

Dependencies

What Are Project Dependencies

TensorFlow

OpenCV

Streamlit

Streamlit webRTC

Heroku

Steps Involved

Data Exploration

Exploring the dataset and analyzing in an unstructured way to uncover initial patterns, characteristics, and points of interest.

Data Preprocessing

Data Augmentation & Data Generator

Training Models

Deepface, Simple CNN and CNN using deep layers

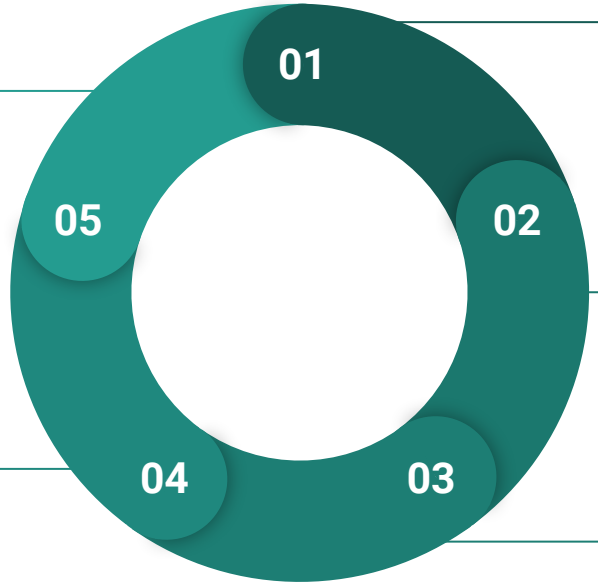
Model Evaluation & Testing

Visualization of the model performance

Testing the Model using Webcam Feed

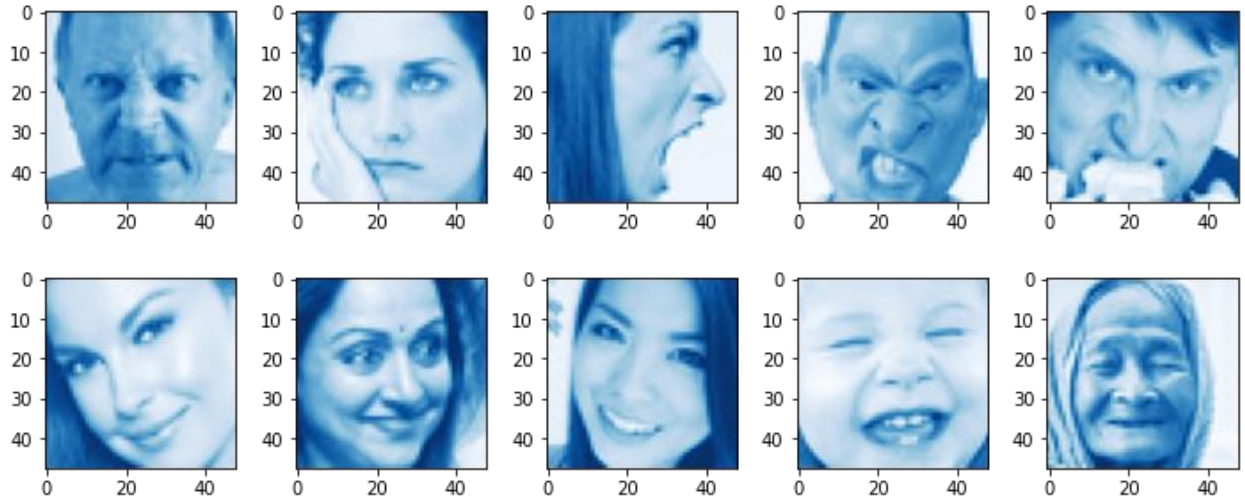
Model Deployment

Streamlit and Heroku



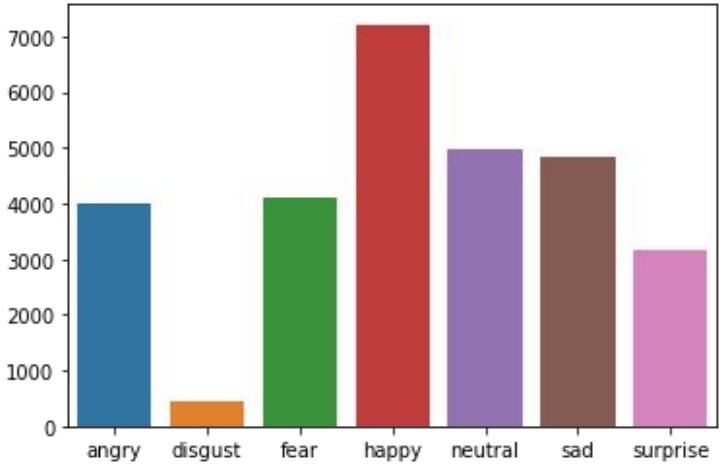
Data Exploration

- We started with importing the data and required libraries.
- The training set consists of 28,709 examples and the test set consists of 3,589 examples.
- The shape of images is (48, 48, 3).



Data Preprocessing-

- Checked the number of images in the training set.
- The images of different emotions do not have the same distribution
- Standardize the images by dividing them with the highest pixel value (1./255)



Data Augmentation

