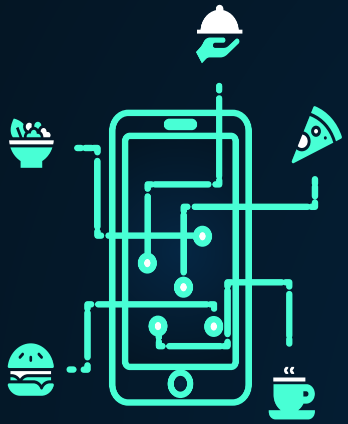
Kombuder Bishon

Team Members:
Giri Prasath
S. Shreevignesh
Ashwin Gopinath
Sudhansh Yelishetty

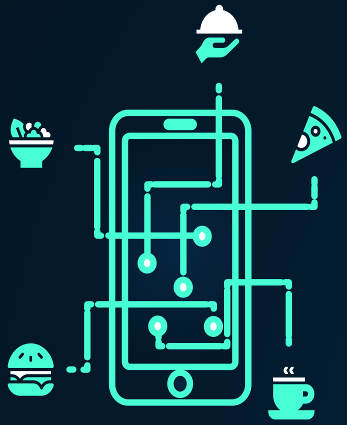
Project topic

Facial recognition/ verification: Automatic Naming of Characters in TV Video



Objective

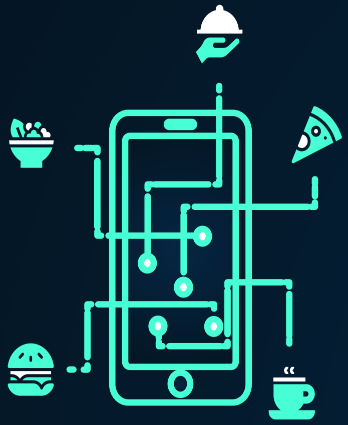
Automatic naming of characters in TV Video, and increasing its accuracy by combining multiple sources of information, both visual (using concepts of facial detection) and textual (Subtitles).



Method

The method of naming the characters include:

1. automatic generation of time stamped annotation by using subtitles and transcripts.
2. strengthening the supervisory information by identifying when characters are speaking.
3. using complementary cues of face matching to propose common annotations for face tracks

