VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnanasangama, Macche, Santibastwada Road
Belagavi-590018, Karnataka

A
Project Report
on

Enhanced Security System with Breach Detection using Deep Learning

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering
in
Electronics & Communication Engineering
by

1DS17EC014:	**AVINASH REDDY P**
1DS17EC032:	**GANESH REDDY G**
1DS17EC034:	**G RAMYA**
1DS17EC153:	**DHEERAJ P**

Under the guidance
of

Dr. Mahesh Kumar N
Project Guide
Assistant Professor, ECE Dept., DSCE, Bengaluru

Department of Electronics & Communication Engineering
(An Autonomous College affiliated to VTU Belgaum, accredited by NBA & NAAC)
Shavige Malleshwara Hills, Kumaraswamy Layout,
Bengaluru-560078, Karnataka, India

2020-21

Dedicated to my

Beloved

Family, Professors, Relatives and Friends

Certificate

Certified that the project work entitled "**Enhanced Security System with Breach Detection using Deep Learning**" carried out by **Avinash Reddy P** (1DS17EC014), **G Ganesh Reddy** (1DS17EC032), **G Ramya** (1DS17EC034), **Dheeraj P** (1DS17E153) are bonafide students of Dayananda Sagar College of Engineering, Bangalore, Karnataka, India in partial fulfillment for the award of Bachelor of Engineering in Electronics & Communication Engineering of the Visvesvaraya Technological University, Belagavi, Karnataka during the academic year 2020-21. It is certified that all corrections / suggestions indicated for project work have been incorporated in the report deposited to the ECE department, the college central library & to the university. This project report **(17EC8DCPR2)** has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

_____	_____
Dept. Project Coordinator Convener	Project Guide
Dr. K.N. Pushpalatha	Dr. Mahesh Kumar N
_____	_____
Head of the Department	Dr. C.P.S. Prakash
Dr. T.C.Manjunath	Principal

External Project Viva-Voce

Name of the project examiners :

1 : Signature : _____

2 : Signature : _____

Declaration

Certified that the project work entitled, "**Enhanced Security System with Breach Detection using Deep learning**" is a bonafide work that was carried out by ourselves in partial fulfillment for the award of degree of Bachelor of Engineering in Electronics & Communication Engg. of the Visvesvaraya Technological University, Belagavi, Karnataka during the academic year 2020-21. We, the students of the project group/batch no. R33 hereby declare that the entire project work has been done on our own & we have not copied or duplicated any other's work. The results embedded in this project report has not been submitted elsewhere for the award of any type of degree.

Student Name-1 : Mr. AVINASH REDDY P

USN : 1DS17EC014

Sign : 

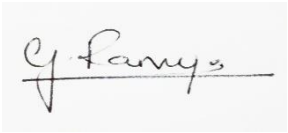
Student Name-2 : Mr. G GANESH REDDY

USN : 1DS17EC032

Sign : 

Student Name-3 : Ms. G RAMYA

USN : 1DS17EC034

Sign : 

Student Name-4 : Mr. DHEERAJ P

USN : 1DS17EC153

Sign 

Date : 18/05/2021

Place : Bengaluru -78

Acknowledgement

We are extremely grateful to the Chairman - **Dr. Hemachandra Sagar**, Vice Chairman – **Dr. Premachandra Sagar** and Secretary – **Mr. Galiswamy** of DSCE for providing us with adequate facilities and the infrastructure in order to complete the project. We express our gratitude towards **Dr. C.P.S. Prakash**, Principal of DSCE for providing us an opportunity to conduct this project as a part of our curriculum in the partial fulfillment of the degree course.

We convey our heartiest thanks to **Dr. T.C. Manjunath**, HOD, Electronics and Communication Engineering for his cordial support, valuable information and guidance, which helped us in completing this task through various stages. We take this opportunity to express our profound gratitude and deep regards to our guide **Dr. N Mahesh Kumar** for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

Our deepest thanks to the Project Convener – **Dr. K.N. Pushpalatha** and Department Project Coordinators – **Dr. Vimala, Dr. A Rajagopal, Prof. Deepa N P** and **Prof. Manasa R K** for their valuable suggestions and help.

We are would also like to thank the teaching staff at the Department of Electronics and Communication, DSCE without whom this project would have been a distant reality. We would also like to extend our regards to all the nonteaching staff of the department for their timely support.

We convey our sincere gratitude to our family and friends for their continuous and unparalleled love, help and support.

We would like to thank God, the Almighty, who has granted countless blessings, knowledge, and opportunities in our academic pursuits.

Avinash Reddy P

Ganesh Reddy G

Ramya G

Dheeraj P

Abstract

Our experiment is used to detect infiltrators and to detect suspicious activities in high security demanding area. Security breach of restricted locations has become a threat that everyone intends to eliminate.

Manual labor in security is faulty as it would have many human errors and other factors like sleep, food, leisure, and shifts are involved, in addition to that a security person cannot concentrate on all the areas at the same time and detecting an infiltration is almost impossible. We intend to use image processing based on deep learning (convolution neural networks) for image identification combined with machine learning algorithms to analyze and detect a breach based on image, height, thermals, and iris.

The project will be retrofitted with the existing security cameras and will also be attached to a large-scale thermal camera to detect if a person is in an invisible area like pip- lines or in a disguise, to second that it will also minimize and overcome the effect of jammers. We can enhance our project to raise alarms when there is a malfunction in some systems or if some suspicious activities are taking place. Furthermore, this can also be used to locate terrorists from a group of locals and send info to the base, thus enhancing our security furthermore to a great extent.

This project can also be implemented in other fields as well, like corporate buildings, hospitals, malls etc., with various tuning.

Keywords : *Deep Learning, Computer Vision , Deep Neural Networks, Facial Recognition, Night Vision Recognition, AI Security, Feature extraction, Data sets.*

Table of Contents

List of Figures

List of Tables

Nomenclature and Acronyms

Abbreviations :

IEEE	Institute of Electrical & Electronics Engineers
DSCE	Dayananda Sagar College of Engineering
ECE	Electronics & Communication Engineering
IDE	Integrated development Environment
DNN	Deep Neural Network
CNN	Convolution neural network
128 D	128-dimensional vector
SVM	Scalar Vector Model
Open CV	Open Computer Vision
IR	Infrared
LED	Light Emitting Diode
NVC	Night Vision Camera
NOD	Night Optical Device
NVG	Night Vision Googles
SD Card	Secure Digital Card
HDMI	High-Definition Multimedia Interface

CHAPTER 1

INTRODUCTION

1.1 Overview

Technology used in securing incredibly important places has changed a lot since the last few years and will continue to change in the coming years. Security is especially important when it comes down to smart applications. The new and emerging concept of smart security offers a convenient, comfortable, and safe way for securing extremely sensitive areas.

However, the proposed security system offers many more improvisations when compared to the conventional systems and these are easy to update in the present security systems. This will focus on image processing using deep learning where we intend to detect faces from the frames obtained using the hardware camera modules sending data to the microcontroller or raspberry pi modules where the image processing algorithms are present.

Then we are having thermal images to compare so that the image can be compared even at the dark and jammers cannot stop or block this kind of systems. Similarly, we are having other two detections one is height detection (height of a person is compared) and other is eye iris detection. All these variables will be retrieved, and a machine learning decision model will give a final output. This will give a heavy and healthy security system. While comparing the data if the system did not find any match, then automatically a message will be sent to the officers.

In other cases, any suspicious activity or person was found inside or near the restricted area then an alarm is automatically activated. Differentiation of civilian, an officer or a stranger can be easily done even from far distance using this technology. This system can also integrate a covid-19 detector using the thermal cameras.

1.2 Literature survey

Over the years there has been many advancements in automation pertaining to security and security systems, we have gone through some of these advancements and improvements to come up with an efficient project.

In the reference [1] and [2] they proposed facial detection and facial recognition using open Computer vision. The open CV has machine learning Libraries which are used in our program to develop an efficient security system which is helpful in overcoming the difficulties faced while using the existing conventional security system

The usage of all the various methods and algorithms such as Open CV DNN , Haar Cascade Frontal Face Classifier etc. are used during implementation of the project, and it also includes the applications and solutions regarding Open computer Vision. All these are taken from reference [3] and [4]

From reference [5] and [6], we have learnt about night vision cameras and night vision camera modules which we use in our project for recognition purposes and how it detects when there

is a suspicion in our surrounding and how we can identify it and prevent it. Their configurations connection and features etc.

In the reference [7] they have given a brief description about Raspberry pi how and where is it used, and the pin configuration in Raspberry pi and detailed description about all the pins present in raspberry pi. Uses of Raspberry pi, Advantages, disadvantages, features and pin configuration.

Reference [8] talks about Open CV with python, what are the libraries present in open CV are used in Python. How we can use them.

The Knowledge about the PyCharm which is an integrated development environment [IDE]. The tools used in PyCharm which are used for Visualization and Understanding the Deep Learning models is given in reference [9]

In Reference [10] and [11] we Learn about Tensor flow which is an open-source framework developed to run machine learning, deep learning, and other statistical and predictive analytics workloads. Tensor Flow is a Python-friendly open-source library for numerical computation that makes machine learning faster and easier.

Later many ideas were put forth, from our reference [12] we will get to know about the practical solutions from Preprocessing to Deep Learning.

1.3 Objectives / Scope / Aim of the project work

- To enhance the existing security system using Deep learning.
- Introducing night vision recognition system to improve security
- Continuous monitoring to give alert notifications based on conditions.

1.4 Methodology

The 1st program is used to create datasets of the persons we need to recognize, we used Open CV's hara Cascade frontal face classifier to detect a person's face, from the live video stream of our webcam and night vision camera separately.

Later we captured photos of our faces at various angles to create a dataset for each person which will be Used later for training. The datasets were placed in folders named based on the person's name and stored in a folder called dataset for captures from normal camera and dataset for night vison camera. The faces in the dataset are located 1st with the Dnn module

Which contains a pretrained deep learning model to detect faces using face landmarks and which is much more accurate than the Haar cascade.

In the 2nd program, we create a program to extract the embeddings from all the images in the dataset to quantify the face. The OpenCV Dnn module is used for this. To improve the accuracy, we use Triple loss function.

In the 3rd program we used a machine learning algorithm called SVM, which is a recognizer model.It recognizes the person's face from the data set and displays it accordingly. Next live video detection is done which follows all the steps above. IN case of any intrusion or suspicion different alert notifications are given. These alert notifications are given through LEDs and speakers.

1.5 Organization of the project report

The project work undertaken by us is organized in the following sequence as follows.

