Real-Time Eye-Based System Navigation Control

Group members:
Wasim Ishaq Khan - S20160010107
Kaustubh Pandey - S20160010041
Aakash Shukla - S20160010001

Guide:
Dr. Prerana Mukherjee
Dr. Shiv Ram Dubey

Outline

- Agenda
- Introduction
- Motivation
- Work done till first evaluation
- Work done so far
- Timeline
- Results
- Conclusion
- Literature review

Agenda

- Our project aims at providing a platform for the differently-abled persons to make use of the technology.
- We aim at providing a system that will make use of the eye-movement and eye-blinks of the user for interaction with the computer system.
- In this project, we intend to develop a real-time system navigation control which uses the eye movement and eye-blinks of the user to give system commands.

Introduction

- All the available devices require manual control and cannot be used by persons impaired in movement capacity.
- Therefore, there is a need for developing alternative methods of communication between human and computer that would be suitable for the persons with motor impairments.
- The user will be provided with a virtual cursor which he/she can use to give commands to the system.
- Based on the eye movements and the eye blinks, the user will be able to perform different tasks.

Motivation

- Human-Computer Interface (HCI) can be described as the point of communication between the human user and a computer.
- Normal users can interact with a computer using the available input devices.
- But for differently-abled persons, using the available input devices is not convenient.
- Developing alternative means of interacting for such persons who cannot speak or use their limbs (cases of quadriplegia etc.) is very important.
- Our project is one such effort in the area of addressing the issues faced by such persons.

