FACE RECOGNITION UNLOCKING SYSTEM FOR VEHICLES

A report submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Technology

in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

by

S.LAVAN KARTHIK	**2011CS010266**
S.NITHIN SAI	**2011CS010404**
P.VENNELA REDDY	**2011CS010241**
S.SAI KUMAR GOUD	**2011CS010185**

Under the esteemed guidance of

Mrs. S. Sowmya

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

School of Engineering

MALLA REDDY UNIVERSITY

Maisammaguda, Dulapally, Hyderabad, Telangana 500 010

2024

FACE RECOGNITION UNLOCKING SYSTEM FOR VEHICLES

A report submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Technology

In

COMPUTER SCIENCE & ENGINEERING

BY

S LAVAN KARTHIK	2011CS010266
S NITHIN SAI	2011CS010404
P VENNELA REDDY	2011CS010241
N SAIKUMAR GOUD	2011CS010185

Under the guidance of

Mrs. S. Sowmya

Assistant Professor

Department of COMPUTER SCIENCE AND ENGINNERING

School of Engineering

MALLA REDDY UNIVERSITY

Maisammaguda, Dulapally, Hyderabad, Telangana 500100

2024

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project report entitled **"FACE RECOGNITION UNLOCKING SYSTEM FOR VEHICLES"**, submitted by **S Lavan Karthik (2011cs010266), S Nithin Sai (2011cs010404), P Vennela Reddy(2011cs010241), N Sai Kumar Goud(2011cs010260),** in fulfilment of the requirement for the awardof degree of **BACHELOR OF TECHNOLOGY in Computer Science & Engineering** is recorded of bonafide work carried out by them. The result embodied in this report havebeen verified and found satisfactory.

Signature of Guide
Mrs.S.Sowmya
Assistant Professor
Department of CSE
School of Engineering
Malla Reddy University

Head of the Department
Dr. Meeravali Shaik
Head of the Department
Department of CSE
School of Engineering
Malla Reddy University

External Examiner

DECLARATION

We, the students of '**Bachelor of Technology in Department of Computer Science and Engineering'**, session: 2020 - 2024**, School of Engineering, Malla Reddy University, Maisammaguda, Kompally, Medchal – Malkajgiri District,Hyderbad – 500 100** hereby declare that the work presented in this Project Work entitled **FACE RECOGNITION UNLOCKING SYSTEM FOR VEHICLES** is the outcome of our own bonafide work and iscorrect to the best of our knowledge and this work has been undertaken taking careof Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

Place: Hyderabad

Date:

S Lavan Karthik	**2011cs010266**
S Nithin Sai	**2011cs010404**
P Vennela Reddy	**2011cs010241**
N Sai Kumar Goud	**2011cs010185**

ACKNOWLEDGEMENT

We extend our sincere gratitude to all those who have contributed to the completion of this project report. Firstly, We would like to extend our gratitude to **Dr. V. S. K Reddy**, Vice-Chancellor, for his visionary leadership and unwavering commitment to academic excellence.

We would also like to express my deepest appreciation to our project guide **Mrs S.Sowmya**, Professor, whose invaluable guidance, insightful feedback, and unwavering support have been instrumental throughout the course of this project for successful outcomes.

We are also grateful to **Dr. Meeravali Shaik**, Head of the Department of CSE, for providing us with the necessary resources and facilities to carry out this project. We would like to thank **Dr. Kasa Ravindra**, Dean, School of Engineering, for his encouragement and support throughout my academic pursuit.

My heartfelt thanks also go to **Dr. Harikrishna Kamatham**, Associate Dean School

of Engineering for his guidance and encouragement.

We are deeply indebted to all of them for their support, encouragement, and guidance, with

out which this project would not have been possible.

S Lavan Karthik	2011cs010266
S Nithin Sai	2011cs010404
P Vennela Reddy	2011cs010241
N Sai Kumar Goud	2011cs010185

ABSTRACT

A Vehicle with a key is a problem which we are facing in our day to day life. According to today era every vehicle is operated with key. So our intension is to create a Smart Unlock System Based on Facial Recognition Technology. In today's era of automation and smart devices, there is crucial need to alter the security measures of system as privacy and security are notable issues in the information system. In traditional system many of the vehicle doors are having mechanical lock which was restricted on the number of keys. We will propose Smart Vehicle-Door Unlock System based on Face Recognition to enhance the security. In this system camera sensor is used to capture the face and image matching algorithm will be used to detect the authenticated faces. Only the person whose face is matched can be able to unlock the door. This system will not only enhance the security but also help the system without physical key.

LIST OF FIGURES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

In recent years, advancements in technology have revolutionized the automotive industry, leading to the development of innovative features aimed at enhancing both convenience and security for vehicle owners. Among these advancements is the integration of biometric authentication systems, such as face recognition, into automotive security systems. One such application is the Face Detection Unlocking System For Vehicles Using Machine Learning. This project aims to leverage the capabilities of face recognition technology, utilizing the Hash cascade Algorithm, to automatically control the locking and unlocking of car doors based on the identity of the individual approaching the vehicle.

By harnessing the power of computer vision and machine learning algorithms, this system offers a seamless and secure solution to access control for automobiles. Traditional car locking systems typically rely on physical keys or remote key fobs, which may be prone to theft, loss, or unauthorized duplication. In contrast, a face recognition-based approach offers a more sophisticated and robust method of authentication, eliminating the need for physical keys while enhancing security and user convenience.

The Haar cascade Algorithm, a variant of the Haar Cascade Algorithm, is employed for efficient face detection within the captured video feed from the vehicle's onboard camera system. This algorithm is renowned for its speed and accuracy in detecting objects, making it well-suited for real-time applications such as automatic door locking systems. Once a face is detected, the system utilizes advanced facial recognition techniques to match the detected face with pre-registered profiles stored in the system's database. Upon successful verification of the individual's identity, the system triggers the automatic unlocking of the vehicle's doors, allowing seamless entry for authorized users.

This project encompasses the fusion of cutting-edge technologies including computer vision, machine learning, and biometric authentication to create a robust and intelligent Face Detection Unlocking System for Vehicles. By employing face recognition powered by the Haar cascade Algorithm, this system offers enhanced security, convenience, and user experience, marking a significant advancement in automotive access control systems.

In automotive innovation, the integration of biometric authentication systems has emerged as a pioneering solution for enhancing vehicle security and convenience. One such innovation is the Face Detection Unlocking System for Vehicles Traditional methods of vehicle access, such as physical keys or remote key fobs, have limitations in terms of security and user experience. These methods can be susceptible to theft, loss, or unauthorized duplication, prompting the need for more sophisticated and reliable alternatives.

1.2 Overview

The Face Detection Unlocking System for Vehicles addresses these concerns by leveraging cutting-edge facial recognition technology to authenticate individuals seeking access to the vehicle. By employing sophisticated algorithms and onboard cameras, this system accurately identifies authorized users based on their facial features. The concept is simple yet powerful: as an authorized individual approaches the vehicle, the onboard camera captures their face and compares it against a database of pre-registered faces. Upon successful recognition, the system automatically unlocks the car doors, providing seamless entry without the need for physical keys or manual intervention. This innovative system not only enhances security by eliminating the risk of unauthorized access but also improves user convenience by streamlining the entry process. Additionally, it offers a futuristic and sophisticated alternative to traditional locking mechanisms, aligning with the evolving expectations of modern vehicle owners. In this project, we delve into the implementation of the Face Detection Unlocking System for Vehicles, exploring the underlying technologies, design considerations, and practical applications. By harnessing the power of facial recognition, this system represents a significant step forward in the intersection of automotive engineering and biometric authentication, ushering in a new era of intelligent vehicle access control.

Security Concerns with Traditional Locking Systems: Traditional car locking systems, such as physical keys or remote key fobs, are susceptible to theft, loss, or unauthorized duplication, compromising the security of vehicles.

1. Inconvenience and Accessibility Issues: Users often face inconvenience and accessibility issues associated with traditional locking systems, such as the need to carry and locate physical keys or key fobs.

2. Risk of Unauthorized Access : Unauthorized individuals may gain access to vehicles through stolen or lost keys, posing a security risk to vehicle owners and their belongings.

Need for Enhanced Security Measures There is a growing demand for enhanced security measures in vehicles to mitigate the risk of theft and unauthorized access. Advancements in Biometric Authentication Advancements in biometric authentication technologies, such as facial recognition, offer a promising solution to address security concerns and enhance the user experience in vehicle access control systems.

3. Facial Recognition in Automotive Security: There is a need to explore the integration of facial recognition technology into automotive security systems to provide a secure and convenient method of vehicle access.

4. Accuracy and Reliability of Face Recognition: The accuracy and reliability of facial recognition algorithms in real-world scenarios, particularly in varying lighting conditions and with diverse facial features, need to be evaluated to ensure effective implementation in automatic car door lock systems.

CHAPTER 2
LITERTAURE SURVEY

Face Recognition is the ability to detect and recognize a person by their facial characteristics. Face is a multidimensional and hence requires a lot of mathematical computations. Face recognition system is very essential and important for providing security, mug shot matching, law enforcement applications, user verification, user access control, etc and is mostly used for recognition for various applications. These all applications require an efficient Face recognition system. There are many methods that are already proposed and have low recognition capability, high false alarm rate. Hence the major task of the research is to develop face recognition system with improved accuracy and improved recognition time of an face recognition system. This paper proposes a hybrid face recognition algorithm by combining two face recognition techniques by integrating (PCA) principle Component Analysis, (LDA) Linear Discriminant Analysis. Jacobi method is used to compute Eigenvector that are necessary for PCA and LDA algorithms. Face Recognition system will be implemented on Embedded system based Raspberry pi 3 board. All over the world Drowsiness has been the significant cause of horrible accidents which is causing deaths and fatalities injuries. Day by Day fatal injuries numbers are increasing globally. From the past many years, researchers have concluded drivers with a lack of sleep and more tiredness which causes drowsiness of the driver. this paper shows a new experimental model is designed for detecting drowsiness of driver is presented to reduce accidents caused by this problem which increases transport safety. In this work, two ways are used to detect the drowsiness of a person effectively. First is captured and eye retina detection and facial feature extraction are done and blinking values are calculated then threshold values are set. Secondly, the Aurdino module is used which is integrated with elastomeric sensors for real-time calculation of driver hand pressure on the car steering wheel and the threshold value is set. The result from both methods is taken as input for taking the final decision and alerting the driver.