A brief introduction to the work was presented in the introductory chapter in chapter 1 which Consists of overview of the project objectives of our project. As well as the motivation and Basic methodology of the project is seen in Chapter 1.

Block diagram and working principle of project work undertaken by us is presented in the chapter 2.

Flow chart of the project along with the algorithm as well as a overall block diagram are seen in the Chapter 3.

Hardware and Software tools Description employed in our project work is depicted in the chapter 4 in detail.

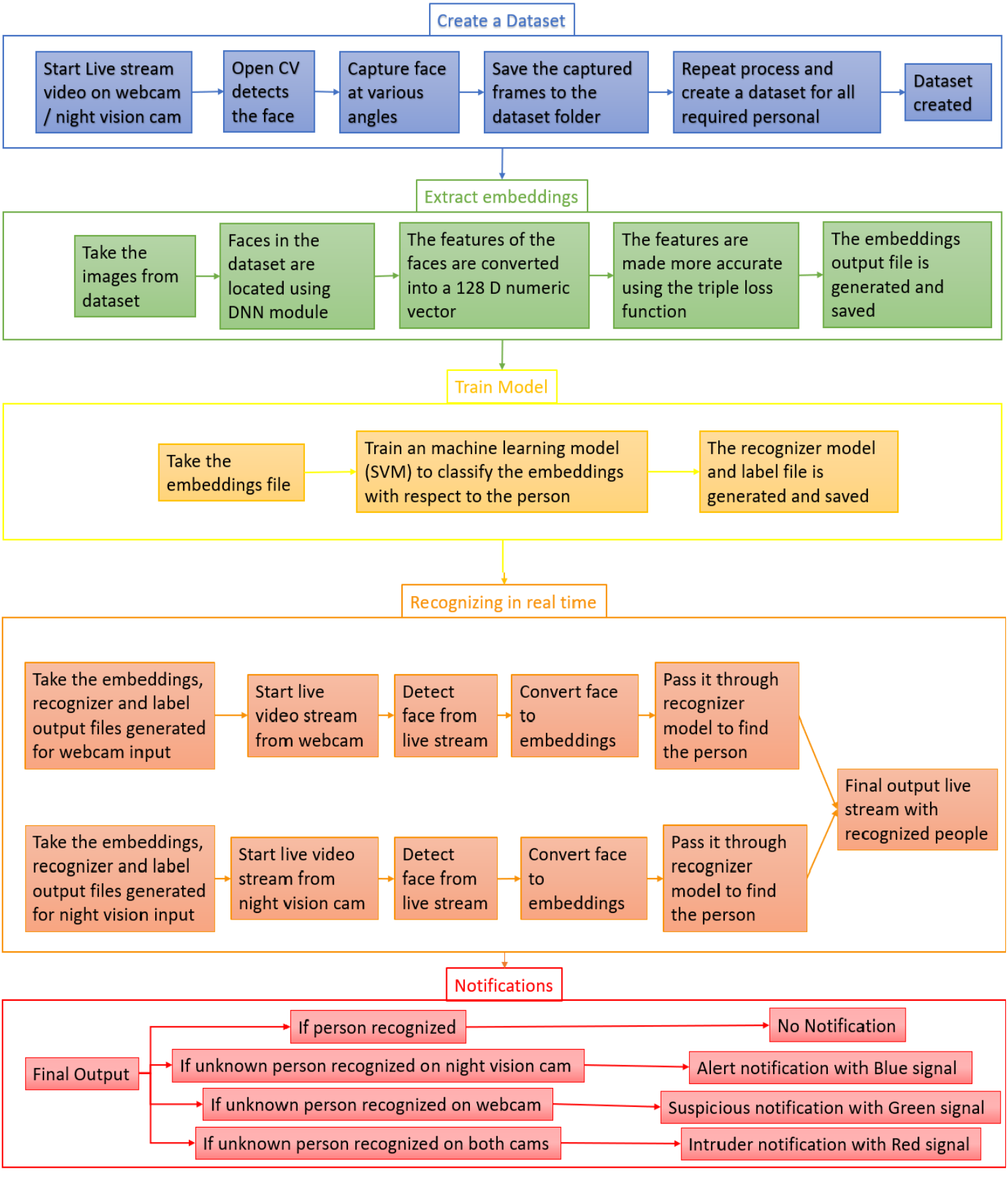
In Chapter 5 we can see in detail discussions of results of both simulation as well as the Hardware results

In chapter 6 we have discussed the Applications advantages and disadvantages in our current project.

Finally, the report concludes with the conclusion and future work in chapter 7.

CHAPTER 2

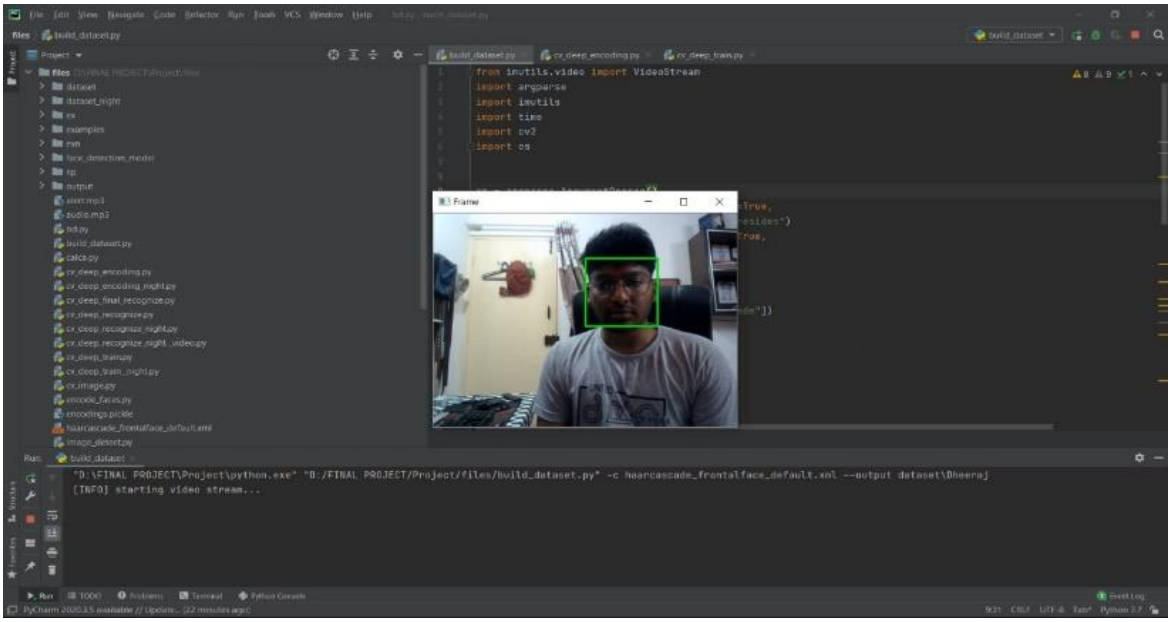
BLOCK DIAGRAM AND WORKING PRINCIPLE



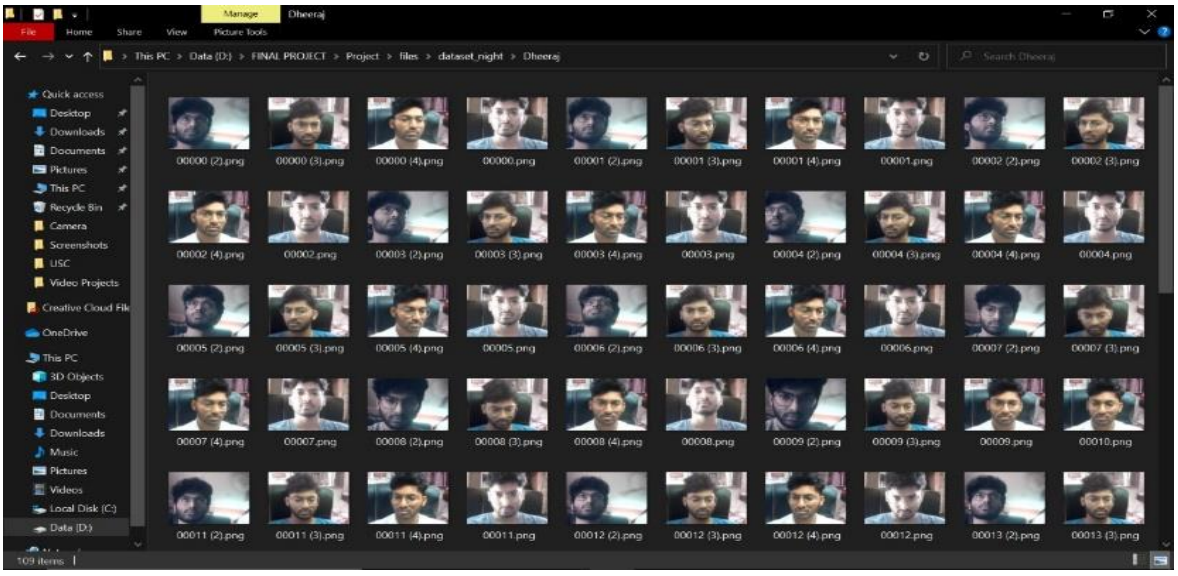
Creating Dataset

Our project is based on basic deep learning techniques, during the 1st phase major work of collecting data for datasets, embedding, and encoding images and training models takes place.

In this model we 1st designed a program to create datasets of the persons we need to recognize, we used Open CV and hara cascade frontal face classifier to detect a person's face, using our webcam. Later we captured photos of our faces at various angles to create a dataset for each person which will be used later for training. The datasets were placed in folders named based on the person's name and stored in a folder called dataset.



