NET GUARD

### A Major Project

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

By

Shashwat Chauhan (Registration No.:10320210080) Deepanshu Dhama (Registration No.:10320210088)

Sunny Dahiya (Registration No.:10320210063)

### Submitted to Dr. Arvind Kumar

(Associate Professor)



### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING FACULTY OF ENGINEERING

SRM UNIVERSITY DELHI-NCR

### Plot No.39, Rajiv Gandhi Education City, Sonepat, Haryana 131029 MAY 2024



**(Established under Haryana Private University Act, 2006 as amended by Act no. 8 of 2013)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**MAJOR PROJECT REPORT**

**Course Code :** CS 4114

**Program :** B. Tech. CSE-CORE

**Course Title :** Major Project

**Project Title :** Net Guard

**Semester & Year :** VIII & IV

**Name :** Shashwat Chauhan, Deepanshu Dhama,

Sunny Dahiya

**Registration No. :** 10320210080, 10320210088, 10320210063

**Supervisor :** Dr. Arvind Kumar

***SRM University, Delhi-NCR, Sonepat***



***Projects Committee***

***Computer Science and Engineering Department***

**APPROVAL FOR BINDING OF PROJECT REPORT MAJOR PROJECT (CS 4114)**

**B.Tech. Computer Science and Engineering**

**GROUP NO: 11 SESSION: 2020 / 2024**

***To be filled in by student***

Student name(s)

Signature

: SHASHWAT

CHAUHAN,

:

DEEPANSHU DHAMA, SUNNY DAHIYA

Registration Number : 10320210080, 10320210088, 10320210063

Project title : NET GUARD

***To be filled in by supervisor\****

**This is to certify that the project submitted and presented by the above-mentioned student(s) is**

□ **COMPLETE & ACCEPTED** □ **INCOMPLETE** □ **NOT ACCEPTED**

**and the draft of graduation project report has been corrected from all content flaws, typing errors and language mistakes.**

Supervisor name and Signature : Examiner ‘s Name and Signature : HOD’s Signature :

Date:

Date:

Date:

**\*NOTE**: The student may proceed with Ring binding ONLY after this section has been completed. The student

shall submit three **copies** of Ring bound draft report for Minor/Major Project to the panel of examiners on the day of final presentation.

***For Departmental Office use only***

Date received : Signatures :



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## CERTIFICATE

This is to certify that Shashwat Chauhan, Deepanshu Dhama and Sunny Dahiya with registration no. 10320210080, 10320210088 and 10320210063 respectively of the Computer Science and Engineering Department have satisfactorily completed teamwork on Major Project.

(Subject Code: CS4114)

**Project Name:** Net Guard

**Year and Semester:** 4th Year & 8th Semester **Name of Supervisor:** Dr. Arvind Kumar

**SUPERVISOR PROJECT CO-ORDINATOR HOD(CSE)**



**(Established under Haryana Private University Act, 2006 as amended by Act no. 8 of 2013)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CANDIDATES’ DECLARATION**

We hereby declare that the work completed & presented in the project titled “**Net Guard**” for the award for the degree of Bachelor of Technology in Computer Science & Engineering and submitted in the Department of Computer Science & Engineering of SRM University, Delhi-NCR, Sonepat, Haryana (India) is an original record of our work carried out under the supervision of **Dr. Arvind Kumar** as major project (CS4114) in 8th semester during the academic year 2023-24. The content presented in this project has not been submitted for the award of any other degree of this or any other educational institution.

**Shashwat Chauhan**

**(10320210080)**

**Deepanshu Dhama**

**(10320210088)**

**Sunny Dahiya (10320210063)**

##### ACKNOWLEDGEMENT

We would like to thank and acknowledge our project supervisor “**Dr. Arvind Kumar**”, who helped & guided us in making this project complete and successful.

We would also like to express gratitude towards SRM University for giving us this opportunity to do a project on “**Net Guard**”. Without their support and suggestions, would not have been completed. Your valuable guidance and suggestions helped us in various phases of the completion of the project. We will always thankful to you in this regard.

We would also like to thank our college professors for providing additional assistance and valuable support while creating this project. Lastly we would like to thank our colleagues and peers who helped and supported us throughout this project.

We ensure that this project was completed by us only and is not copied.