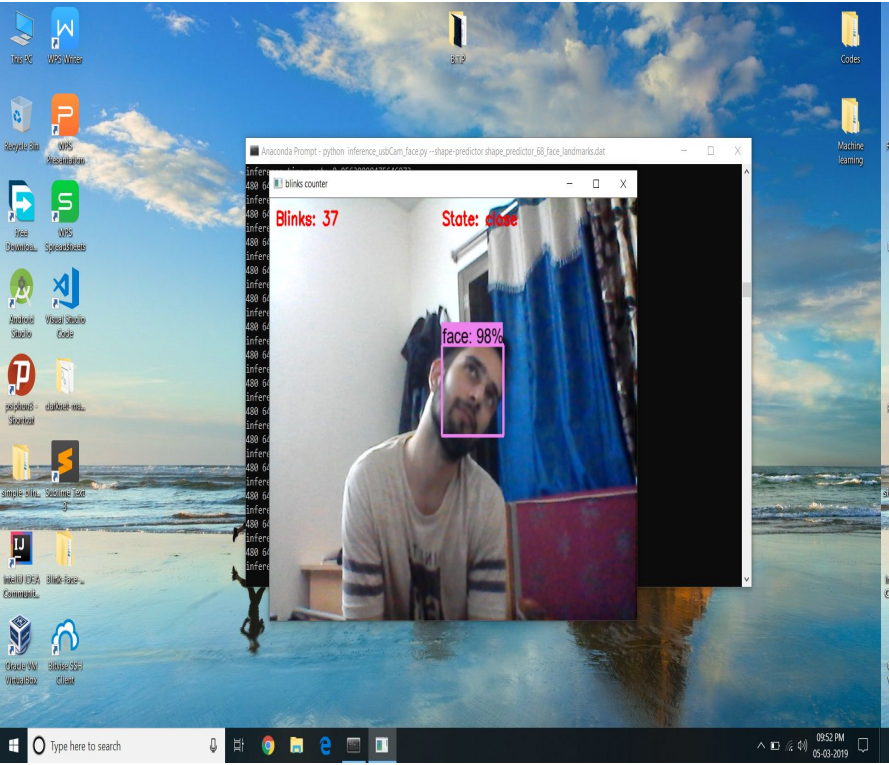
Work Done Till Last Evaluation

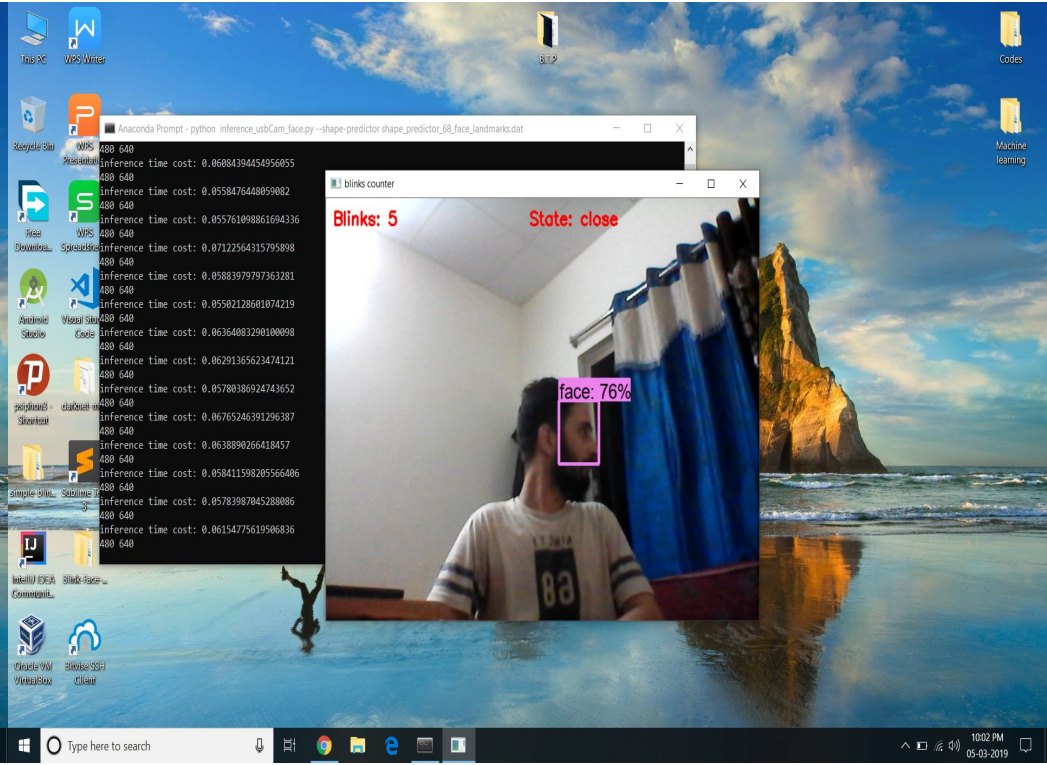
- Face detection (using SSD).
- Eye region extraction (Using dlib).
- Eye blink detection (Using CNN).

Dataset

- Dataset consists of 2200 images corresponding to five persons.
- Each image is rescaled to 64*64.
- From each image 5 parts are extracted viz. left-eye, right-eye, face, face mask, ground truth coordinates.
- The dataset is then converted to .npz format for model training.