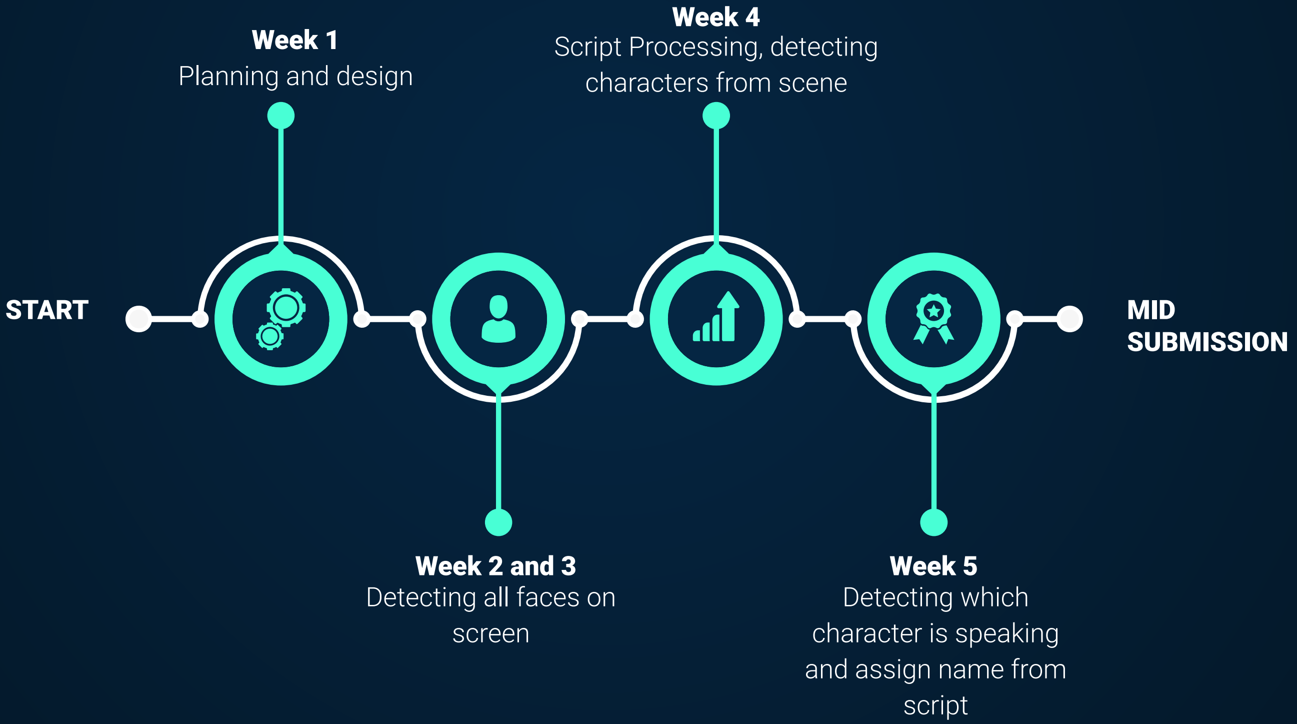


Goals

We are planning as mentioned below:

- **First week**: Planning, designing and solidifying the pipeline of implementation.
- **Second and Third week**: Detecting all faces on the screen.
- **Fourth week**: Implementing script processing and detecting characters in scene from script.
- **Fifth week**: Detecting mouth movements to find who is speaking in the scene and match the characters name from the processed script.

OUR GOAL



START

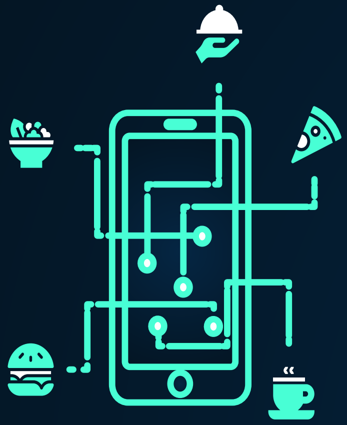
Week 1
Planning and design

Week 2 and 3
Detecting all faces on screen

Week 4
Script Processing, detecting characters from scene

Week 5
Detecting which character is speaking and assign name from script

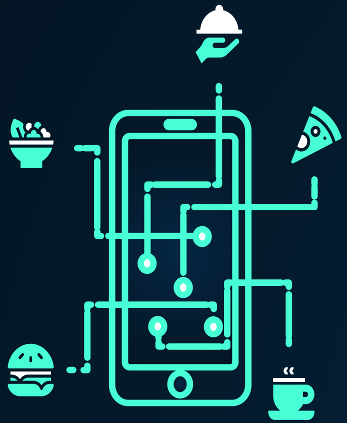
MID SUBMISSION



Detecting all faces on screen

Experiments:

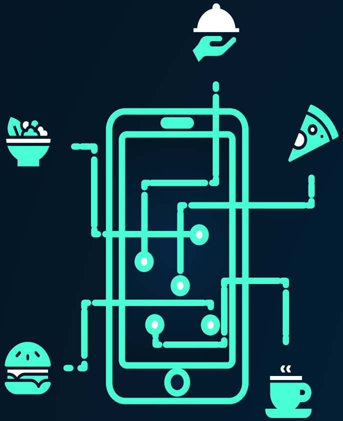
1. First we tried using a Haar Cascade Classifier specifically for full frontal faces, as mentioned in the paper. But this method was slow and unreliable as it gave many False-Positives. In addition to this these, the Haar cascade classifier did not give any distinction between the facial features (eg. mouth movements). Thus, to make the process faster and more reliable, we shifted to a Deep Neural Network-based architecture.



Detecting all faces on screen

Experiments:

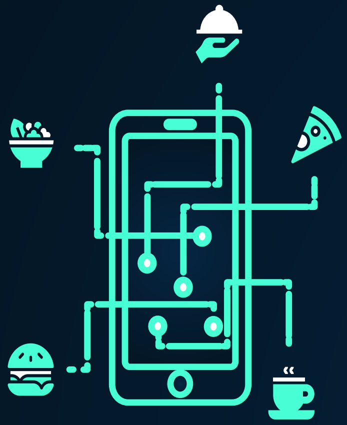
2. The deep neural network (resnet10) method was the fastest method of all the methods we experimented with. But the results were unreliable, the output had lots of false positives and took many many iterations to learn facial features to detect. It could not predict and detect distinct facial features, similar to the Haar implementation. Thus, we had to tradeoff speed to accuracy and detecting facial features and choose an implementation which uses dlib data and functions.