Shashwat Chauhan (10320210080)

Deepanshu Dhama (10320210088)

Sunny Dahiya (10320210063)

4th Year / 8th Semester

##### ABSTRACT

On the highway rapidly growing and progressive Technology In development for over years: face recognition and detection system become a legacy for many applications, from security monitoring to human-machine system interaction. The system uses a specially designed deep neural network architecture to improve the detection of faces and objects in a scene. The model is trained on multiple datasets to ensure performance across different environments, platforms, lighting, and different faces. . This research contributes to ongoing efforts to improve computer vision capabilities in real-world environments and situations. In real-world situations, including crowded areas and difficult lighting, it has also attempted to demonstrate the power of the desired method. Learning is widely used to improve the capabilities of models, allowing them to adapt to new situations and understand from previously recorded information. Optimization techniques such as model quantization and parallelization are used to achieve faster results. The system's many features make it suitable for use in security, augmented reality (AR), human-computer interaction, and many other fields.

##### LIST OF FIGURES

S.No FIGURES

1. Fig. 1: Research Flow
2. Fig. 2: Facial Conversion
3. Fig. 3: Detection of Faces
4. Fig. 4: Gray Scale Image
5. Fig. 5: Gray Scale Depth
6. Fig. 6: Hog Version
7. Fig. 7: Twin Brother Photos
8. Fig. 8: Marking Points
9. Fig. 9: Detecting Faces
10. Fig. 10: Work Flow of Model
11. Fig. 11: Single Face Detection
12. Fig. 12: Two Faces Detection
13. Fig. 13: Unknown Face Detection
14. Fig. 14: Detecting Faces

Page No.

13

14

14

15

15

16

17

17

18

19

21

21

22

22

## TABLE OF CONTENTS

S.No CONTENTS

1. APPROVAL
2. CERTIFICATE
3. DECLARATION
4. ACKNOWLEDGEMENT
5. ABSTRACT
6. LIST OF FIGURES
7. INTRODUCTION
8. LITERATURE SURVEY
9. METHODOLOGY
   1. DATASETS
   2. IMPLEMENTING ALGORITHM
   3. TOOLS AND TECHNIQUES
   4. HARDWARE & SOFTWARE REQUIREMENTS
   5. EVALUATE
10. RESULTS AND DISCUSSIONS
11. CONCLUSION AND FUTURE SCOPE
12. REFERENCES

Page No.

2

3

4

5

6

7

9

11

13

14

14

20

20

21

23

24

25

##### INTRODUCTION

Digital imaging technology have developed rapidly in the last few years. The increase in computing power has led to significant advances in the field of computer vision. The ability to quickly and accurately recognize faces and objects in a dynamic, immersive environment holds great promise for improving security, performance, and user experience. Among these, facial recognition and real-time object detection have gained importance in many fields, from surveillance and security to interactive technology.

The main purpose of real-time face and object detection is to detect and identify faces and objects in the visual field. The technology has applications in many areas, including human- computer interaction, augmented R, security monitoring and self-management. The challenge is not only to achieve high accuracy, but also to speed up processing time, which is an important aspect of applications that require immediate and effective response.

The development of face and object detection has changed from the management of traditional methods based on manual operations to deep learning, such as network neural networks (CNN), which have been proven to be especially good in Hierarchical terms.

The representation is learned from raw image data and allows the elimination of difficult features important for accurate detection. Integrating deep neural networks into real-time systems requires solving computational problems to ensure zero accuracy.

Tensor Flow is an open-source tool which is used for machine learning and artificial intelligence. It uses a deep learning concept which is mainly used in the interface of the application. It uses Tensor Flow to generate a CNN algorithm to identify hand gestures. Keras is used as the interface to access the Tensor Flow library. A user-friendly interface for training and building the translator and implementing CNN is done using Keras. It is also responsible for data abstraction as to reduce major complexities. It helps with the concepts of deep learning and necessary tools.

As the demand for vision processing continues to increase, so does the development of real-time dynamic face detection and object detection capabilities. The findings not only contribute to the educational debate in computer vision, but also have implications for companies seeking to exploit the potential of data processing: Fast, accurate.

##### LITERATURE SURVEY

The story of real-time face and object discovery is evolving thanks to advances in deep learning, computer vision, and the increasing demand for paper. This literature review provides an overview of important research and applications conducted in the field.

[1] Evolution of traditional methods and deep learning: Initial face and object detection was based on handcrafted and traditional machine learning methods. However, the limitations of these methods, especially in handling complex situations and changing lighting conditions, have led to the transition to deep learning.

[2] Deep Learning Architecture: The era of convolutional neural networks (CNN) has revolutionized face and object detection. The seminal work of Krizhevsky et al. and AlexNet demonstrate the effectiveness of deep neural networks in image classification. Later designs, including (GoogLeNet, ResNet, and VGGNet), were modified and extended for object search tasks introduce the concept of regional communication to improve accuracy and efficiency.

[3] Transfer Learning: Transfer learning has become an important technique for training face recognition and object detection models. Models pre-trained on large datasets like ImageNet can be a powerful and useful starting point. Improving these models of the project plan helps to adapt to some features and improve details in real situations.