It is a foregone conclusion that property crimes will hit ten million. Among such, within the steal list, the vehicle is flat-top and sometimes controls entire components of the earth. There have been some recent technical advances and new strategies are being developed to overcome this downside. The ways involved in the detection of vehicle theft are noted for interference by some or everything, as well as shields. This research helps to prevent the vehicle stealing from third parties with the help of RFID card and authorized key. Here, RFID reader connected to the car

and if anyone enter into car and they requires the card authorization. Keypads were connected to the engine and the authorized person can only enter into the car by pressing passwords. If the password is incorrect, buzzer will produce noise continuously. The proposed system gives the regular updates to the car owners regarding the location of car by using GPS technology. Face recognition have gained a great deal of popularity because of the wide range of applications such as in entertainment, smart cards, information security, law enforcement, and surveillance. It is a relevant subject in pattern recognition, computer vision, and image processing. Two major methods are used for features extraction, which can be classified into appearance-based and Model-based methods. Appearance-based methods use global representations to identify a face. Model-based face methods aim to construct a model of the human face that capture facial variations. Image similarity is the distance between the vectors of two images. This paper contains Four sections. The first section discusses face recognition applications with examples. The second section discuss the common feature face recognition methods. The third section discuss distance measurement classifiers. The fourth section discuss different face recognition databases.

CHAPTER 3

SYSTEM ANALYSIS

System requirements are the functionality that is needed by a system in order to satisfy the customer's requirements. System requirements are abroad and a narrow subject that could be implemented to many items. The requirements document allows the project team to have a clear picture of what the software solution must do before selecting a vendor. Without an optimized set of future state requirements, the project team has no effective basis to choose The best system for your organization.

3.1 Existing System:

The existing car unlocking system relies primarily on conventional methods such as physical keys or remote key fobs for access control. These methods, while widely used, are prone to security vulnerabilities, including theft, loss, or unauthorized duplication of keys. Moreover, manual authentication processes can be time-consuming and inconvenient for users, leading to potential inefficiencies and security risks.

Disadvantages of Existing System:

- But in these methods users have to wait for long time in making a queue at time they enter the office. Many biometric systems are available but the key authentications are same is all the techniques.
- Every biometric system consists of enrolment process in which unique features of a person is stored in the database and then there are processes of identification and verification.

3.2 Proposed System:

The proposed car unlocking system introduces approach based on facial recognition technology and the Haar cascade algorithm. This system aims to enhance security and convenience by replacing traditional authentication methods with biometric-based identification. By leveraging facial recognition, the proposed system offers a more secure and efficient means of granting access to authorized users. The Haar cascade algorithm, known for its robustness and accuracy in facial recognition tasks, forms the core of the proposed system's authentication mechanism. This algorithm generates secure hashes from facial features, ensuring reliable

identification of authorized individuals while protecting against unauthorized access attempts. This interface provides users with intuitive controls for initiating the authentication process and accessing the vehicle. Through the combination of facial recognition technology, the Haar cascade algorithm, and a Tkinter-based GUI, the proposed system offers a comprehensive solution for secure and convenient car unlocking.

Advantages of Proposed System:

- Face recognition technology provides a high level of security by accurately identifying authorized individuals based on their unique facial features. Unlike traditional keys or key fobs that can be lost, stolen, or duplicated, facial features are difficult to forge, enhancing the overall security of the vehicle.
- Face recognition eliminates the need for physical keys or key fobs, providing a more convenient and accessible method of vehicle access. Users no longer need to carry or locate keys, making the entry process seamless and hassle-free, especially in situations where keys may be misplaced or forgotten.

CHAPTER 4

SYSTEM REQUIREMENTS & SPECIFICATIONS

4.1 Database

Central to the Facial Recognition System is a Database that securely stores the facial data of authorized drivers. When a new face is detected, the system compares it against the database to determine if it matches an authorized user. This database is crucial for ensuring only permitted individuals can unlock the vehicle.

4.2 Algorithm

Face detection a widely popular subject with a huge range of applications. Modern day Smartphones and Laptops come with in-built face detection software, which can authenticate the identity of the user. There are numerous apps that can capture, detect and process a face in real time, can identify the age and the gender of the user, and also can apply some really cool filters. The list is not limited to these mobile apps, as Face Detection also has a wide range of applications in Surveillance, Security and Biometrics as well. But the origin of its Success stories dates back to 2001, when Viola and Jones proposed the first ever Object Detection Framework for Real Time Face Detection in Video Footage. **Haar Cascade Algorithm**

Haar Cascade is a machine learning object detection algorithm used to identify objects in images or video. It is named after the Haar wavelets, which are mathematical functions used in the transformation process.

- Positive Images: Positive Images are a type of image that we want our classifier to identify.
- Negative Images: Negative Images are a type of image that contains something else, i.e., it does not contain the objects we want to detect.

The first real-time face detector also used the Haar classifiers, which we are introducing here. Finding objects in pictures and videos is done by a machine learning programme known as a Haar classifier or a Haar cascade classifier.

This is about taking a gentle look on the Viola-Jones Face Detection Technique, popularly known as Haar Cascades, and exploring some of the interesting concepts proposed by them.

This piece of work was done long before the Deep Learning Era had even started. But it’s an excellent work in comparison to the powerful models that can be built with the modern day Deep Learning Techniques. The algorithm is still found to be used almost everywhere. It has fully trained models available on GitHub. It’s fast. It’s pretty accurate (at least when I try it).

According to Woody Bledshoe, Helen Chan Wolf, and Charles Bisson were the first ones to do the first ever Face Detection on a Computer back in the 1960s**.**

A person had to manually pinpoint the coordinates of facial features such as the pupil centers, the inside and outside corner of eyes, and the widows peak in the hairline. The coordinates were used to calculate 20 distances, including the width of the mouth and of the eyes. A human could process about 40 pictures an hour in this manner and so build a database of the computed distances. A computer would then automatically compare the distances for each photograph, calculate the difference between the distances and return the closed records as a possible match. So what is Haar Cascade? It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and

Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001. The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them. The model created from this training is available at the OpenCV GitHub repository has the models stored in XML files, and can be read with the OpenCV methods. These include models for face detection, eye detection, upper body and lower body detection, license plate detection etc. Below we see some of the concepts proposed by Viola and Jones in their research.

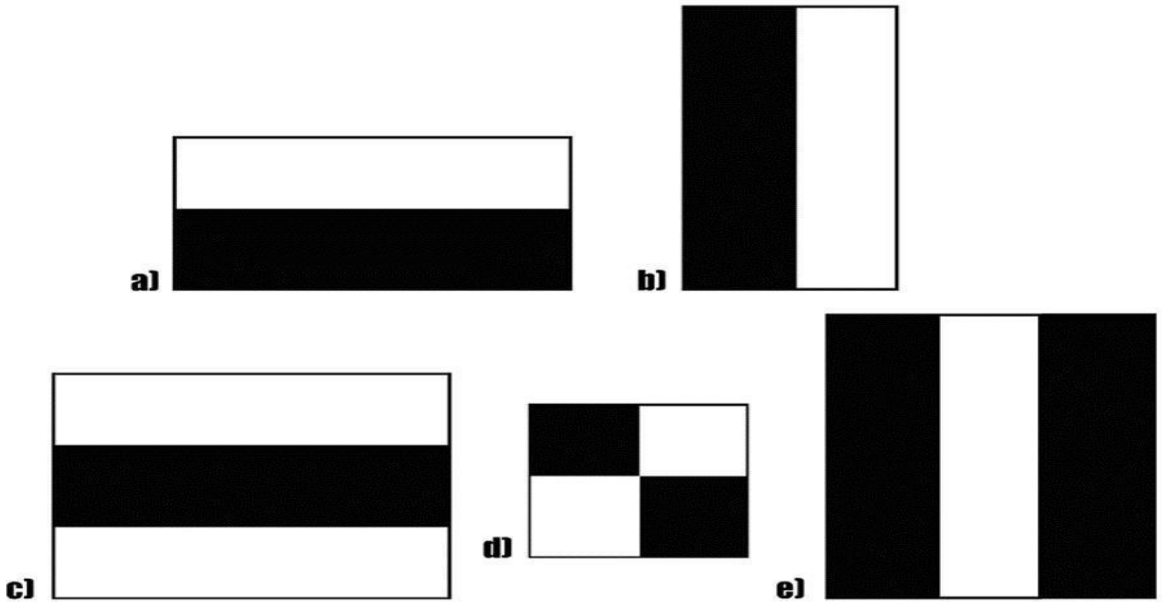

Fig 1. A sample of Haar features used in the Original Research Paper published by Viola and Jones.

The first contribution to the research was the introduction of the haar features shown above. These features on the image makes it easy to find out the edges or the lines in the image, or to pick areas where there is a sudden change in the intensities of the pixels.
The sum of all the image pixels lying in the darker area of the haar feature and the sum of all the image pixels lying in the lighter area of the haar feature.

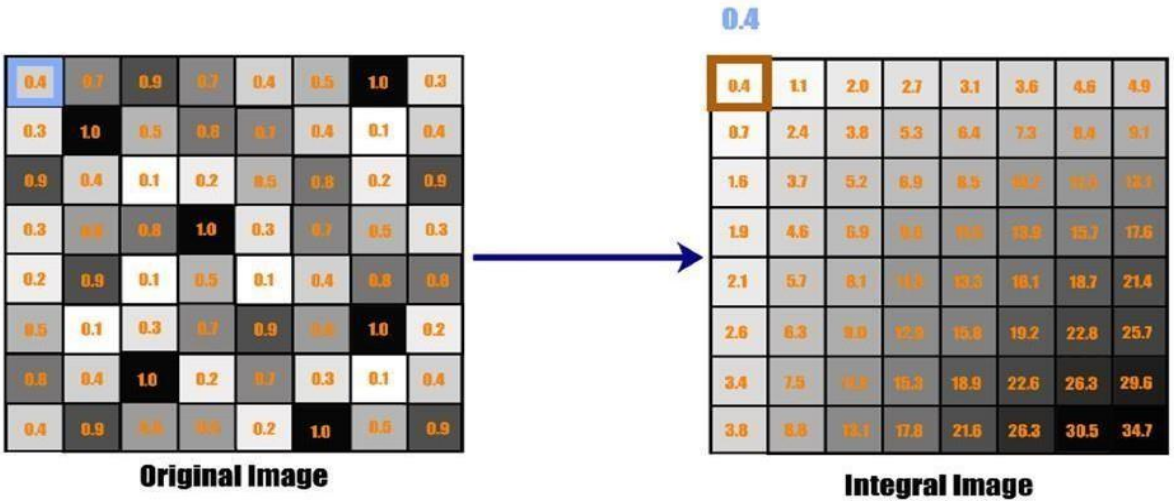

Fig 2.The GIF shows the making of an Integral Image. Each pixel in an Integral image is the sum of all the pixels in its left and above.

A sample calculation of Haar value from a rectangular image section has been shown here. The darker areas in the haar feature are pixels with values 1, and the lighter areas are pixels with values 0. Each of these is responsible for finding out one particular feature in the image. Such as an edge, a line or any structure in the image where there is a sudden change of intensities. For ex. in the image above, the haar feature can detect a vertical edge with darker pixels at its right and lighter pixels at its left.