Data Augmentation is a technique used to artificially add the data by creating modified data from the existing one.

Data Generator

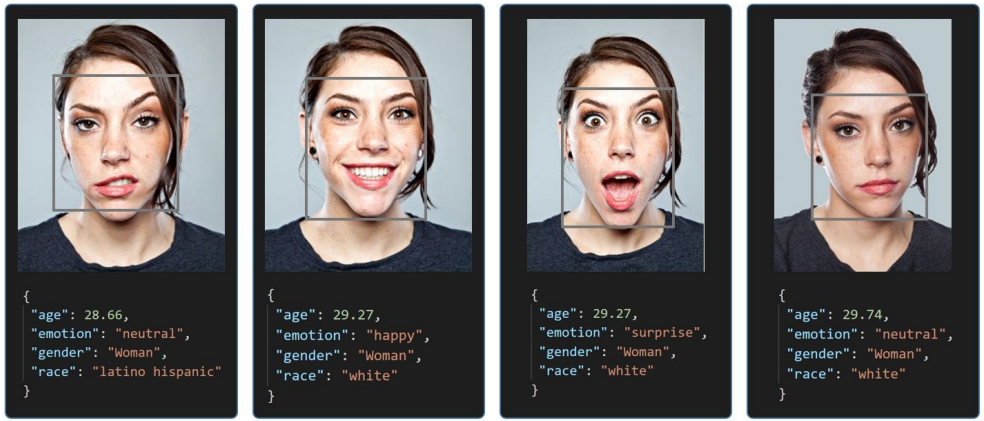
We used a data generator to generate the train and validation data to flow from the directory having image size 48x48, color mode as grayscale, batch size of 32 and class mode as categorical.

Training Models

DeepFace Model

- Deepface is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for python.
- DeepFace is a deep learning facial recognition system created by a research group at Facebook.
- The program employs a nine-layer neural network with over 120 million connection weights and was trained on four million images uploaded by Facebook users.

deepface



Model Evaluation - DeepFace

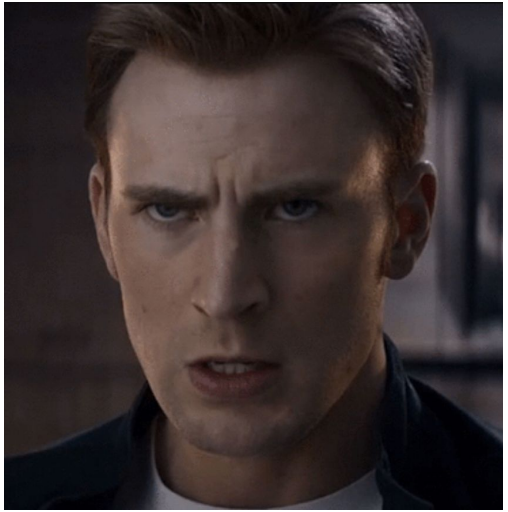
Demography

age: 44

emotion: 'happy'

race: 'white'

gender: 'Man'



Demography

age: 31,

emotion: 'sad',

race: 'white',

gender: 'Man'

In first image the deepface model is detecting the correct emotion of the person in the image whereas in the second case the person is angry but the model is predicting it as sad.