2 (a) Output for Haar Cascade Frontal face classifier

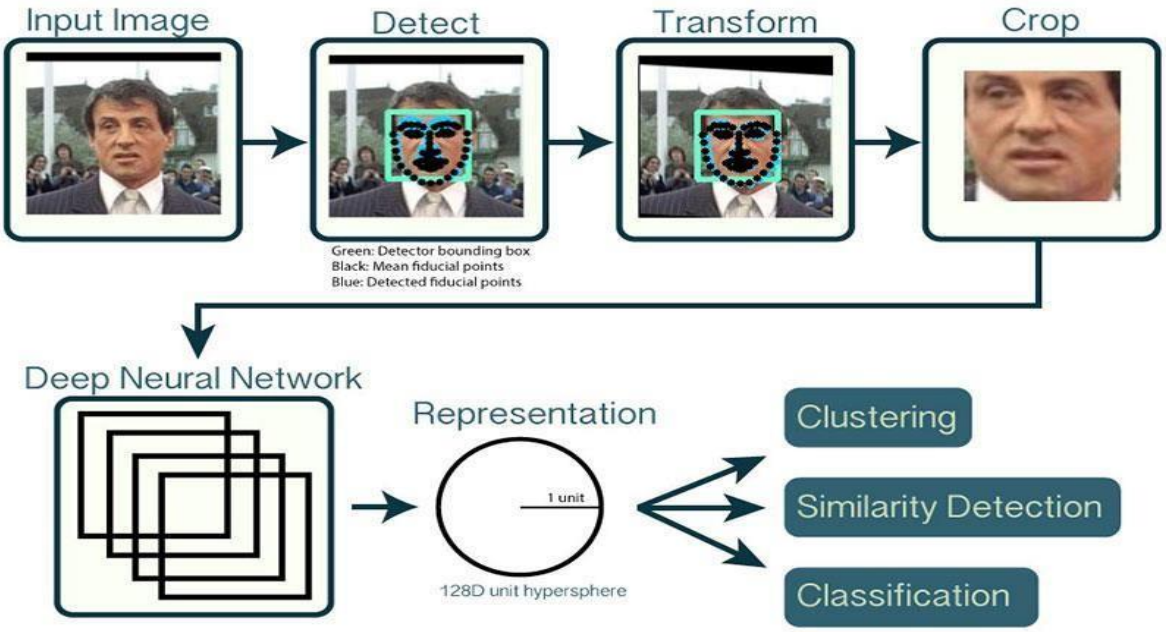


2 (b) Data set of a person is created

Extract Embedding

Secondly, we create a program to extract the embeddings from all the images in the dataset to quantify the face. The OpenCV DNN module is used for this. The faces in the dataset are located 1st with the DNN module which contains a pretrained deep learning model to detect faces using face landmarks and which is much more accurate than the hara cascade.

Once the images are detected we extract the embedding for each face, this is 128-dimension vector also called 128d which quantifies the face itself. This file which contains all the embedding is stored on the disk which is later used to train the recognition



2 (c) Extract Process

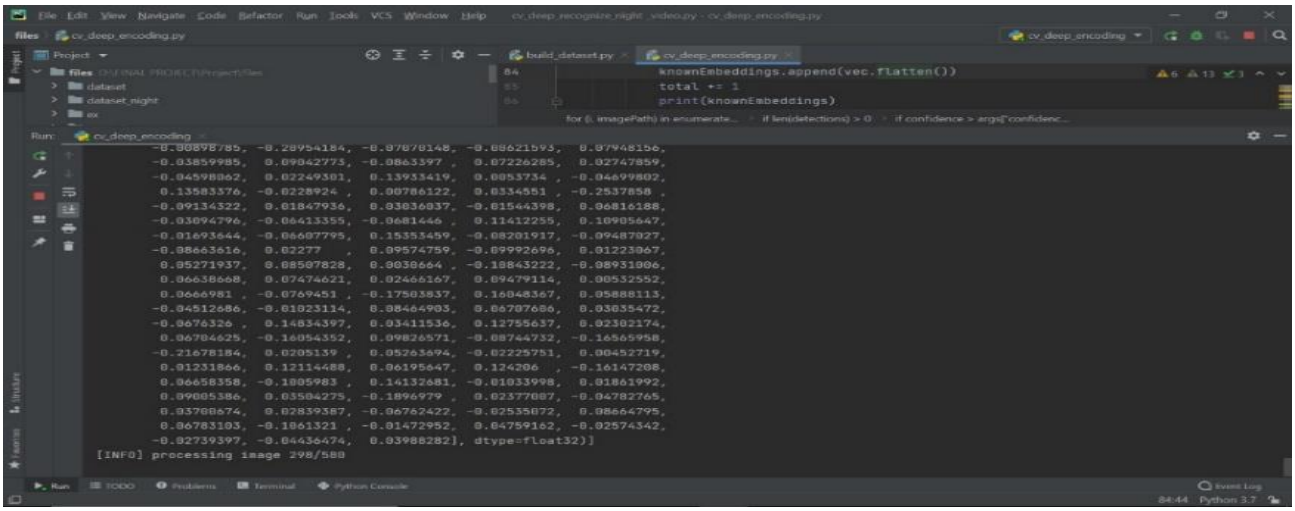


Fig 2 (d) Quantization of facial features

Train Model

The 3rd program is created for training purposes. The input data or the embedding file is used to train the model, 1st there is an anchor, or the current image let us say image of person A all its embeddings are present on one side, then there is a positive image which means it's a different image for person A. When a positive image is founds, the weights are tweaked using triple loss function in such a way that the embeddings of these two images are close together.

When the anchor compares the current image of person A with and negative image which means an image of some other person than A, the embeddings are pushed far away from these two images. This process is continued and a final file containing all the tweaked 128d embeddings of vector is stored on disk.

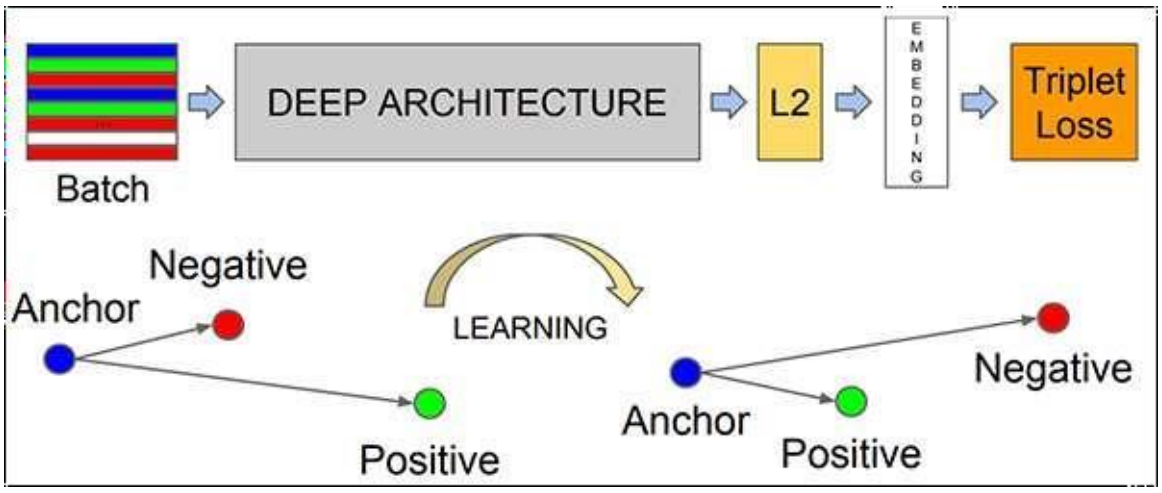


Fig 2 (e)

The embeddings are made accurate using the triple loss function. 1st there is an anchor, or the current image let us say image of person A all its embeddings are present on one side, then there is a positive image which means it's a different image of person A. When a positive image is found the weights are tweaked in such a way that the embeddings of these two images are close together.

When the anchor compares the current image of person A with and negative image which means an image of some other person than A, the embeddings are pushed far away from these two images. This process is continued and a final file containing all the tweaked 128d embeddings of vector is stored on disk.

Fig 2 (f)

Fig 2 (g)

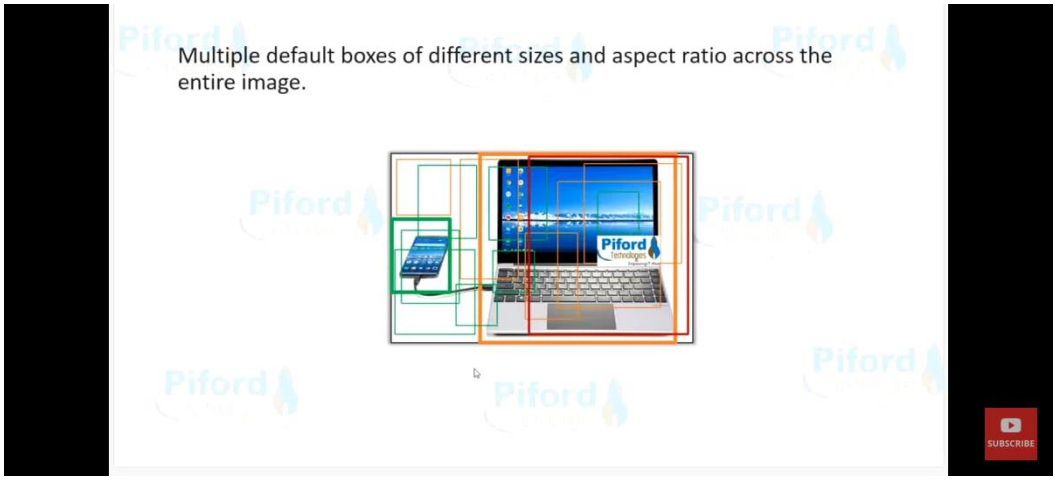


Fig 2 (h)

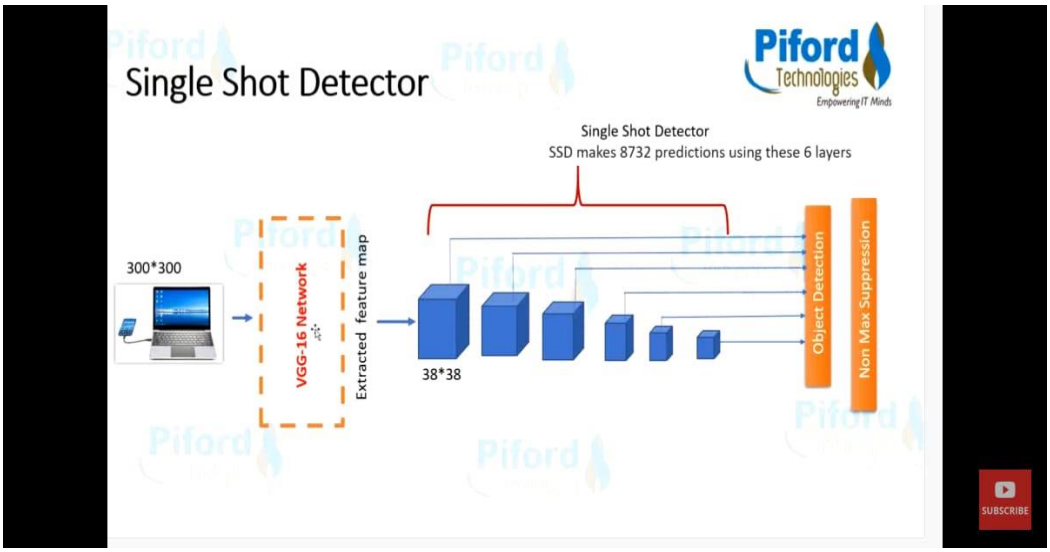


Fig 2 (i)

The 3rd program is used to train an SVM model on top of the embeddings created in the prior step to classify the embeddings to the right person. As the embedding values are far apart a basic classification model like SVM is enough to classify the faces. After the SVM model has been trained we will dump the output file on the disk. A label encoder file containing all the persons names is also created to classify the persons and show their names above the recognized face.