[4] Real-time optimization technology: Real-time implementation needs to be done better than the design model. Model pruning and quantization are powerful tools (like GPUs) that have been invented to simplify the inference process. These optimization points strike a balance between complexity and efficiency of the operation.

Model pruning, and Quantization are efficient hardware accelerators (e.g., GPUs) have been explored to streamline inference processes.

These optimizations aim to strike a balance between complexity and modular computational efficiency.

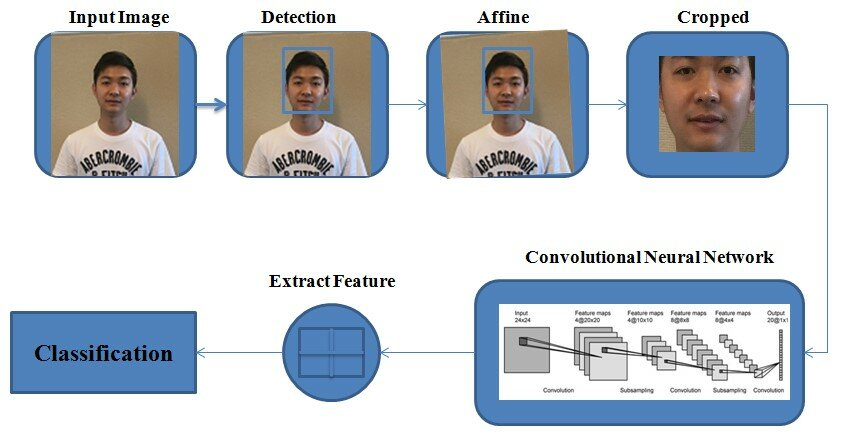
[5] Challenges and future directions: Despite significant progress, challenges remain, such as addressing competition, changes in lighting and illumination, and ensuring privacy in face detection applications. More futuristic and advanced research trends towards exploring 3D products are changing and addressing ethics when using time sensing technologies.

Real-time facial information and object detection reflect fast and rapid changes. The combination of adaptive learning, deep learning, and optimization has created an accurate and efficient system with applications ranging from security to reality. Continuous research continues to solve problems and push the boundaries of what can be achieved in real-time visualization.

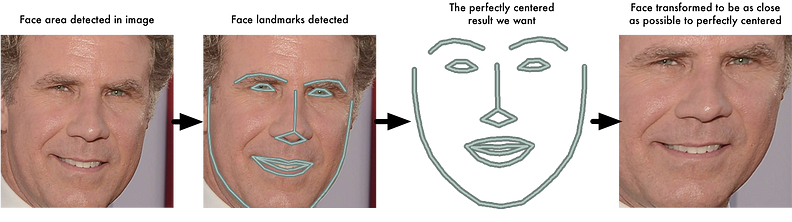
##### METHODOLOGY

Real-time face and object detection is one of the most popular tasks involving computer vision. The ability to find and identify video streams or images of faces or objects in the image. Here's how to create a face and discover the product at that moment:

The research is given in form of Figure 1:



**Fig 1.** Research Flow



**Fig 2.** Facial Conversion

### **DATASETS**

The data set contains 900 images. The dataset is from Jitendra Shah.

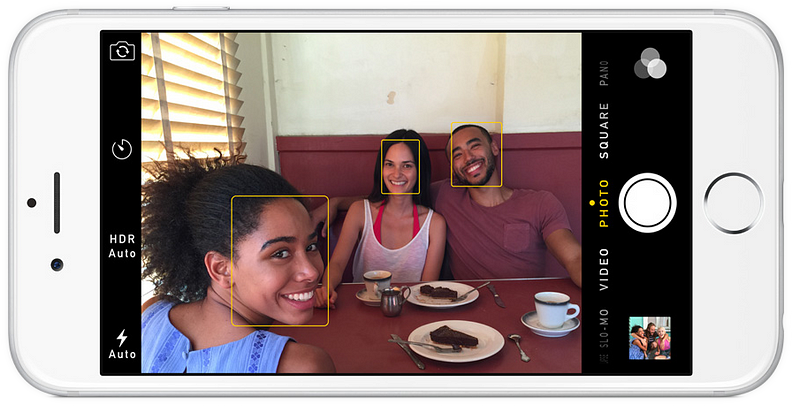
The dataset of 900 images of face were trained on CNN Algorithm. The dataset contains 36 classes for which each file contains 25 images.

The testing data also included 36 classes and contains high quality images for human and computer interaction making it more easily visible and easy to understand.

### **IMPLEMENTING ALGORITHM**

### **Step 1: Discovering all the Faces**

The initial phase of our system is face detection. Because it requires to find multiple faces in the picture before trying to distinguish them. If you use a phone camera, you've definitely seen face detection.