Simple Convolutional Neural Network

Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification.

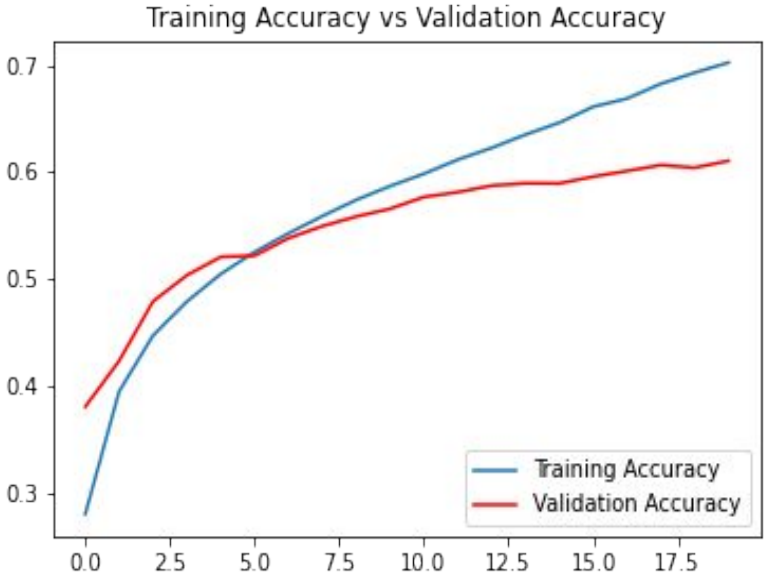
Model: "sequential"

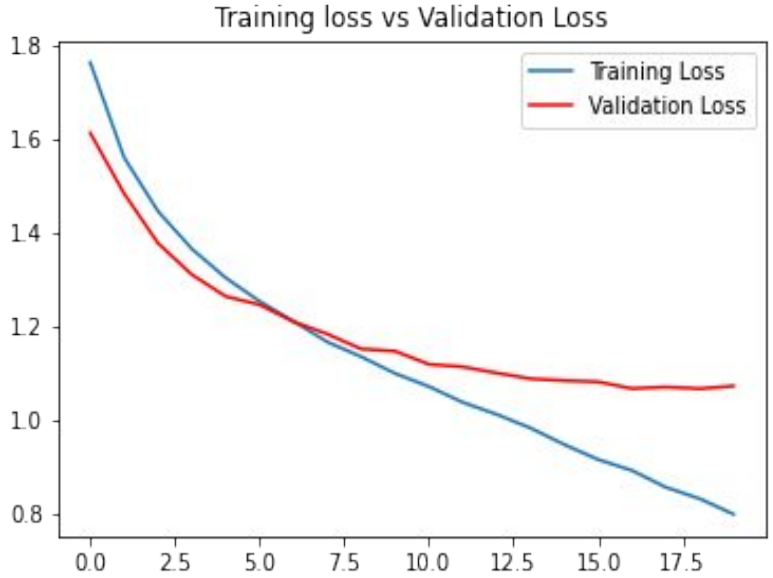
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 32)	320
conv2d_1 (Conv2D)	(None, 44, 44, 64)	18496
max_pooling2d	(None, 22, 22, 64)	0
dropout (Dropout)	(None, 22, 22, 64)	0
conv2d_2 (Conv2D)	(None, 20, 20, 128)	73856
Max_pooling2d_1	(None, 10, 10, 128)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_2	(None, 4, 4, 128)	0
dropout_1 (Dropout)	(None, 4, 4, 128)	0

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dropout_2 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 7)	7175

Total params: 2,345,607
Trainable params: 2,345,607
Non-trainable params: 0

Model Evaluation





We got 70.24% train accuracy and 61.06% validation accuracy using a simple CNN model but this model was not able to predict emotions precisely when we tested it in real time.