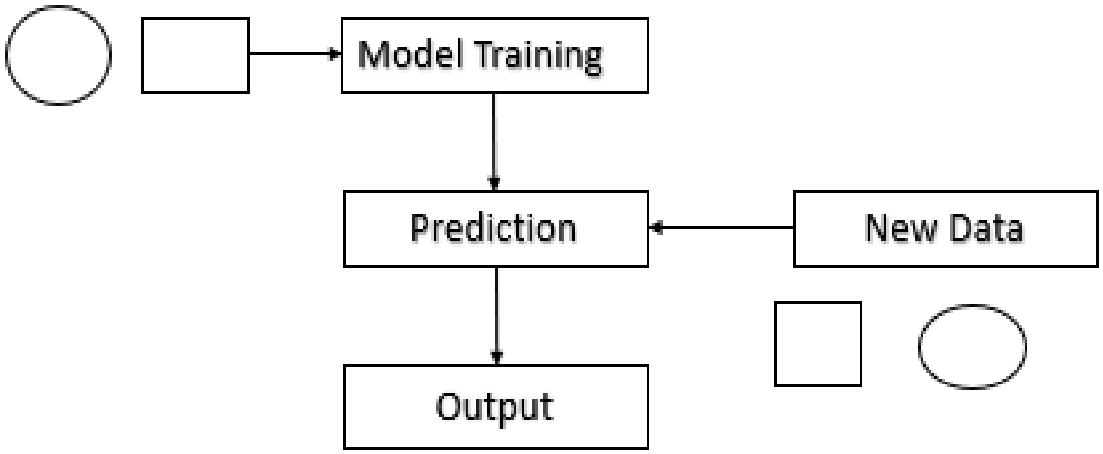


Fig 2 (j)

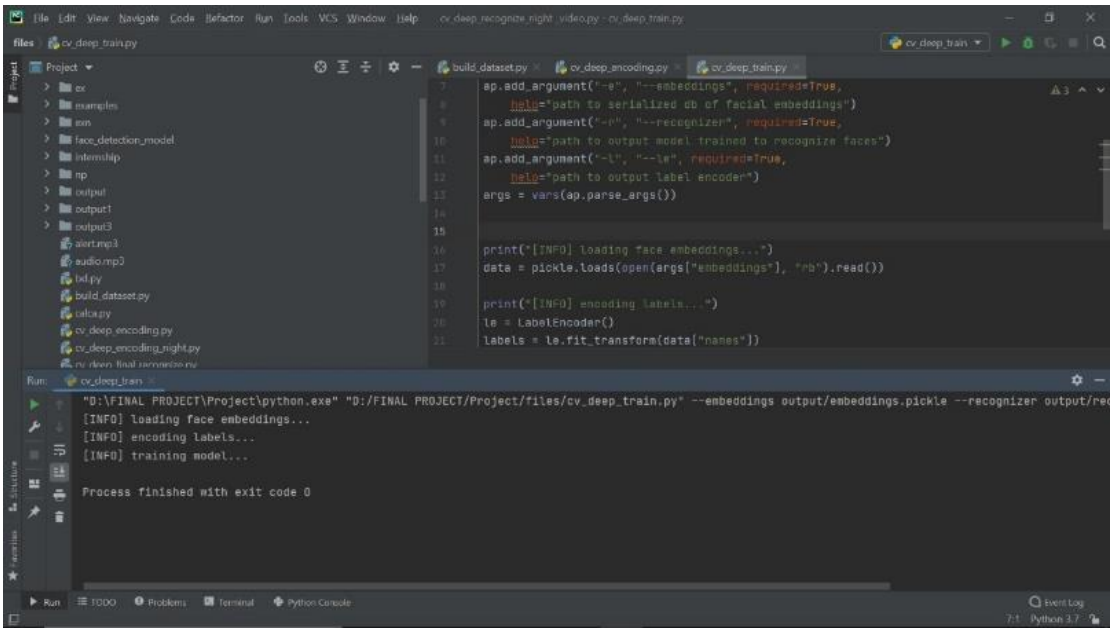


Fig 2 (k) the program for getting the embedding output as well as recognizer output4

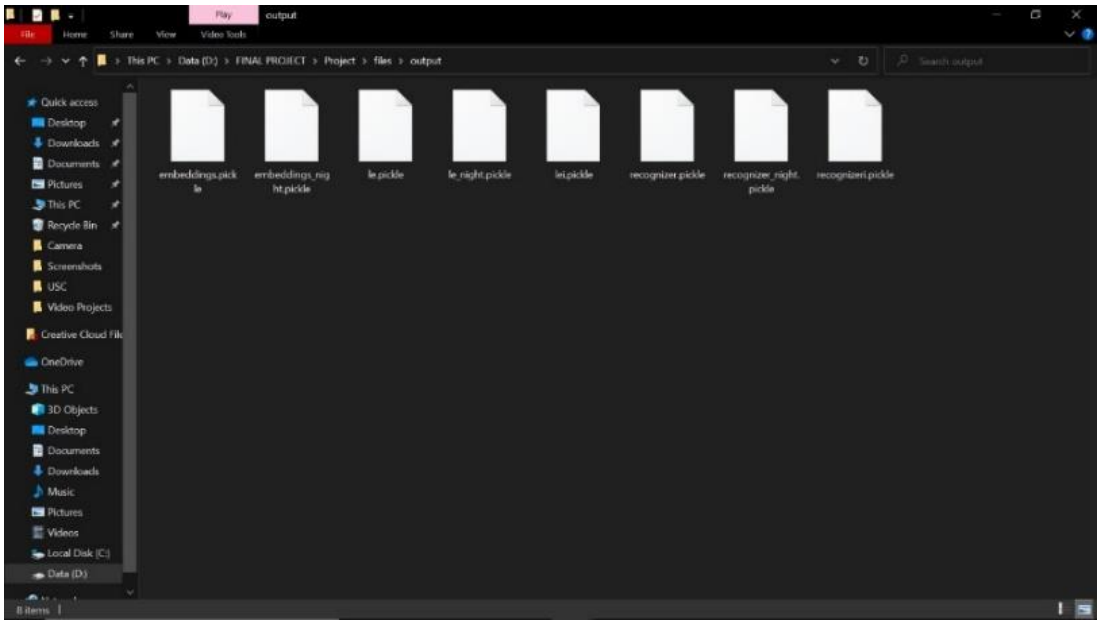


Fig 2 (l) The embeddings output as well as recognizer output are shown in this figure

IR camera

- We made our own IR Camera To cut down the costs and make it more efficient.
- It is made using the normal web cam by removing the IR Filter present inside the web cam.
- An Infrared LED array was made to improve cameras quality during the night.
- In this manner the cost has been cut down to a greater extent and we thus made used of the normal camera to obtain IR night vision camera
- The remaining deep learning model used is same i.e., the deep neural network model used in the first phase and the model is trained until desired accuracy is achieved.

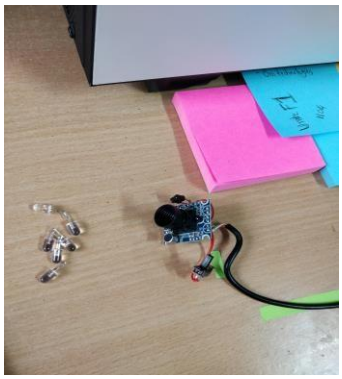


Fig 2 (m)

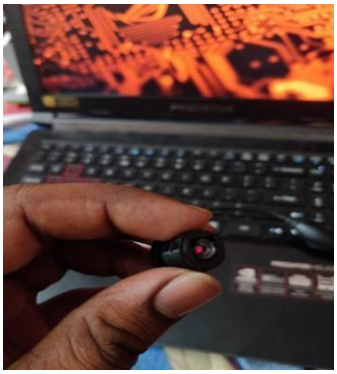


Fig 2 (n)

Fig 2 (o)

Above images shows the process of changing regular webcam into a night vision camera. Fig2 (o) shows the broken IR filter removed from the normal webcam

Recognition in Real Time

The final program is created to recognize the faces in live video stream by using all the output files that are the models created previously. 1st the detection model provided by open cv Dnn module is used to detect faces in the live video, later the embedding file is used for reference to detect faces. Then the SVM model is used to compare the detected faces with the reference embeddings and gives an output. Finally, the label encoder file is used to display the name above the detected face.

Notifications

Once the faces have been recognized in the live stream various notifications are given based on various conditions

If the person is recognized on both cameras no notification is given. If the person is recognized as unknown on night vision camera, an alert notification with a blue signal is given. If the person is recognized as unknown on webcam, a suspicious notification with a green signal is given.If the person is recognized as unknown on both cameras, an intruder notification with a red signal is given.

CHAPTER 3

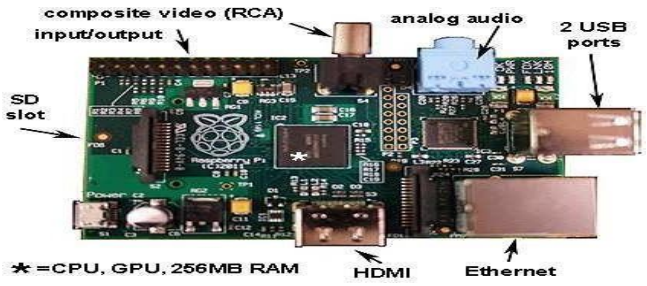
HARDWARE/ SOFTWARE TOOLS /DESCRIPTION/INTERFACING

Hardware Components

1. Raspberry pi
2. IR/Thermal camera module
3. Existing Cameras
4. Sd card

Raspberry pi

The **Raspberry Pi** is a **low** cost, credit-card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.



3 (a)

Night vision camera

A night-vision Camera (NVC), also known as night optical/observation device (NOD) and night-vision goggles (NVG), is an optoelectronic device that allows images to be produced in levels of light approaching total darkness. The image may be a conversion to visible light of both visible light and near infrared, while by convention detection of thermal infrared is denoted thermal imaging.

