Human Gaze Detection

Tejal Rawale
Sreelaya Devaguptam

CS 5330 : Pattern Recognition and Computer Vision

Table of Contents

Problem Definition

Gaze detection is a technique used to track the direction of a person's gaze.

Aim - to identify and track the observer's point of regard (PoR) or gaze direction.

It uses image recognition with sensors and/or deep learning & image processing techniques to perform eye tracking.

DL Models used for comparison :

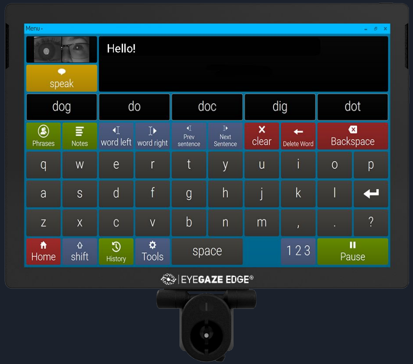
- DLib-64
- DLib-8
- Caffe Model

Compare it with the most popular Object Detection algorithm

provided by Haar Cascade Classifier as a baseline model.

Applications

Safer Cars: To detect whether driver is unfocused on the road while driving.



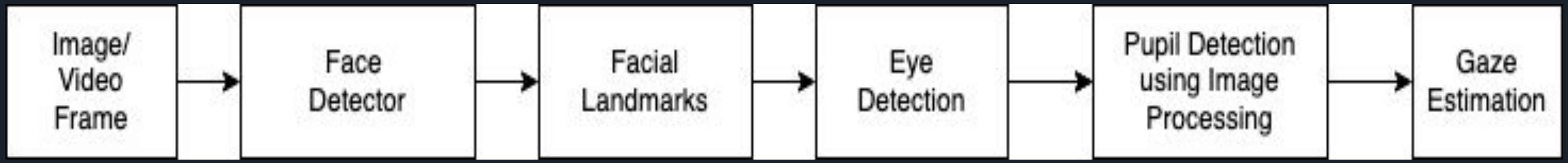
Assistive Technology: Virtual keyboard with eye movements, Eye Edge tracker.

AR / VR: It can be used to make AR/VR experiences more immersive and interactive.

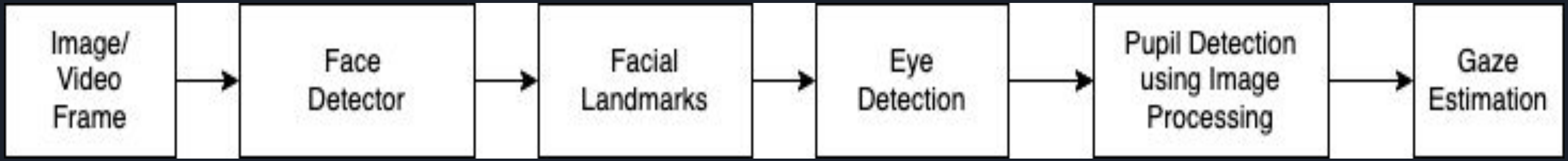
Objectives

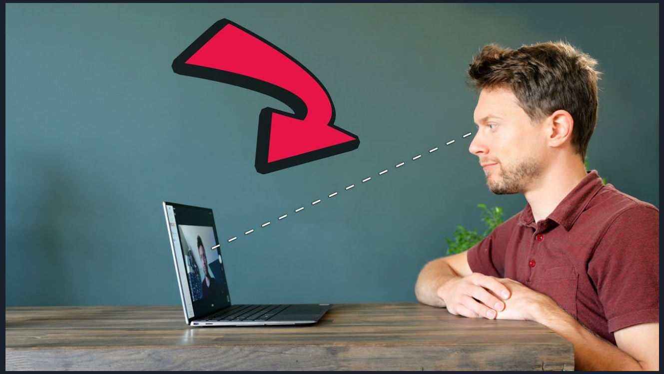
- Existing solutions are great but expensive and complex.
- Create an application which requires minimal resources and is cost effective, at the same time producing optimal results.
- Used OpenCV to create a gaze tracking system which is can be used run on low end devices with basic camera and can be for non-commercial purposes.