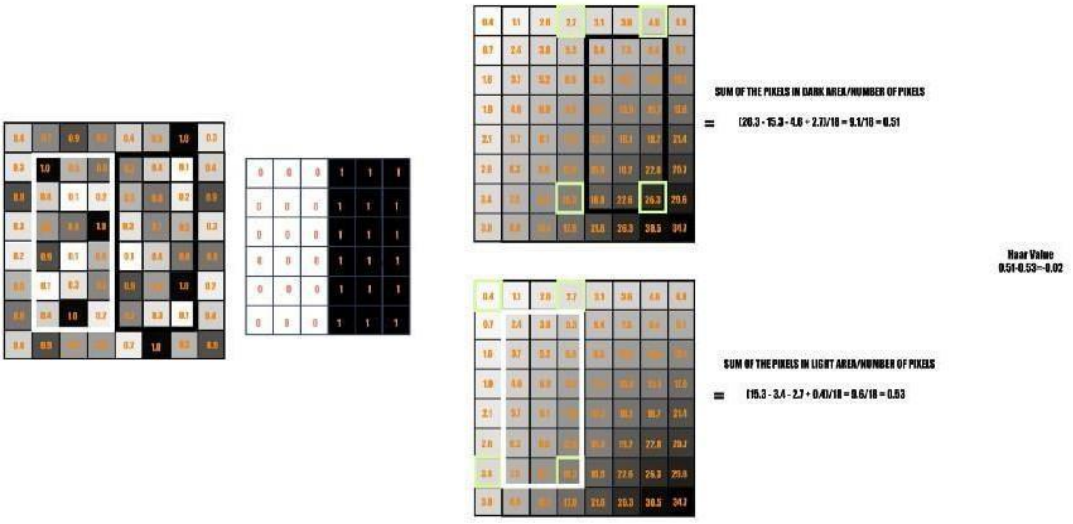

Fig 3. Integral Image is used here to calculate the haar value.

With the Integral Image, only 4 constant value additions are needed each time for any feature size (with respect to the 18 additions earlier). Thus, the time complexity of each addition gradually, as the number of additions does not depend on the number of pixels enclosed anymore.

In the above image, there is no edge in the vertical direction as the haar value is very far from