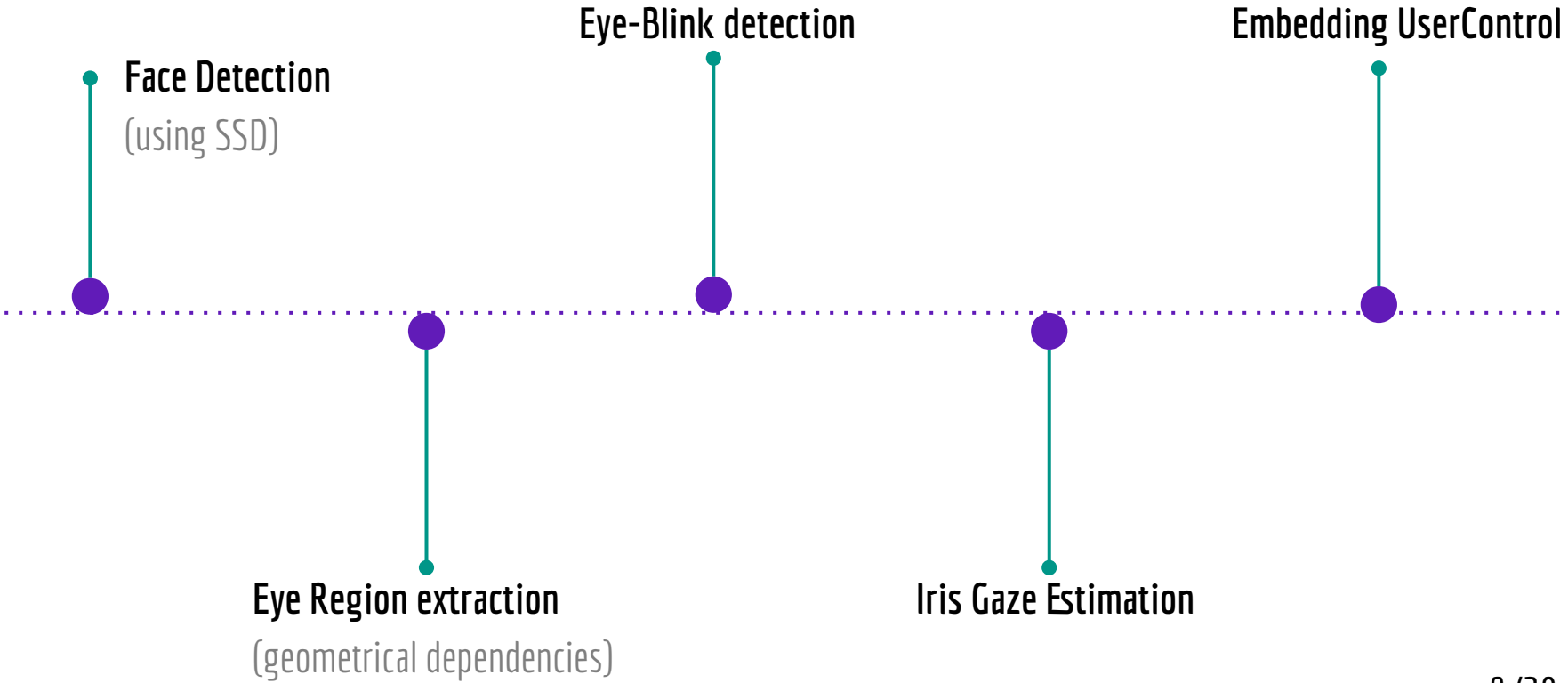
Result Using SSD





Under low-light conditions.

Timeline


Face Detection
(using SSD)
Eye-Blink detection
Embedding UserControl
Eye Region extraction
(geometrical dependencies)
Iris Gaze Estimation

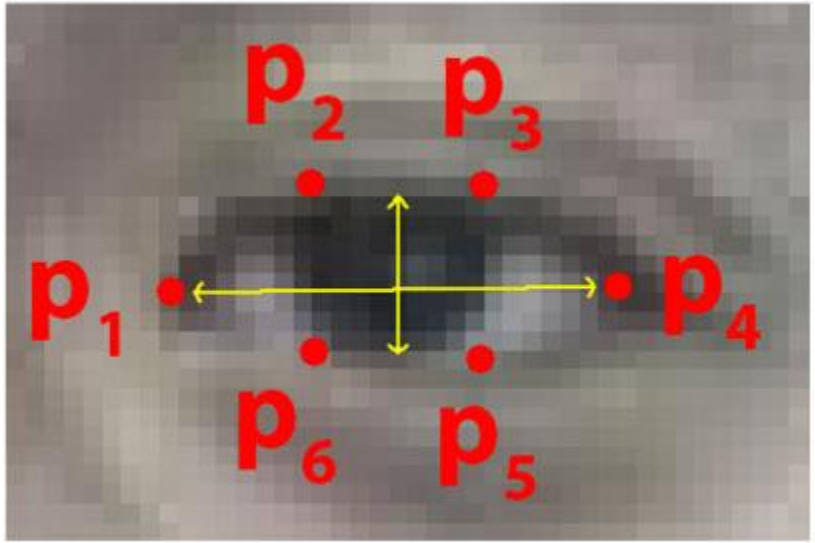
Face Detection Using SSD

- Single Shot Multibox Detector (SSD) is designed for object detection in real-time.
- SSD speeds up the process by eliminating the need of the region proposal network.
- The SSD object detection composes of 2 parts: Extract features maps and apply convolutional filters to detect objects.
- SSD uses VGG16 to extract feature maps. Then it detects objects using the Conv4_3 layer.
- Each prediction composes of a boundary box and 21 scores for each class and we pick the highest score as the class for the bounded object.