Gaze Detection Pipeline

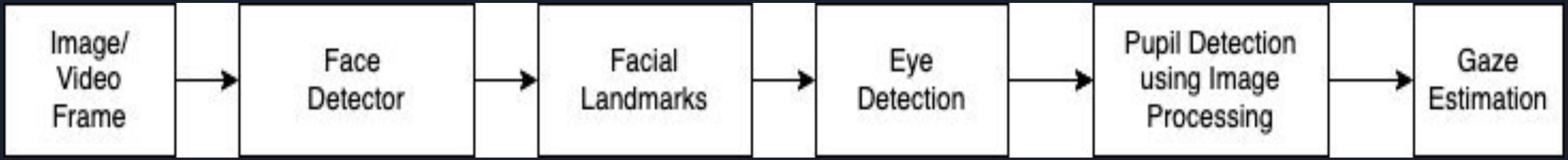


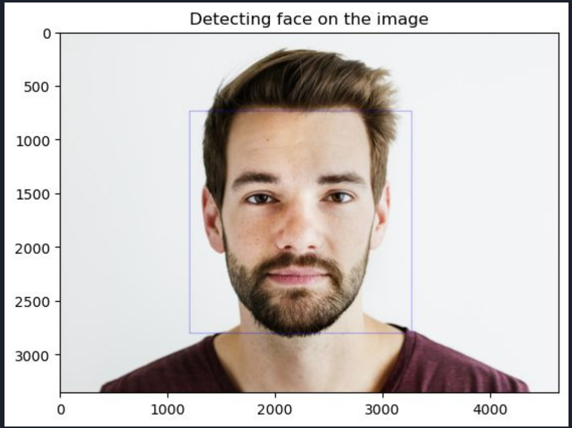
Gaze Detection Pipeline

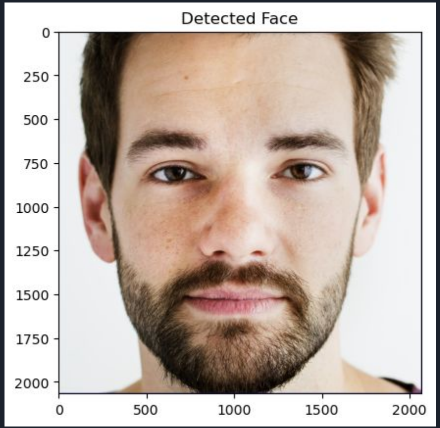




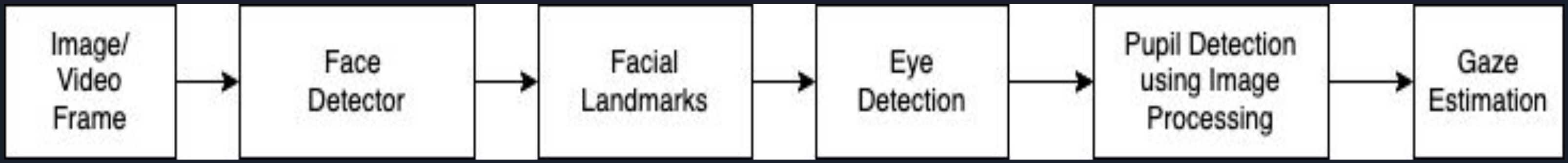
Gaze Detection Pipeline

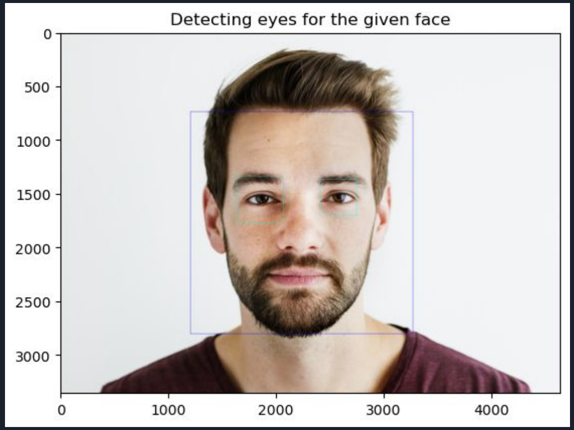




