AI TRESPASSER OVER BREACH MAIL

A PROJECT REPORT

Submitted by

G.UGENDAR RAJ (211420205173)

SYAM SAI.K.G (211420205162)

S.YOGESHWARAN (211420205184)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE, CHENNAI 600 123

(An Autonomous Institution)

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report **"AI TRESPASSER OVER BREACH MAIL"** is the bonafide work of **"G.UGENDAR RAJ (211420205173), SYAM SAI.K.G (211420205162)** and **S.YOGESHWARAN (211420205184)"** who carried out the project under my supervision.

SIGNATURE
Dr. M. HELDA MERCY M.E., Ph.D.,
HEAD OF THE DEPARTMENT

Department of Information Technology
Panimalar Engineering College
Poonamallee, Chennai - 600 123

SIGNATURE
Mrs. MUTHULAKSHMI M.Tech., Ph.D.,
ASSOCIATE PROFESSOR
(SUPERVISOR)

Department of Information Technology
Panimalar Engineering College
Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I hereby declare that the project report entitled "**AI TRESPASSER OVER BREACH MAIL"** which is being submitted in partial fulfillment of the requirement of the course leading to the award of the 'Bachelor of Technology in Information Technology' in **Panimalar Engineering College, An Autonomous Institution Affiliated to Anna University- Chennai** is the result of the project carried out by me under the guidance and supervision of **Mrs. K.MUTHULAKSHMI, M.TECH.,(Ph.D) Associate Professor in the Department of Information Technology**. I further declare that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

(**G.UGENDAR RAJ**)

(SYAM SAI KG)

(S.YOGESHWARAN)

Date: **Place:** Chennai

It is certified that this project has been prepared and submitted under my guidance.

Date: (**Mrs. K.MUTHULAKSHMI M.Tech., Ph.D.)**

Place: Chennai (Associate Professor/ IT)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind cooperation and support from many, for successful completion. We wish to express our sincere thanks to all those involved in completing this project.

Our sincere thanks to **Our Honorable Secretary and Correspondent, Dr.**

P. CHINNADURAI, M.A., Ph.D., for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors, Mrs. C. VIJAYA RAJESHWARI and Dr.C.SAKTHI KUMAR, M.E., Ph.D.** and **Dr.SARANYA SREE SAKTHIKUMAR., B.E., M.B.A., Ph.D.** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and for providing us ample time to complete our project.

We express our indebtedness and gratitude to our staff in charge, **Mrs.K. MUTHULAKSHMI M.Tech.,(Ph.D)** Associate Professor, Department of Information Technology for her guidance throughout the course of our project. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

TABLE OF CONTENTS

ABSTRACT

The use of artificial intelligence (AI) technology is rapidly expanding and has already made a significant impact on various industries. However, with the increasing reliance on AI, there is a growing concern about the potential for AI systems to breach privacy laws and ethical standards. This paper presents a new AI trespasser system that uses breach mail technology to detect and prevent potential privacy violations by AI systems. The system operates by continuously monitoring AI systems for any signs of data breaches and sends out alerts to the relevant parties in the form of breach emails. These breach emails contain detailed information about the nature of the breach, its severity, and any necessary actions that need to be taken to resolve the issue. This system provides a proactive and effective solution to the problem of AI privacy breaches and will be valuable to organizations that rely on AI systems to manage sensitive information. Here the Authorized Facial features of the existing members have been captured and trained using OpenCV, and the generated model dataset will be deployed to detect the unauthorized trespasser which in turn triggers a breach mail if the detected facial features are not found in the generated/trained dataset model.

Keywords: ***Machine Learning, Trespasser Detection, Trigger Mail Alert, Artificial Intelligence, Data Science, Open CV, CNN, Dataset Training.***

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. No	Abbreviation	Expansion
1	SVM	Support Vector Machines
2	OPEN CV	Open Source Computer Vision
3	CNN	Convolutional neural network
4	DCNN	Deeply Convoluted neural network
5	LDA	Linear Discriminant Analysis
6	GPU	Graphical processing unit
7	PC	Personal Computer
8	RAM	Random access memory
9	OS	Operating System

INTRODUCTION

1.1. INTRODUCTION:

Facial recognition technology is an artificial intelligence that uses algorithms to identify and verify an individual's identity based on their facial features. This technology has become increasingly popular for security purposes, as it offers a quick and efficient way to verify the identity of individuals in a variety of settings. In terms of security, facial recognition technology can be used for a range of applications, including surveillance, access control, and authentication. For example, it can be used to identify known criminals or suspects in real time, to monitor large crowds in public spaces, or to control access to secure areas.

Facial recognition technology works by analyzing the unique features of an individual's face, such as the distance between their eyes, the shape of their nose, and the contours of their jawline. This information is then compared to a database of known faces to determine a match.