3(b)

Existing Camera

A **camera** is an optical instrument used to capture an image. At their most basic, cameras are sealed boxes (the camera body) with a small hole (the aperture) that allow light in to capture an image on a light-sensitive surface (usually photographic film or a digital sensor). Cameras have various mechanisms to control how the light falls onto the light-sensitive surface. Lenses focus the light entering the camera, the size of the aperture can be widened or narrowed to let light into the camera, and a shutter mechanism determines the amount of time the photo-sensitive surface is exposed to the light.

3(c)

Sd Card

A Secure Digital (SD) card is a tiny flash memory card designed for high- capacity memory and various portable devices, such as car navigation systems, cellular phones, e-books, PDAs, smartphones, digital cameras

3(d)

Software Components

1. Python
2. PyCharm
3. Dib
4. Pandas
5. NumPy
6. Open CV

Python

Python is a high-level programming language designed to be easy to read and simple to implement. It is open source, which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to Java and .NET virtual machines.

PyCharm

PyCharm is an integrated development environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.

Pandas

pandas are a **software** library written for the **Python** programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series.

NumPy

NumPy (pronounced /ˈnʌmpaɪ/ (NUM-py) or sometimes /ˈnʌmpi/ (NUM-pee)) is a library for the **Python** programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays

Open CV

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

CHAPTER 4

ALGORITHM OR FLOWCHART

Program 1: Building a Dataset

- Import all the libraries required
- Create argument parser so that we can pass arguments externally
- Load the Haar cascade classifier which is required for face detection
- Create a command to start the live video stream
- Create an infinite loop, to capture frames
- Grab each frame from video frame and clone it
- Resize the frame, for faster face detection
- Convert frames to grayscale
- Create a box around the detected face
- Display the modified video stream
- Create an if clause so that when we press k, the frame is saved and if we press q the loop breaks
- Destroy all windows
- Close program

Program 2 : Extract embeddings of face

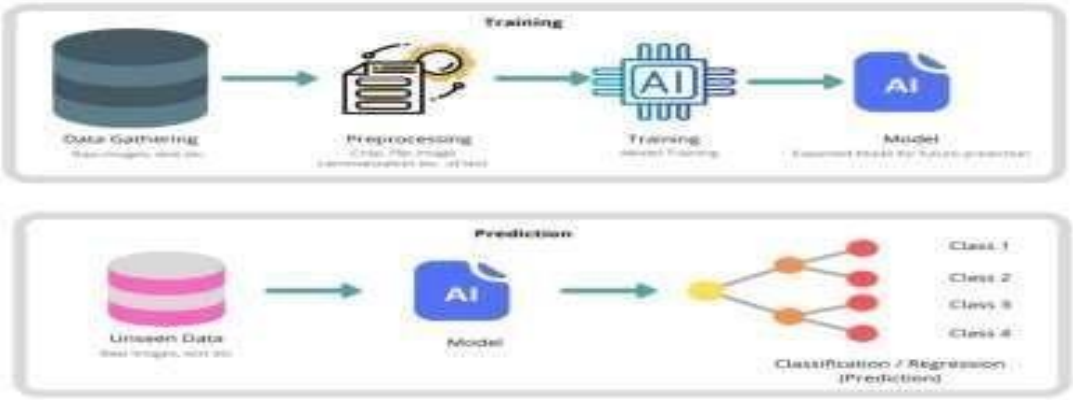
- Import all the libraries required
- Create argument parser so that we can pass arguments externally
- Load detector model and embedding model
- Create lists to save embeddings and corresponding names
- Loop over the dataset to create embeddings
- Resize all the images from the dataset once loaded
- Create a blob of the image and store it
- Pass the blob through the loaded detector to detect faces
- Create another blob this time for the faces detected and not the entire image
- This blob is passed through the embeddings model to create the embeddings
- All the embeddings and names are added to the blank list created before
- Finally, an embeddings output file is created
- Close the program

Program 3: Training the model

- Import all the libraries required
- Create argument parser so that we can pass arguments externally
- Load the embedding file created previously
- Extract names from the embedding file
- As the data is numeric, we can train it using SVM
- Train the model so that the corresponding embeddings referrer to the correct person
- Save the recognition model to disk
- Close the Program

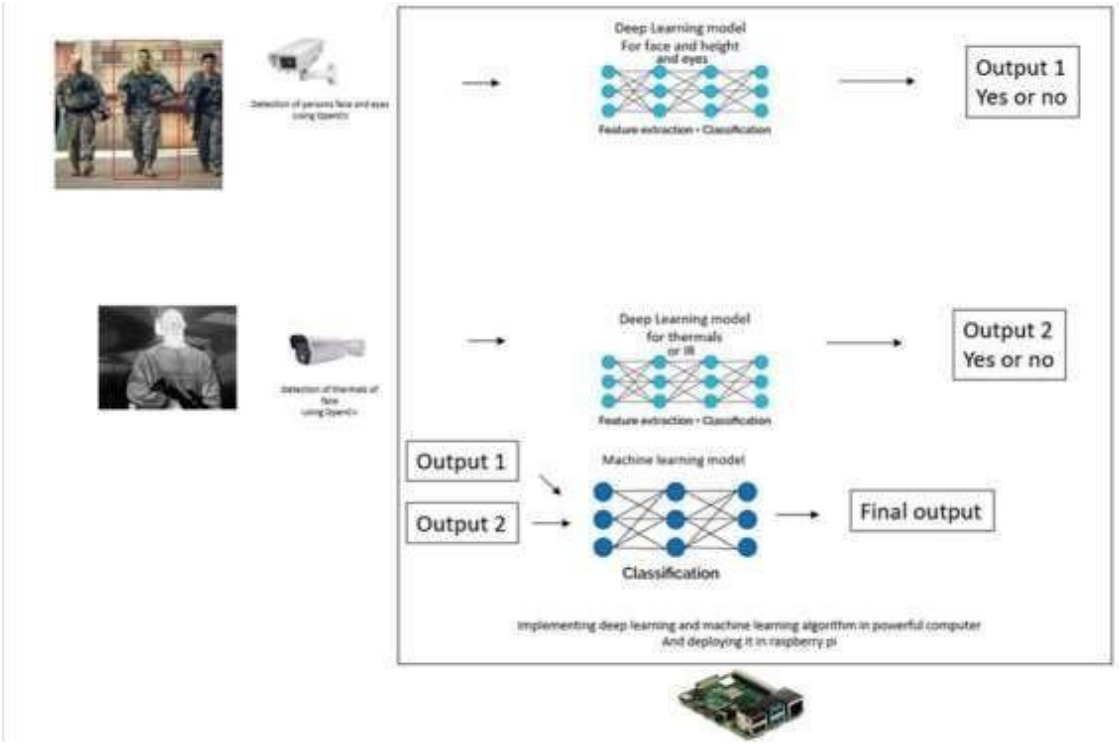
Program 4: Recognizing the Program in Live video

- Import all the libraries required
- Create argument parser so that we can pass arguments externally
- Load the detector model, embedding model, labels and recognizer model created in the previous steps.
- Start video stream
- Capture the frames and resize it
- Create a blob to detect where the face is using detector
- Create a for loop
- Once the faces have been created create another blob this time only for the detected face
- Calculated the facial embeddings using the embedding model
- Send this embedding to the recognizer model to classify it
- Place a box around the face with the received output name and its probability
- Finally display the modified frames as a stream
- Close the program



4 (a) Pipeline Diagram

4 (b) Overall Working



CHAPTER 5

RESULTS AND DISCUSSIONS

Simulation Results

Output 1: The Input of known Person is given, and it is identified with his name.

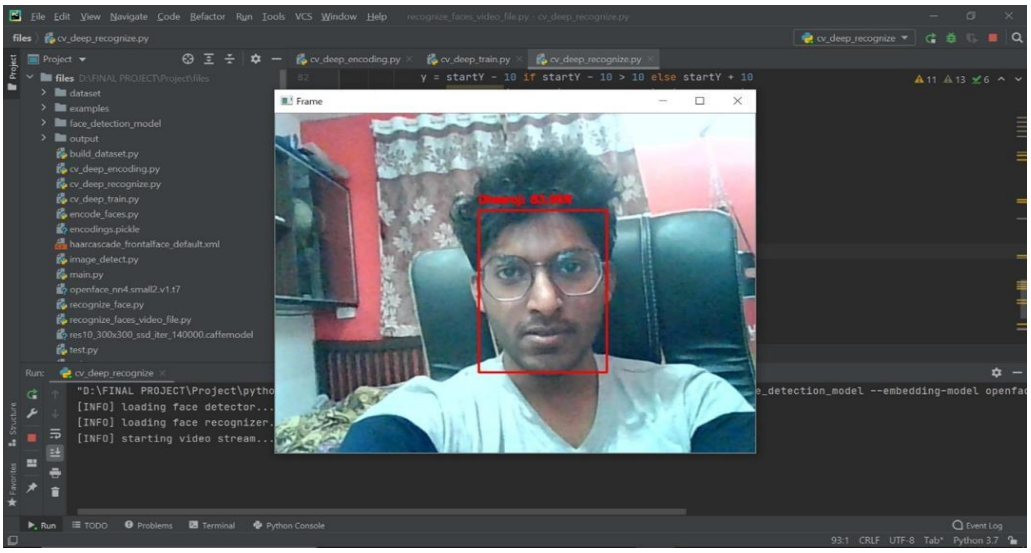


Fig 5(a)

Output 2: The Input of Unknown Person is given, and it is identified as Unknown.