**Fig 3.** Detection of Faces

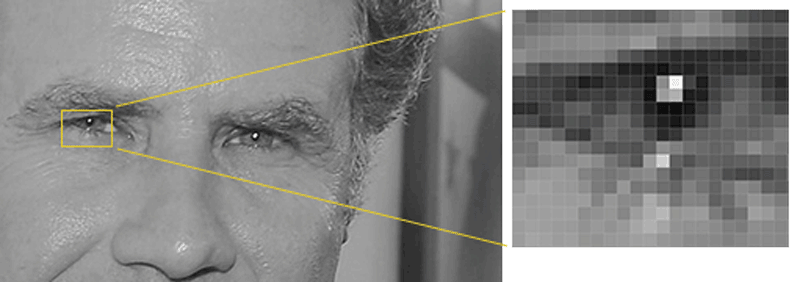
Face spotting is a new feature for cameras. When a camera of phone pick out numerous faces it ensures that all the faces present are in focus before taking the picture. But we use this for a different operation — finding areas within image we want to pass on to the second step for are modular architecture to detect faces in any image, we’ll start by

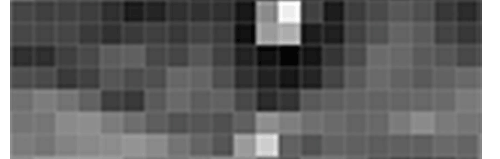
converting our image [b/w] - black / white because we actually don’t require color data to find faces :-



**Fig 4.** Gray Scale Image

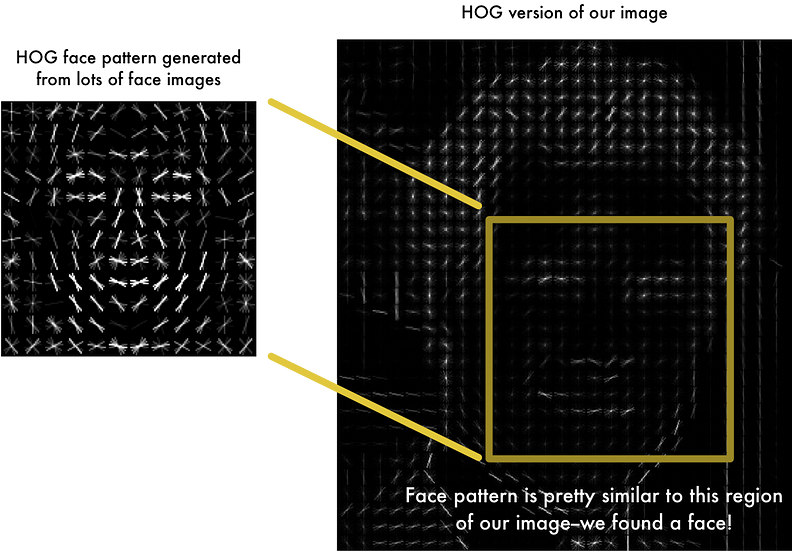
Then we overlook at every one of pixel within our image at a time. For any given pixel there we try to look at the pixel that is directly surrounding that pixel to some extent.





**Fig 5.** Gray Scale Depth

Main aim is to work out how dark the pixel currently is compared to the pixel directly neighboring it. Then we tend to draw an arrow in the way of which the image is slightly darker.



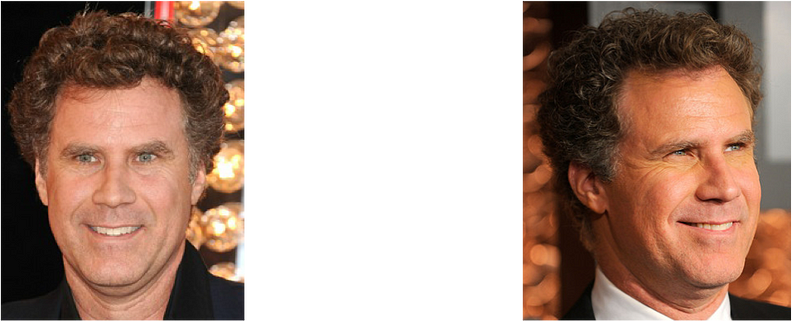
**Fig 6.** Hog Version

As shown the one pixel and the pixels surrounding, the image is becoming darker towards upper right side of the spectrum. The conclusion is we turn sample image in simple architecture that encapsulates the basic architecture of any given facial structure in a 2-D way. The main image is developed in a [representational HOG that collects the main features of image despite of the images brightness and saturated orientation] To detect the face in any HOG, all we have to do is find the part of our image that looks exact to a renowned HOG pattern that is extracted from a group of older training datasets.