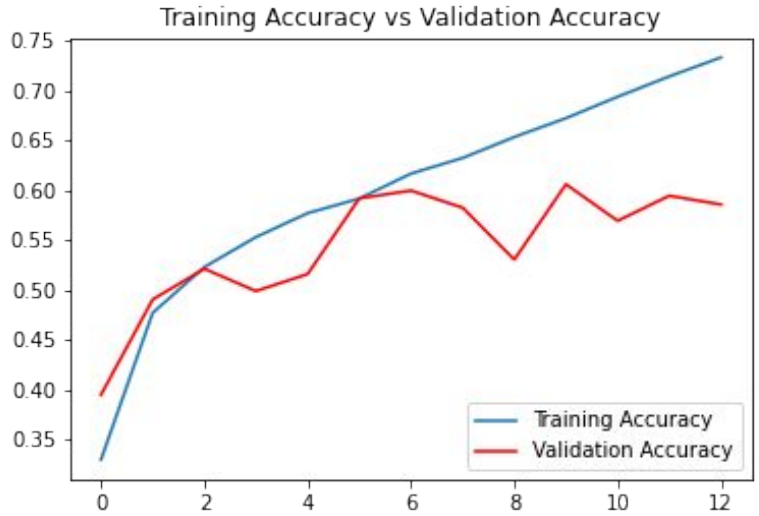
CNN with deep layers

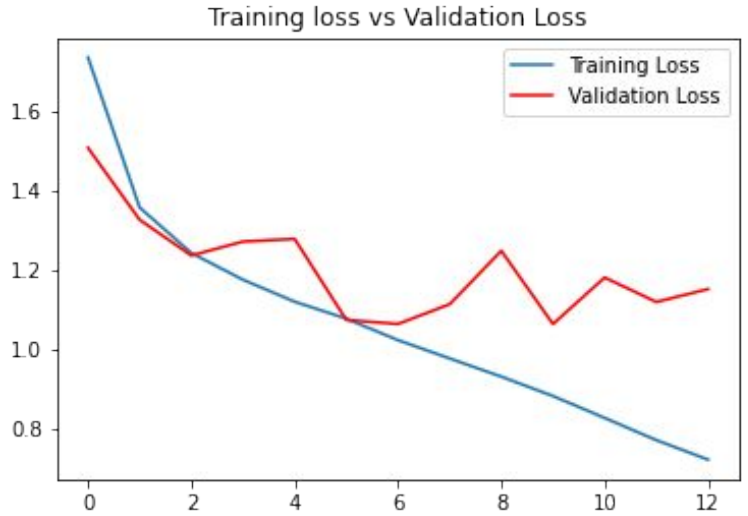
Added more layers to the model to make it deep for better results.

Due to requirement of gpu and to train the model in less time we have used kaggle notebook to train this model.

Architecture	Compilation & Call Back	Training
Sequential model Four Conv2D layers- 64, 128, 512, 512 nodes Activation - ReLU, Max Pooling, Dropout, Batch Normalization Flatten Layer Two fully connected layer- Dense layers - 256, 512 nodes Output Activation - Softmax 7 nodes in output layer	Compile model using three parameters: Optimizer - Adam(lr=0.001) Loss - categorical cross-entropy Metrics - Accuracy Early Stopping to halt the training of Neural Networks at the right time	Trained model with the following parameters: training data steps_per_epoch = 28709 // 32 validation data validation_steps = 7178 // 32 epochs = 30 Training stopped early after 13 epochs

