A Novel and Effective Queue Management Tool using Computer Vision.

YASH-19BCB0046

UTPAL PRAJAPATI-19BCE0759

in partial fulfilment for the award of the degree of
B. Tech

Under the guidance of

Prof. Mohan Kumar P

School of Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Abstract

India is the second-largest country when it comes to population. Hence, we often see overcrowding in public places. Managing such a volume of the population is never easy. The problem of a massive population is continuously increasing, and with the outbreak of the pandemic COVID - 19, we all have understood how difficult it is to manage the crowd and how important it is to maintain social distancing in it. To address this issue, we plan to build a queue manager using object detection and tracking. We primarily focus on the Case Study of a Hospital. In rural hospitals, generally, the number of people waiting to see a doctor is huge. Hence a lot of manpower is wasted in maintaining decorum and queueing, for the Checkup. Our aim is to automate this process. The queue manager will be identifying a person (in the frame of vision of the camera) and giving him/her a number tag. People will follow the order of their tag number. Once the person goes into the doctor's room (out of the field of vision of the camera), the number of the subsequent people gets decremented. We plan to use YOLOv3 for Object detection, which can be further curated using an additional CNN layer, for better accuracy. Our System will process and work with the live feed, that is brought in through the camera. We believe that this invention is not only limited to Hospitals but also can be widely and effectively used in other places such as – Popular food joints, Car repair services, Government Offices, Courts, Air Traffic Control, etc. We will also be exploring the wide applications that this project brings with it.

1. Introduction

Nowadays the digitalization of the world created many benefits for society. And due to this digitalization, we are meeting many challenges to make our lives more luxurious. As a result, we are in a rapid development phase in all kinds of fields. One of the developing fields is image processing which we are using in our daily day lives like when we take photos in our mobile phone we increase the brightness and recently there are also apps which help us to remove or blur the parts we don't like in the pictures, there is also the possibility of changing the background of the picture like these there are many areas in which the image processing is used by us without knowing that we are doing it in our daily lives. This project is about detecting the number of faces in the images to know the number of people present in a video for convenience. We have also extended this to face recognition in a live feed. We created a python code for detecting the number of faces and storing these faces to see the faces of the people present in the image. Here in this project, we are using python language to code the program to detect the faces in the video and webcam. And for the python to detect the face we have to download the OpenCV library and the haar cascade classifier for easy calculations. OpenCV is a library for the programming language used for 2D and 3D feature toolkits, ego-motion estimation, facial recognition system, gesture recognition, human-computer interaction (HCI), mobile robotics, motion understanding, object identification, segmentation and recognition, stereopsis stereo vision: depth perception from 2 cameras, structure from

motion (SFM), motion tracking, augmented reality. We can download OpenCV from the home website available publicly to everyone. The OpenCV library application is written in C++ language which is to say its primary interface is C++ but it still retains a less comprehensive though extensive older C interface. It works on a wide variety of hardware platforms, including the Intel Atom platform, Intel Core processor family, and Intel Xeon processor family. OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the following mobile operating systems: Android, iOS, Maemo, and BlackBerry. We can download OpenCV of the official releases from SourceForge or take the latest sources from GitHub. Apart from OpenCv various other libraries have been used like pickle, numpy etc. Facial recognition generally involves two stages: Face detection: A photo or webcam video is searched to find a face, then the image is processed to crop and extract the person's face for easier recognition. Furthermore after face detection our code assigns an unique ID number to people based on their time of arrival in the queue. As soon as the person in the front steps in the IDs are adjusted by decrementing one from all the IDs behind. This indeed helps in the effective queue management and reduces the need of a human for managing the queue. This can have a variety of applications like in temples where we have long line of bhakts waiting or a hospital or also in managing lines in malls and other public places during covid. This method efficiently manages the crowd and also reduces a lot of human efforts and need for manpower.