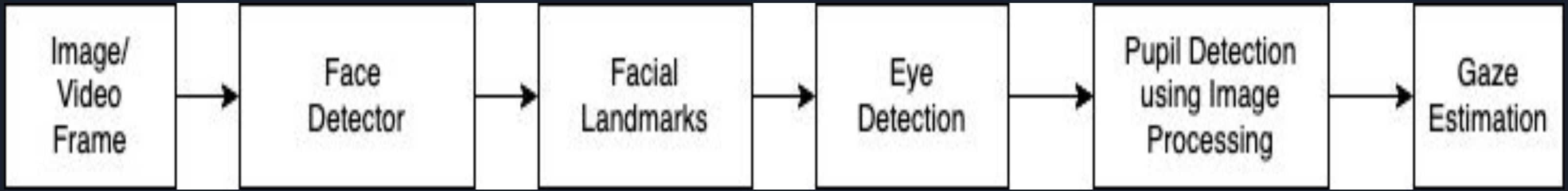


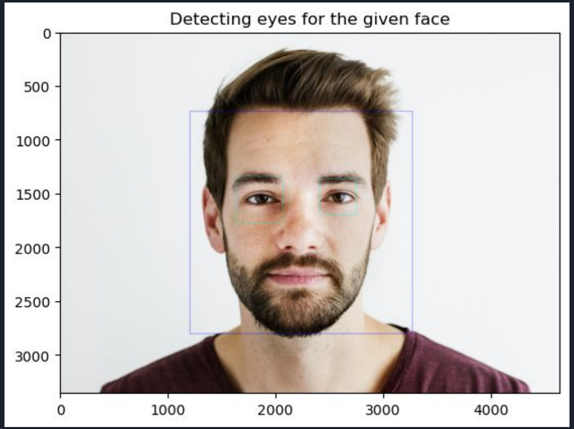
Gaze Detection Pipeline

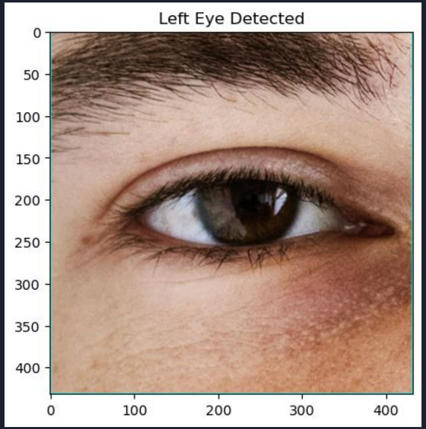




Gaze Detection Pipeline







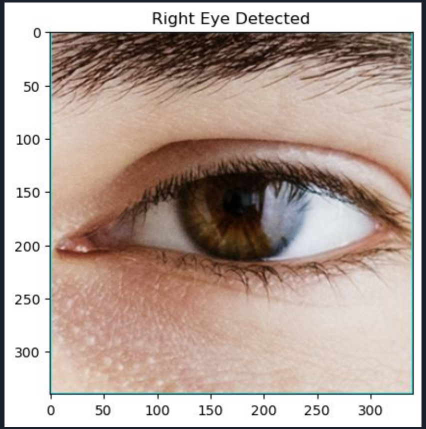
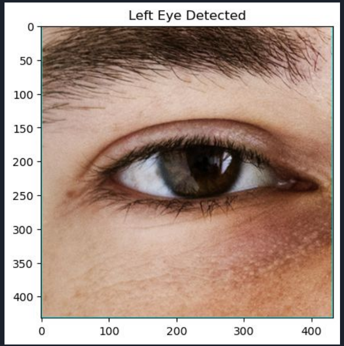


