Emotion Detection
of
The Occupants of A Car

Team Members

Rajath J | PES1201801707

Tanishq Vyas | PES1201800125

Mohit Gaggar | PES1201800112

Mentor Faculty

Dr. Natarajan S

Intel Technovation | 2020

Introduction

Emotion and Behavior detection of occupants of a car.

The huge number of accidents that happen these days are due to drivers not making decisions in the absolute best condition possible due to him or her being angry or sad is a big issue.

This issue can be solved by developing an emotion recognition system not only for the driver but also the passengers in the car. Also getting to know how the passengers are experiencing their ride, is a great help for many a driver and for many civilian transportation companies (uber, ola .etc.).

We will try to solve this issue through this project by developing an end to end emotion detection system which detects the emotions of the occupants of the car in real time and notifies the user about his/her emotions so he/she can act about it and be safer on the road.

This project focuses on developing a client server system where the client is the camera which captures video of the occupants of the car and sends it over to the server which detects emotions of the people in the car in real time and gives corresponding notifications based on their emotions. This enables us to make computation heavy tasks. The car at the end is intact and doesn't require much computing power.

Instead of an end to end approach of deep learning the team has made use of a stepwise approach wherein we breakdown the whole process into two main chunks, namely

face detection & **emotion classification**

Thus, helping us in increasing our accuracy. The whole idea is to give good results even in low lighting conditions and face obstruction due to caps, glasses, or any kind of noise.

Steps involved in the process/ Working of our Program:

1. Recording video in real time:

We make use of **Ip WebCam**, an application that allows people to turn their android phone into an internet camera.

For our project prototype we are using the already existing camera in the phones owned by occupants of the car.

This application records the video in real time and sends it to a local network server for processing.

This application was used because it was simple to use and easily accessible to anyone with an android phone. It was also easy to access the video on the same local network and thus making the recorded video accessible easily to the client for use.

By using this app no external hardware is needed which helps optimize the process.

This application also takes care of low lighting conditions as it has an inbuilt enhanced night vision mode.

2. Sending video to server for emotion detection:

The client uploads the video clip originally recorded by the camera on the android phones of occupants that is accessed by it on the local server setup by Ipwebcam to Google Drive.

Google api client was used for this task as this client allows us to access the google drive api with ease.

The python module pydrive was used to act as an interface between the client, google drive and the server as it is very easy to authorize the client to access data on the drive and is very compatible with google api client.

For the task of uploading videos to google drive, the client uses pydrive and python (uploads 4 seconds of video snippet for every 30 seconds of video recorded by the camera to decrease the amount of processing to be done and make it more feasible) after which the server downloads the videos from google drive for further processing and emotion detection.

3. Detecting faces in video:

For the task of detecting faces in the video so emotion recognition can take place the **MTCNN** face detection algorithm was used with the help of the freely available python pip package for the same.

MTCNN was used instead of the usual and more trivial haar-cascade method as it is much faster and produces better results.

Since the input is in video format, we extract the frames from the video and faces are extracted from these frames.

Since the video might contain multiple faces the faces are extracted separately and stored.

4. Emotion detection:

The faces extracted using MTCNN are passed through the convolutional neural network which is trained for the task of emotion detection.

The **model (CNN)** contains 4 convolutional and 3 dense ‘elu’ activated layers and is trained on thousands of images of each of the 5 categories i.e. angry, sad, happy, neutral and surprised.

All the layers were normalized using the batch normalization technique which helps decrease the training time of each epoch and produces better accuracy.