2. Hardware / Software Requirements

Hardware Requirements:

The hardware requirements for the project are:

1) System WebCam - Processing at 480p 4:3 at 30 FPS
2) 4 GB RAM
3) No GPU required
4) Around 10 GB of free ROM Memory

Software Requirements:

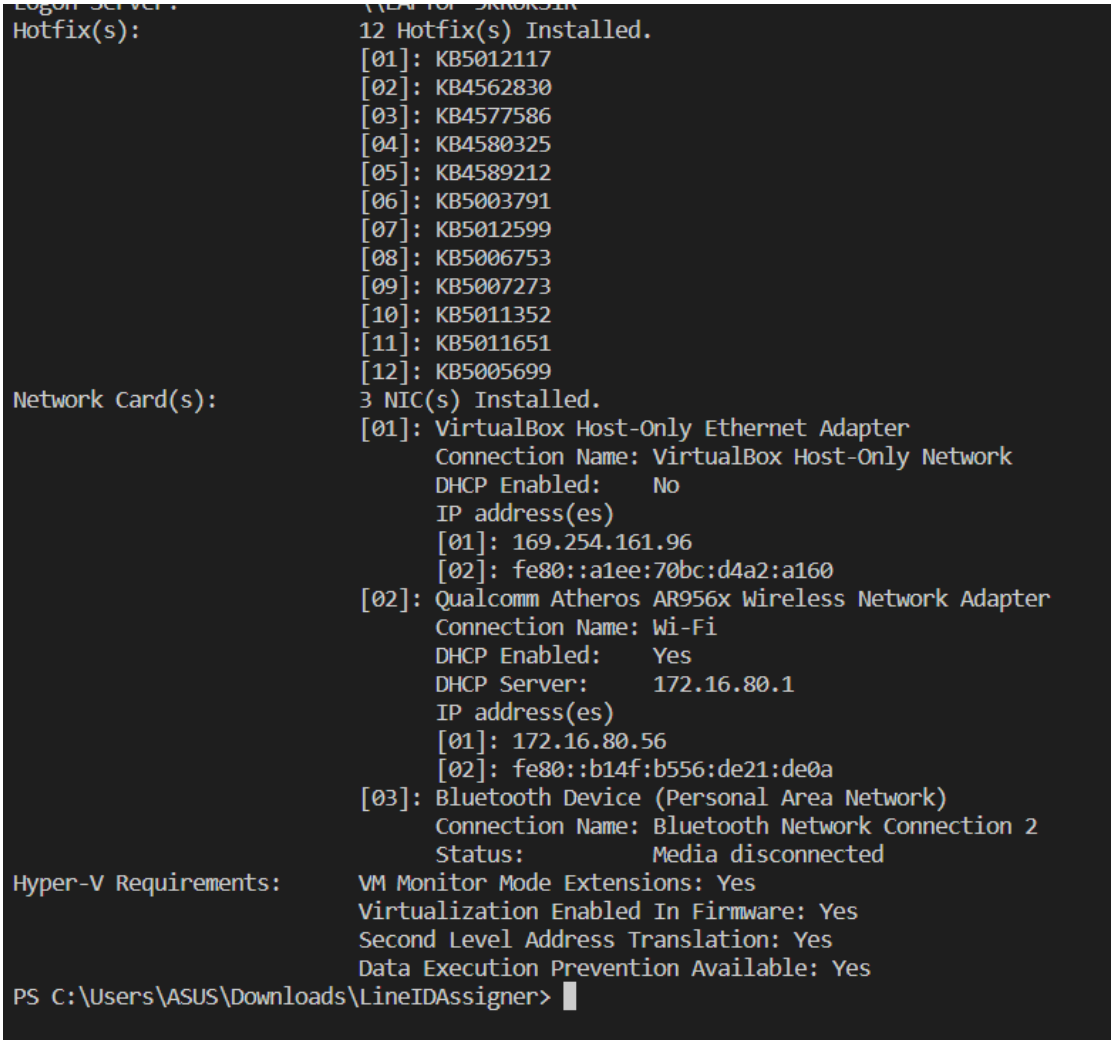
The software requirements for this project are:

1) Python3
2) Computer Vision
3) Face Detection Library
4) Numpy
5) Pickle

System Requirements:

Some more system requirements include:

```
Host Name:                 LAPTOP-5KRUKS1R
OS Name:                   Microsoft Windows 10 Home Single Language
OS Version:                10.0.19044 N/A Build 19044
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          N/A
Registered Organization:   N/A
Product ID:                00327-35101-26458-AAOEM
Original Install Date:     26-03-2021, 01:46:32
System Boot Time:          20-04-2022, 13:04:35
System Manufacturer:       ASUSTeK COMPUTER INC.
System Model:              VivoBook 15_ASUS Laptop X507UAR
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 142 Stepping 9 GenuineIntel ~2304 Mhz
BIOS Version:              American Megatrends Inc. X507UAR.303, 21-05-2019
Windows Directory:         C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume7
System Locale:             en-us;English (United States)
Input Locale:              00004009
Time Zone:                 (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi
Total Physical Memory:     8,074 MB
Available Physical Memory: 1,371 MB
Virtual Memory: Max Size:  12,695 MB
Virtual Memory: Available: 3,412 MB
Virtual Memory: In Use:    9,283 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\LAPTOP-5KRUKS1R
```



3. Existing Systems

3.1 Drawbacks in the current approach.

Some of the existing approaches for this queue management that we have gone through are:

[1] This paper works on live feed that transfers data into an SSD Object Detection Algorithm. The use of SSD Object detection is favoured over other Algorithms like YOLOv3 as they are computationally expensive, and hence are not responsive to embedded systems. This Paper has taught us, how to work efficiently with live feed transfer, and how to compromise between accuracy and computational capacity.

[2] The proposed Petrol Pump Queue Management System uses a supervised classification algorithm of machine learning. This is to identify the type of vehicle and be able to estimate

the tank size to calculate automatically the time required for refuelling. The main idea is to display the timer count on the screens so that the drivers of the vehicles that are in the queue will know the time remaining for each vehicle and accordingly they will be able to decide the choice of a queue when there are multiple queues at the petrol pumps. Alongside, this system will help the workers in the petrol pump station to organize their work as they will be able to track which car is finishing first. In this paper, we try and emulate the System that has been created for Petroleum, in our Hospital Queue Management System.

[3] In this paper, they have designed an adaptive queue management-based object detecting (QMOD) scheme for achieving low latency AI services with limited computation resources. The implementation of the QMOD scheme is done by utilizing Python, PyTorch, OpenCV and YOLOv3. They have introduced a target delay requirement of AI services as a control parameter and mapped such requirements into the optimal queue size. Then, the maximum queue size is dynamically calculated by referring to the weighted average of processing time and the target delay requirement. We learnt from this paper, the amount of accuracy that the algorithms like You Look Only Once bring with them which will help us in our Algorithm Choice.

4. Proposed/Developed Model

4.1 Design

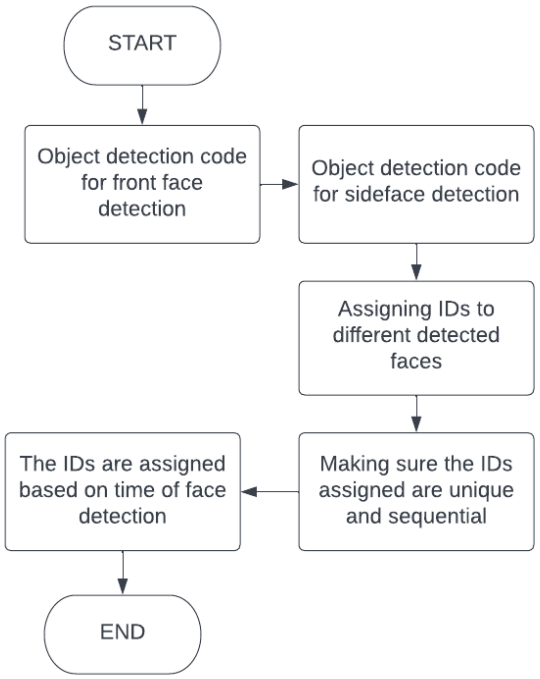


Fig.1 Design

4.2 Module Wise Description

We have basically written the whole code using python language. Various python libraries have been used for the same:

1) OpenCV:- Used for the live feed capturing through WebCam .
2) Face Detection Library :- Used for the face detection of a person from front as well as sideways.
3) Pickle:- This is used for converting the python objects into 0s and 1s. I.e we are serializing and deserializing the python objects.
4) Numpy:- This has been used in order to assign the IDs properly and also flushing out and decreasing the ID value once the person has moved out of the frame.