4.3 Design

4.3.1 System Architecture

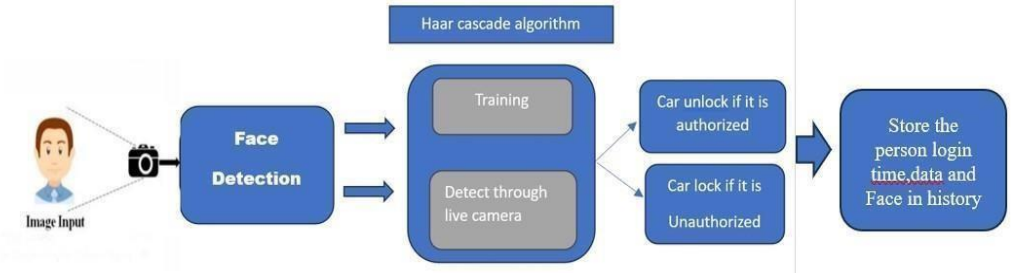

Fig 4. System architecture of Face Detection Unlocking System For Vehicles

4.3.2 Use Case

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

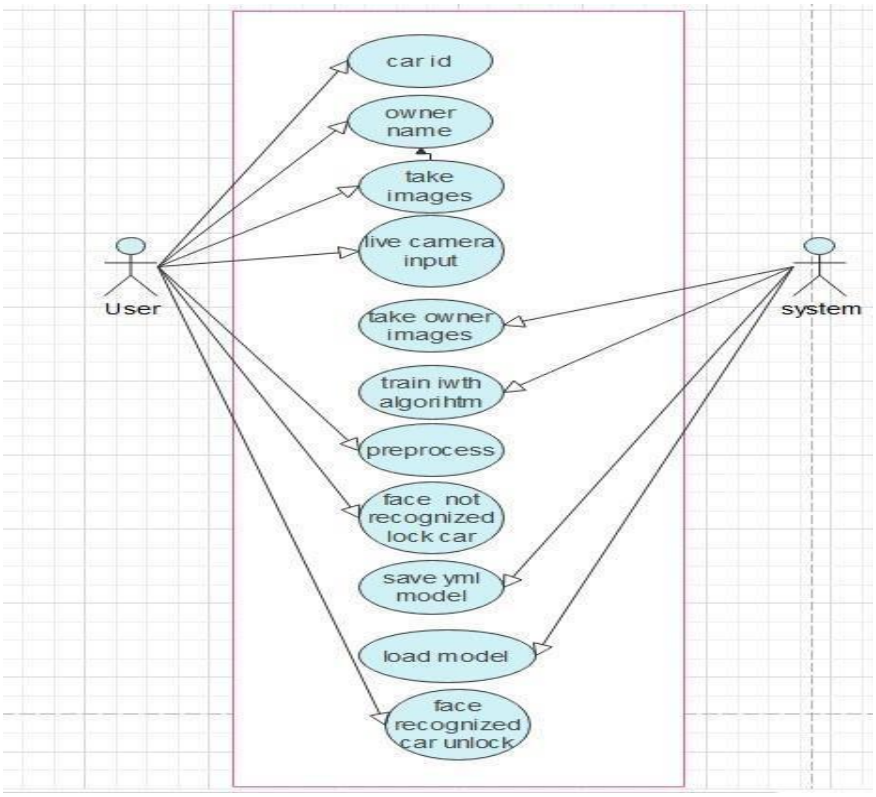

Fig 5. Use Case Diagram

4.3.3. Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

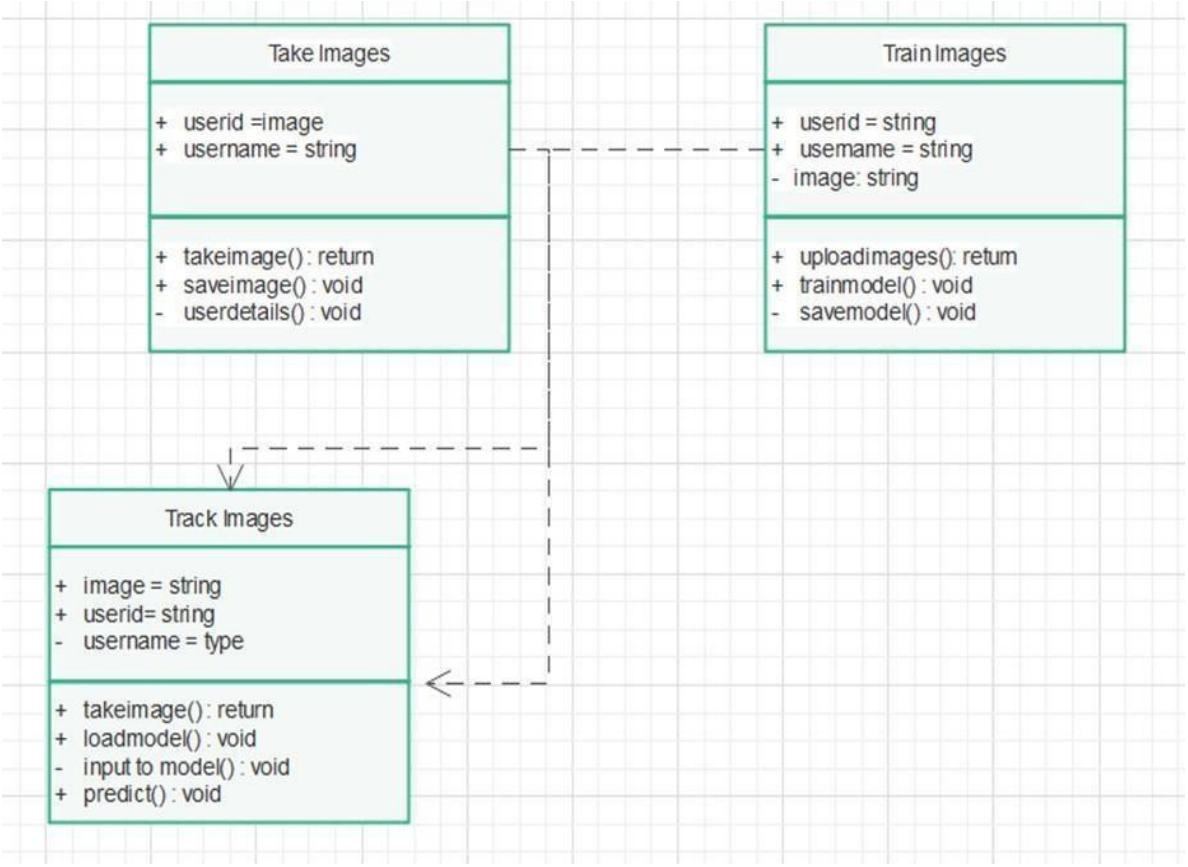

Figure 6. Class Diagram

4.3.4. Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

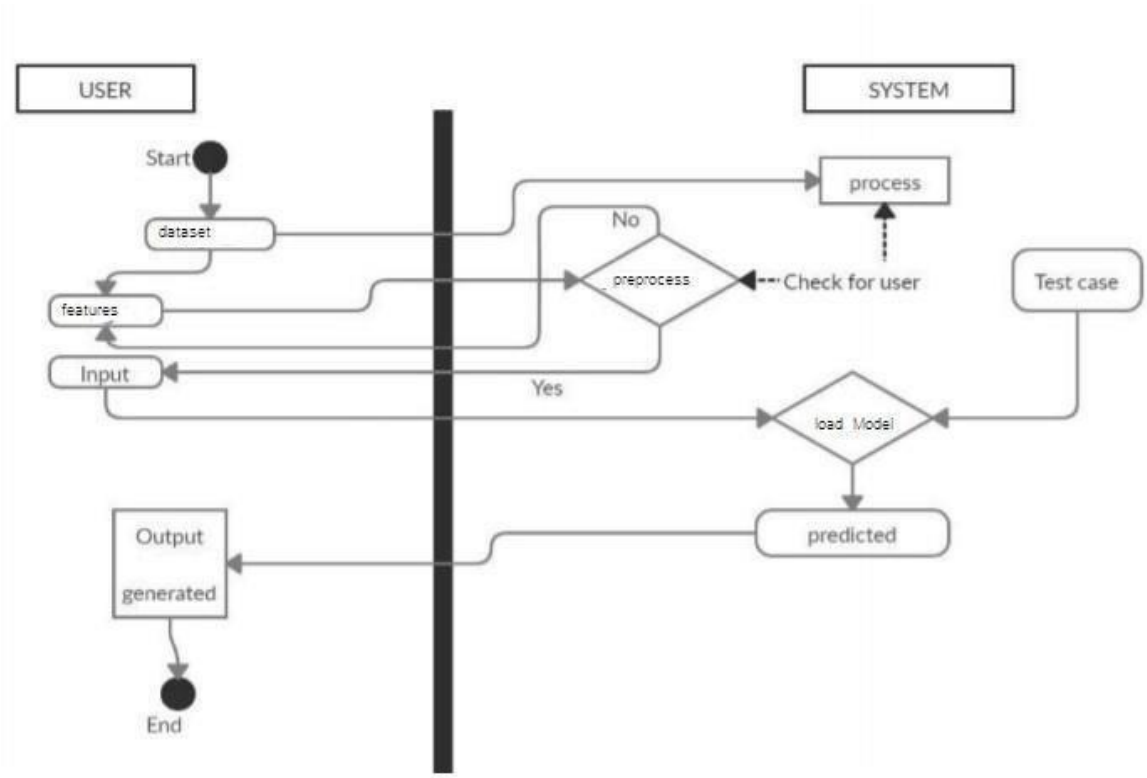

Figure 7. Activity Diagram

4.3.5. Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

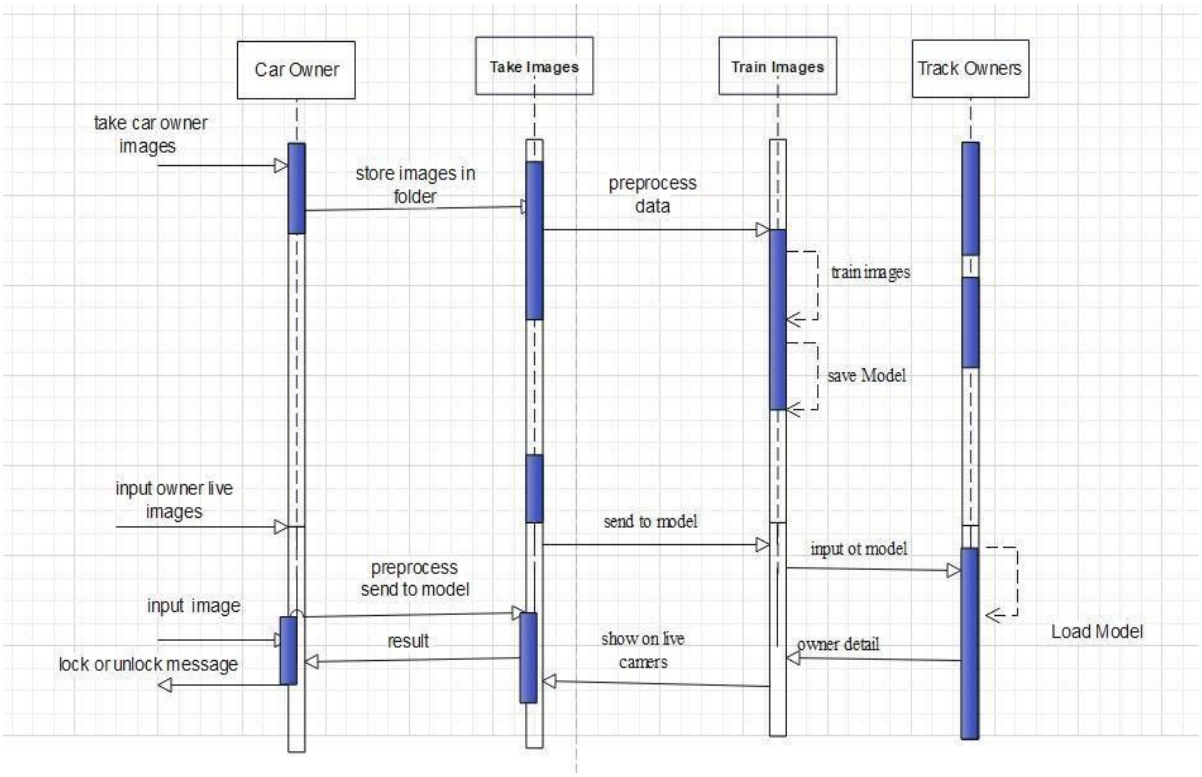

Fig 8. Sequence Diagram

4.4 Modules

4.4.1 Modules Description

System design is the process of creating a system's architecture, parts, and interfaces to ensure that it satisfies the needs of its users.

Thus, in order to examine the design of this project, we first go through the specifics of establishing the concept of drone detection through a few fundamental modules that would clearly describe the workings of the system that would come from the development.

User module:

In this module user will open camera and track images of every car owner to whom he want to lock his car through this application. For each tracking process it will take 60 images and then user should close training process. Once the process is done data will be stored in a folder 0,1,2..etc for each user new folder is created.

Training Process:

Once taking images process is done images from folder and for each image training process is done using opencv and yml file is stored in folder. This yml file is used for testing new images.

Detection Process:

In this process when user opens camera it will track live images of user and convert user image to gray colour and check with face recognition model and then boxes are drawn on each face and features are verified with trained model and output is displayed with car owner name number.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features in the image.

4.5 System Requirements

4.5.1 Hardware Requirements

- System : i5
- Hard Disk : 1 TB.
- Input Devices : Keyboard, Mouse
- Ram : 4 GB.

4.5.2 Software Requirements

- Operating system : Windows XP/7/10.
- Coding Language : Python
- IDE : Anaconda

4.5.3 Functional Requirements

1. **User Authentication:**

   The system shall capture and analyze facial features to authenticate users. It should have a user registration process for initial facial data enrollment. The system shall support multiple user profiles for shared vehicle usage.

2. **Face Detection Algorithm:**

   Implement a machine learning-based face detection algorithm. The algorithm shall adapt to varying lighting conditions and facial expressions. Achieve a high level of accuracy in identifying authorized users.

3. **Security Measures:**

   Ensure secure storage and transmission of facial data. Implement encryption protocols to protect user data. Have a mechanism to detect and prevent spoofing attempts.

4. **Registration of User Faces:**

   Users shall have the ability to register their faces with the system.During registration, the system shall capture multiple images of the user's face from different angles for better recognition accuracy.

5. **Multi-User Support:**

   The system shall support multiple users registering their faces for the same vehicle.Each user shall have the option to register and manage their face data independently.

6. **Notifications:**

   In case of multiple failed authentication attempts, the system shall alert the owner with a notification. The system shall also notify the owner if an unrecognized face is detected attempting to access the vehicle.

7. **Continuous Learning and Improvement:**

   Implement mechanisms for the system to learn and adapt over time. Regularly update the face detection algorithm based on new data and user interactions.

8. **Scalability:**

   Ensure the system is scalable to accommodate different vehicle models and types Facilitate easy integration into various automotive platforms.

4.5.4 Non-Functional Requirements

1. **Performance:**

   The system shall respond to face detection requests within 5 second under normal operating conditions.

2. **Availability:**

   The system shall be available 99.9% of the time, allowing for scheduled maintenance.

3. **Fault Tolerance:**

   The system should be resilient to minor failures and recover gracefully without compromising security.

4. **Accessibility:**

   The system shall comply with accessibility standards, ensuring usability for users with disabilities.

4.6 Testing

Code testing:

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

Specification Testing:

Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

Unit testing:

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

4.6.1. Black Box Testing

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

The above Black Box can be any software system you want to test. For example : an operating system like Windows, a website like Google ,a database like Oracle or even your own custom application. Under Black Box Testing , you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation. Black box testing – Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.
- Functional testing – This black box testing type is related to functional requirements of a system; it is done by software testers.
- Non-functional testing – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- Regression testing – Regression testing is done after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

4.6.2 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.White box testing is also known as clear, open, structural, and glass box testing.

It is one of two parts of the "box testing" approach of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "whitebox" was used because of the see-through box concept. The clear box or whitebox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

Steps to verify white box testing:

- Internal security holes

- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

4.6.3 Integrating Testing

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interactionbetween integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

Bottom-up Integration

This testing begins with unit testing, followed by tests of progressively higher-levelcombinations of units called modules or builds.

Top-down Integration

In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter.In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations. Table 6.5shows the test cases for integration testing and their results

4.6.4 System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner

design of the code or logic. System testing is important becauseof the following reasons:System testing is the first step in the Software Development Life Cycle, where theapplication is tested as a whole. The application is tested thoroughly to verify that it meets the functional andtechnical specifications. The application is tested in an environment that is very close to the production environment where the application will be deployed. System testing enables us to test, verify, and validate both the businessrequirements as well as the application architecture.

CHAPTER 5
METHODOLOGY

5.1 Technologies Used

Machine learning (ML) is an area of artificial intelligence (AI) that enables computers to "learn" for themselves over time from training data and develop without explicit programming. Data patterns can be found by machine learning algorithms, which can then use this information to learn and develop their own predictions. Algorithms and models used for machine learning, in essence, gain experience.

In contrast, machine learning is a process that is automated and gives computers the ability to solve issues with little to no human involvement and make decisions based on prior experiences.While machine learning and artificial intelligence are frequently used synonymously, they are actually two distinct ideas. Machine learning, a subset of AI that enables intelligent systems to autonomously learn new things from data, is what allows intelligent systems to make decisions, gain new abilities, and solve problems in a way that is similar to people. AI is the more general notion. Machine learning methods are used in image classification to examine the existence of objects in a picture and classify it.

The foundation of computer vision and image recognition is this specific problem. Machines don't examine an image in its entirety. They just examine pixel patterns or vectors while analysing a picture. The elements will subsequently be classified and given labels in accordance with the various rules that were established during algorithm configuration.

There are two types of Image Classification techniques:

If you are facing various pictures, and you only want to know whether the object in them is a cat or not, the problem you will have to handle is a binary classification. You only need to label one class of items for all images, or not to label it. The binary classification model is in charge of computing the presence or the absence of the object.

If there is more than one category, you handle a multiclass classification problem which implies the creation of multiple labels that will correspond to various objects. The machine will then predict which single object class is contained in the photographs or pictures, among a set of predefined labels. These techniques both mainly rely on the way the reference images are labeled. The following parts of this article will give a more detailed presentation of the way image classification works.

Machine learning is the ability that gives the computer to learn without being explicitly programmed. There are two types of machine learning:

Supervised Learning: supervised learning is the learning of the labelled data. It is the types of machine learning that maps the input and output based on the examples input-output pairs. In supervised learning each training data having pairs of input and desired outputs values. Supervised learning algorithm analyzes the training data and produces a function which can be used for mapping of new data.

Supervised Learning The output to solve the supervised learning algorithm are as:

- Determine the types of data, before doing anything else the user should understand which types of data set is to be used for training the data.
- Gathered the training data sets either in form of human experts or from measurements.
- Determine the feature of inputs from the learned data and depends on the inputs it changed into feature vector; number of features should not be large but should contains enough information to accurately predict the outputs.
- Check the learned function and the learned algorithm for example we use support vector machines or decisions tree.
- Complete the design and run the trained data sets.
- Analyzed the output and verify the data sets to get the accurate outputs.