Image Processing Techniques

Precise Eye Region - Remove Eyebrows



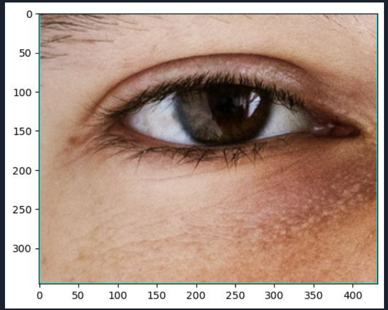
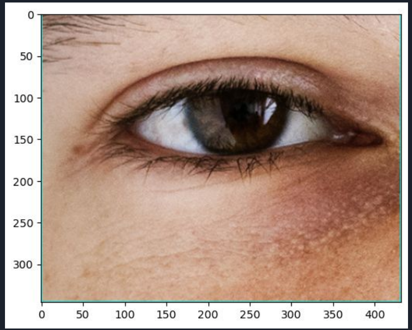
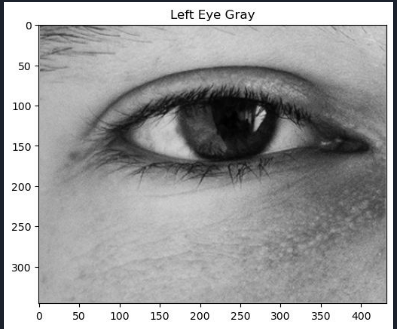


Image Processing Techniques

Convert to Binary Image to detect Pupil





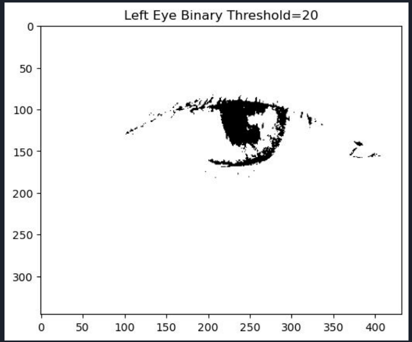
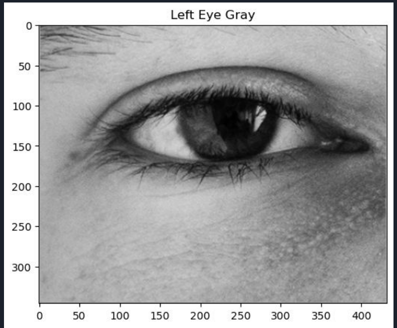
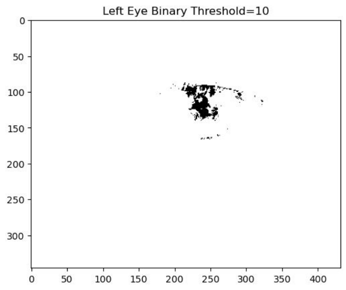
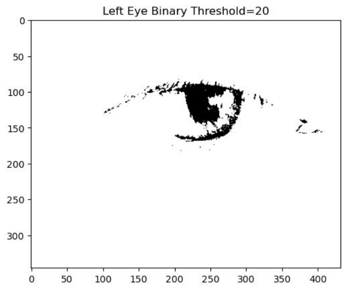