Detecting all faces on screen

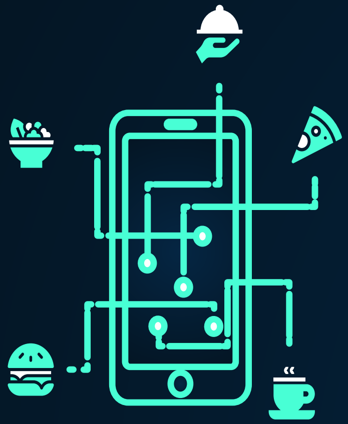
Experiments:

3. At last, we decided to work on the dlib implementation, which even though was slow in comparison to the other methods, had a high accuracy in detecting faces. In addition dlib had much more robust functions which worked on detecting specific keypoints of the faces, which allowed us to detect and localize different facial features and detect independent movement of these facial features (eg. lip movement, explained in the following slides). Using shape priors trained on the 300-W (Faces in The Wild) dataset, we were able to leverage HOG features in our implementation when it came to face and feature detection and localization.



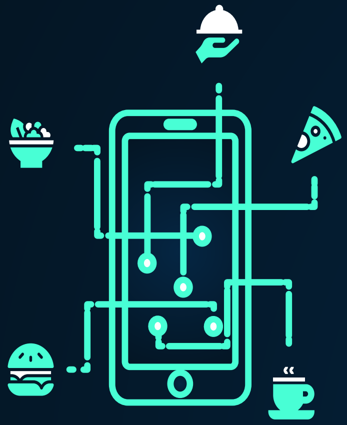
Detecting Lip movements

1. Dlib allowed us to localise facial features with keypoints. Using those keypoints, we used the keypoints around the mouth to detect lip movement in order to identify the speaker. However, this started giving false positives as the keypoints moved even when the head was moving or when the camera moves.



Detecting Lip movements

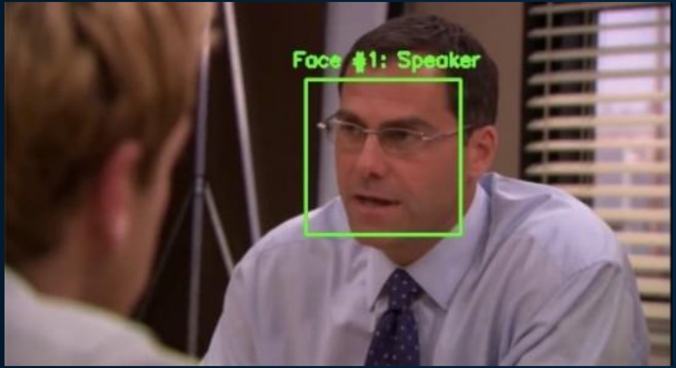
2. In order to tackle this issue, we cropped the mouth region and used it as our region of interest. Now we compute the changes in keypoints position within the ROI to detect lip movement. This solved the previous problem.



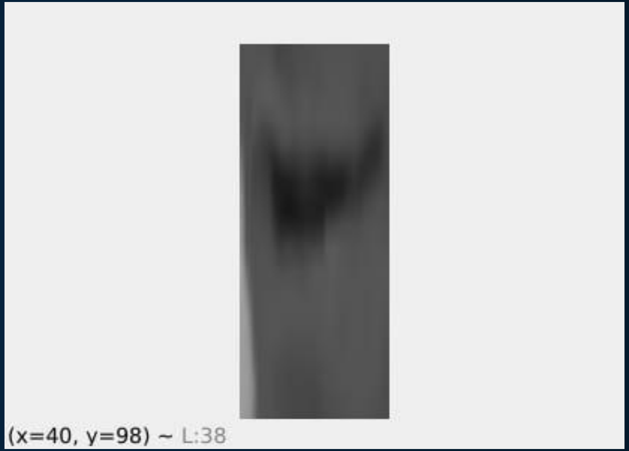
Detecting Lip movements

3. The metric used for identifying lip movements is the Euclidean distance of the key points. As of now, the decision rule we use to identify the speaker is to find the lip on the current frame with the highest movement. We are exploring alternate ways of doing this in the form of thresholding etc.