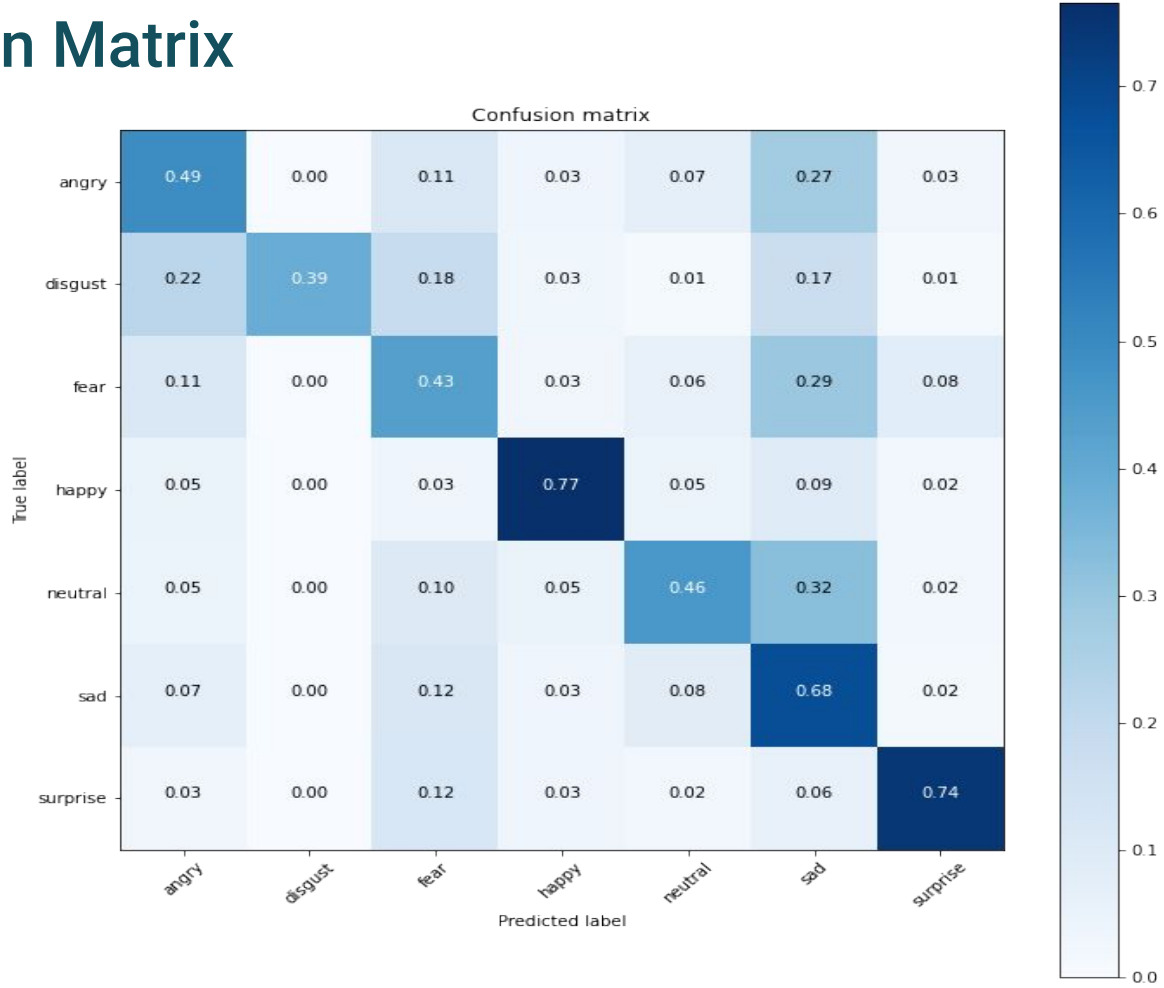
Model Evaluation





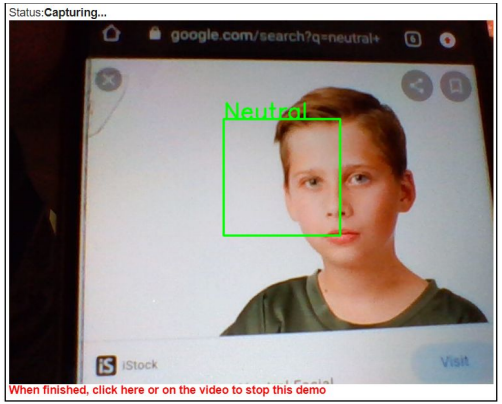
73.27% Train accuracy & 58.55% validation accuracy