While facial recognition technology can be an effective tool for security purposes, there are also concerns about its potential misuse and infringement on privacy. Some critics argue that the technology could be used for mass surveillance or profiling and that it may not be accurate or reliable enough to be used as a sole means of identification. As a result, there are ongoing debates about the appropriate use of facial recognition technology and the need for regulations to ensure its responsible use. Facial recognition technology is a type of biometric authentication that uses algorithms to identify and verify a person's identity based on their facial features. In terms of security, AI facial recognition is used to enhance security measures and prevent unauthorized access to sensitive areas or data. For example, in high-security environments such as airports, AI facial recognition technology can be used to compare the facial features of individuals against a database of known individuals, in order to identify and prevent the entry of individuals who are not authorized to be there. Similarly,

AI facial recognition can be used to monitor security footage in real-time and alert security personnel if an unrecognized face is detected. However, using AI facial recognition for security purposes has also raised concerns about privacy, surveillance, and potential misuse of the technology. As a result, many countries have implemented regulations to govern the use of facial recognition technology and ensure that it is used in a responsible and ethical manner.

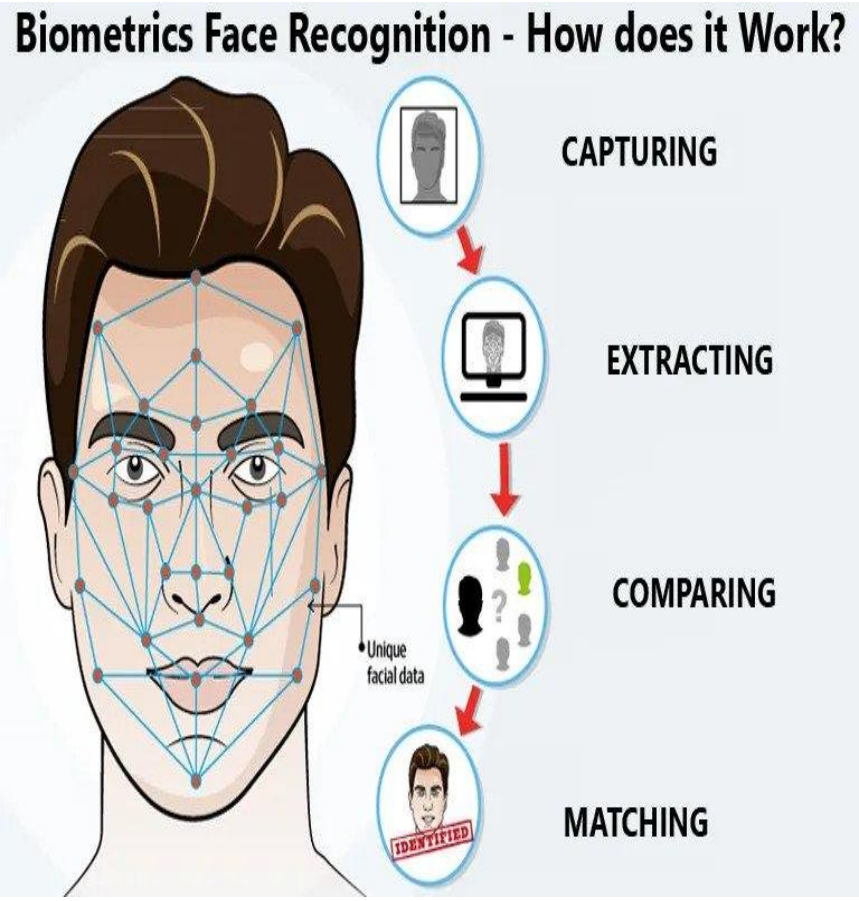

Fig 1.1. Formal Face Recognition Method.

1.2. OVERVIEW OF THE PROJECT:

- An AI face recognition project typically involves the development and deployment of software that can identify and verify individuals based on their facial features. The project may involve the following steps:

1. Data collection: Collecting a large dataset of images of faces to be used for training the AI model. This may involve sourcing images from public sources or collecting images through the use of cameras or other imaging devices.
2. Data preprocessing: Cleaning and preparing the collected images for use in training the AI model. This may involve tasks such as image cropping, resizing, and normalization.
3. Model training: Training an AI model using preprocessed image data. This typically involves using machine learning algorithms such as convolutional neural networks (CNNs) to learn to recognize different facial features and patterns.
4. Model testing: Evaluating the performance of the trained AI model on a set of test data. This helps to ensure that the model can accurately identify and verify individuals.
5. Deployment: Integrating the trained AI model into a software application or system for use in real-world scenarios. This may involve deploying the model to run on cloud-based servers or on edge devices such as smartphones or surveillance cameras.