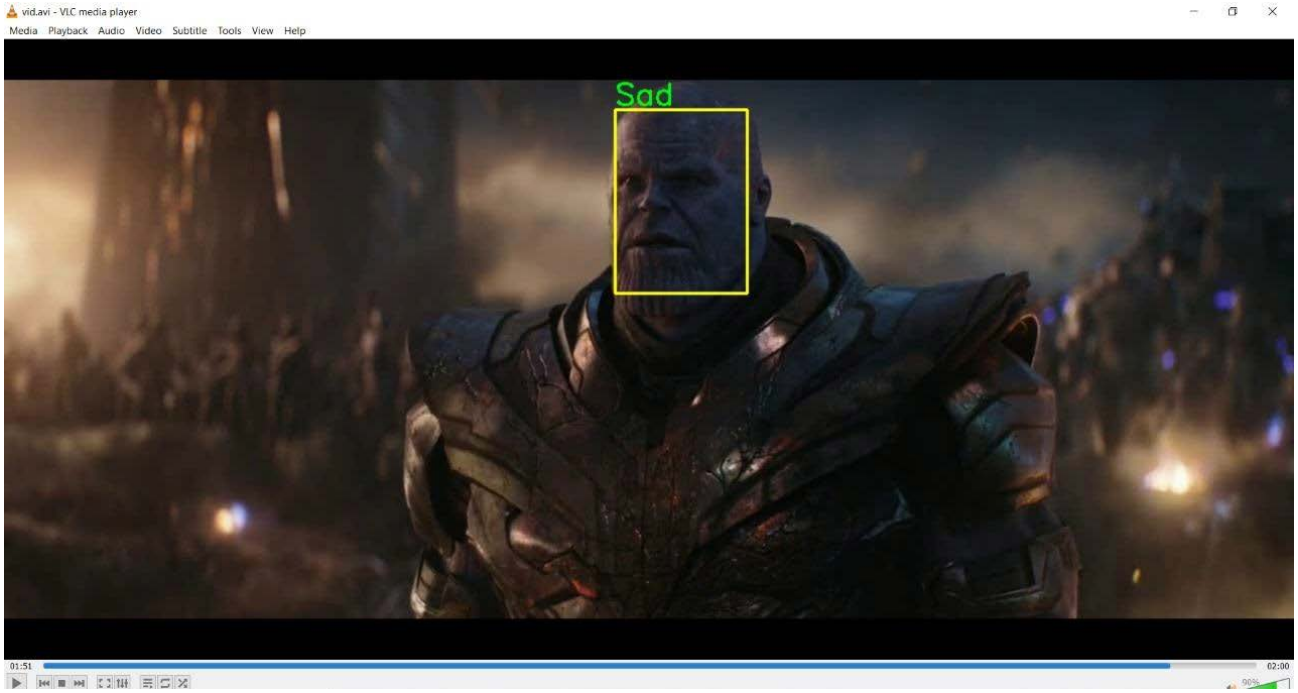
A certain amount of dropout was also added to each of the layers to decrease time taken for training without any real decrease in accuracy.