Detecting Lip movements

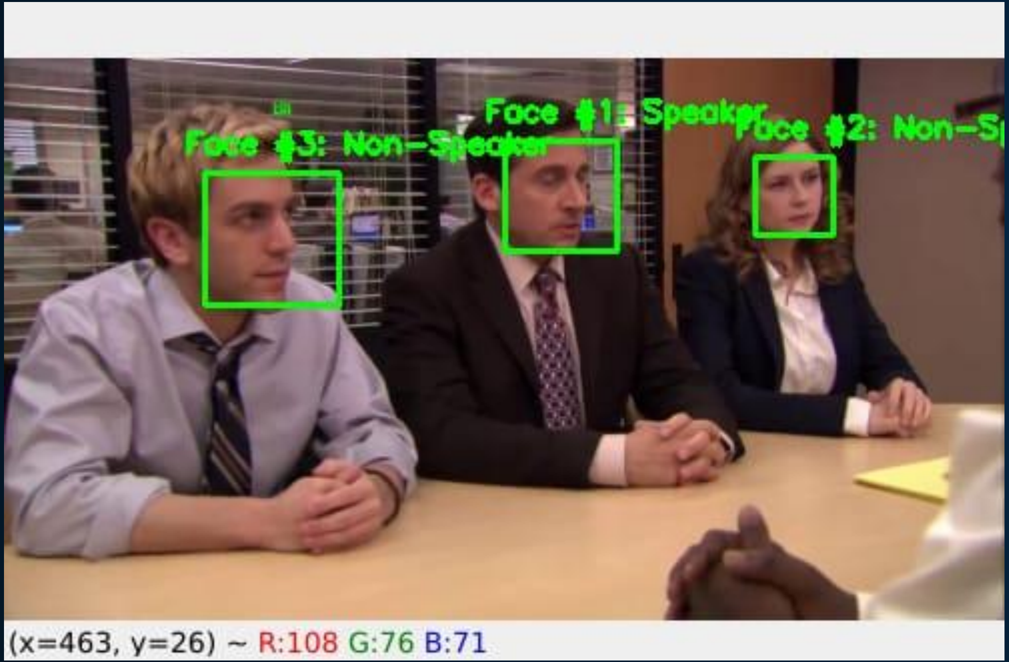


Face of the speaker detected

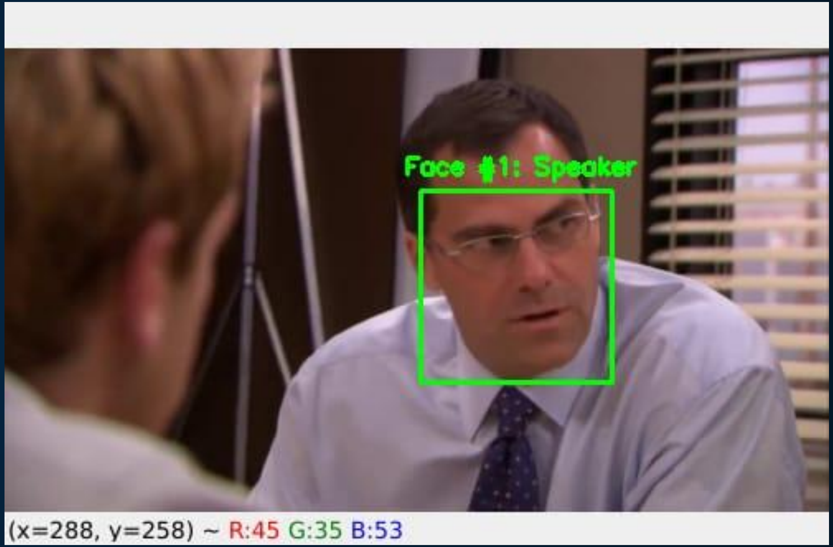


Cropped mouth region

Working Examples

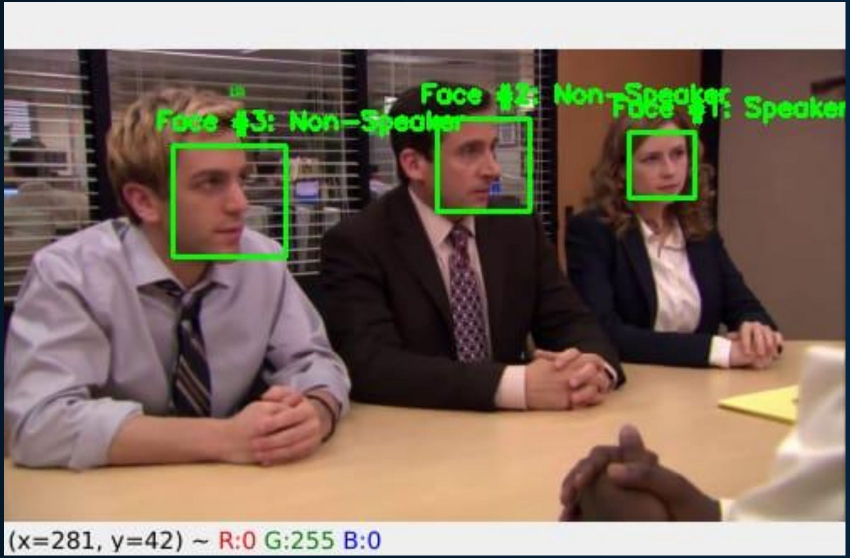


```
(x=463, y=26) ~ R:108 G:76 B:71
```