Unsupervised Learning:

Unsupervised learning is a type of machine learning that helps in finding the previously unknown patterns in the data set without any known labels. It is known as self- organization and allows modelling probability densities of given inputs.

unsupervised Learning Some of the algorithm used in unsupervised learning are:

- Clustering
- Anomaly detection
- Neural networks
- Approach for learning latent variable models
- Non labelled data

Semi Supervised Machine Learning algorithm:

It’s like the middle man which have some labeled data and some unlabeled which can be prosed by the both the structured and unsupervised learning. The algorithms have been compared based upon the parameters: Size of the dataset and Number of technical indicators used. Accuracy and F-measure values have been computed for each algorithm. Long term model has been used to compute the accuracy and Fmeasure.

Reinforcement Learning:

This type of learning is used to reinforce or strengthen the network based on critic information. That is, a network being trained under reinforcement learning, receives some feedback from the environment. However, the feedback is evaluative and not instructive as in the case of supervised learning. Based on this feedback, the network performs the adjustments of the weights to obtain better critic information in future.

This learning process is similar to supervised learning but we might have very less information.

Python

Python is high level language and it is also integrated version of the program. Python is an object-oriented approach and its main aim to help programmers to write the code clearly, logical code for small and large scale of project.

Python is dynamically typed and garbage collected it also support multiple programming and it is both procedure and object oriented and also functional programming. And structural programming also supported. It has many built in function it also supports filter, map and reduce function. All the machine learning algorithm and the libraries are being supported by the python programming language. Python also support list, dict, sets and other generators.

Python code can be run in different platform such as anaconda, PyCharm etc.

The main goal of this programing language is as follows:

- Python is simple, object-oriented programming language.
- The language and implementation should provide support for software engineering principles such as strong type library preset for different machine learning algorithm, and all other algorithm in simple manner.
- Coding will be smooth in python and the data analysis can be easily done in python.

Python technology is both a programming language and a platform.

The python Programming Language the python programming language is a high-level language that can be characterized by all of
the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Python programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Python byte codes the platform-independent codes interpreted by the interpreter on the Python platform. The interpreter parses and runs each Python byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

Anaconda

Anaconda is free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine Learning applications, Large- scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. It is developed and maintained by Anaconda, Inc. The distribution incudes datascience packages suitable for Windows, Linux, and macOS. Packaged versions are required and are managed by the package management system anaconda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which

includes only conda, Python, the packages they depends on, and a small number of other packages.

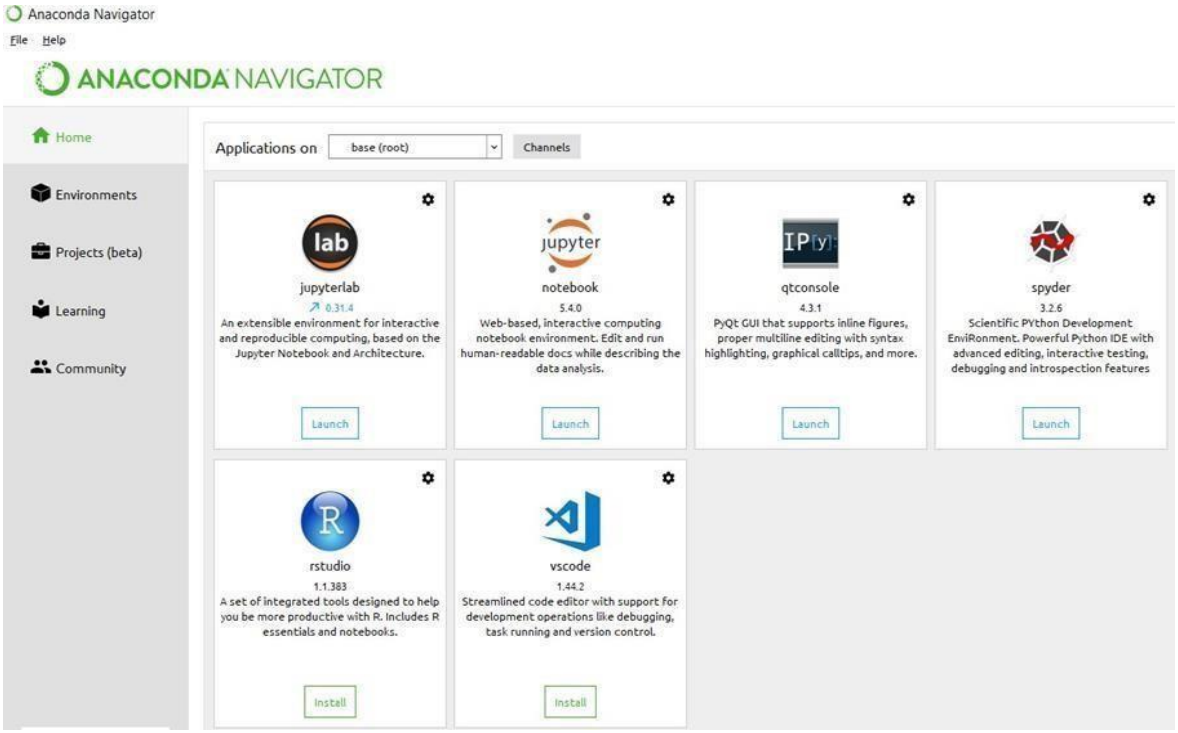

Fig 9 Anaconda navigator home tab

import numpy as np

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.

- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.
- A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays.

  In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.

Logical Design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams.

Physical Design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified / authenticated, how it is processed, and how it is displayed as output. In Physical design, following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing Requirements,
5. System control and backup or recovery.

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

Import time

This module provides various time-related functions. For related functionality, see also the datetime and calendar modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

An explanation of some terminology and conventions is in order. The epoch is the point where the time starts, and is platform dependent. For Unix, the epoch is January 1, 1970, 00:00:00 (UTC). To find out what the epoch is on a given platform, look at time.gmtime.The term seconds since the epoch refers to the total number of elapsed seconds since the epoch, typically excluding leap seconds. Leap seconds are excluded from this total on all POSIX-compliant platforms. The functions in this module may not handle dates and times before the epoch or far in the future. The cut-off point in the future is determined by the C library; for 32-bit systems,

it is typically in 2038.Function strptime() can parse 2-digit years when given %y format code. When 2-digit years are parsed, they are converted according to the POSIX and ISO C standards: values 69–99 are mapped to 1969–1999, and values 0–68 are mapped to 2000–2068.UTC is Coordinated Universal Time (formerly known as Greenwich Mean Time, or GMT). The acronym UTC is not a mistake but a compromise between English and French.DST is Daylight Saving Time, an adjustment of the timezone by (usually) one hour during part of the year. DST rules are magic (determined by local law) and can change from year to year. The C library has a table containing the local rules (often it is read from a system file for flexibility) and is the only source of True Wisdom in this respect. The precision of the various real-time functions may be less than suggested by the units in which their value or argument is expressed. E.g. on most Unix systems, the clock “ticks” only 50 or 100 times a second.On the other hand, the precision of time() and sleep() is better than their Unix equivalents: times are expressed as floating point numbers, time() returns the most accurate time available (using Unix gettimeofday() where available), and sleep() will accept a time with a nonzero fraction (Unix select() is used to implement this, where available).The time value as returned by gmtime(), localtime(), and strptime(), and accepted by asctime(), mktime() and strftime(), is a sequence of 9 integers. The return values of gmtime(), localtime(), and strptime() also offer attribute names for individual fields.See struct_time for a description of these objects.Changed in version 3.3: The struct_time type was extended to provide the tm_gmtoff and tm_zone attributes when platform supports corresponding struct tm members.

Import os

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the file input module. For creating temporary files and directories see the temp file module, and for high-level file and directory handling see the shutil module.

Notes on the availability of these functions:

- The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function os.stat(path) returns stat information about path in the same format (which happens to have originated with the POSIX interface).
- Extensions peculiar to a particular operating system are also available through the os module, but using them is of course a threat to portability.
- All functions accepting path or file names accept both bytes and string objects, and result in an object of the same type, if a path or file name is returned.
- On VxWorks, os.popen, os.fork, os.execv and os.spawn*p* are not supported.