Eye Region Extraction

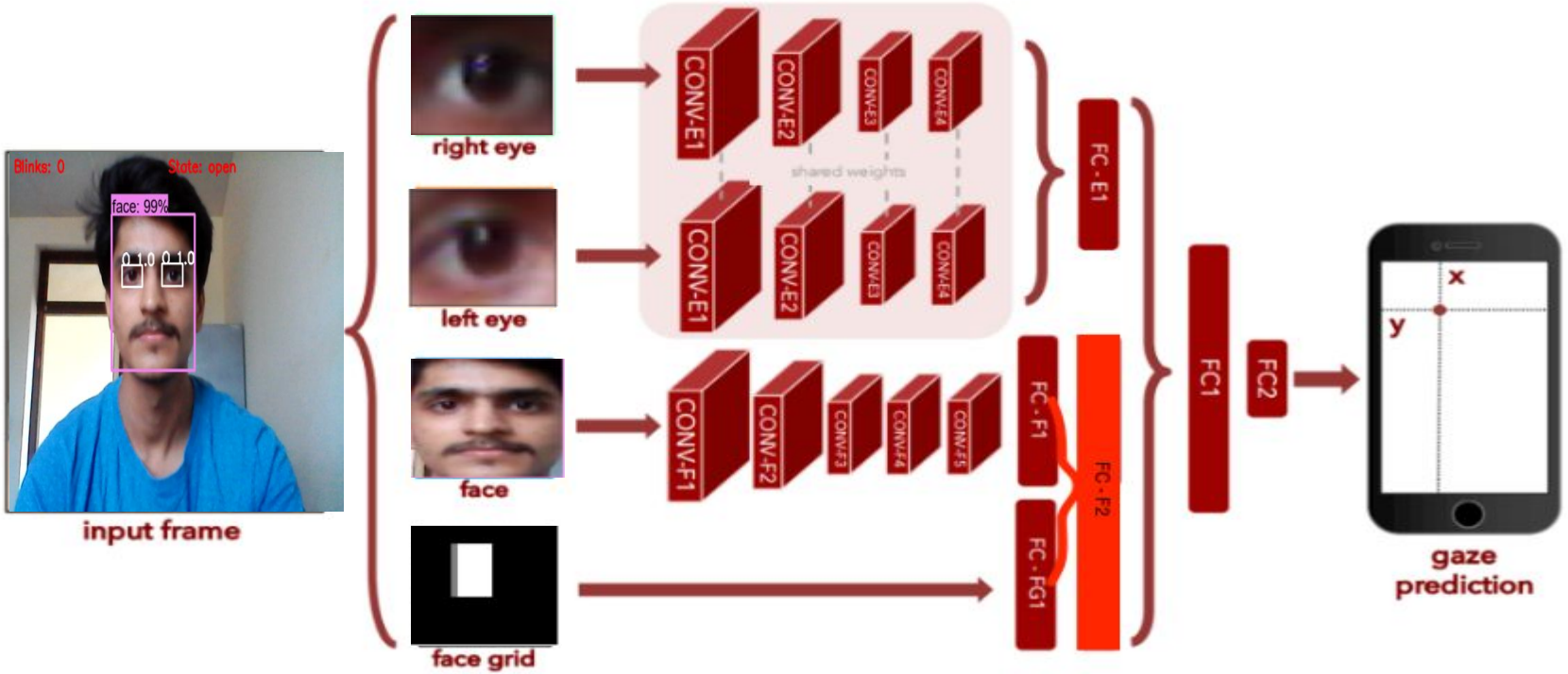
- The bounding box obtained from SSD will be used by dlib for eye-region extraction.
- We are going to use dlib and OpenCV to detect facial landmarks in an image.
- The facial landmark detection implemented inside dlib produces 68 (x, y)-coordinates that map to specific facial structures.
- Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you are looking at the person), and then working clockwise around the remainder of the region.