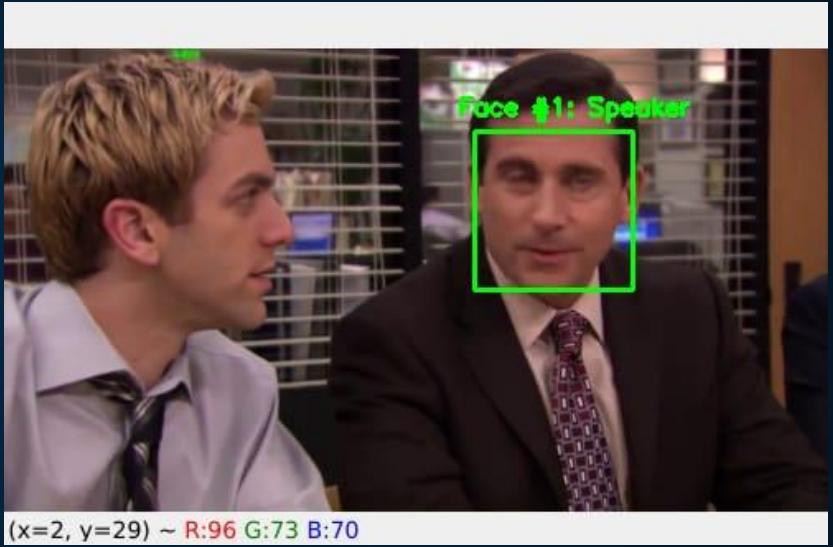
```
(x=288, y=258) ~ R:45 G:35 B:53
```

Exceptions



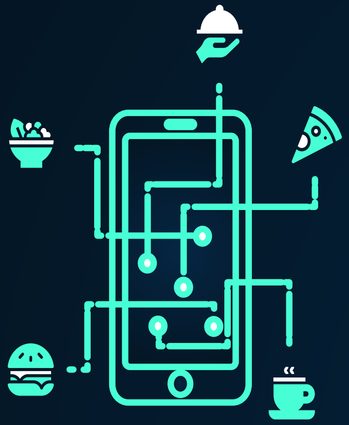
```
(x=281, y=42) ~ R:0 G:255 B:0
```

False positive on speaker identification



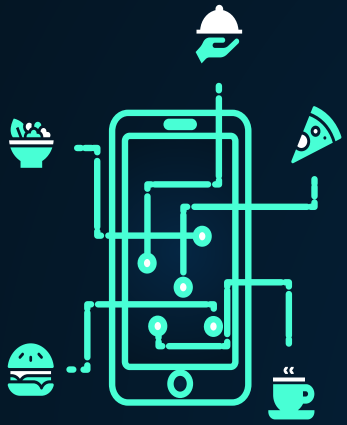
```
(x=2, y=29) ~ R:96 G:73 B:70
```

Ryan’s face not detected because it is turned sideways



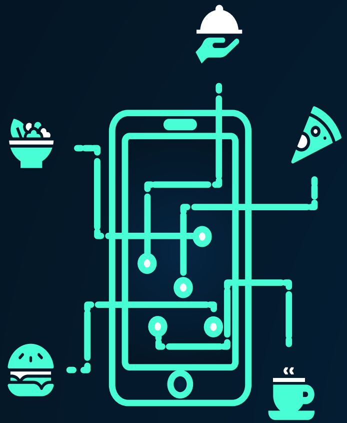
Script Processing

1. We make use of transcript and subtitle files for script processing. The transcript gives information as to who speaks what. The script gives us information as to when each dialogue is spoken.