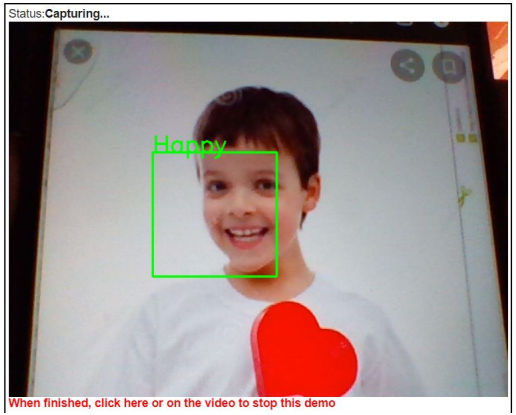
Confusion Matrix



Testing Model using Webcam

- Tested model in real-time using face detection
- Used Haar-cascade for face detection
- Used OpenCV to read frames and for image processing





Face Detection with Haar Cascade

- It is an Object Detection Algorithm used to identify faces in an image or a real time video.
- The algorithm uses edge or line detection features proposed by Viola and Jones.
- The model created from this training is available at the OpenCV GitHub repository.
- The repository has the models stored in XML files, and can be read with the OpenCV methods.
- These include models for face detection, eye detection, upper body and lower body detection, license plate detection etc.

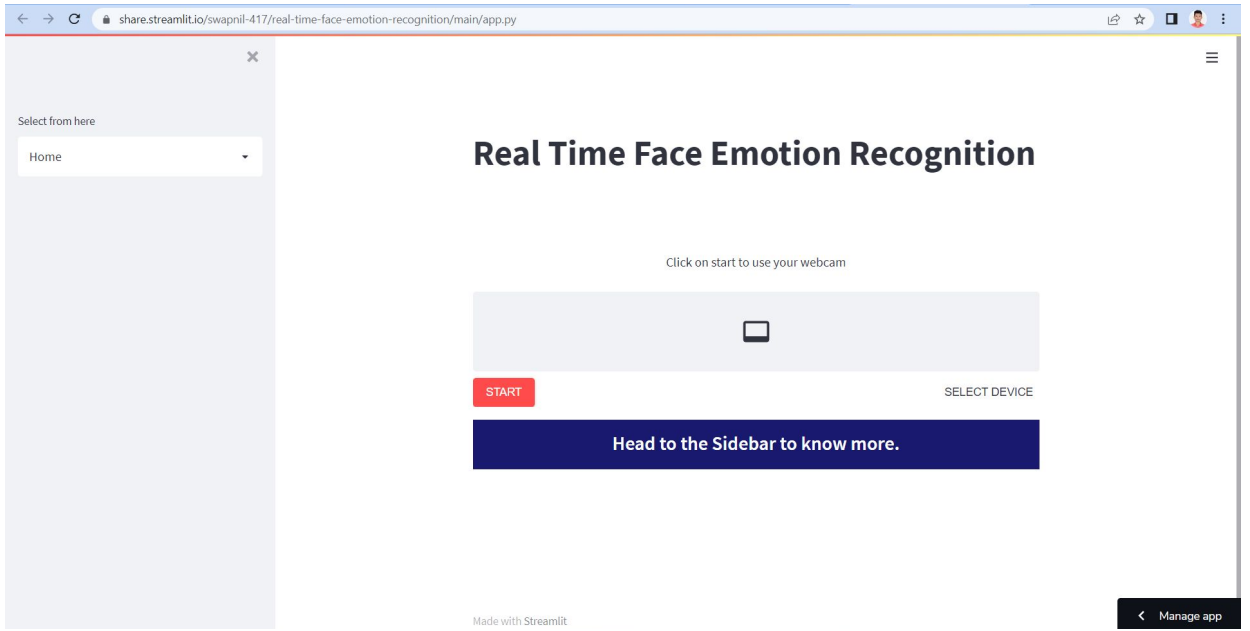
OpenCV GitHub Repo link:- https://github.com/opencv/opencv/tree/master/data/haarcascades