4.3 Implementation

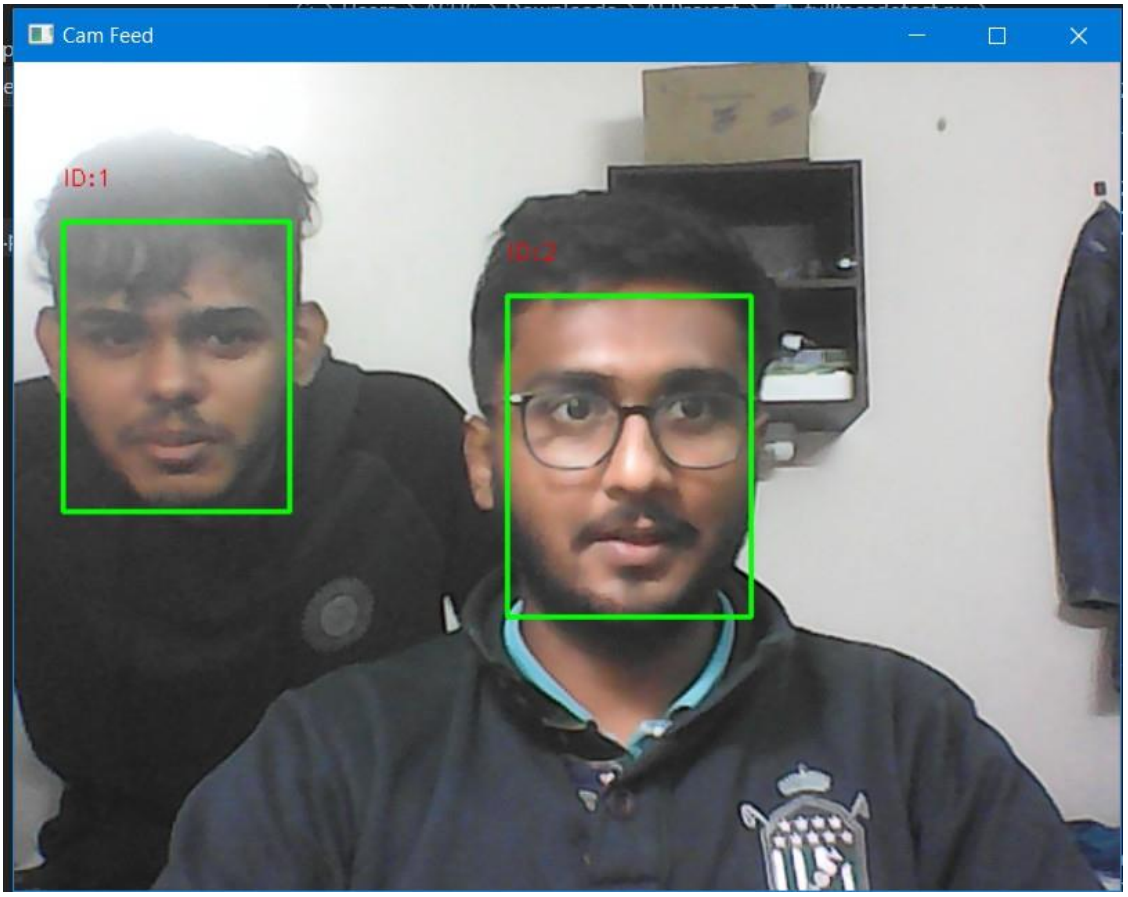


Fig. 2 Person 1 and Person 2 in the frame, the face is successfully detected.