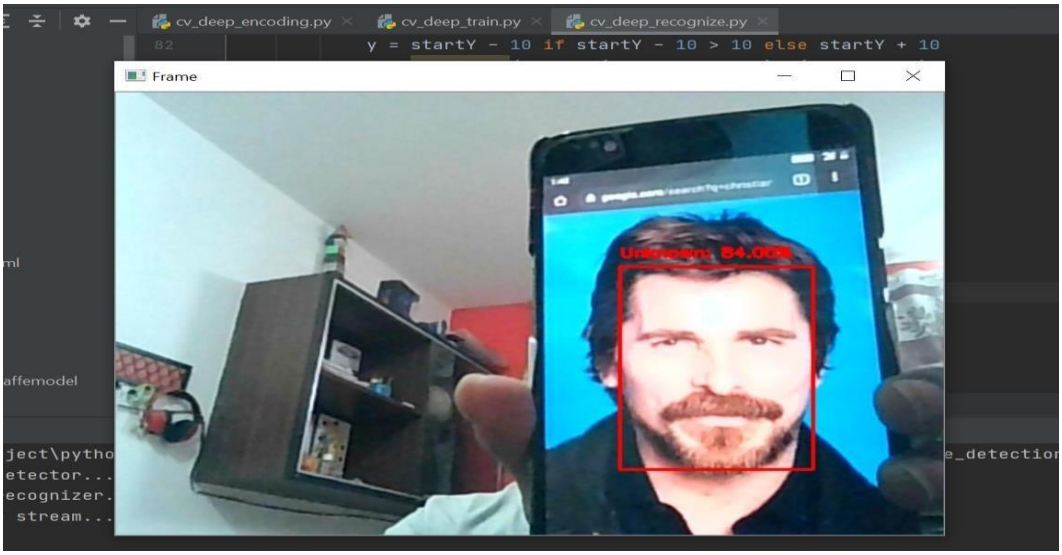


Fig 5(b)

Output 3: The known input is given and thus output is obtained using night vision

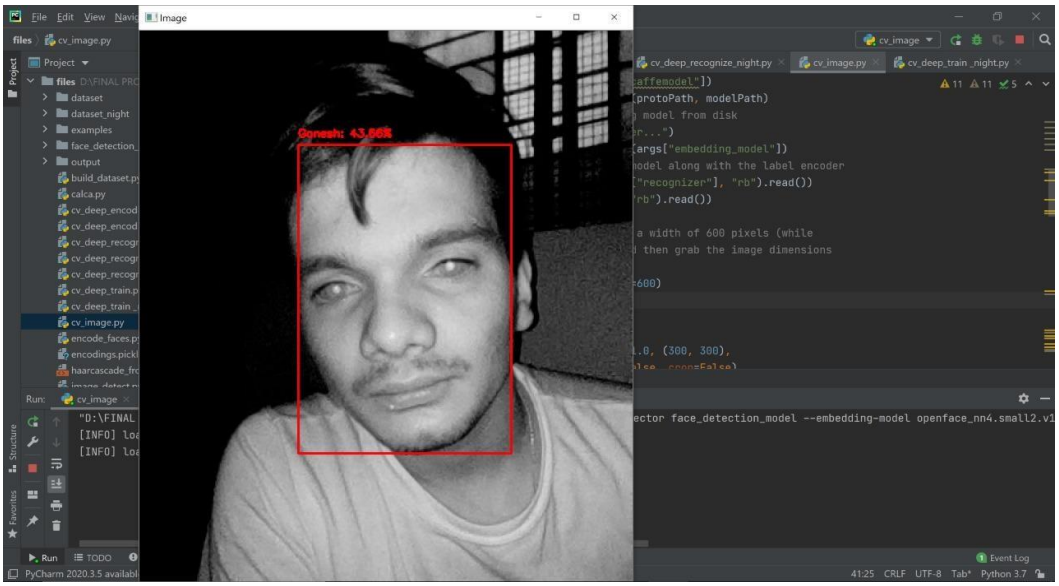


Fig 5(c)

Output 4: A group photo of 9 people is given and only the known person is detected

Fig 5(d)

Output 5: Live stream video is given, and multiple known people are identified

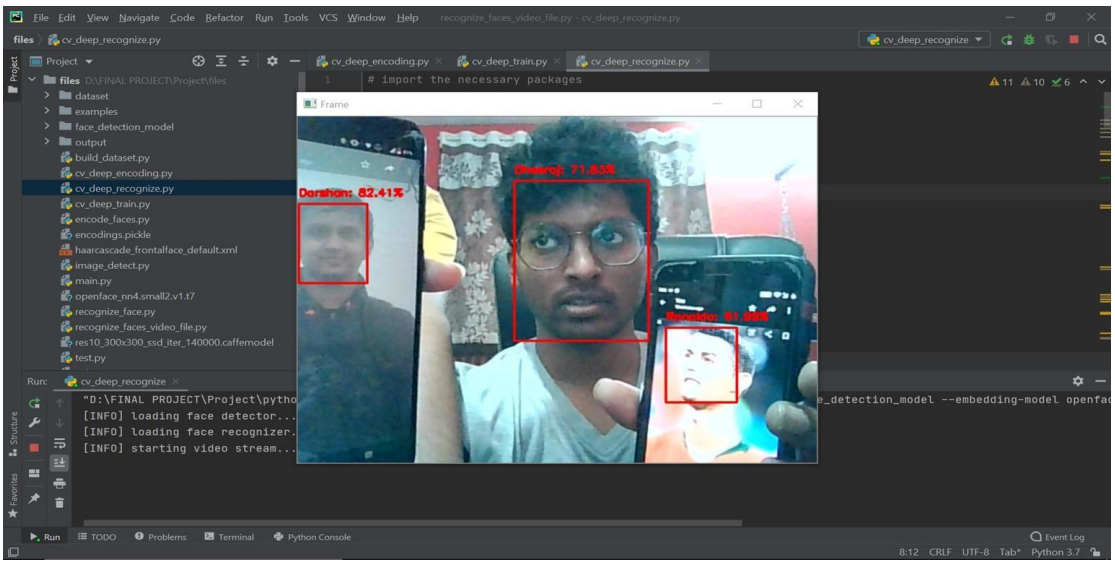


Fig 5(e)

Output 6: Multiple Faces are detected on live stream on both the Normal camera as well as Night vision camera

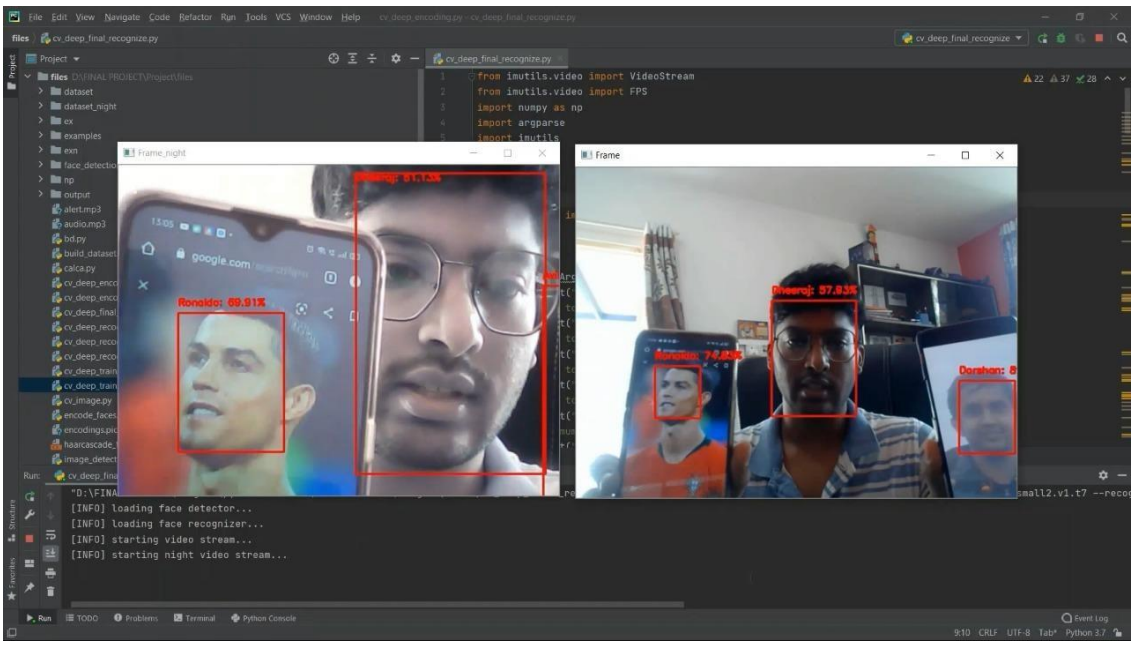


Fig 5(f)

Output 7: The unknown person is detected on both cameras, even the known person is detected

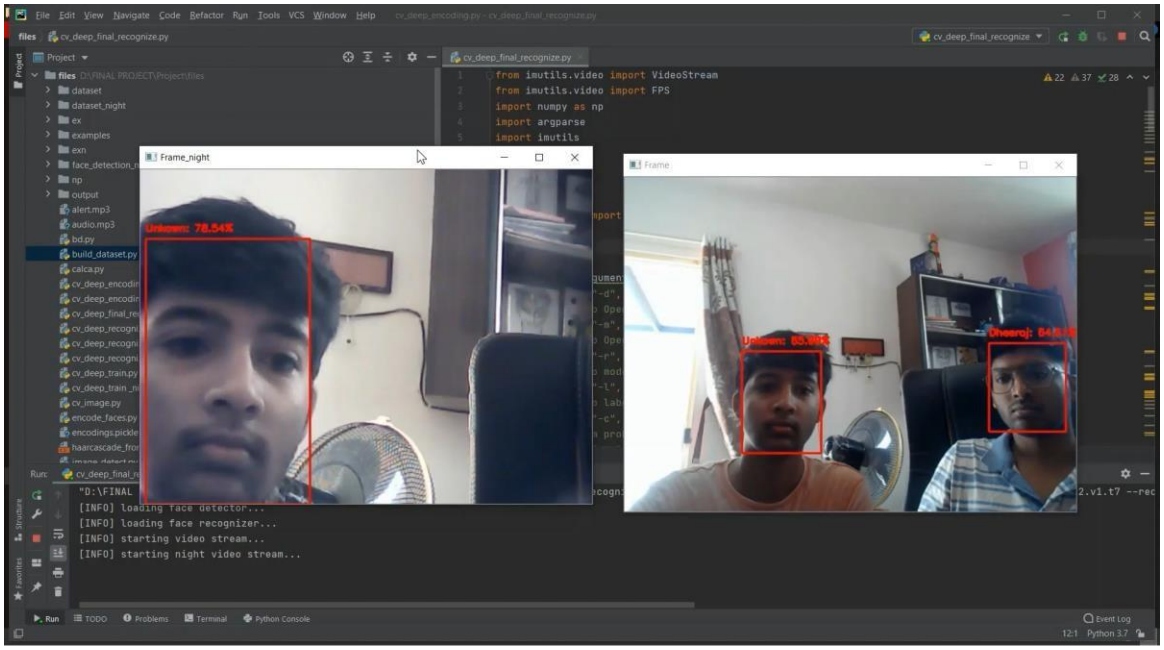


Fig 5(g)

Results

We tested the program, and we were able to detect our faces and with varying accuracy as seen in figures. The model was good enough to detect faces with high accuracy as seen in fig 5(a). It also was able detect an unknown face and refer it as unknown as seen in fig5(b). It was also able to detect faces at a time with good accuracy as seen in fig5(d). It was able to detect the face even in night vision camera as shown in fig 5(c).

In fig 5(e) we were able to detect the faces of multiple known people in live stream in normal camera. Whereas in 5(f) multiple Faces are detected on live stream on both the Normal camera as well as Night vision. In 5(g) The unknown person is detected on both cameras, even the known person is detected.