5.2 Source Code

```
# -*- coding: utf-8 -*-


import tkinter as tk
from tkinter import Message ,Text
import cv2,os
import shutil
import csv
import numpy as np
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time
import tkinter.ttk as ttk
import tkinter.font as font
from tkinter import messagebox

window = tk.Tk()

window.title("Face Recognition unlocking system")

dialog_title = 'QUIT'
dialog_text = 'Are you sure?'



# window.configure(background='green')

# Open and convert the image to a format compatible with Tkinter

image = Image.open("rec.jpg")
image = image.resize((window.winfo_screenwidth(), window.winfo_screenheight()),
Image.ANTIALIAS)
background_image = ImageTk.PhotoImage(image)

# Create a Canvas widget
canvas = tk.Canvas(window, width=image.width, height=image.height)
canvas.pack()

# Add the background image to the Canvas
canvas.create_image(0, 0, anchor=tk.NW, image=background_image)



window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)
```

```
message = tk.Label(window, text="Face Recognition unlocking system", fg="black",
font=('times', 30, ' bold '), bd=0, highlightthickness=0)

message.place(x=500, y=60)

lbl = tk.Label(window, text="Enter
ID",width=20  ,height=2  ,fg="black"    ,font=('times', 15, ' bold ') )
lbl.place(x=400, y=200)

txt = tk.Entry(window,width=20  ,fg="black",font=('times', 15, ' bold '))
txt.place(x=700, y=215)

lbl2 = tk.Label(window, text="Enter Name",width=20  ,fg="black"      ,height=2
,font=('times', 15, ' bold '))
lbl2.place(x=400, y=300)

txt2 = tk.Entry(window,width=20  ,fg="black",font=('times', 15, ' bold ')  )
txt2.place(x=700, y=315)

lbl3 = tk.Label(window, text="Notification : ",width=20  ,fg="black"    ,height=2
,font=('times', 15, ' bold underline '))
lbl3.place(x=400, y=400)

message = tk.Label(window, text=""    ,fg="black"  ,width=30  ,height=2,
activebackground = "yellow" ,font=('times', 15, ' bold '))
message.place(x=700, y=400)

lbl3 = tk.Label(window, text="Notification : ",width=20  ,fg="black"     ,height=2
,font=('times', 15, ' bold  underline'))
lbl3.place(x=400, y=650)


message2 = tk.Label(window, text="" ,fg="black"   ,activeforeground =
"green",width=30  ,height=2  ,font=('times', 15, ' bold '))
message2.place(x=700, y=650)

lbl4= tk.Label(window, text="Notification : ",width=20  ,fg="black"    ,height=2
,font=('times', 15, ' bold  underline'))
lbl4.place(x=400, y=750)


message3= tk.Label(window, text="" ,fg="black"   ,activeforeground =
"green",width=30  ,height=2  ,font=('times', 15, ' bold '))
message3.place(x=700, y=750)
```

```
# def clear():
#     txt.delete(0, 'end')
#     res = ""
#     message.configure(text= res)

# def clear2():
#     txt2.delete(0, 'end')
#     res = ""
#     message.configure(text= res)

def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass

    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError, ValueError):
        pass

    return False

def carownerimage():
    Id=(txt.get())
    name=(txt2.get())
    if(is_number(Id) and name.isalpha()):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector=cv2.CascadeClassifier(harcascadePath)
        sampleNum=0
        while(True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x,y,w,h) in faces:
                cv2.rectangle(img,(y,y),(x+w,y+h),(255,0,0),2)
                #incrementing sample number
                sampleNum=sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage/ "+name +"."+Id +'.'+ str(sampleNum) +
".jpg", gray[y:y+h,x:x+w])
                # Inside the carownerimage() function, just before cv2.imwrite()
                print("Image dimensions (x, y, w, h):", x, y, w, h)
```

```
                print("Gray image dimensions:", gray.shape)

                #display the frame
                cv2.imshow('frame',img)
            #wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
            elif sampleNum>60:
                break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Saved for ID : " + Id +" Name : "+ name
        row = [Id , name]
        with open('carownerdetails\carownerdetails.csv','a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
        message.configure(text= res)
    else:
        if(is_number(Id)):
            res = "Enter Alphabetical Name"
            message.configure(text= res)
        if(name.isalpha()):
            res = "Enter Numeric Id"
            message.configure(text= res)

def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector =cv2.CascadeClassifier(harcascadePath)
    faces,Id = getImagesAndLabels("TrainingImage")
    recognizer.train(faces, np.array(Id))
    recognizer.save("TrainingImage\Trainner.yml")
    res = "Image Trained"#+",".join(str(f) for f in Id)
    message.configure(text= res)

def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    print(imagePaths)

    #create empth face list
    faces=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
```

```
        #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id from the image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces,Ids

def Detectcarowner():
    var=0
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read("TrainingImage\Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    df=pd.read_csv("carownerdetails\carownerdetails.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names =  ['Id','Name','Date','Time']
    attendance = pd.DataFrame(columns = col_names)
    while True:
        ret, im =cam.read()
        gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
        faces=faceCascade.detectMultiScale(gray, 1.2,5)
        for(x,y,w,h) in faces:
            cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
            Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
            #print(conf)
            if(conf < 50):
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
                timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa=df.loc[df['Id'] == Id]['Name'].values
                tt=str(Id)+"-"+aa
                attendance.loc[len(attendance)] = [Id,aa,date,timeStamp]
                var=0
            else:
                Id='Unknown'
                tt=str(Id)
                var=1
            if(conf > 75):
                noOfFile=len(os.listdir("Image Unknown"))+1
                cv2.imwrite("ImagesUnknown\Image"+str(noOfFile) + ".jpg",
im[y:y+h,x:x+w])

            cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
```

```
        attendance=attendance.drop_duplicates(subset=['Id'],keep='first')
        cv2.imshow('im',im)
        if (cv2.waitKey(1)==ord('s')):
            break
    ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour,Minute,Second=timeStamp.split(":")
    fileName="Detected\Detected_"+date+"_"+Hour+"-"+Minute+"-"+Second+".csv"
    attendance.to_csv(fileName,index=False)
    cam.release()
    cv2.destroyAllWindows()
    #print(attendance)
    res=attendance
    print(res)
    if var ==1:
       popup()
       var=0
    else:
        dd="Face Detected Unlock the door"
        message2.configure(text= res)
        message3.configure(text= dd)


def popup():
    result = messagebox.askquestion("Popup Title", "Do you want to continue?")
    if result == 'yes':
        print("You clicked Yes!")
        dd="Allow to unlock"
        message3.configure(text= dd)
    else:
        print("lock the car")
        dd="Not allow to unlock"
        message3.configure(text= dd)

# clearButton = tk.Button(window, text="Clear",
command=clear  ,fg="red"   ,bg="yellow"  ,width=20  ,height=2 ,activebackground =
"Red" ,font=('times', 15, ' bold '))
# clearButton.place(x=950, y=200)
# clearButton2 = tk.Button(window, text="Clear",
command=clear2  ,fg="red"   ,width=20  ,height=2, activebackground = "Red"
,font=('times', 15, ' bold '))
# clearButton2.place(x=950, y=300)
takeImg = tk.Button(window, text="OwnerImage",
command=carownerimage  ,fg="red"    ,width=20  ,height=3, activebackground = "Red"
,font=('times', 15, ' bold '))
takeImg.place(x=200, y=500)
```

```
trainImg = tk.Button(window, text="Train Images",
command=TrainImages  ,fg="red"    ,width=20  ,height=3, activebackground = "Red"
,font=('times', 15, ' bold '))
trainImg.place(x=500, y=500)
trackImg = tk.Button(window, text="Detectcarowner",
command=Detectcarowner  ,fg="red"    ,width=20  ,height=3, activebackground = "Red"
,font=('times', 15, ' bold '))
trackImg.place(x=800, y=500)
quitWindow = tk.Button(window, text="Quit",
command=window.destroy  ,fg="red"    ,width=20  ,height=3, activebackground = "Red"
,font=('times', 15, ' bold '))
quitWindow.place(x=1100, y=500)
#popup_btn = tk.Button(window, text="Click Me", command=popup)
#popup_btn.pack(pady=20)

window.mainloop()
```

CHAPTER 6

EXPERIMENTAL RESULTS

6.1.1. Login page

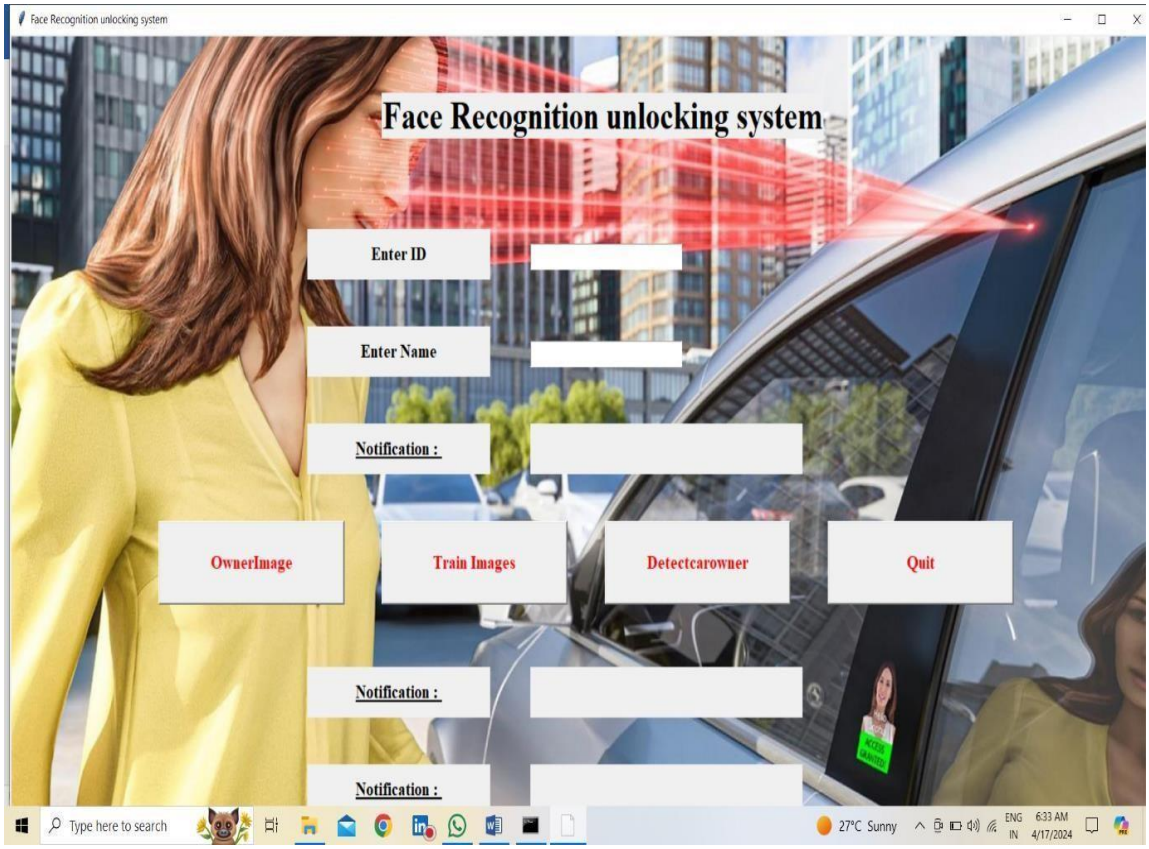

Fig 10. Login page

Description:

After running the python code file we will get the output as follows that is the output page of the required document that the python file consist of the machine learning as well as the front Tkinter code with the predefined libraries.

To Run the python we have to follow those steps:

1. Open anaconda command prompt then go to the project folder with the command of cd and give the path.
2. Then run by command **python train.py**