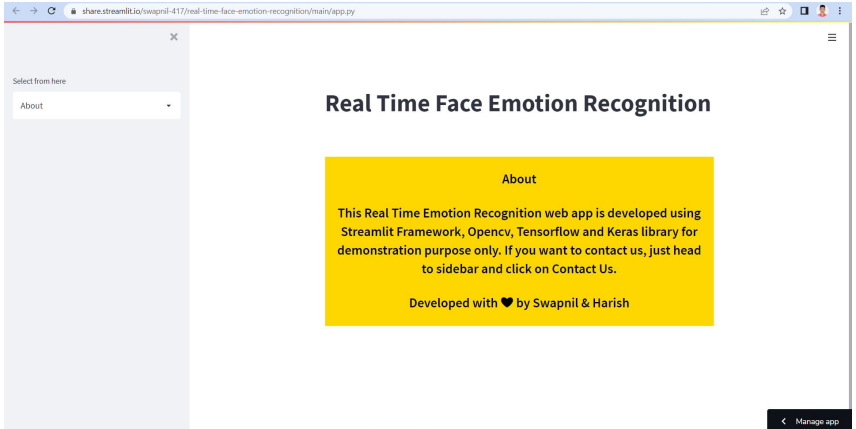
Model Deployment

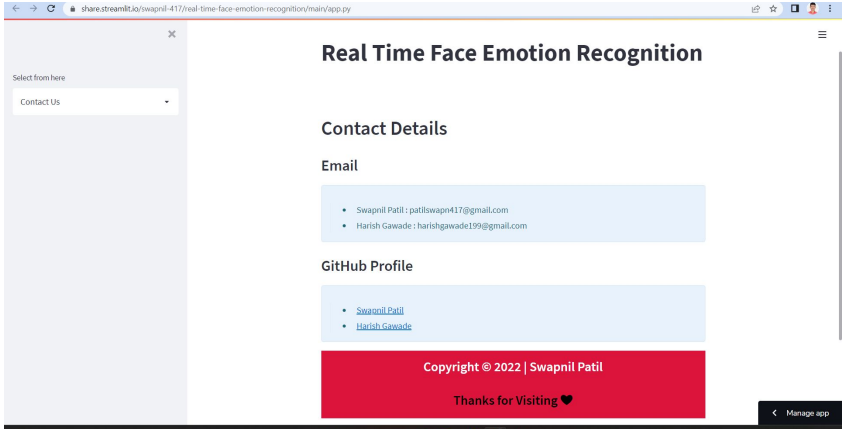
- We have created a front end using streamlit for web app.
- Used streamlit-webrtc which helped to deal with real time video streams.
- Image captured from the webcam is sent to the Video transform function to detect emotions.
- We deployed the model on the streamlit sharing cloud platform.

Streamlit Link - https://share.streamlit.io/swapnil-417/real-time-face-emotion-recognition/main/app.py