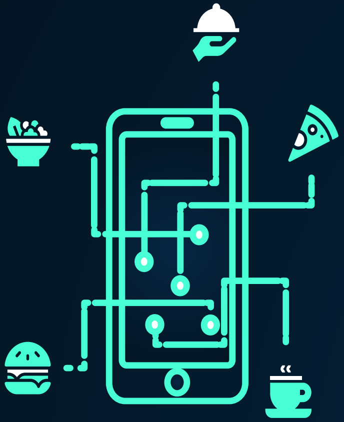
Script Processing

2. We write a script that takes the fps of the video and the subtitle files It will process the files and return a list of speakers in each frame of the video. This is done using the starting and ending timing of every dialogue provided in the subtitle file and converting them into starting and ending frames using the fps



Script Processing

3. After processing, the name of the speaker will be displayed along with a bounding box around his/her face.

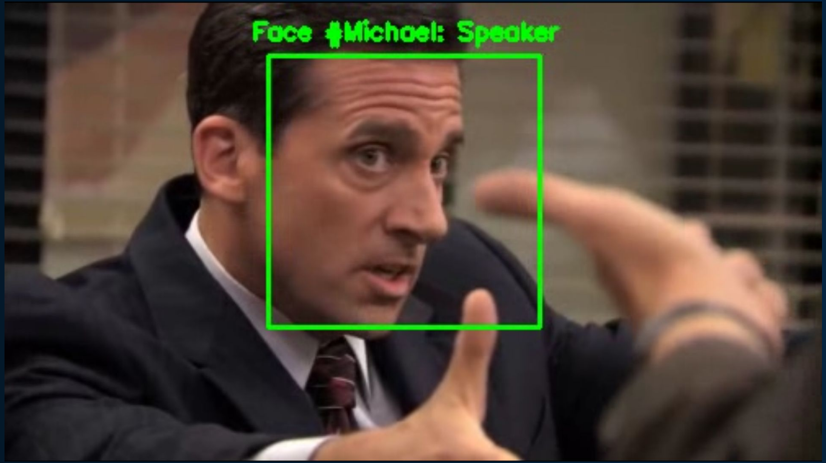


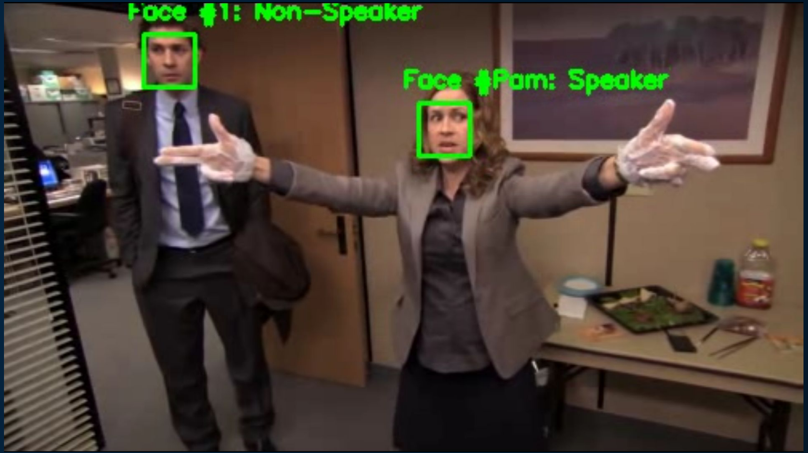
Detecting facial features and assigning names