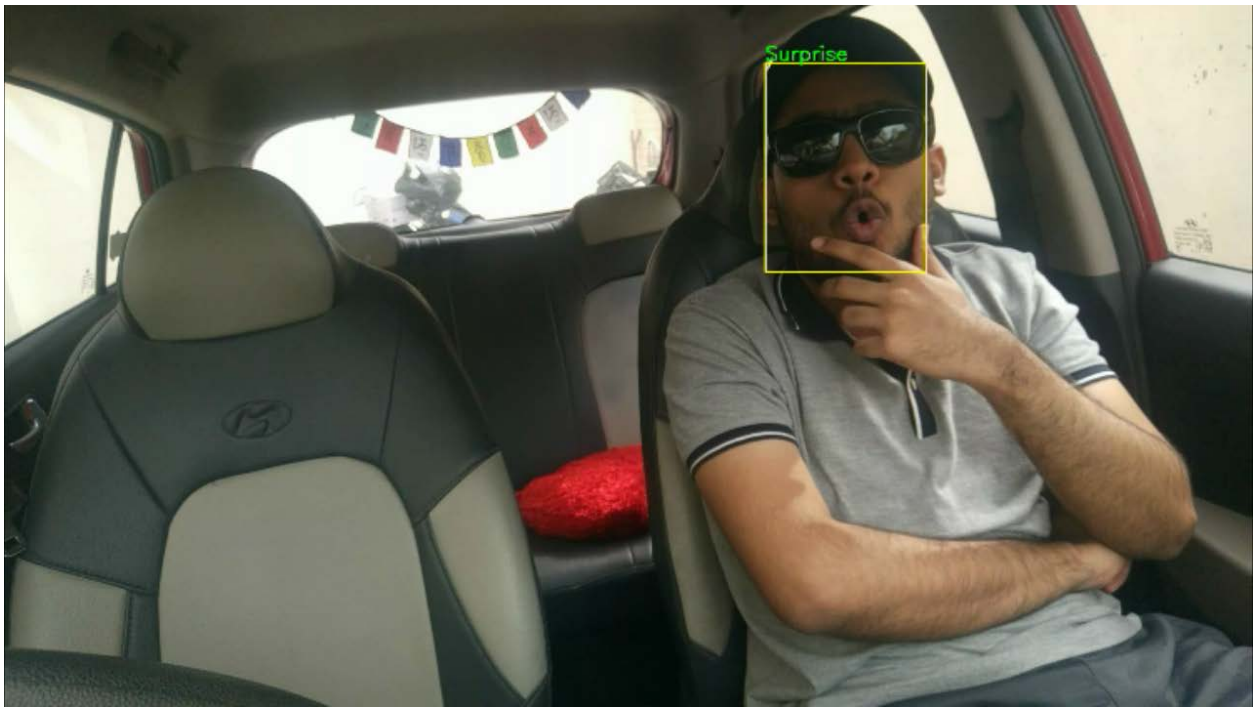
The mode uses categorical cross entropy loss function as it is found to be the best performing function for such a problem and uses adam optimizer which acts as a replacement algorithm for stochastic gradient descent and helps train the model.

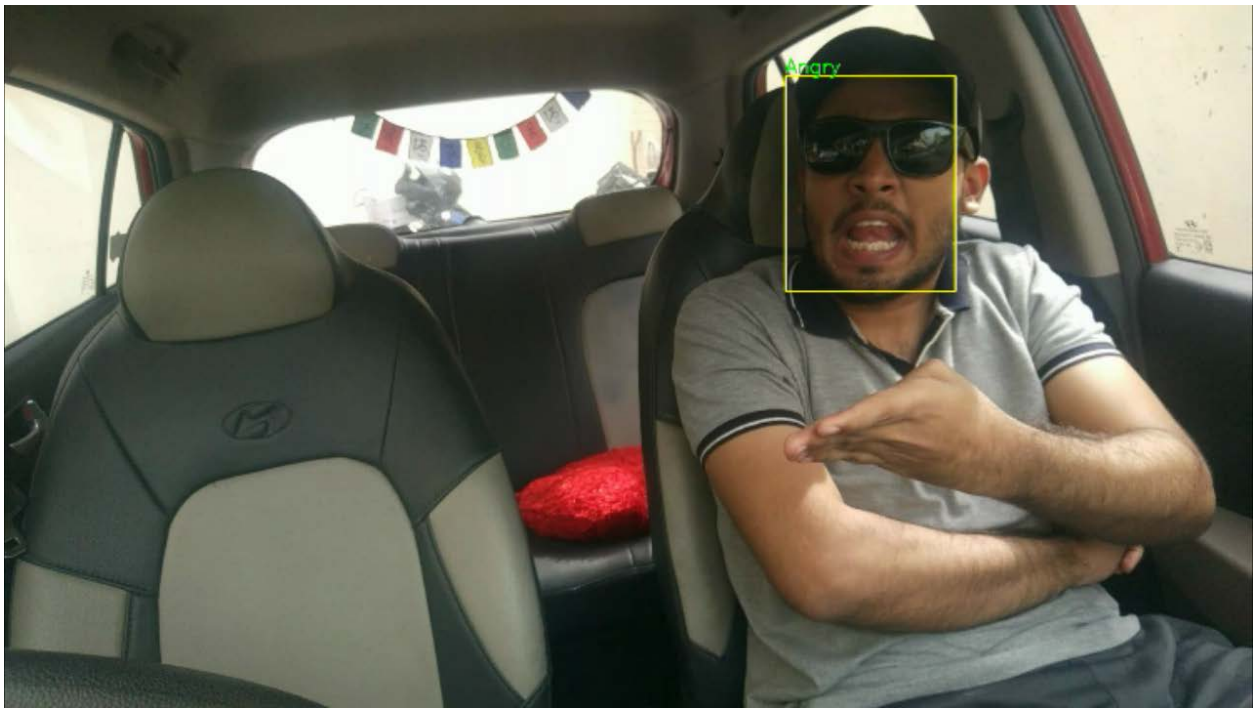
The **model** can achieve a validation accuracy of approximately 82 percent and testing accuracy of 80 percent.