6.1.2. Enter the details

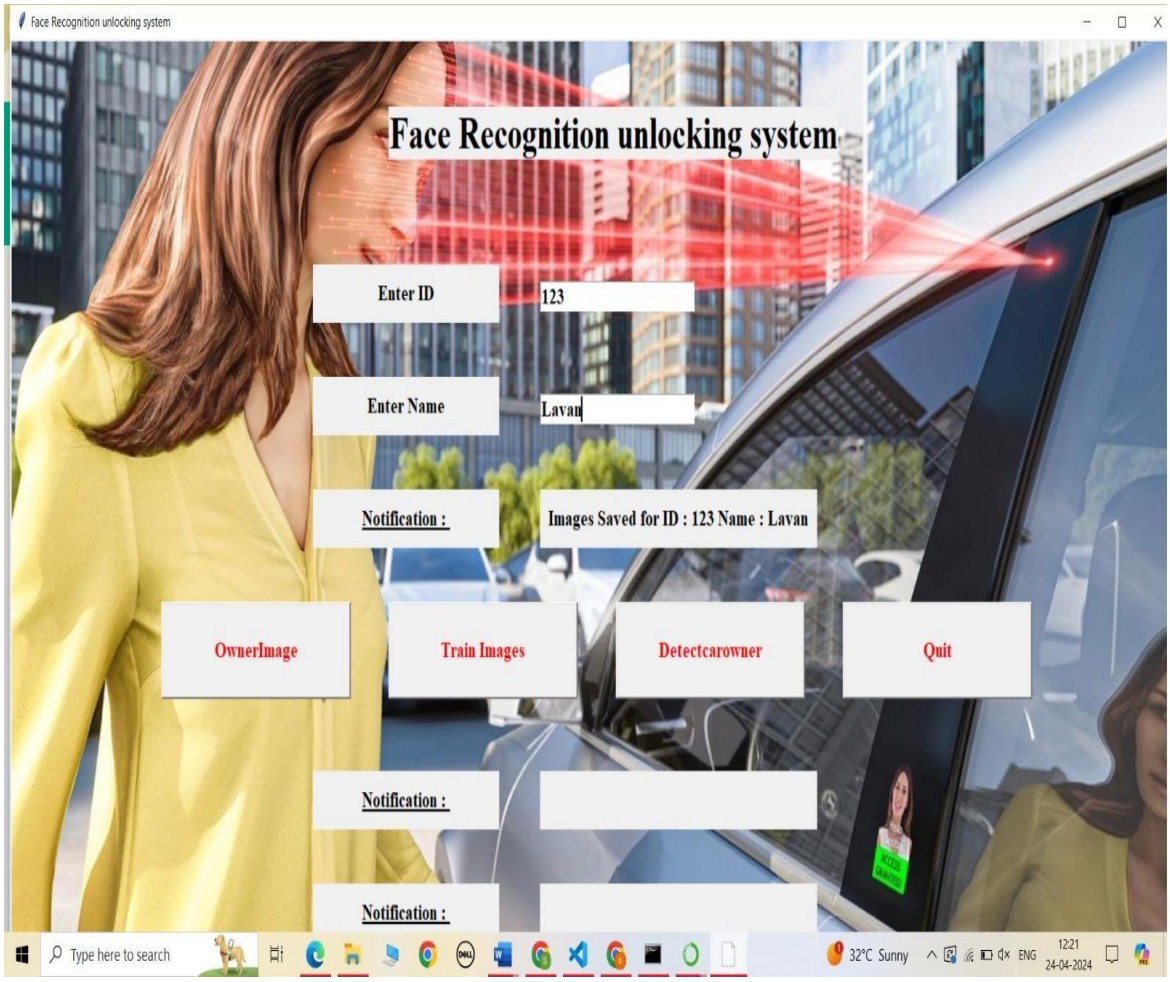

Fig 11 Enter the details

After running the project then the page will be open that contains of the name and mobile number to register then it will going to me recognise the image for training.

Give the name and mobile number as for registration process.

6.1.3 Camera opens to take images

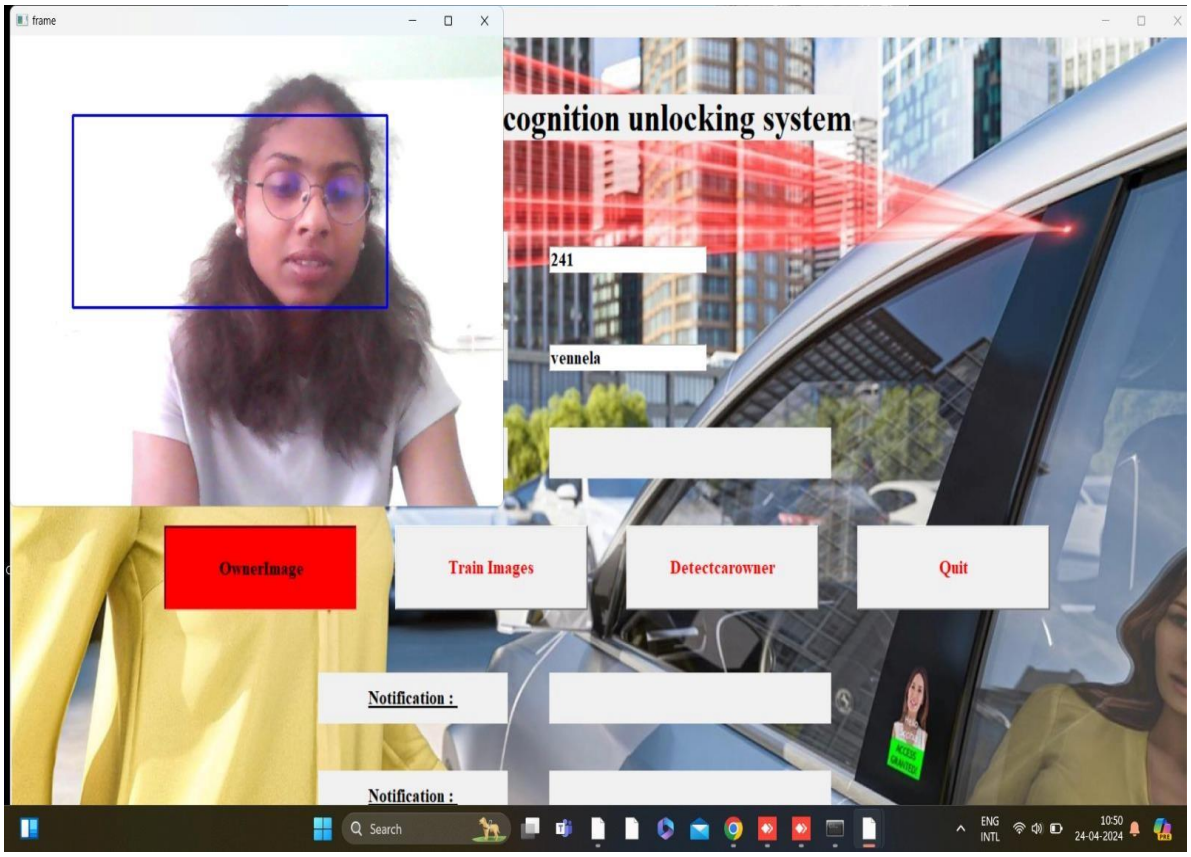

Fig 12. Taking images

Then the camera will be open when we click the take owner image button then it will take images as input and going to be store in the project file training image folder.

6.1.4. Train the Images

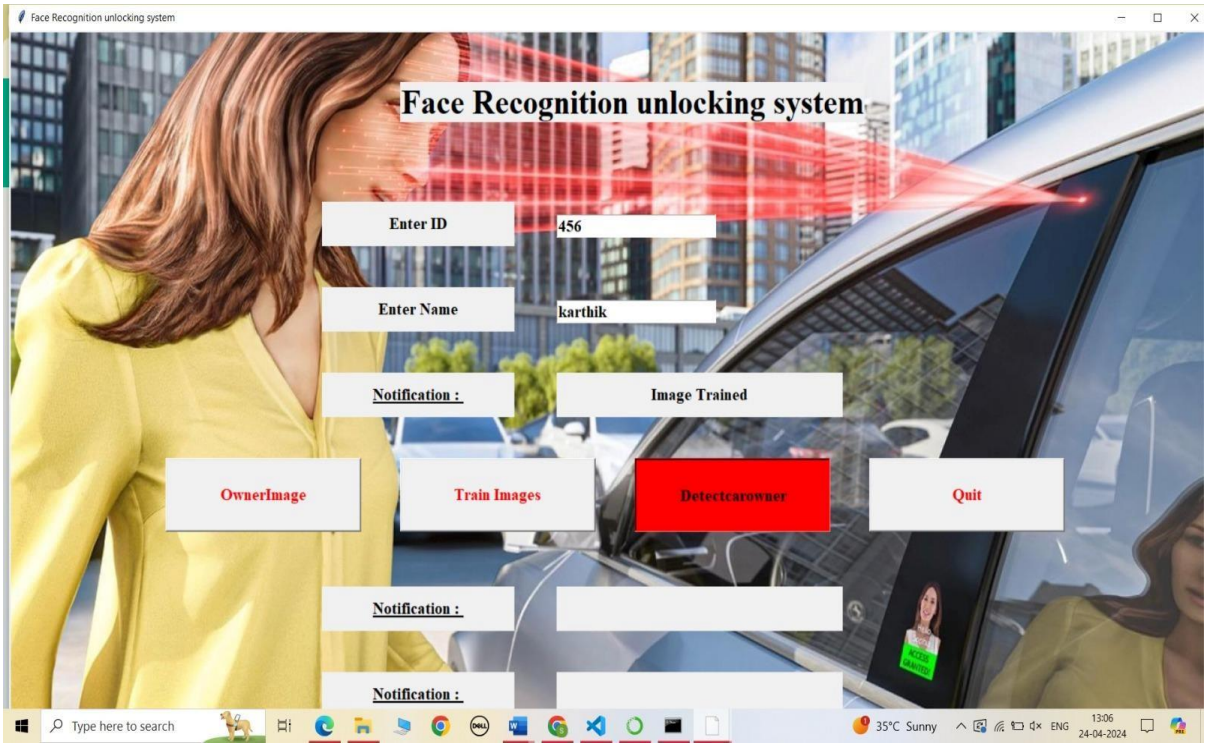

Fig 13 Train the person

After taking the images then we are going we have to train the images by clicking the button of the train image then the images will be trained then the trained images will have different frequency.

6.1.5. Details of unauthorized face

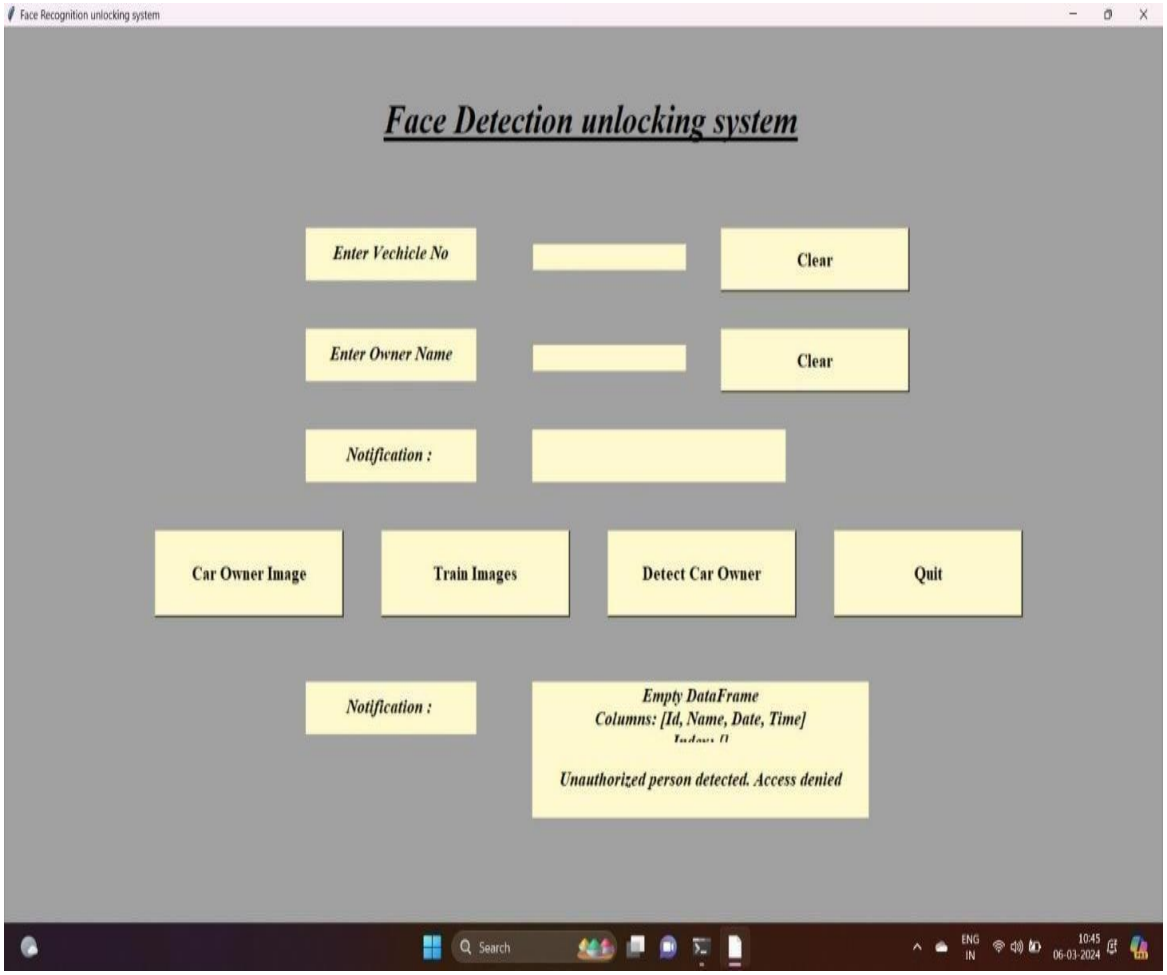

Fig 14. Unauthorized person

After training the images we are going to detect the images through the camera at real time then according to the secure access if the certain person is authorised only the door will be will unlocked. If the person is unauthorised then we get notify the popup if we allow only then the door will be unlocked.