Eye Region Extraction

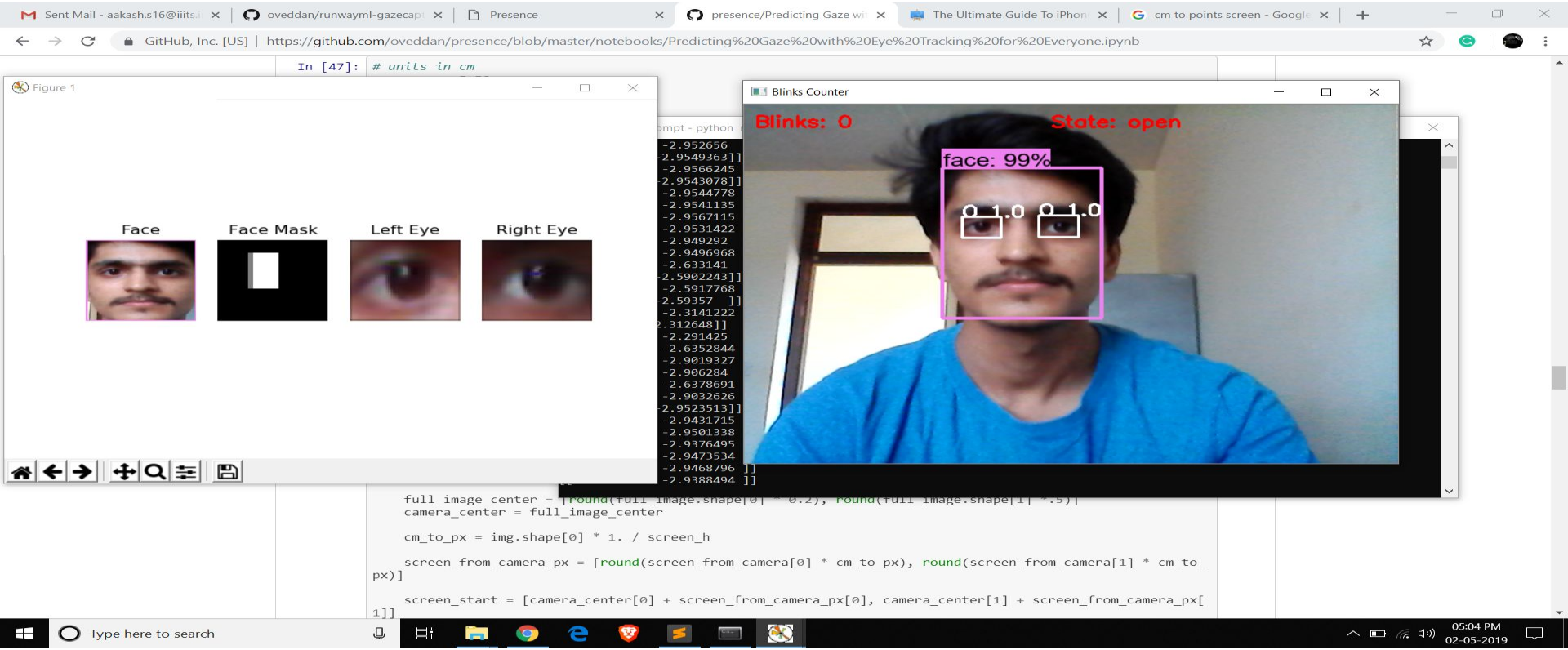


The 6 facial landmarks associated with the eye.