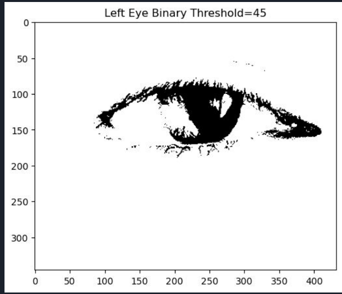
Image Processing Techniques

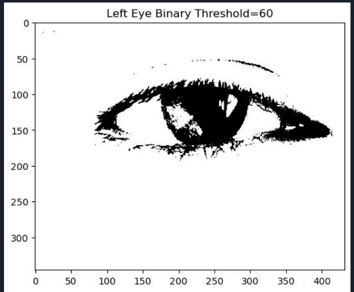
Different threshold for Binary Images

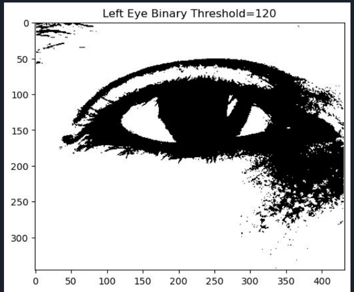












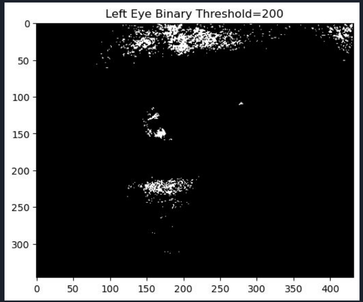
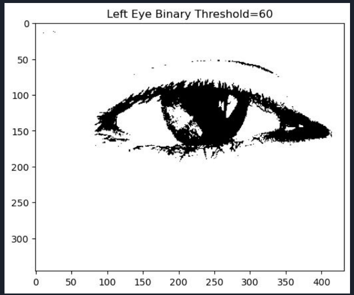
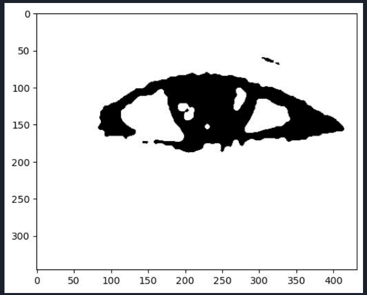


Image Processing Techniques

Pupil Detection



Binary Image



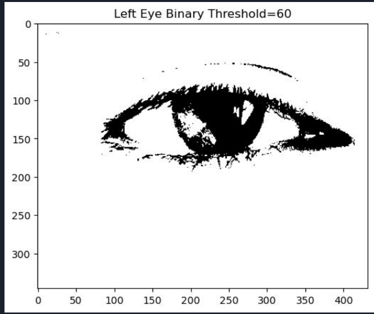
Removing Noise + Smoothening

1. Simple Blob Detector

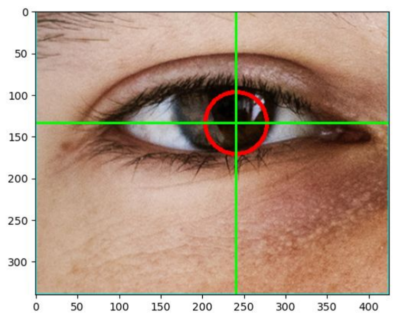
- Blob is a group of connected pixel.
- Our intuition was that the algorithm
- will easily identify the pupil. However, algorithm rarely detected any matches.
- We tried to enhance the image by applying erosions and dilations to reduce the noise in the eye image and we make the image smoother using blur.
- Unfortunately, we did not find significant key-points for many testing instances.