**Step 2: Posture and experiment with angular faces**

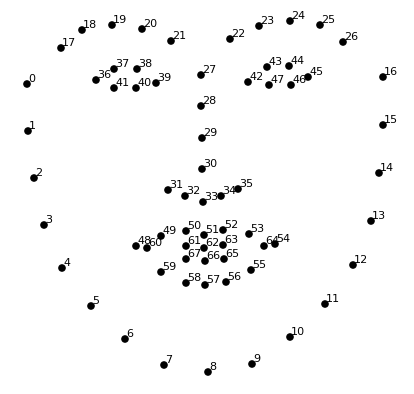
We have already isolated the faces in our image but now arises a problem where we have to deal with the faces turned in different directions looks dissimilar to computer system:

We see the 2 figures as humans, still the computer will look at the image totally dissimilar individuals.



**Fig 7.** Twin Brothers Photo

To overcome the problem, we try to distorting each image so that the lips as well the eyes are every time in center of the picture. Which makes it very easier for us to work off from in the next step on how to operate we will be using an algorithm called "Facial Cue Prediction". The concept is that we see sixty eight particular points (areas) located in each face, the bottom of the chin, the corner of both eyes, inside of each eye and furthermore. Then we train a ML algorithm to find the sixty eight unique features in each face:



**Fig 8.** Marking Points

The sixty eight unique points are located on any face. Here is the result below of our operation:

Now it doesn’t even matter how the face is facing, centrality of the mouth, eyes are always in the exact position of the image making and further steps more efficient and totally optimal.

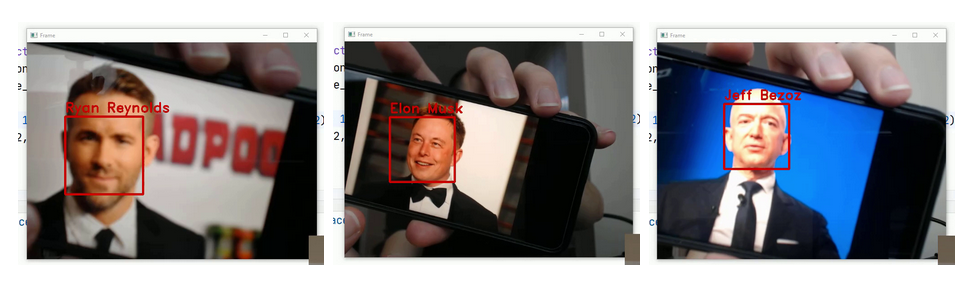
**Step 3: Enciphered Facial Expressions**

Easiest way in recognizing a face is to compare the face which is unknown. Pictures of all the people tagged. When we see that the previously marked face bears resemblance to our unknown face, it must be human. The answer is to develop the deep convolutional neural networks on trained datasets. However, the similar images like last time, we will coach it to create hundred and twenty eight parameters for each face instead of 68 preliminary tasks the tutoring phase works by looking at three face image at a time: 1. Upload a graphical image of a known person in the dataset 2. Upload different graphical image of the same person 3. Upload a unique graphical image of an unknown person Real-time face and object detection is an inveterate task that involves using computer vision Hteecrhen'siques to identify and locate faces or objects in a video stream or series of images. A general methodology for real-time face and object detection:

Further steps are:

1. Optimization for Real-Time: Optimize the model for real-time Environment, considering factors such as speed and accuracy trade-offs. Consider model quantization, pruning, or using lighter architectures for faster inference.

2. Integration with Frameworks: Integrate the trained model with a deep learning framework suitable real-time applications, such as ONNX.

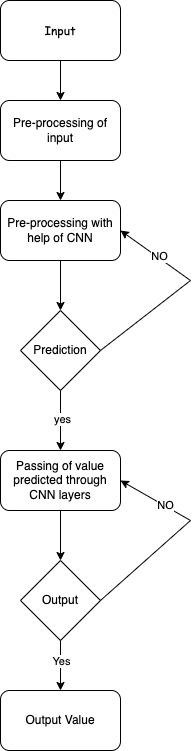


**Fig 9.** Detecting faces

3. Hardware Acceleration: Utilize hardware acceleration techniques (e.g., GPU, FPGA) to speed up the tangential process, ensuring real-time capabilities.

4. Post-processing: Use post-processing techniques to optimize and filter test results. Non-maximum suppression (NMS) is used to remove unnecessary boxes.

5. Testing and Evaluation: Test the model of a separate test to evaluate its performance. Use metric scores such as precision, F1, and recall to measure the model's accuracy.



**Fig 10.**Workflow of Model

* 1. **TOOLS AND TECHNIQUES**

It uses python and Pycham Software and the library we used are numpy and some data from kagle.