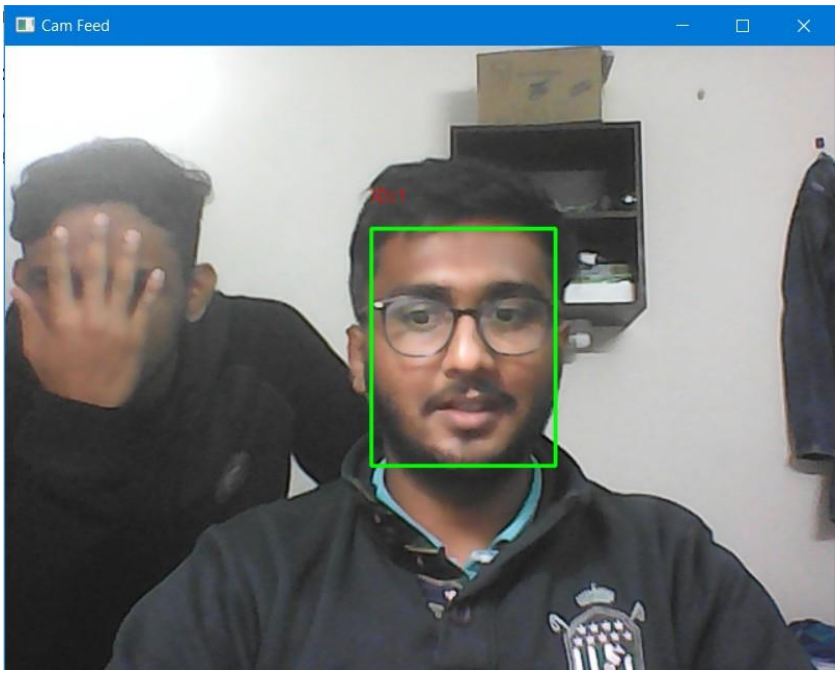


Fig. 3 If Person 1 moves out of the frame, then the model will assume that the Person has served his turn, and would move to the next person in the frame.

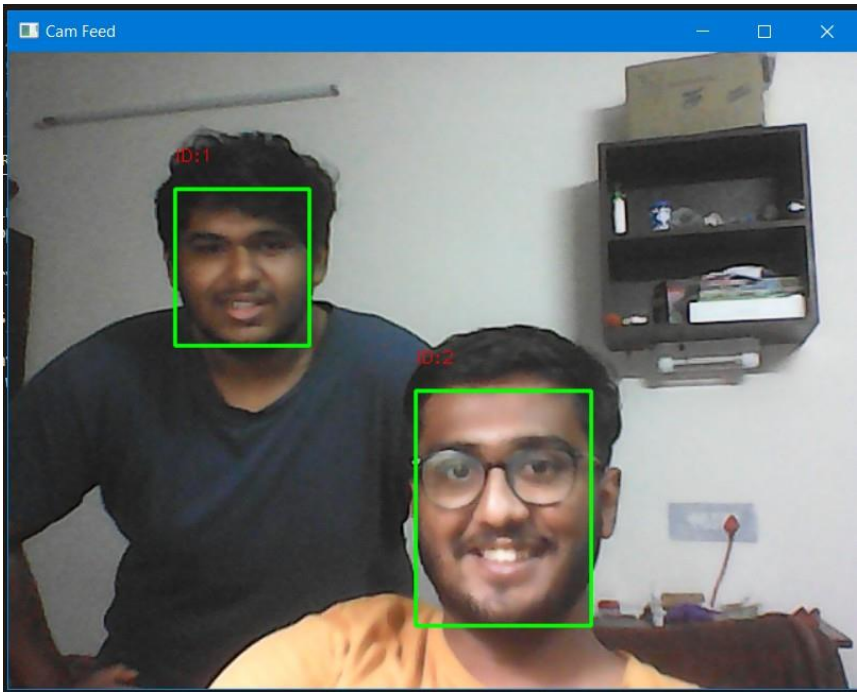


Fig. 4 The model would assess a new face too with ease, and provide an ID.