Image Processing Techniques

Pupil Detection



Binary Image

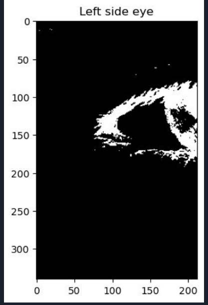


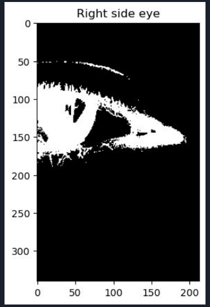
Pupil Detected

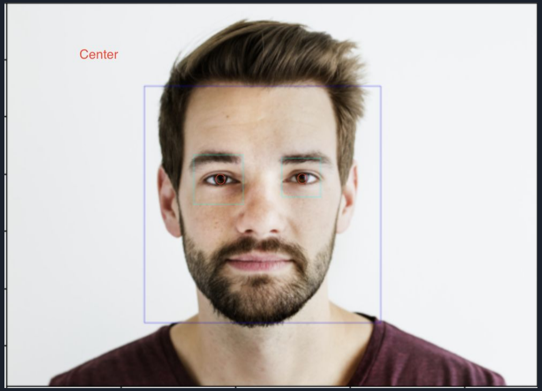
2. Contour Detection

- Contours are the line joining all the points along the boundary of an image that have the same intensity.
- Thus, contours can be used to identify the boundary of the pupil.
- By selecting the largest contour, we can then be confident that we have identified the pupil.
- This method detected the pupil efficiently given that the eye is correctly detected.

Image Processing Techniques







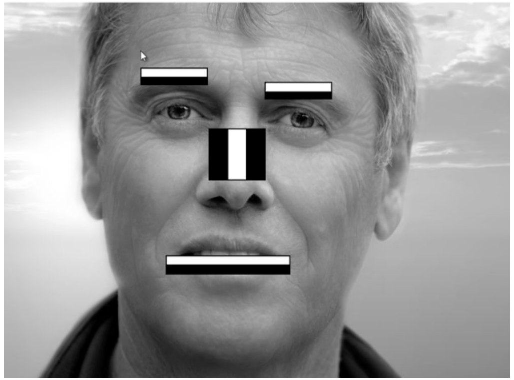
Gaze Estimation

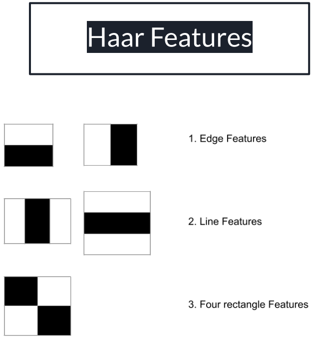
- Using eye and pupil information, we compute a ratio based on the pixels values, to determine if the gaze is left or right or center oriented.

Baseline Model

Haar Cascade Classifier

- A Haar cascade classifier is a type of object detection algorithm which is based on one of the popular methods of object detection proposed by Viola and Jones in 2001.
- The algorithm uses a series of simple Haar-like features to detect face features in images, which are then combined into a cascade of increasingly complex classifiers.
- Each haar feature is mapped to facial features like nose, mouth, eyebrows, etc. For each feature, it finds the best threshold which will classify the faces as positive and negative.
- Haar Cascade Classifier is trained on dataset that contains positive and negative samples - Face and Non-Face values.
- Hyper parameters : minSize, scale, and number of neighbors.