Hardware Implementation

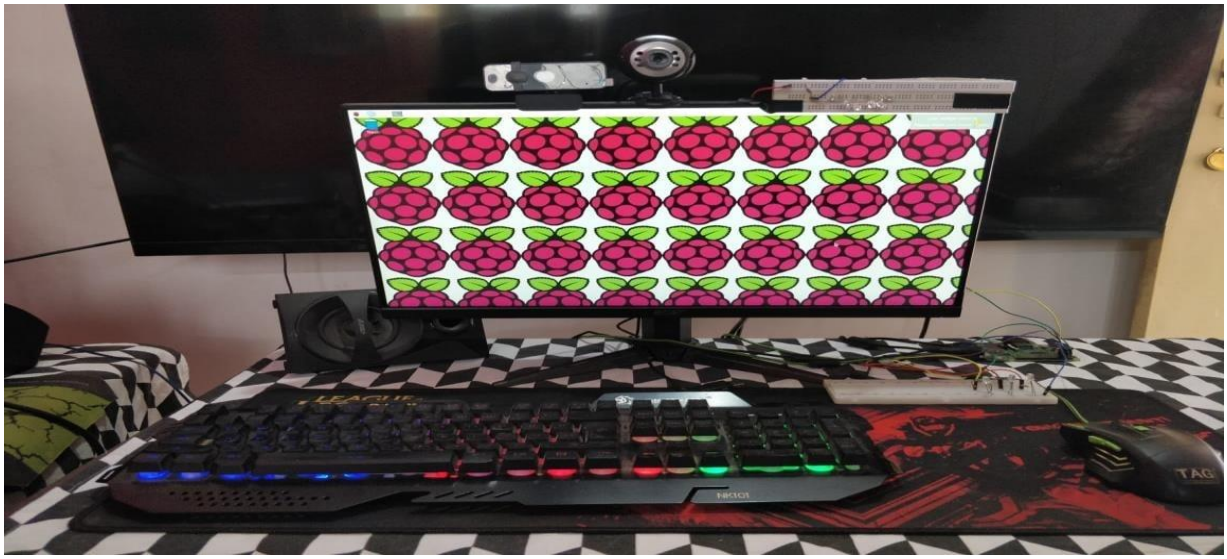


Fig 5(h)

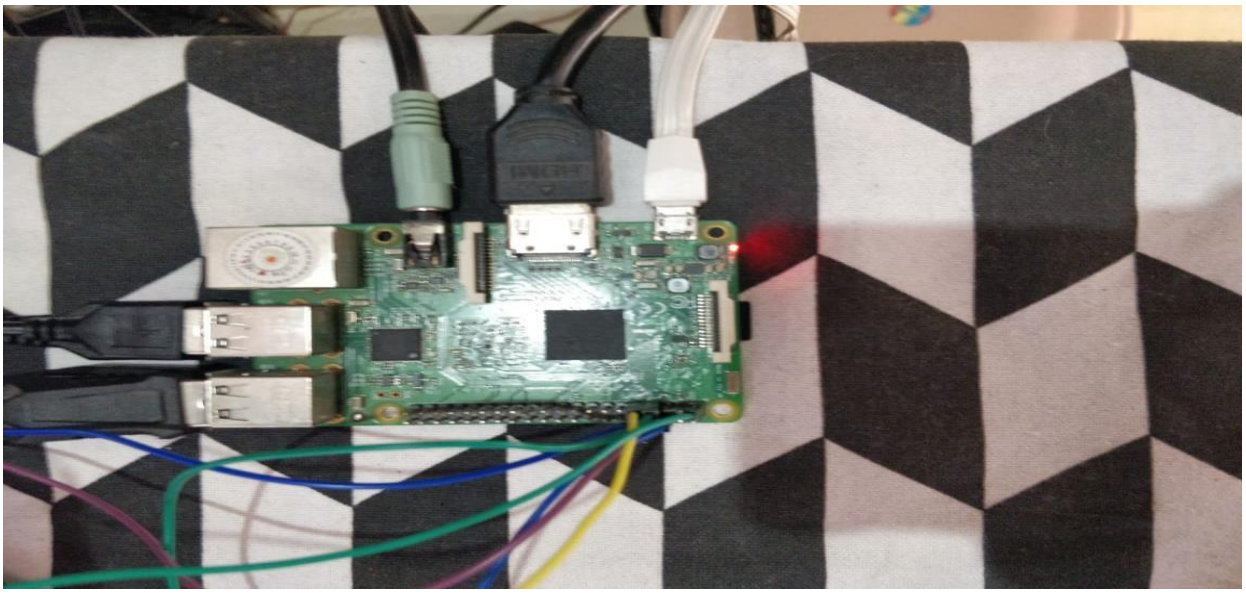


Fig 5(i)

- As we can see from Fig 6(i) we have connected our Raspberry pi to our monitor through a cable, and speakers through 3.5 mm jack. The power supply is given by a 2.4-amp Adaptor.

- The Other connections are the cameras. Both the cameras are connected i.e., normal camera as well as night vision camera and the remaining are mouse and keyboard connections.

- As shown in fig 6(l) 6(m) 6(n)We have connected 3 different LEDs to indicate 3 different levels of infiltration and we also have IR LED array to get more precession in the night vision camera.

Hardware Results

Output 1: -An image of group of 10 people is given and its only detecting the person present in our dataset

Fig 5(j)

Output 2: Known person is detected both in Normal camera as well as Night vision camera on live stream, no notification is raised.

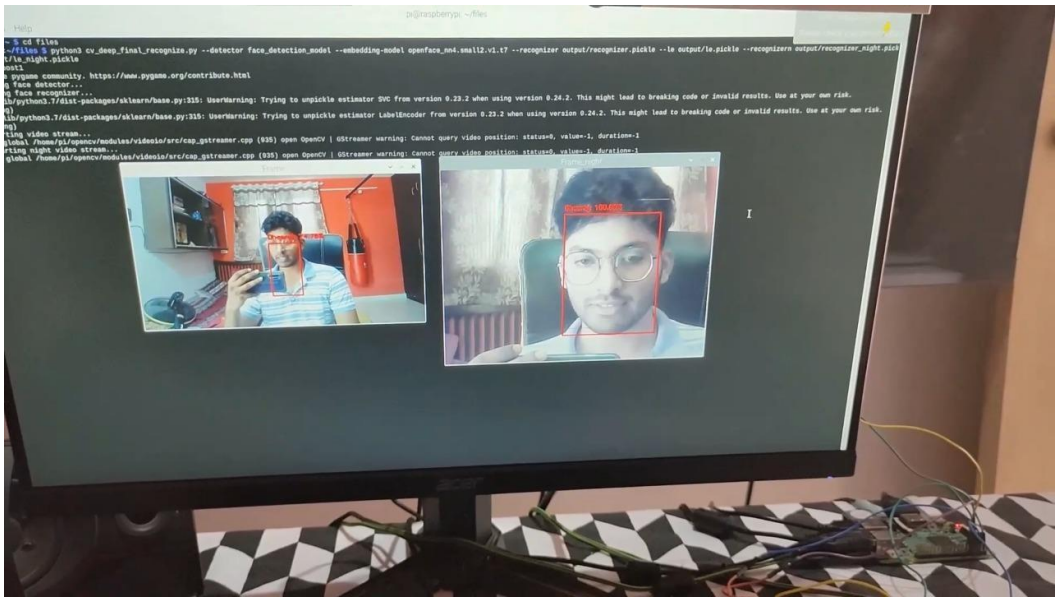


Fig 5(k)

Output 3: Multiple people are detected on both Normal camera as well as Night vision camera on live stream, no notification is raised.

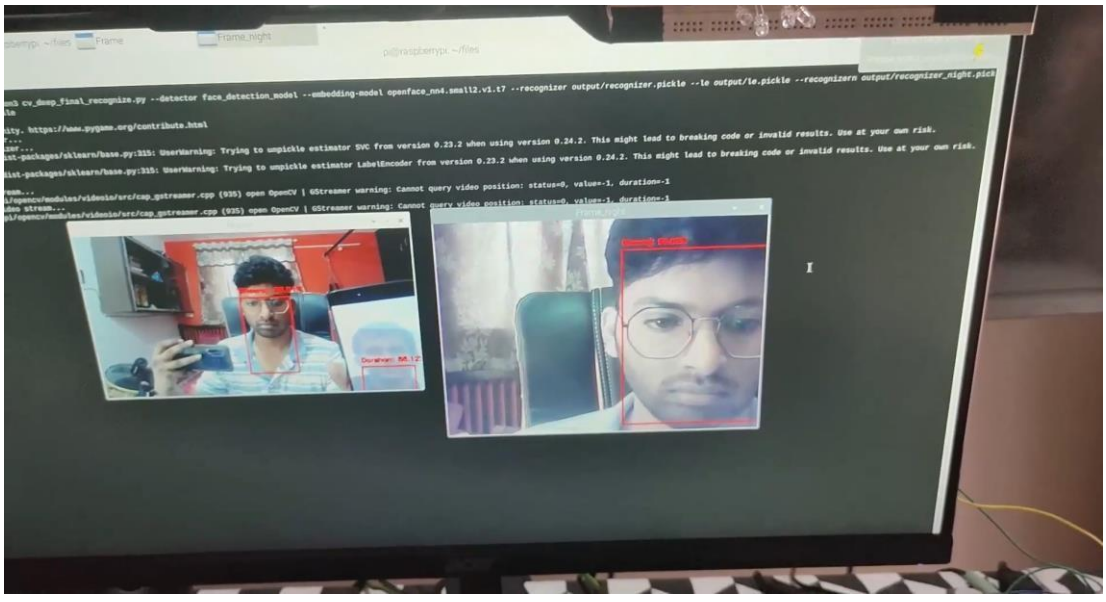


Fig 5(l)

Output 4: - An unknown person has been detected on the live stream only on the night vision camera hence a notification saying "Alert" is given, with a BLUE signal.