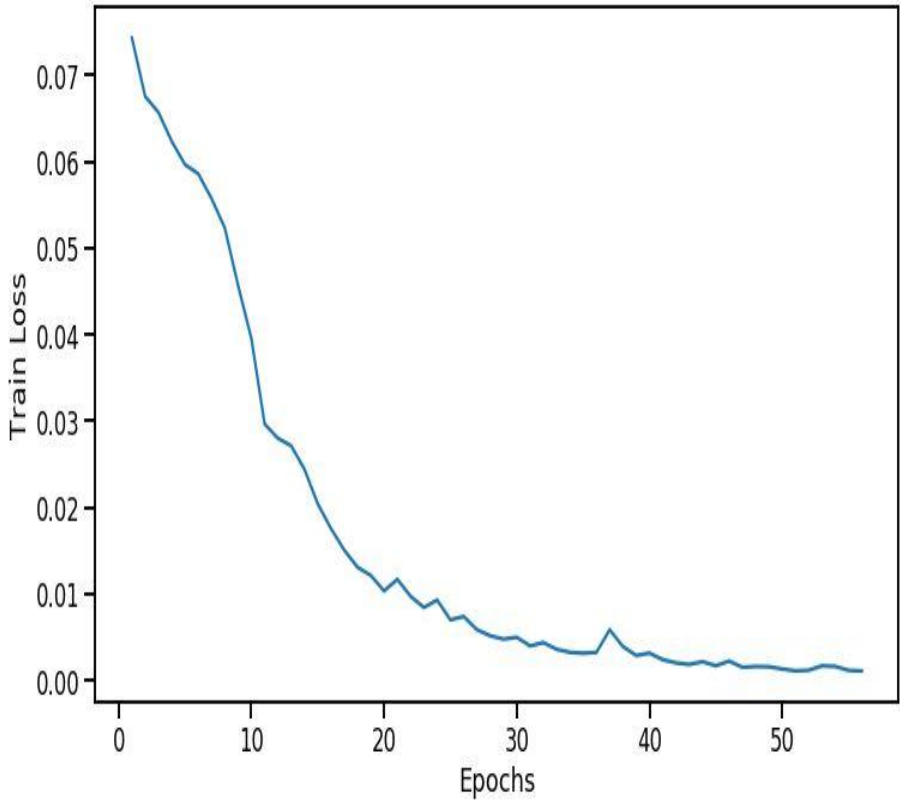
Model Architecture

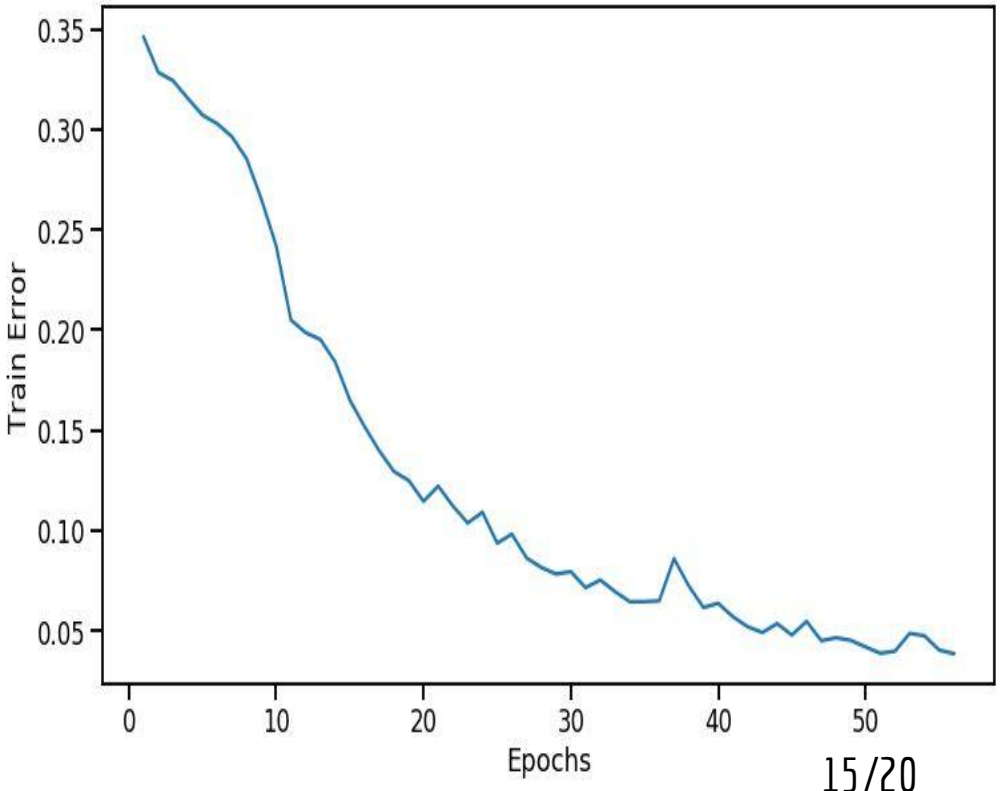


Results

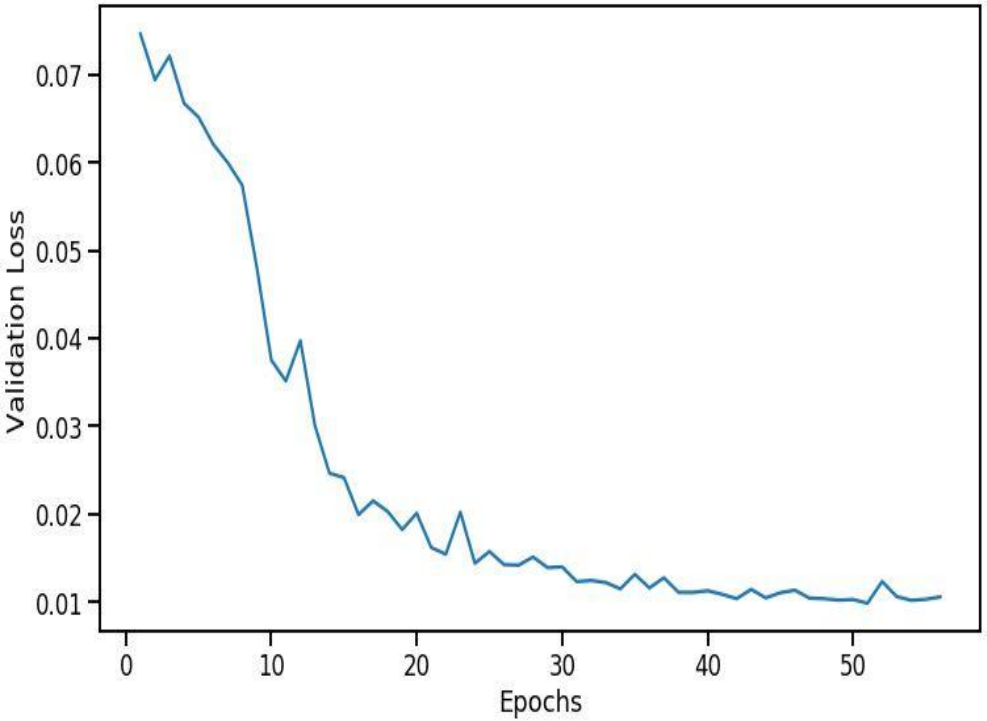

Sent Mail - aakash.s16@iiits.
oveddan/runwayml-gazecap
Presence
presence/Predicting Gaze wi
The Ultimate Guide To iPhon
cm to points screen - Google
GitHub, Inc. [US] | https://github.com/oveddan/presence/blob/master/notebooks/Predicting%20Gaze%20with%20Eye%20Tracking%20for%20Everyone.ipynb
In [47]: # units in cm
Figure 1
Face
Face Mask
Left Eye
Right Eye
Blinks Counter
Blinks: 0
State: open
face: 99%
1.0
1.0
full_image_center = [round(full_image.shape[0] * 0.2), round(full_image.shape[1] * .5)]
camera_center = full_image_center
cm_to_px = img.shape[0] * 1. / screen_h
screen_from_camera_px = [round(screen_from_camera[0] * cm_to_px), round(screen_from_camera[1] * cm_to_px)]
screen_start = [camera_center[0] + screen_from_camera_px[0], camera_center[1] + screen_from_camera_px[1]]
Type here to search
05:04 PM
02-05-2019