AI face recognition technology has a wide range of potential applications, including security and surveillance, access control, and biometric authentication. However, it is important to consider the ethical and privacy implications of using such technology and to ensure that it is deployed in a responsible and transparent manner.

- AI face recognition projects involve the use of artificial intelligence and machine learning algorithms to analyze and recognize human faces in images or videos. These

projects can be used for a variety of purposes, such as:

1. Security: AI face recognition technology can be used to enhance security measures by verifying the identity of individuals before allowing access to secure areas or data.
2. Biometric authentication: AI face recognition can be used as a biometric authentication method to replace traditional passwords or PINs, providing a more secure and convenient way to access devices or applications.
3. Personalization: AI face recognition technology can be used to personalize experiences for individuals, such as recommending content or products based on their facial features.
4. Marketing: AI face recognition can be used to collect demographic data on individuals, such as age and gender, which can be used for targeted marketing campaigns.
5. Healthcare: AI face recognition can be used in healthcare to detect and diagnose medical conditions based on facial features.

AI face recognition projects typically involve training machine learning models on large datasets of labeled images or videos to recognize and classify facial features. These models can then be used to analyze new images or videos and make predictions about the identities or characteristics of individuals in the footage.

The use of artificial intelligence (AI) technology is rapidly expanding and has already made a significant impact on various industries. However, with the increasing reliance on AI, there is a growing concern about the potential for AI systems to breach privacy laws and ethical standards. This paper presents a new AI trespasser system that uses breach mail technology to detect and prevent potential privacy violations by AI systems. The system operates by continuously monitoring AI systems for any signs of data breaches and sends out alerts to the relevant parties in the form of breach emails. These

breach emails contain detailed information about the nature of the breach, its severity, and any necessary actions that need to be taken to resolve the issue. This system provides a proactive and effective solution to the problem of AI privacy breaches and will be valuable to organizations that rely on AI systems to manage sensitive information. Here the Authorized Facial features of the existing members have been captured and trained using OpenCV, and the generated model dataset will be deployed to detect the unauthorized trespasser which in turn triggers a breach mail if the detected facial features are not found in the generated/trained dataset model.

1.3 PROJECT SYNOPSIS:

Project Title:

AI Trespasser over Breach Mail

2.Project Description:

The aim of this project is to develop an AI facial recognition system for security applications. The system will use machine learning algorithms to analyze and recognize faces from a database of known individuals. The system will also be able to detect and alert security personnel to any unknown individuals or potential threats and trigger an alert mail containing the desired alert message and the picture of the unauthorized invader to the mentioned mail address. The project will be completed within a period of 4 months.

3. Project Team:

The project team will consist of a project manager, software developers, machine learning experts, and a security consultant.

4. Project Deliverables:

The project will deliver a fully functional AI facial recognition system that can be used for security applications. The system will be able to analyze and recognize faces with high accuracy and will include a user-friendly interface for security personnel. The system will also include a trigger mail feature that can get a picture of the unauthorized person and a database of known individuals and the ability to add or remove individuals from the database.

5. Project Methodology:

The project will follow an Agile development methodology with weekly sprints and regular stakeholder meetings. The system will be developed using Python and various machine-learning libraries. The team will use test-driven development (TDD) to ensure high-quality code and functionality.

6. Project Budget:

The estimated project budget is Rs.50,000, which includes hardware and software costs, salaries for the project team, and other expenses such as training and travel.

7. Project Risks and Mitigation:

The main risks associated with the project include the accuracy of the facial recognition algorithm and potential legal and ethical concerns regarding the use of facial recognition technology. To mitigate these risks, the team will conduct extensive testing and validation of the algorithm and will work closely with legal experts to ensure compliance with all applicable laws and regulations.

8. Project Impact:

The facial recognition system developed in this project will have a significant impact on security applications. It will provide a more efficient and accurate way to recognize individuals and detect potential threats. The system can be used in various settings such as airports, border checkpoints, and other high-security areas.

9. Conclusion:

This project has the potential to revolutionize security applications by providing a more efficient and accurate way to recognize individuals. The facial recognition system developed in this project will have a positive impact on security and safety in various settings.

1.4 PROJECT SCOPE:

The scope of an AI facial recognition project may include the following elements:

1. Project Objectives: The primary goal of the project is to develop a facial recognition system that can accurately identify individuals from digital images or video footage.
2. Data Collection: The project will involve collecting a large dataset of facial images and videos to train the AI model. This may involve collaborating with partners to access existing databases or collecting new data through cameras or other sources.
3. AI Model Development: The project will involve developing an AI model that can recognize facial features and match them with stored images or data. This may involve selecting an appropriate AI algorithm, testing and refining the model, and integrating it into a larger system.
4. User Interface Design: The project will involve designing a user interface for the facial recognition system, including a dashboard for managing the system and a user

interface for end users to access the system.