* 1. **HARDWARE & SOFTWARE REQUIREMENTS**

**Hardware Requirements:**

Camera: You must have a camera which is able to take high-quality pictures. It could either be an in-built webcam or a connected external one. High resolution cameras might be more accurate.

Processor (CPU): The CPU processing power is very important for real-time image processing and machine learning inference. A modern multi-core processor that supports parallel processing would be recommended for the highest performance.

Memory (RAM): Adequate RAM is essential for loading and manipulating images, running machine learning models, and managing the application’s interface. At least 8GB of RAM is suggested, but more may be desired to keep it functioning well.

Storage: Although the app itself might not require much storage space, having enough disk space can be helpful when saving training data, model weights or any other resources necessary in the app development process.

Audio Input (Optional): If there are any audio input features present in this software such as speech recognition you will need a microphone or audio input device.

**Software Requirements:**

Operating System: The project can run on all the major Operating Systems such as Windows, macOS or Linux.

Python: For running this project, you should have python in your computer. Ensure that your computer has Python version which is compatible (for example: Python 3.x).

Python Libraries: You can use pip as your package manager to install these necessary python libraries. These should consist of NumPy, OpenCV, Tkinter, Keras, Pillow, PyAudio, Speech Recognition and TensorFlow mentioned in the description plus any other dependencies.

CUDA (Optional): If you want GPU acceleration while training or making predictions using TensorFlow then CUDA drivers and libraries for your GPU may be required as well.

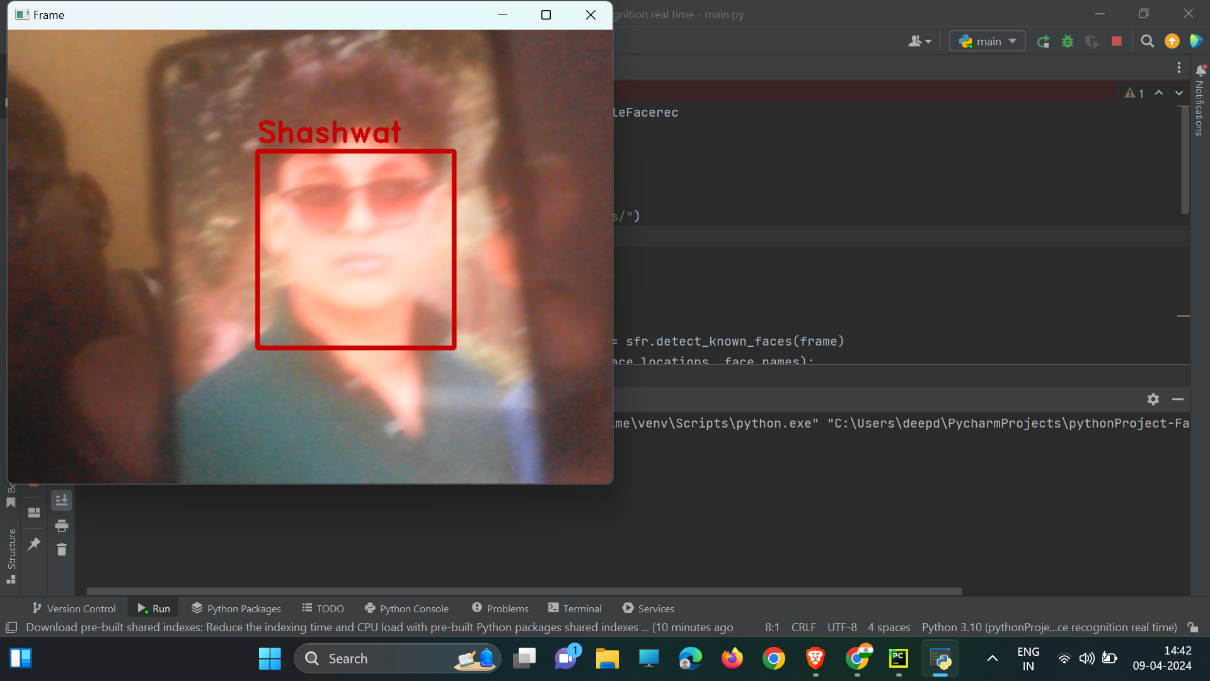
Development Environment: A text editor or an integrated development environment (IDE) where python code can be written and executed is a necessity. Some popular ones are PyCharm, Visual Studio Code and Jupyter Notebook.

Additional Dependencies: Depending on what is needed by the particular application and environment you may need additional system libraries or packages to install.