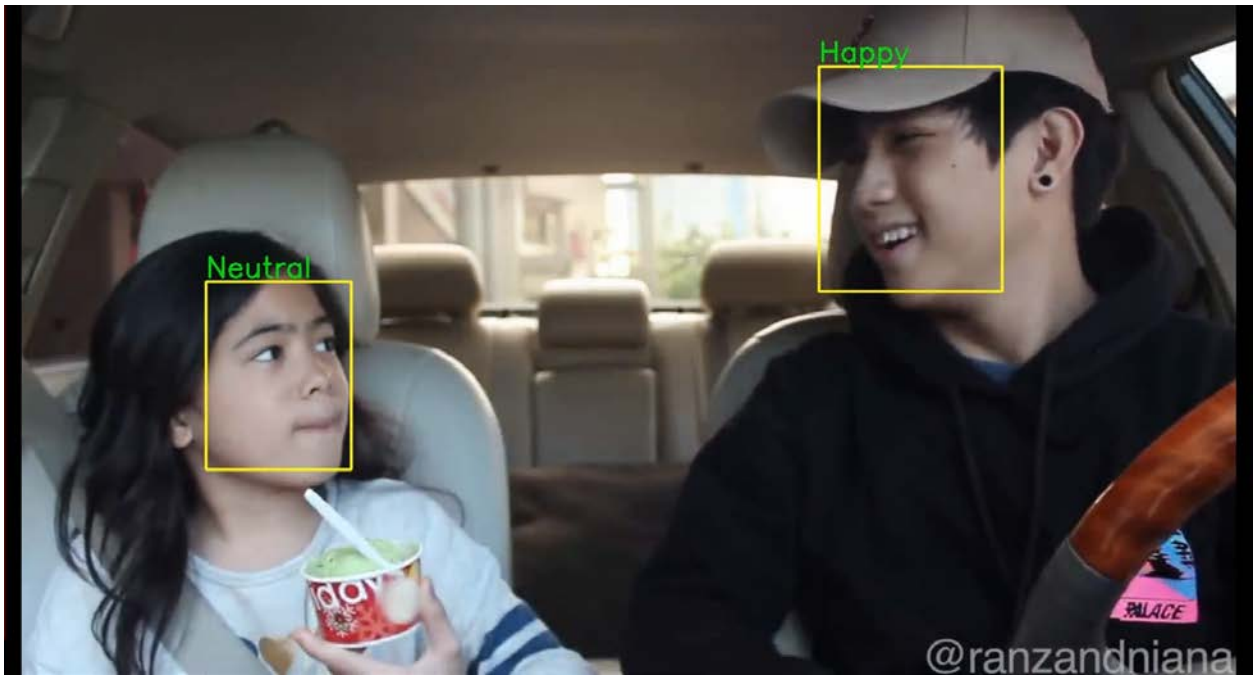
The result of emotion detection is to give the image one of the 5 tags ie Angry, Happy, Neutral, Sad, Surprised. In the initial 4 second video emotion detection is done on the same face as many times as it appears and the final output of emotion detection on the 4 second video is a dictionary with key as the emotions and values as the number of times those emotions were detected on the face in that duration.