6.1.6 Authorized person

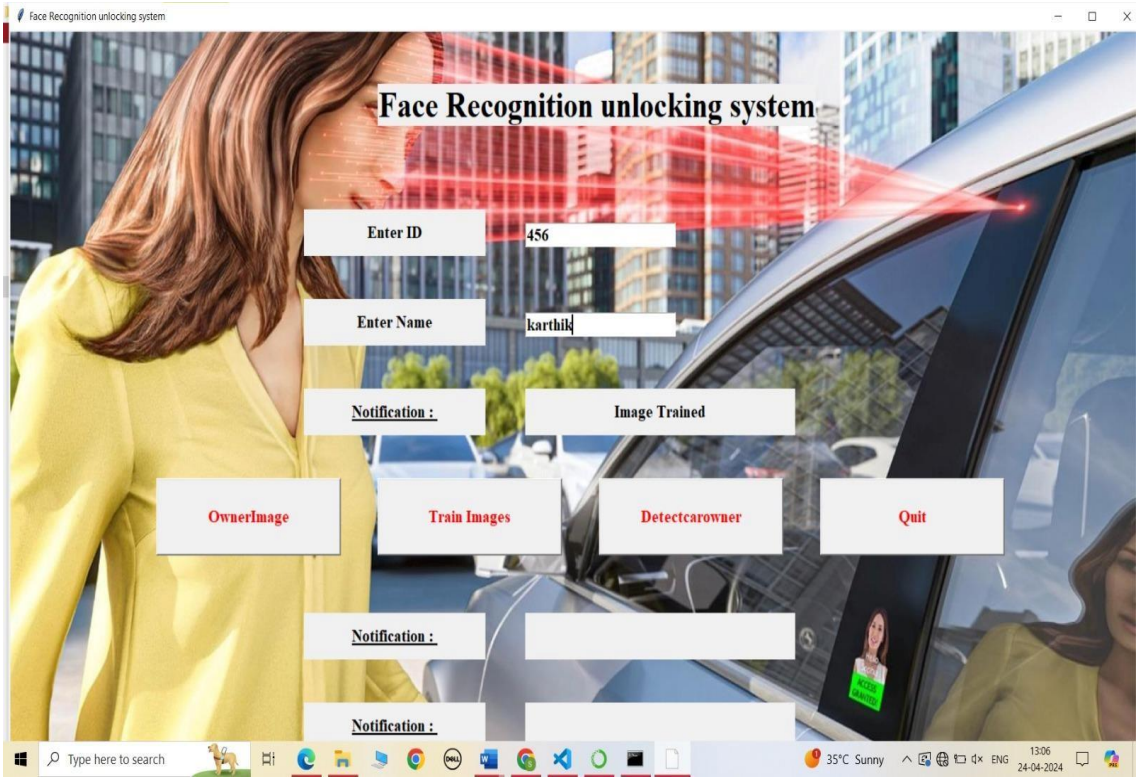

Fig 15 Authorised person

After training the images we are going to detect the images through the camera at real time then according to the secure access if the certain person is authorised only the door will be will unlocked.

6.1.6 Images Stored in the folder

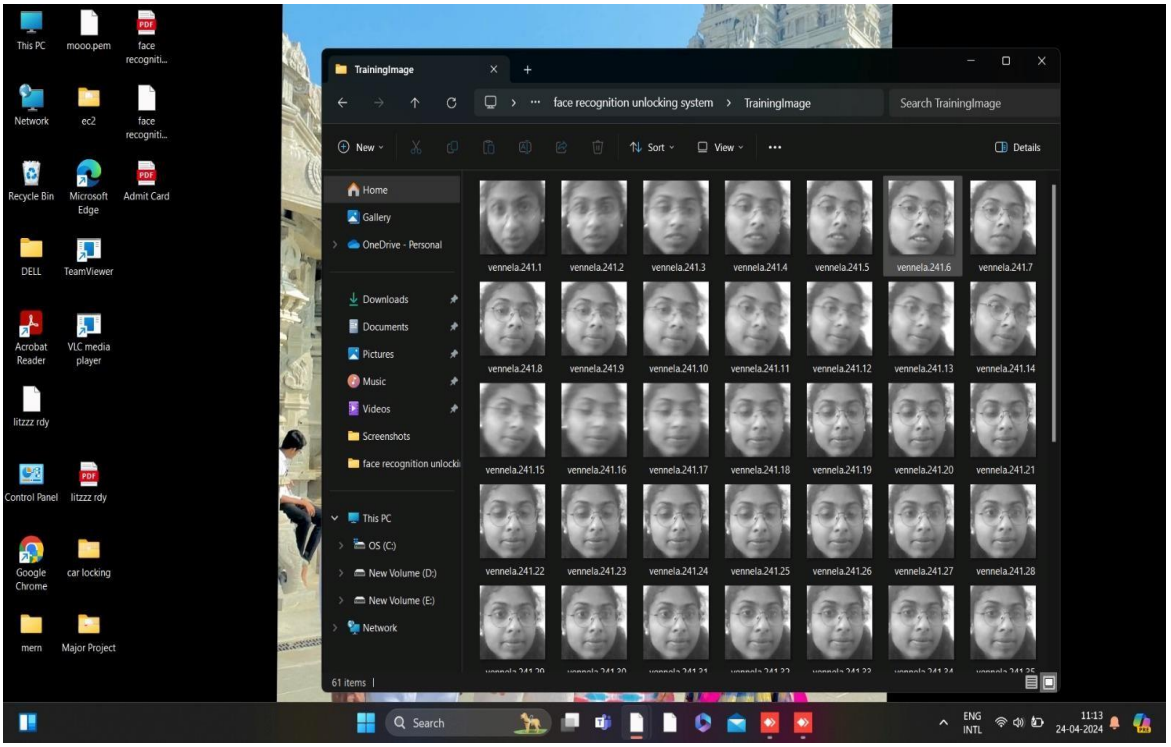

Fig 16.Images stored

CHAPTER 7

CONCLUSION

In conclusion, the implementation of a face recognition-based car unlocking system offers numerous benefits in terms of security, convenience, user-friendliness, safety, and technological innovation. By leveraging advanced facial recognition technology, this system provides a secure and seamless method of vehicle access control, eliminating the need for traditional keys or key fobs. Users benefit from a more convenient and accessible entry process, while also enjoying enhanced safety features and personalized access settings. Moreover, the integration of face recognition technology into vehicles demonstrates a commitment to technological advancement and future-proofing, positioning vehicles as intelligent systems capable of adapting to evolving user expectations and security standards. Compliance with regulations regarding data protection and privacy further enhances the credibility and reliability of these systems. As the automotive industry continues to embrace technological innovation, the adoption of face recognition-based car unlocking systems is expected to increase, offering an exciting opportunity for manufacturers to differentiate their vehicles and enhance the overall driving experience.

Future Scope

While face recognition-based car unlocking systems offer significant advantages, there are areas for future work and improvement to further enhance their functionality and effectiveness. Some potential areas for future research and development include, Improved Accuracy and Reliability for Further research and development efforts can focus on improving the accuracy and reliability of face recognition algorithms, particularly in challenging lighting conditions and with diverse facial features. Integration with multi-factor authentication, exploring the integration of face recognition technology with other biometric or authentication methods, such as fingerprint recognition or voice recognition, to enhance security through multi-factor authentication.

Real-time adaptation and personalization for developing algorithms that enable real-time adaptation and personalization of vehicle settings based on the recognized user's preferences, such as seat positions, climate control settings, and entertainment options. Enhanced privacy and data protection to implementing robust privacy and data protection measures to ensure secure storage and processing of facial recognition data, as well as compliance with evolving

REFERENCES

[1]. N. R. Borkar and S. Kuwelkar, "Real-time implementation of face recognition system", 2017 International Conference on Computing Methodologies and Communication (ICCMC), pp. 249-255, 2017

[2].K.Satish, A.Lalitesh, K.Bhargavi, M.S.Prem and T.Anjali, “Driver Drowsiness Detection”, 2020.

[3]. Rajasekhar Kommarju, Rangacharya Kommanduri, S. Rama Lingeswararao, and Cherukumalli srivalli, “IOT Based Vehicle(Car) Theft Detection”,Image Processing and Capsule Networks, pp. 620-628, January 2021.

[4]. Nawaf Hazim Barnouti, Sinan Sameer Mahmood AI-dabbagh,Wael Esam Matt “Face recognition” 2016.

[5]. Javier Ruiz Del Solar, Rodrigo Verschae, and Mauricio Correa. Face recognition in unconstrained environments: A comparative study. In ECCV Workshop on Faces in Real Life Images: Detection, Alignment, and Recognition, Marseille, France, October 2015.

[6]. W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld,“Face recognition: A literature survey,” ACM Computing Surveys, 2010, vol. 35, no. 4, pp. 399-458.

[7]. C. Nandakumar, G. Muralidaran and N. Tharani, "Real Time Vehicle Security System through Face Recognition", International Review of Applied Engineering Research, vol. 4, pp. 371-378.

[8]. Z. Zhu and F. Chen, "Fingerprint Recognition-Based Access Controlling System for Automobiles", proceedings of the 4th International Congress on Image and Signal Processing, Oct. 2011.

[9]. H.K.Ekenel and R.Stiefelhagen, Analysis of local appearance based face recognition: Effe cts of feature selection and feature normalization. In CVPR Biometrics Workshop, New York, USA, 2006