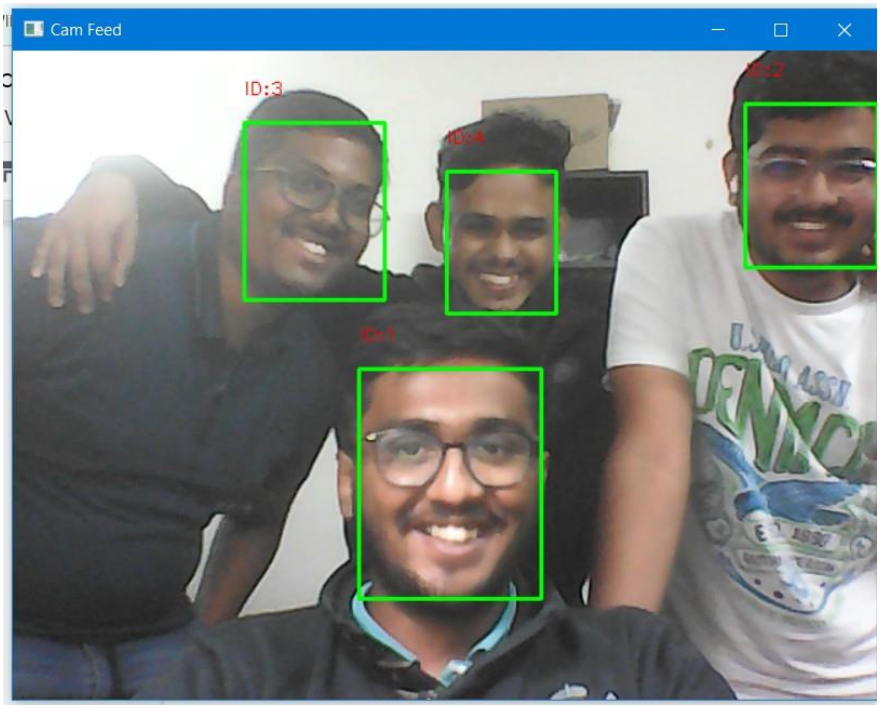


Fig. 5 The model works effortlessly with a group of people.

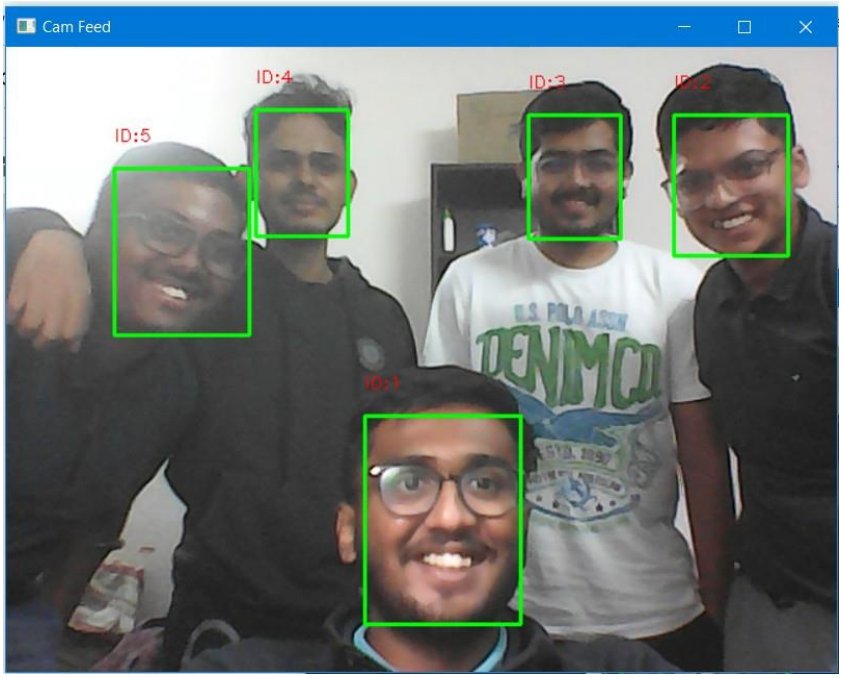


Fig. 6 The model successfully and accurately detects faces, and tracks the object.