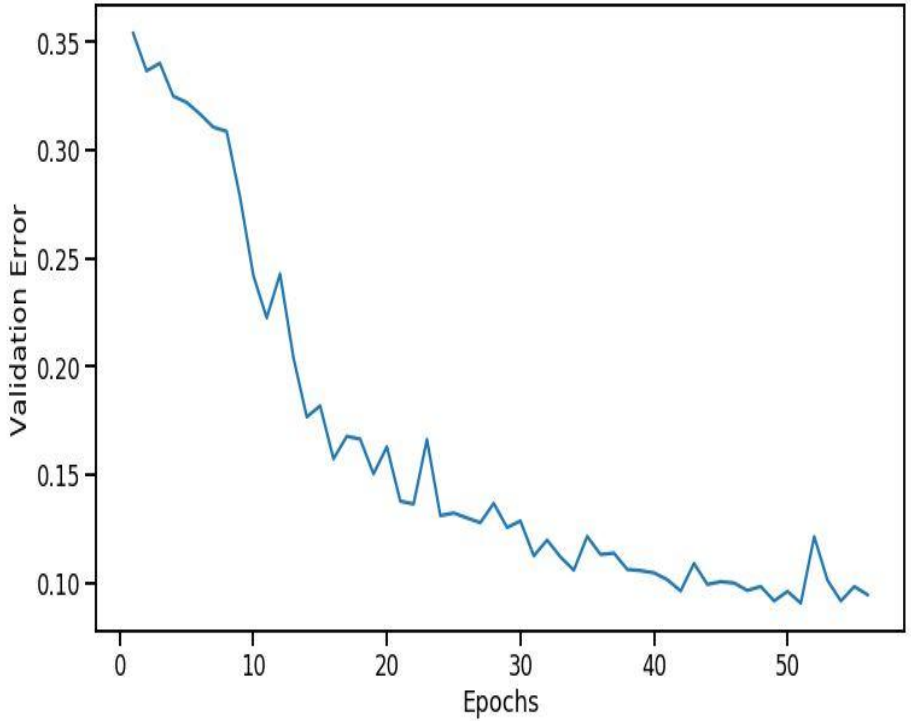
Results


Train Loss
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
50
Epochs

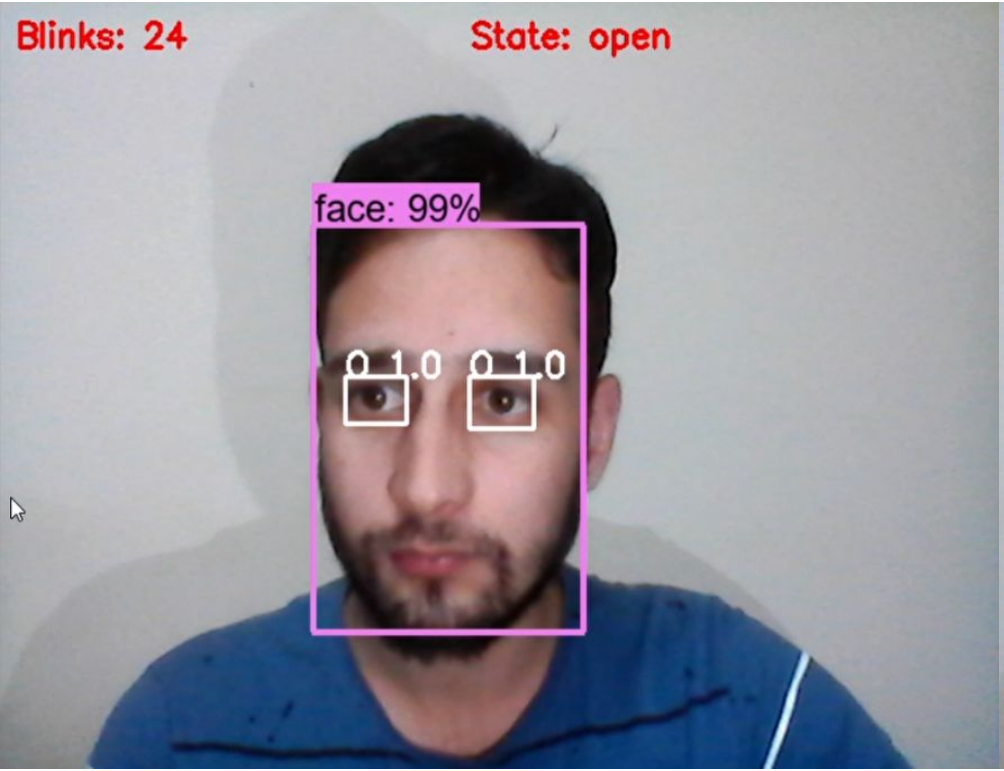

Train Error
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
10
20
30
40
50
Epochs

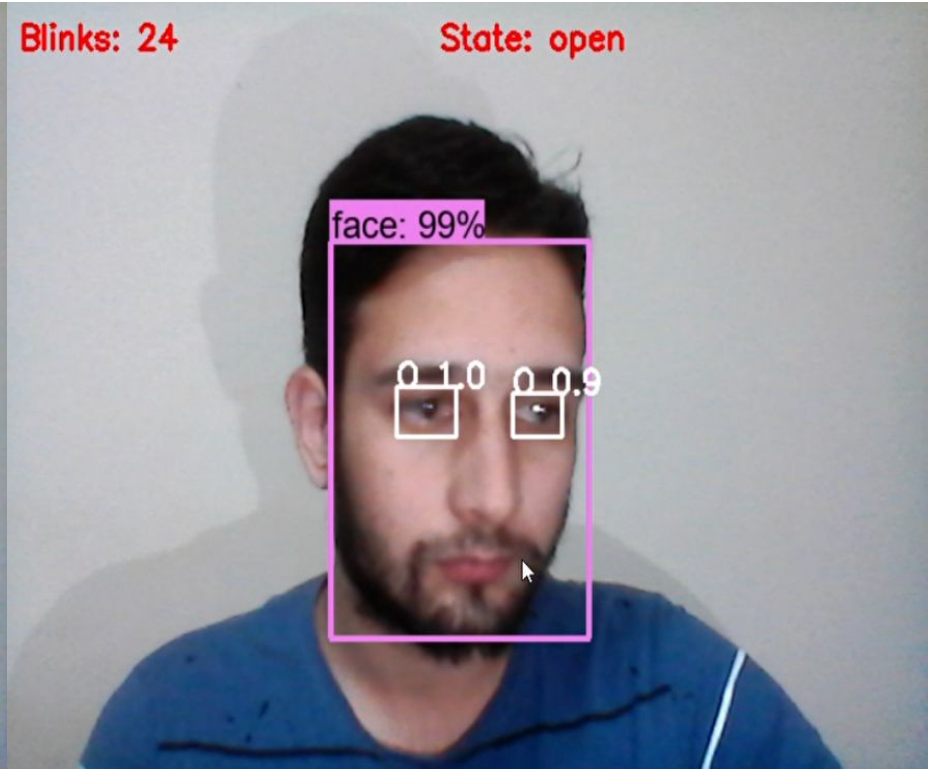
Results


Validation Loss
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
10
20
30
40
50
Epochs


Validation Error
0.35
0.30
0.25
0.20
0.15
0.10
0
10
20
30
40
50
Epochs

Demo


Blinks: 24
State: open
face: 99%
1.0
1.0


Blinks: 24
State: open
face: 99%
1.0
0.9

Key Challenges

- Integrating eye-tracking and eye-blinking.
- Blink detection thresholding is tricky.
- Navigation control with rapid eye movements is difficult and challenging without much fatigue.

Literature Review

- P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," pp. 511–518, 2001
- J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2015
- W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in ECCV, 2016
- D. E. King, "Dlib-ml: A machine learning toolkit," J. Mach. Learn. Res. , vol. 10, pp. 1755–1758, Dec. 2009
- K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. M. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , pp. 2176–2184, 2016

Thanks