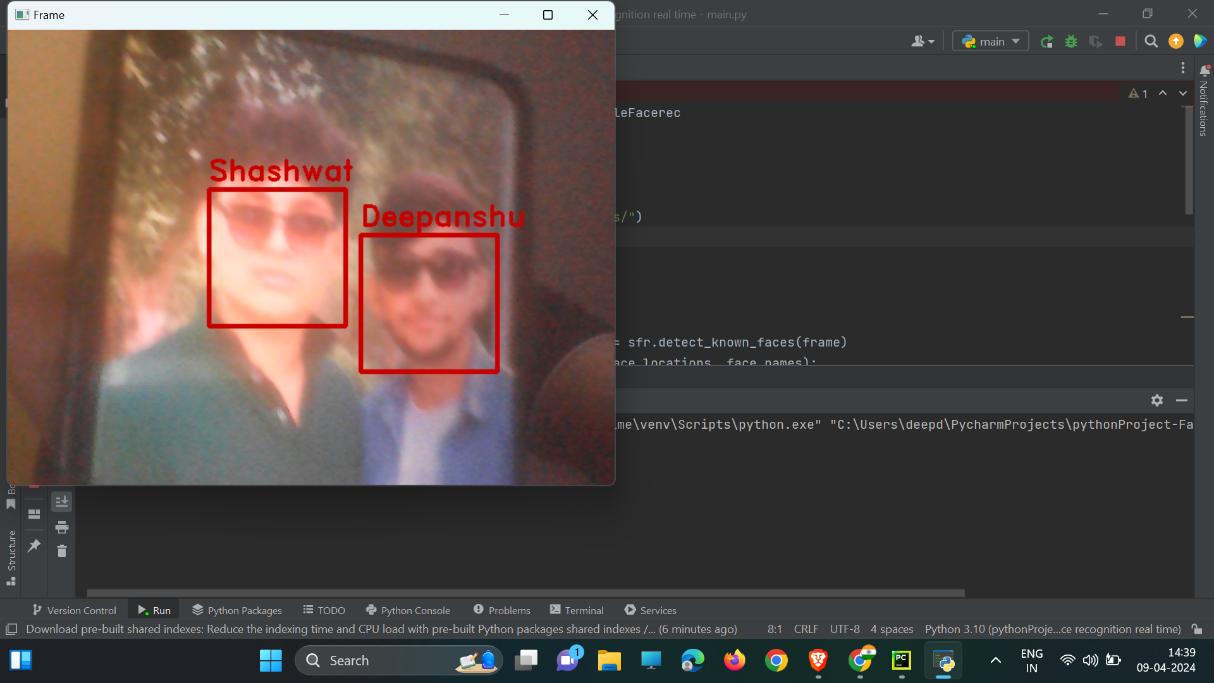
You will be ready to run when you ensure that your hardware along with software meets these requirements.

* 1. **EVALUATE**

The performance and accuracy of the project was measured and epoched. The accuracy of data was measured using a confusion matrix which checks for how to distinguish between faces and how similar types of face are showing the results. So to check how accurately it identifies different faces, the confusion matrix is used. The data used for the confusion matrix was a dataset from Vignesh Rajan Buddy for training the data. The Results for corresponding faces were shown in form in the images below (Fig 3&4):