For Figures 7 and 8, we ran the model on a standing moving queue, which was extracted from Shutterstock. The model works perfectly fine with a moving queue too.

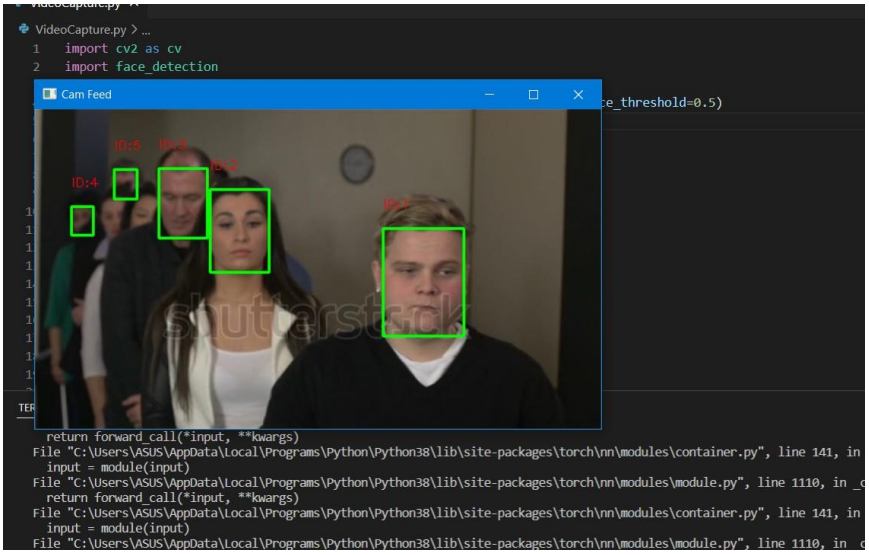


Fig. 7

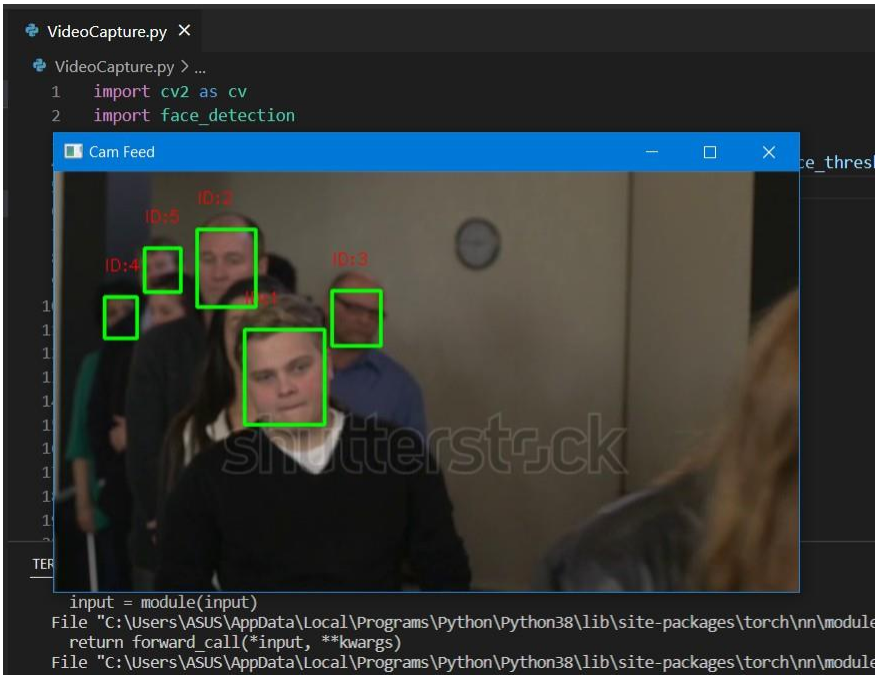


Fig. 8

5. Results and Discussion

We have successfully implemented a queue management system by doing the facial recognition for both videos as well as live feeds. We do queue management by assigning different IDs to people based on the time when the WebCam detects it. Suppose a person comes in the line of the visit of the camera. The facial recognition system concludes that the system will help the visually challenged for several
purposes. The system will take input from the camera and gets the edges of the images. The accuracy of the face detection and recognition of the system has 92% which is comparatively more perfect than the previous detection techniques. From the experimental results, it is seen that the method achieves better results compared to the Eigen-face methods, which are known to be the most successive algorithms. A new facial image can also be simply added by attaching new feature vectors to the reference gallery while such an operation might be quite time-consuming for systems that need training. along with finding faces, we are assigning them with ids that can be used in many applications like counting the number of people. and maintaining queue mechanism in required places.in our project, as we are dynamically assigning the ids it will help to know who is prior at that situation. which is very helpful. In this way, we can use our projects in hospitals, near temples, etc.

6. Conclusion

This project successfully does face recognition be it in a video or a live frame (WebCam). Also moving object facial detection is possible as long as the face is visible in the camera. It also assigns a unique id based on the time of arrival in the frame of the camera. If the face is hidden then the camera doesn't detect it. This is what is the limitation of the project if moving the persons face gets hidden by the camera then its id will be gone and then might be reassigned when the face is again detectable. This can cause a person to wait for his turn longer than his/her actual waiting time. This limitation can be further corrected and improved upon in order to make this more accurate and precise.

References