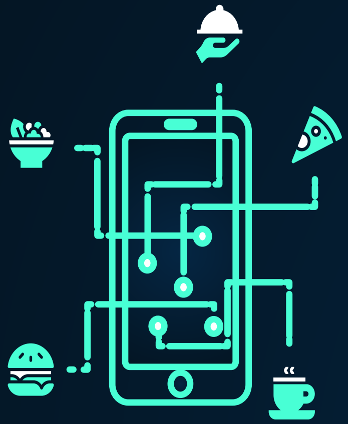
Approach - 1

- In the beginning we only gave names to the speaker, this was implemented by checking who was speaking at that time point and take name from the transcript.

Examples





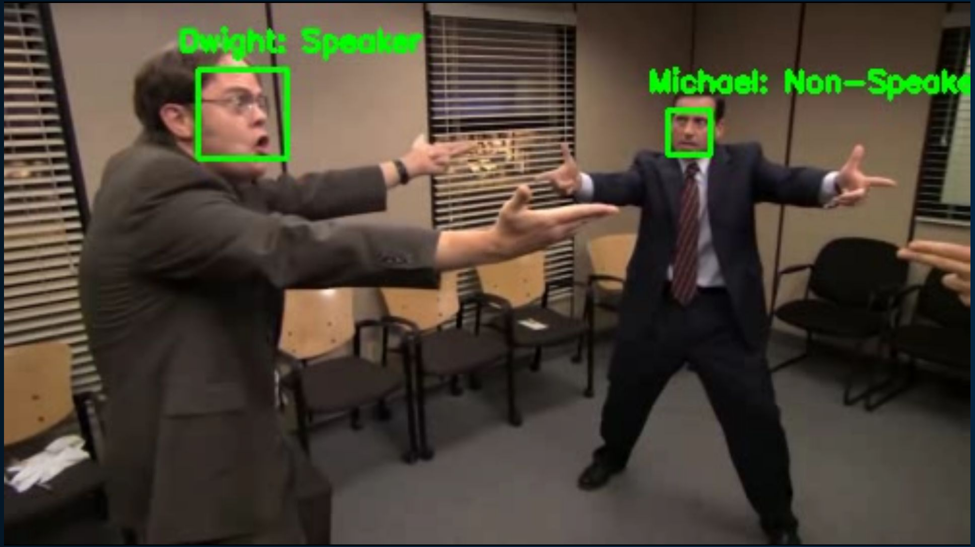


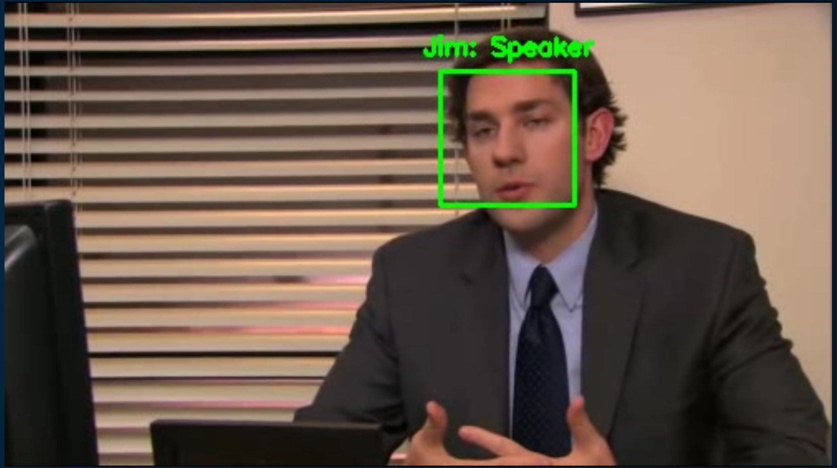
Detecting facial features and assigning names