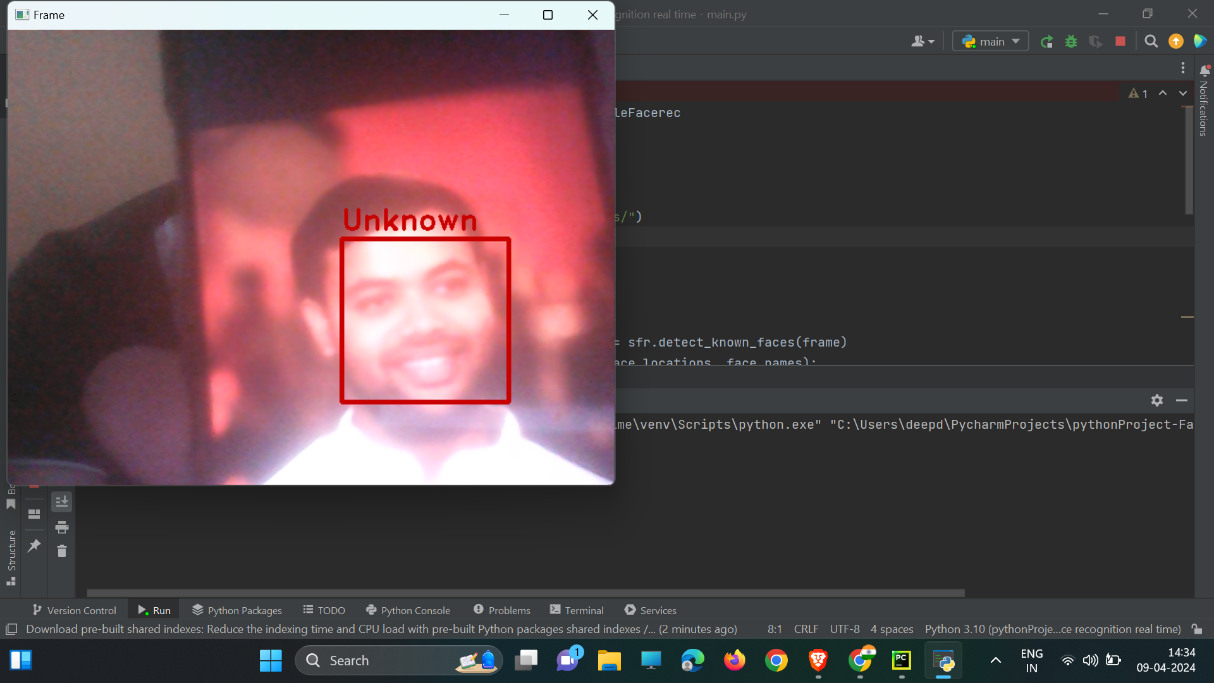


**Fig 11.** Single face detection

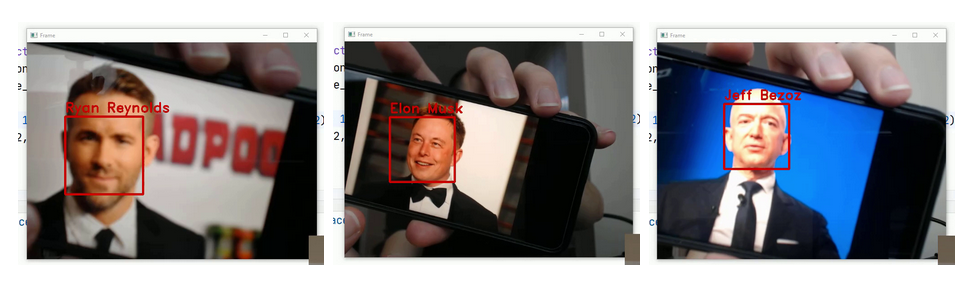


**Fig 12.** Two Face detection

For Recognition, we took faces as a form of input with help of a Data and displayed the corresponding face through the UI. We checked for the images in the dataset in the form of files for each corresponding face and displayed it against each face in the result box.



**Fig 13.** Unknown Face Detection



**Fig 14.** Detecting Faces

##### RESULTS & DISCUSSION

Accuracy measurement: Face detection performance ensures accurate face detection (sensitivity), capturing all real faces (recall), and a mixture of real faces. - Alpha Error Rate: Evaluate numbers of false positives, i.e. when the system is not accurate or there is no face before determining the location.

Speed and Efficiency: Frames per second (FPS): A measure of face detection speed, especially if designed for real-time use. Higher FPS means better real-time and response time, and earlier and stronger when using fewer programs for a better working environment.

- Frames Per Second (FPS): Measure the speed of the face detection system, especially if it is designed for real-time applications. A higher FPS indicates better real-time and a advanced and robust response time while consuming little resources for better environmental performance.

Scale: Detection Across Scales: The system handles faces at different scales, including small large faces and average modeled faces

- Robustness to Variations: Assess the system's robustness to variations in lighting conditions, poses, facial expressions and brightness.

##### CONCLUSION & FUTURE WORK

###### CONCLUSION

In this study, we take the steps of using our dataset, creating a new model from each layer of the Convolutional Neural Network (CNN) using the volume method. That's it, train our CNN model and create a GUI for our application.

As a result, our application can instantly detect faces with 96.5% accuracy. However, detecting and accessing the face takes a few seconds.

Therefore, for future research, we propose to add labels to images, find ways to speed up the process of identifying objects, thereby reducing the waiting time, and improving the accuracy of the model by increasing the methods in the CNN model bigger. Using a non-CNN approach may also be considered as it will give better results.

Tool.

###### FUTURE SCOPE

This project could be improved by adding automatic car opening system from Facial Recognition.

##### REFERENCES

[1] Kortli, Y., Jridi, M., Al Falou, A., & Atri, M. (2020). Face recognition systems: A survey. Sensors, 20(2), 342.

[2] Parmar, Divyarajsinh N., and Brijesh B. Mehta. "Face recognition methods & applications." arXiv preprint arXiv:1403.0485 (2014).

[3] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. ACM computing surveys (CSUR), 35(4), 399-458.

[4] Goudail, F., Lange, E., Iwamoto, T., Kyuma, K., & Otsu, N. (1996). Face recognition system using local autocorrelations and multiscale integration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(10), 1024-1028.

[5] Kamgar-Parsi, B., Lawson, W., & Kamgar-Parsi, B. (2011). Toward development of a face recognition system for watchlist surveillance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(10), 1925-1937