Emotion detection in example clips.




Surprise


Angry



5. Results & Discussion:

The result key value pair generated by the emotion detector is uploaded to drive and also sent to the user by email along with the link to download the output video.

The results are again uploaded to Google drive using the pydrive module in python.

The email functionality is added so the user gets notified. For the task of sending emails we use the python modules email and smtplib for defining mime types of the email and create smtp sessions to send emails.

The user is only notified when some emotion is detected as a majority in that period.

A complete video implementation of our program can be found **here** .

A video of our prototype process can be found **here** .

```
result of 1.avi: and emotions found and its weight
{'Angry': 10, 'Happy': 120, 'Neutral': 384, 'Sad': 22, 'Surprise': 21}

result of 2.avi: and emotions found and its weight
{'Angry': 171, 'Happy': 33, 'Neutral': 230, 'Sad': 242, 'Surprise': 98}

result of 3.avi: and emotions found and its weight
{'Angry': 176, 'Happy': 98, 'Neutral': 270, 'Sad': 135, 'Surprise': 49}

result of 4.avi: and emotions found and its weight
{'Angry': 49, 'Happy': 1, 'Neutral': 133, 'Sad': 45, 'Surprise': 18}

result of 5.avi: and emotions found and its weight
{'Angry': 151, 'Happy': 193, 'Neutral': 393, 'Sad': 45, 'Surprise': 41}

result of 6.avi: and emotions found and its weight
{'Angry': 76, 'Happy': 72, 'Neutral': 504, 'Sad': 30, 'Surprise': 15}
```

FIG : Mailed Results for Test Videos

```
From videos of resolution : 1080p

1. {'Angry': 112, 'Happy': 21, 'Neutral': 86, 'Sad': 149, 'Surprise': 3}
2. {'Angry': 44, 'Happy': 36, 'Neutral': 180, 'Sad': 33, 'Surprise': 15}
3. {'Angry': 5, 'Happy': 0, 'Neutral': 27, 'Sad': 16, 'Surprise': 0}
4. {'Angry': 1, 'Happy': 8, 'Neutral': 216, 'Sad': 28, 'Surprise': 19}
5. {'Angry': 147, 'Happy': 24, 'Neutral': 144, 'Sad': 93, 'Surprise': 1}
6. {'Angry': 51, 'Happy': 36, 'Neutral': 193, 'Sad': 39, 'Surprise': 10}

From videos of resolution : 540p

1. {'Angry': 102, 'Happy': 19, 'Neutral': 108, 'Sad': 130, 'Surprise': 4}
2. {'Angry': 39, 'Happy': 39, 'Neutral': 198, 'Sad': 20, 'Surprise': 15}
3. {'Angry': 6, 'Happy': 0, 'Neutral': 27, 'Sad': 11, 'Surprise': 0}
4. {'Angry': 1, 'Happy': 7, 'Neutral': 224, 'Sad': 16, 'Surprise': 25}
5. {'Angry': 134, 'Happy': 21, 'Neutral': 148, 'Sad': 80, 'Surprise': 0}
6. {'Angry': 38, 'Happy': 33, 'Neutral': 170, 'Sad': 31, 'Surprise': 8}
```