Approach - 2

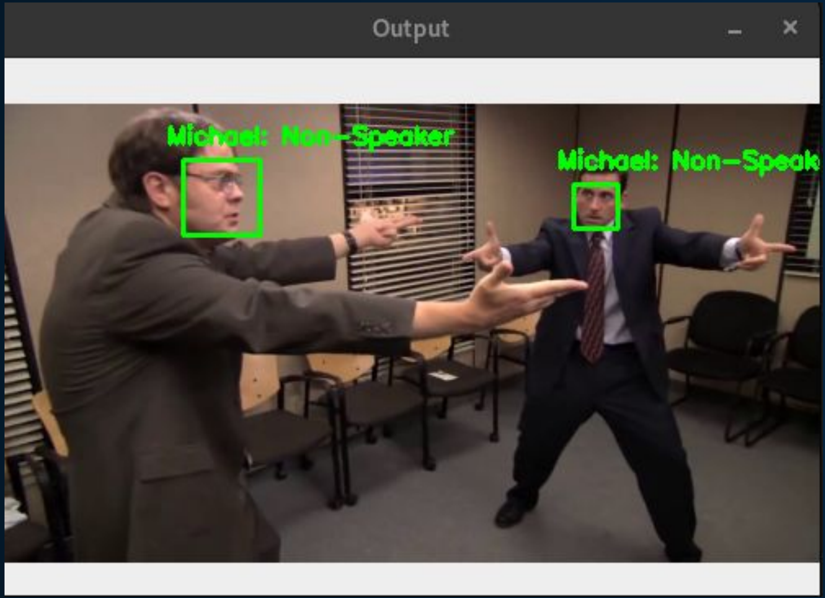
- Then we wanted to give names to the non speakers too, for this we started storing facial features of the characters who get assigned names through the transcript. But this gave us lot of errors as the dataset to refer from is very small and thus has many mistakes in detecting the character

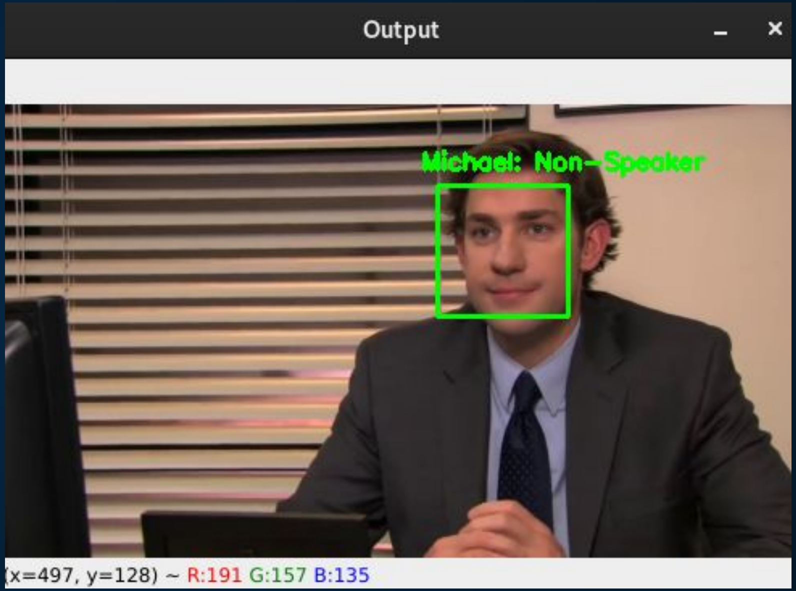
Working Examples



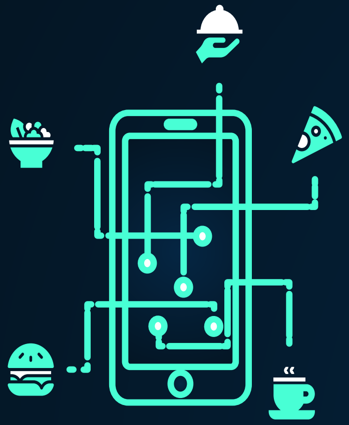


Exceptions





```
(x=497, y=128) ~ R:191 G:157 B:135
```



Detecting facial features and assigning names

Approach - 3

- Apart from our initial dataset, we implemented a method where we considered the transcript's output as our current ground truth and collected faces from the video to aid us in further detection of faces in the sequence.
- The most important disadvantage of this method is of false outputs. This usually happens if a false speaker was detected either through not having a full frontal view of the speaker or incorrect detection of lip movements.