[1] Sabnis, Omkar Vivek, and Lokeshkumar R. “A Novel Object Detection System for Improving Safety at Unmanned Railway Crossings.” *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 2019, doi:10.1109/iconstem.2019.8918786.

[2] Kyritsis, Athanasios I., and Michel Deriaz. “A Queue Management Approach for Social Distancing and Contact Tracing.” *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, 2020, doi:10.1109/ai4i49448.2020.00022.

[3] Iftikhar, Zubair, et al. “Computer Vision Based Traffic Monitoring System for Multi-Track Freeways.” *Intelligent Computing Methodologies Lecture Notes in Computer Science*, 2014, pp. 339–349., doi:10.1007/978-3-319-09339-0_35.

[4] Mahrooqi, Sumaiya Salim Al, and Syed Zakir Ali. “Petrol Pump Queue Management System for Sultanate of Oman Using Artificial Intelligence Technique.” *2019 China-Qatar International Workshop on Artificial Intelligence and Applications to Intelligent Manufacturing (AIAIM)*, 2019, doi:10.1109/aiaim.2019.8632790.

[5] Yang, Yousung, et al. “Adaptive Queue Management in Embedded Edge Devices for Object Detection with Low Latency.” *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, doi:10.1109/ictc49870.2020.9289509.

[6] S. Jianjun, L. Ming and M. Jingang, "Research and application of data sharing platform

integrating Ethereum and IPFs Technology," 2020 19th International Symposiumon Distributed Computing and Applications for Business Engineering and Science (DCABES), 2020, pp. 279-282, doi: 10.1109/DCABES50732.2020.00079.

[7] M. Shah, M. Shaikh, V. Mishra, and G. Tuscano, "Decentralized Cloud Storage Using Blockchain," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 384-389, doi: 10.1109/ICOEI48184.2020.9143004.

[8] P. A. Abdalla and A. Varol, "Advantages to Disadvantages of Cloud Computing for Small-Sized Business," 2019 7th International Symposium on Digital Forensics and Security (ISDFS), 2019, pp. 1-6, doi: 10.1109/ISDFS.2019.8757549.

[9] M. Silva, M. Matos, and M. Correia, "P2CSTORE: P2P and Cloud File Storage for Blockchain Applications," 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), 2020, pp. 1-4, doi: 10.1109/NCA51143.2020.9306714.

[10] Taha, Ali & Salama, Dr-Diaa & Abdelminaam, Salama & Khalid, M & Hosny, Khalid. (2017). Enhancement of the Security of Cloud Computing using Hybrid Cryptography Algorithms. International Journal of Advancements in Computing Technology