FIG : Resolution Respective Comparison

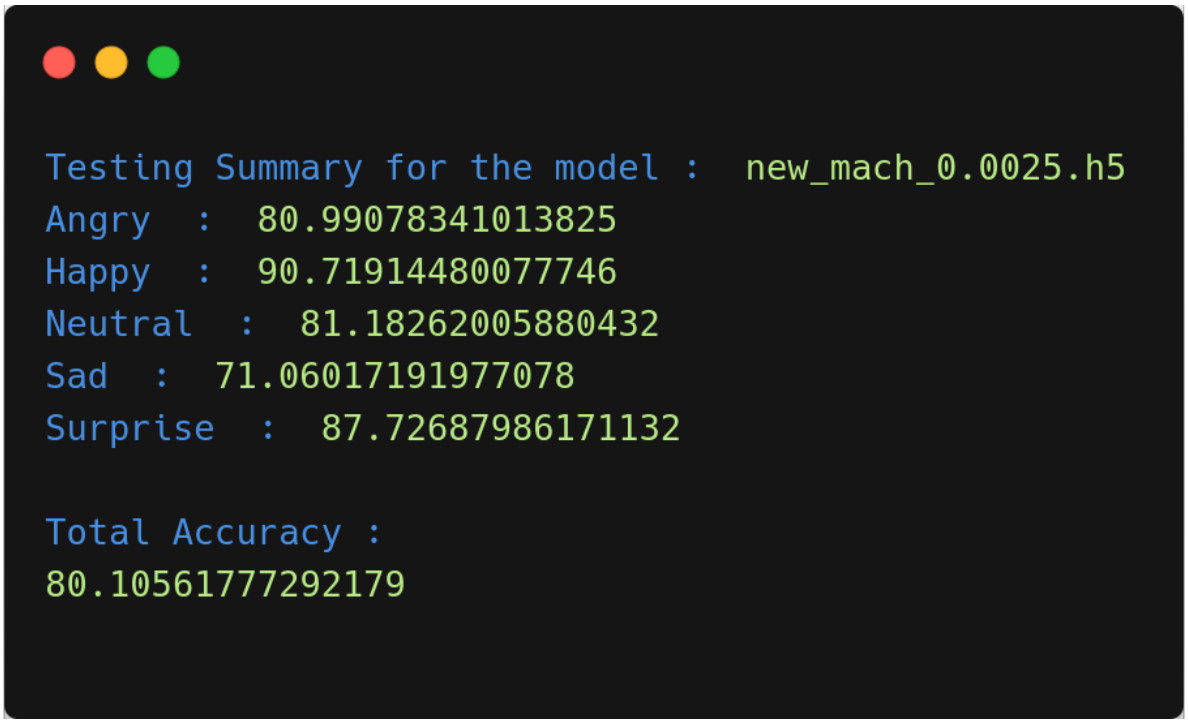


FIG : Testing Accuracy Results

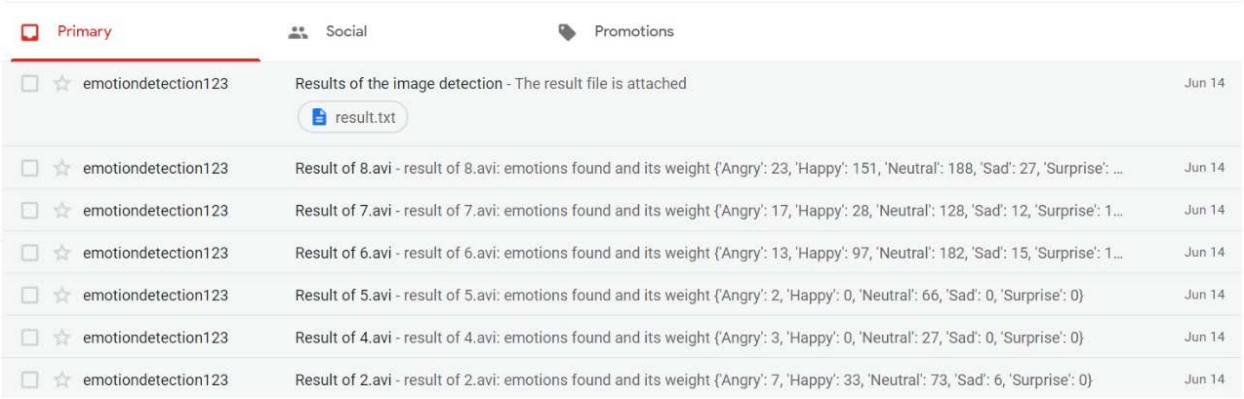


FIG : Mailed Results

6. Future Work

- Adding voice notes to notify users because email is not always convenient.
- Using simpler and faster means to transfer video from client to server.