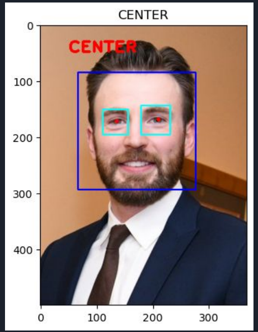


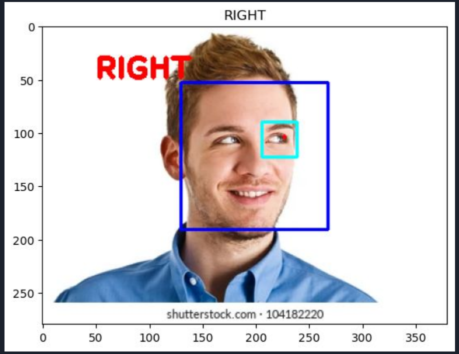


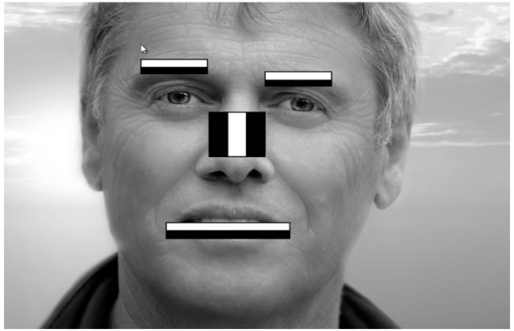
- Advantages
  - Haar Cascade Classifier is similar to CNN models where features are learned during training, while a Haar-Feature is manually determined.
  - Thus, we only train the model to learn the weights of these features. This allows us to train the classifier well with small set of training images.
  - In addition, it also has a higher execution speed, as it involves less computations.
  - With around 200 features used, this model has an accuracy of 95%.

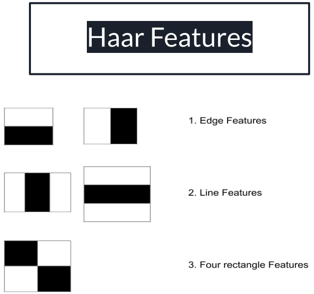
Baseline Model

Haar Cascade Classifier

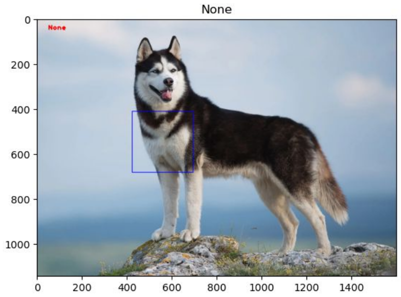


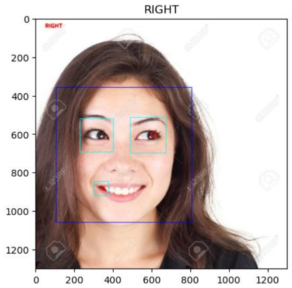






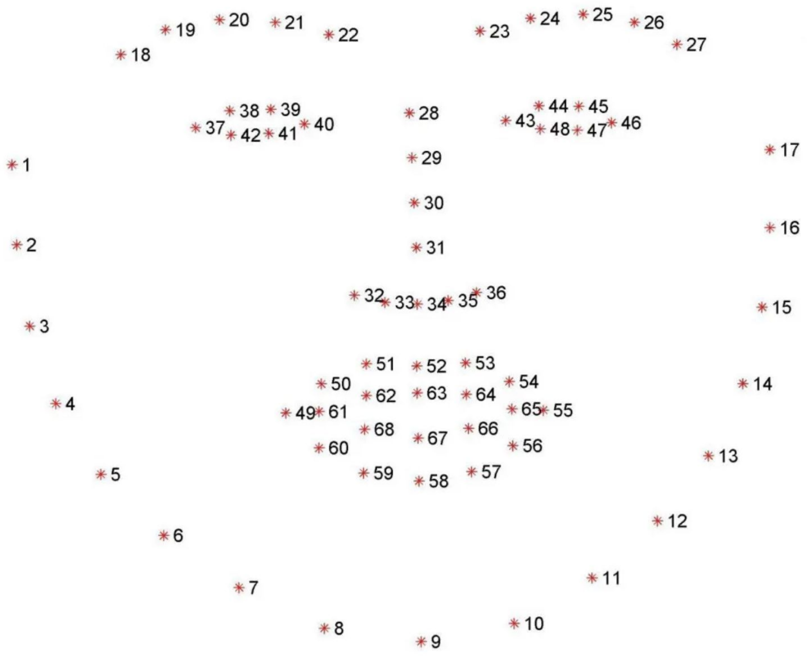
- Drawbacks
    - It is only limited to front facing orientation of the face, it performs poorly when faces are not oriented directly towards the camera.
    - Relatively more false positives than DL models.