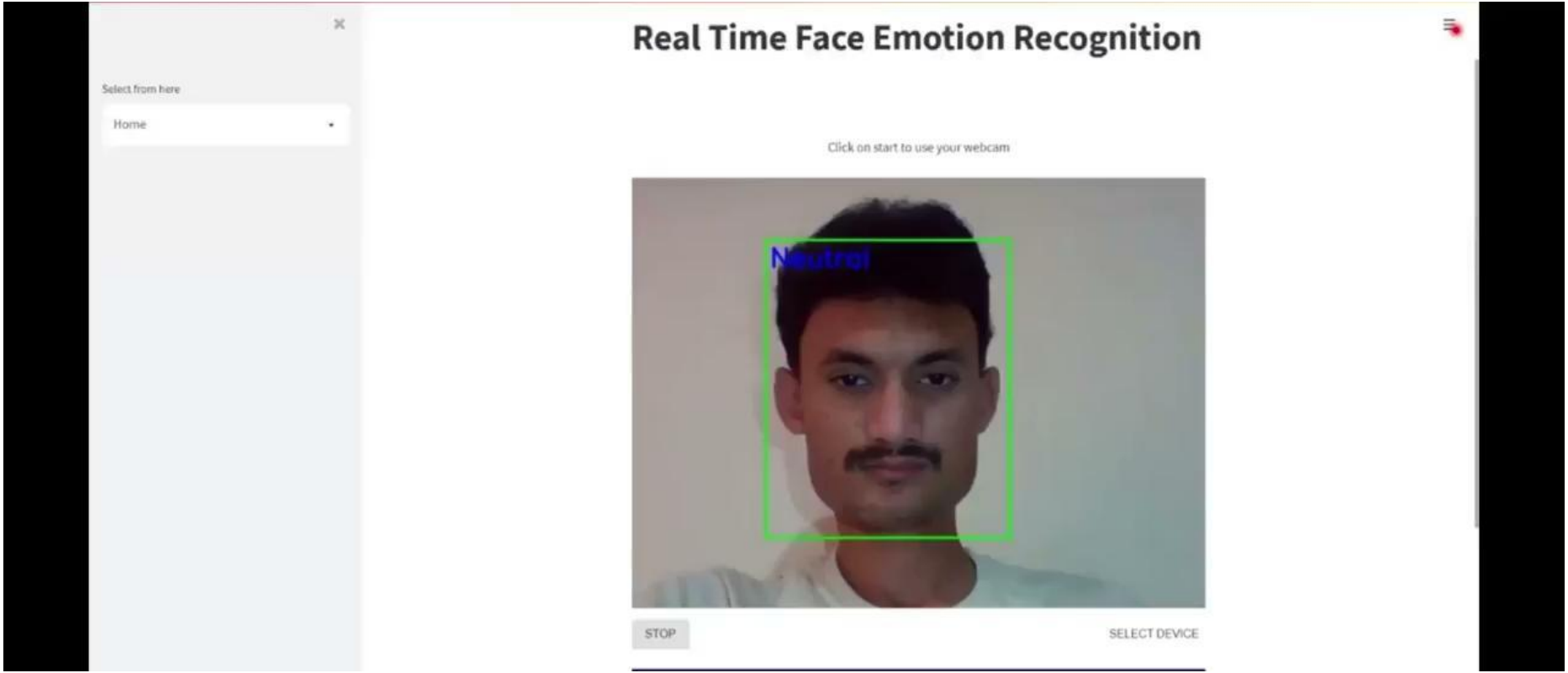
Web App


share.streamlit.io/swapnil-417/real-time-face-emotion-recognition/main/app.py
Select from here
Home
Real Time Face Emotion Recognition
Click on start to use your webcam
START
SELECT DEVICE
Head to the Sidebar to know more.
Made with Streamlit
Manage app


share.streamlit.io/swapnil-417/real-time-face-emotion-recognition/main/app.py
Select from here
About
Real Time Face Emotion Recognition
About
This Real Time Emotion Recognition web app is developed using Streamlit Framework, Opencv, Tensorflow and Keras library for demonstration purpose only. If you want to contact us, just head to sidebar and click on Contact Us.
Developed with ❤ by Swapnil & Harish
Manage app


share.streamlit.io/swapnil-417/real-time-face-emotion-recognition/main/app.py
Select from here
Contact Us
Real Time Face Emotion Recognition
Contact Details
Email
Swapnil Patil : patilswapn417@gmail.com
Harish Gawade : harishgawade199@gmail.com
GitHub Profile
Swapnil Patil
Harish Gawade
Copyright © 2022 | Swapnil Patil
Thanks for Visiting ❤
Manage app

Demo Preview



Challenges

- Large dataset to handle and it containing images
- Limited GPU access on Google Colab and time consuming
- Convolutional Neural Network is so deep and have many parameters. So we made changes very careful.
- Evaluation metrics shown different scenario than real testing of models.
- Running webcam on Google Colab
- Running webapp on streamlit as it do not support webcam
- We tried deploying it on heroku but it was throwing an error of slug size. We successfully reduced the slug size but later it showed some application error which we tried to debug but failed.

Conclusion

- Using a deep learning model based on the architecture of CNN, we constructed a framework to analyze student's emotions.
- Simple CNN model gave 70.24% train accuracy and 61.06% validation accuracy but not able to predict emotions precisely when tested in real time.
- CNN using deep layers has 73.27% train accuracy and 58.55% validation accuracy.
- This model was accurate in predicting tested in real time.
- There were less disgust images in the dataset that's why model confused in detecting disgust.
- This model can help the teachers to adjust their teaching strategies accordingly to improve the efficiency of online teaching.

Q & A