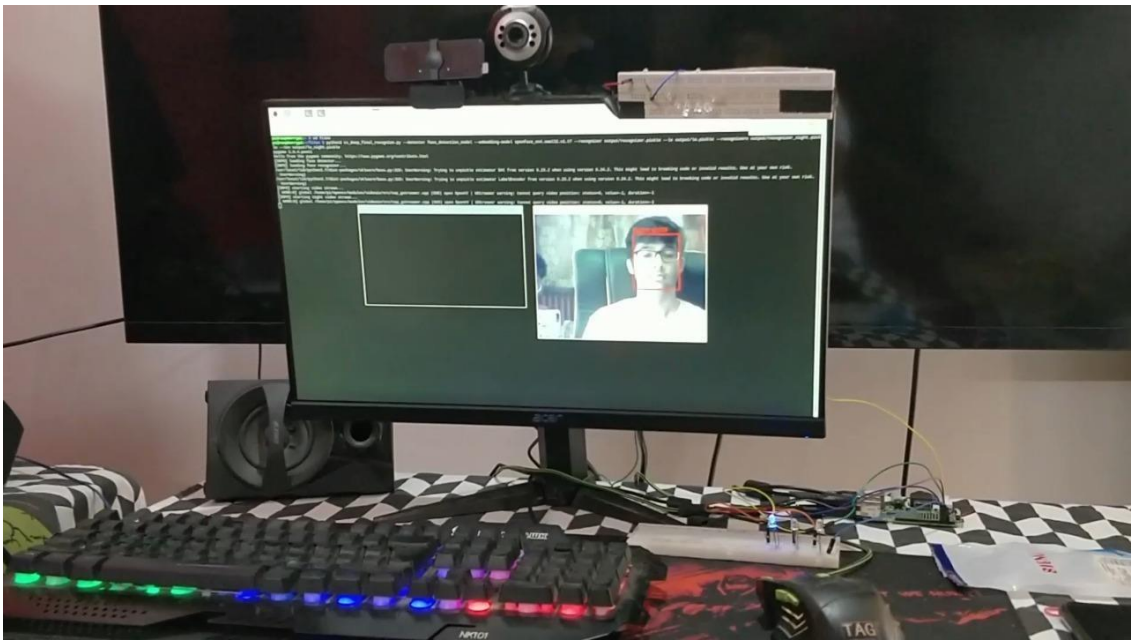


Fig 5(m)

Output 5:- An unknown person has been detected on the live stream only on the webcam hence a notification saying "Suspicious" is given, with a GREEN signal

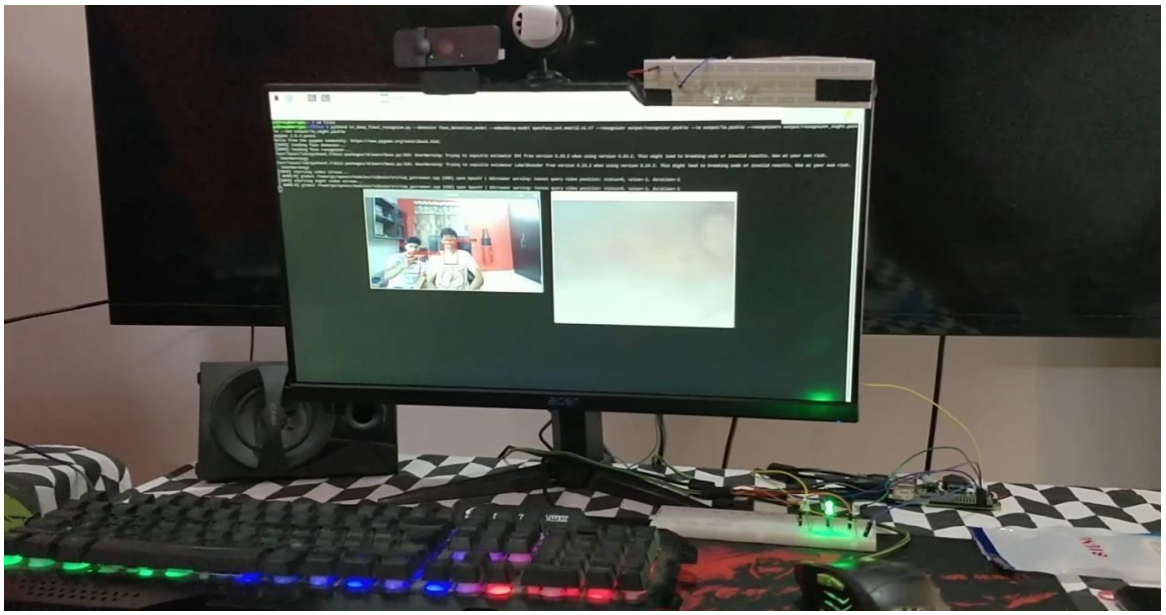


Fig 5(n)

Output 6:- An unknown person has been detected on the live stream on both the camera hence a notification saying "Intruder" is given, with a RED signal.

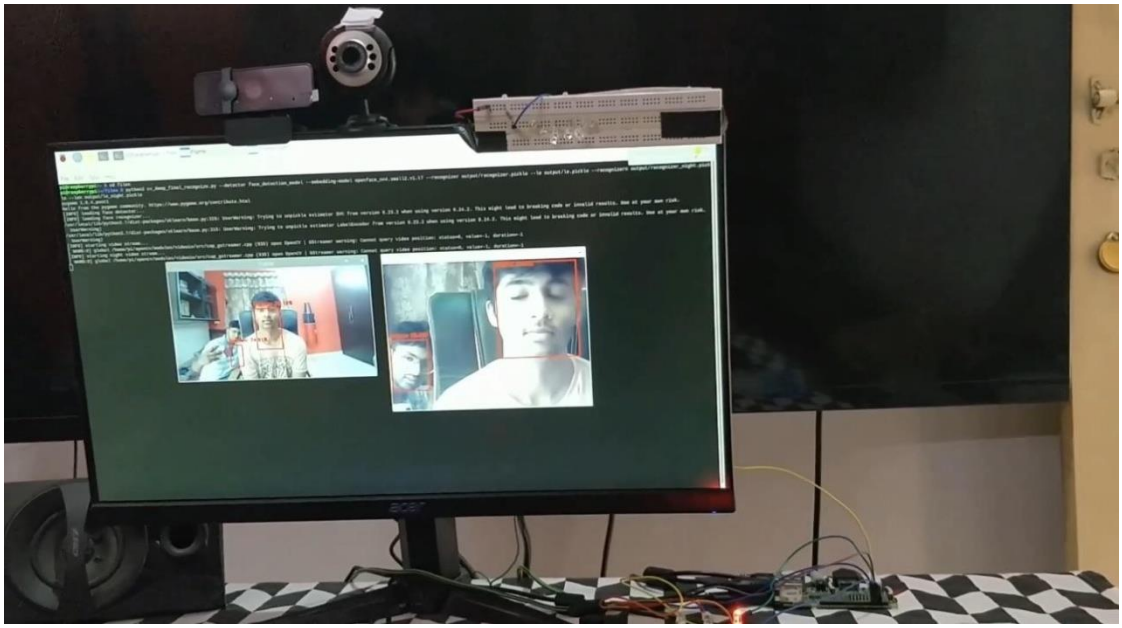


Fig 5(o)

The above figures depict the hardware implementation results of the same pictures taken for simulation results.

In the live video demonstration, we will be able to see how the camera is detecting the Suspicious person or an intruder even more clearly and thus it even gives out voice message if there is any intruder saying suspicious, alert , intruder , and as we have also given the LED connection in our Hardware the respective LED glows along with the Message which is given.

CHAPTER 6

APPLICATIONS, ADVANTAGES AND LIMITATIONS

Advantages of the project work :

1. The use of Image processing for object detection and Machine learning for photo analysis helps us achieve maximum security with least risk.
2. The system can check the status of the secured area and detect the faces in the area at the time of the breach and notify the user both via the cellular network as well as the internet.
3. Deployment of this system is a simple since an exceptionally light set up is installed at the place to be secured leaving the rest of the computation to happen on the cloud.
4. It can be retrofitted in the existing security system, without much hassle.
5. It reduces manual labor and human errors to a great extent.

Application of the project work :

1. Military bases.
2. Highly restricted areas.
3. Hospitals, schools and in government and private working areas

Limitations of the project work :

1. In case of the false positives and false negatives not being handled properly, the system could trigger a false alarm.
2. The addition of new users to verify the input to the model against, the model would have to be retrained to incorporate these changes.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

To conclude, this project will enhance the present security system. The process will be simple, but the security will be improved tremendously. A normal security guard watching over 20 cameras can make many human errors not to forget other factors like fatigue and carelessness.

If we add this project to all the cameras, they will be able to detect and recognize an intruder without any manual labor and notify us immediately. Even if the guard falls asleep the recognition model will be running and there will be no worries about any or intrusion

Though, In some conditions, it might be faulty when there is much similarity between the persons face as the embeddings might cluster them into one. Even though the classification is quite accurate, when there is a large number of people in the frame the accuracy might be lower than excepted.

As the Deep learning models require more processing, when implemented on the raspberry pi, it cannot cope up with the required processing power thus the frame rate is very low.

REFERENCES

[1]. Lahiru Dinalankara. "face detection & face recognition using open computer vision classifires".august 4, 2017.

[2]. Mamata s.kalas." Real time face detection and tracking using OpenCV". 2014.

[3]. Neetu Saini, Sukhwinder Kaur, Hari Singh A Review: Face Detection Methods and Algorithms, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 www.ijert.org Vol. 2 Issue 6, June – 2013.

[4]. ABolle, R., (2002). Face Recognition and Its Application, Biometric Solutions for Authentication in an E-World (pp 83-97). Retrieved at http://andrewsenior.com/papers/SeniorB02FaceChap.pdf on February 6, 2017.

[5]. Tomasz Sosnowski, Grzegorz Bieszczad, Henryk Madura, Mariusz Kastek,"The calibration stands for thermal camera module with cooled infrared focal plane array",2010.

[6]. Bieszczad, G., Sosnowski, T., Orzanowski, T., Kastek, M., „Low voltage differential signal interface for thermal vision camera", Proceedings of the 16th International Conference - Mixed Design of Integrated Circuits and Systems, MIXDES 2009, Pages 648-651(2009)

[7]. Matt Richardson and Shawn Wallace," Getting Started with Raspberry Pi", Frist edition on 2012,second edition on 2014, third edition on 2016.

[8]. Michael Beyeler," OpenCV with Python Blueprints",2015.

[9]. QiangHu , Lei Ma and Jianjun Zhao," Deep Graph: A PyCharm Tool for Visualizing and Understanding Deep Learning Models", 2018 25th Asia-Pacific Software Engineering Conference (APSEC).

[10]. Alexander Sergeev, Mike Del Balso," fast and easy distributed deep learning in TensorFlow", 21 feb 2018.

[11]. Hao Dong, AkaraSupratak, Luo Mai, Fangde Liu, Axel Oehmichen, Simiao Yu, Yike Guo," Tensor Layer: A Versatile Library for Efficient Deep Learning Development", 3 august 2018.

[12].Chris Albon," Machine Learning with Python Cookbook Practical Solutions from Preprocessing to Deep Learning", 2018 Chris Albon