DLib Model

- Uses a training set of labeled facial landmarks on an image and Priors, the likelihood that two input pixels are far apart.
- The specific (x, y) coordinates of the regions surrounding each facial structure are specified in the manual labeling of these images.
- An ensemble of regression trees is trained using this training data to estimate the positions of facial landmarks solely from the pixel intensities without any feature extraction.
- The outcome is a facial landmark detector with high-quality predictions that can be used to identify facial landmarks in real-time
- dlib.get_frontal_face_detector() function, a pre-trained method in the dlib library, to load the face detector.



DLib Models

Depending on how many points it uses to map the face, DLib has a number of variations of the aforementioned model.

Models implemented - the Dlib 68-point Face Detector and the

Dlib 5-point Face Detector

The 68-point face detector is trained on the iBUG 300-W dataset and its mapping can be seen in

The 5-point facial landmark detector condenses the facial information to 2 points for the left eye, 2 points for the right eye, and 1 point for the nose, whereas the 68-point detector localizes regions along the eyes, eyebrows, nose, mouth, and jawline.

68-point version is 8–10% slower than the 5-point detector, but the model size is 9.2MB as opposed to 99.7MB (over 10x smaller).

In practice, the 5-point facial landmark detector performs equally well, despite the fact that the 68-point facial landmark detection may provide a slightly better approximation to the eye centers.

Caffe Model

Advantages:

- Efficient processing: When it comes to image processing, deep learning models are considered to be the best.
- Accurate results: Neural networks are employed whenever the deep learning model is used in image processing applications, and they produce better results than the HAAR cascade classifier.
- Detecting multiple faces: When using other techniques, we occasionally are unable to see multiple faces, but when using the ResNet-10 Architecture, the network model effectively allows us to see the various faces (SSD).

Caffe Model

- Load the face detector enabled by deep learning: cv2.dnn.readNetFromCaffe(), cv2.dnn.readNetFromTensorflow().
- cv2.dnn.blobFromImage() - to retrieve the frame or image.
- opencv_dnn_model.set input() - For pre-processing of the image, which makes it ready to function as the input for the neural network.
- opencv_dnn_model.forward() - To provide the array containing the coordinates of the normalized bounding boxes and the detection confidence value.
- Used a confidence threshold of 0.5. Results improve with a higher detection confidence value.

Experiments

Gaze Estimation using:

- Haar Cascade Face & Eye Detector
- DLib 64-point Face Detector
- DLib 8-point Face Detector
- Caffe Model for Face Detection

Comparisons

- Used a webcam to capture series of the images, which is then processed using a Face Detector and a common image processing pipeline.
- Used Inference time, because the tracker works in real-time, which is expressed in Frames Per Second (FPS), as an evaluation metric to compare the experiments.

- HAAR Cascade - 15 FPS
- DLib 8-point Detector - 2FPS
- DLib 64-point Face Detector - 10 FPS
- Caffe Model - 7 FPS

Conclusion

- Haar cascades - fast, less precise (as it detects only front facing faces and also have large false pos)
- DLib - more precise than Haar cascades but slower.
- Caffe - more precise than Haar cascades and DLib. GPU inference can speed up a model that may otherwise be very slow depending on its depth and complexity.

Future Work

- Using Head Positioning information to refine eye detection and gaze estimation and
- Training a model to detect pupil directly.
- To use gaze tracking to address real-world issues, like accessibility, example virtual keyboard interaction based on eye movements.

Work Distribution

Sreelaya

Code — 50%

Worked on Deep Learning Models - DLib8, DLib64, Caffe and evaluation.

Report — 50%

Worked on writing and presenting concise information + 3 literature review.

Presentation — 50%

Created slides for corresponding topics.

Tejal

Code — 50%

Worked on Haar Cascade Classifier & detailed Application of Image Processing techniques & evaluation.

Report — 50%

Worked on writing and presenting concise information + 3 literature review.

Presentation — 50%

Created slides for corresponding topics.

Thank you!

Feel free to ask any questions. :)