5. Testing and Validation: The project will involve testing the facial recognition system to ensure that it is accurate, reliable, and meets performance requirements. This may involve testing the system with various image types, different lighting conditions, and in different environments.

6. Deployment and Maintenance: Once the facial recognition system is developed and tested, it will need to be deployed and maintained over time. This may involve installing the system on hardware or cloud servers, updating the system as needed, and providing ongoing technical support.

7. Legal and Ethical Considerations: The project will need to consider legal and ethical considerations around facial recognition, such as privacy concerns and the potential for bias or discrimination.

Trespassing happens for any number of reasons. Bad actors looking to gain access to secured information or intending to do physical harm can enter seemingly “off-limits” areas and suddenly businesses are dealing with a security breach. Whether the trespassers’ motivations are malicious or otherwise, it is important for businesses to be alerted when someone is in a place they shouldn’t be. From airports and hospitals to theme parks and entertainment venues, and any type of private property in between, businesses can leverage AI-based video content analytics to extend surveillance security applications and further protect their facilities against trespassers.

1.5. OBJECTIVE OF THE PROJECT:

The primary objective of an AI trespasser over breach mail system is to develop a process that can detect and alert designated individuals or groups when an unauthorized

individual or entity breaches a restricted area or perimeter. The system uses AI and machine learning algorithms to analyze data from sensors, cameras, or other sources to detect when a trespasser is present in a restricted area.

The alert generated by the system can be sent through email or other forms of communication, providing real-time information to designated personnel who can respond quickly and take appropriate action to address the situation.

The objectives of an AI trespasser over breach mail system include:

1. Improved Security: By detecting and alerting authorized personnel to unauthorized access, the system can help to improve security in restricted areas or perimeters.

2. Faster Response Time: The real-time alert generated by the system can help to reduce response times, allowing authorized personnel to respond quickly to potential threats.

3. Reduced Risk of Loss or Damage: By identifying and responding to potential security breaches, the system can help to reduce the risk of loss or damage to assets or property.

4. Enhanced Situational Awareness: The system provides authorized personnel with real-time information about potential security breaches, helping to enhance situational awareness and facilitate quick decision-making.

5. Integration with Other Security Systems: The system can be integrated with other security systems, such as access control systems or surveillance cameras, to provide a comprehensive security solution.

Overall, the objective of an AI trespasser over breach mail system is to provide an effective and reliable solution for detecting and responding to potential security breaches, helping to improve safety and security in restricted areas or peri.

CHAPTER 2

LITERATURE SURVEY

2.1. LITERATURE SURVEY:

PAPER I TITLE: A novel DeepMaskNet model for face mask detection and masked facial recognition

AUTHOR: Naeem Ullah , Ali Javed , Mustansar Ali Ghazanfar , Abdulmajeed Alsufyani , Sami Bourouis

ABSTRACT: Coronavirus disease (COVID-19) has significantly affected the daily life activities of people globally. To prevent the spread of COVID-19, the World Health Organization has recommended the people to wear face mask in public places. Manual inspection of people for wearing face masks in public places is a challenging task. Moreover, the use of face masks makes the traditional face recognition techniques ineffective, which are typically designed for unveiled faces. Thus, introduces an urgent need to develop a robust system capable of detecting the people not wearing the face masks and recognizing different persons while wearing the face mask. In this paper, we propose a novel DeepMasknet framework capable of both the face mask detection and masked facial recognition. Moreover, presently there is an absence of a unified and diverse dataset that can be used to evaluate both the face mask detection

and masked facial recognition. For this purpose, we also developed a largescale and diverse unified mask detection and masked facial recognition (MDMFR) dataset to measure the performance of both the face mask detection and masked facial recognition methods. Experimental results on multiple datasets including the cross-dataset setting show the superiority of our DeepMaskne framework over the contemporary models

PAPER II TITLE: Masked Face Recognition for Secure Authentication

AUTHORS: Aqeel Anwar, Arijit Raychowdhury

ABSTRACT: With the recent world-wide COVID-19 pandemic, using face masks have become an important part of our lives. People are encouraged to cover their faces when in public area to avoid the spread of infection. The use of these face masks has raised a serious question on the accuracy of the facial recognition system used for tracking school/office attendance and to unlock phones. Many organizations use facial recognition as a means of authentication and have already developed the necessary datasets in-house to be able to deploy such a system. Unfortunately, masked faces make it difficult to be detected and recognized, thereby threatening to make the in-house datasets invalid and making such facial recognition systems inoperable. This paper addresses a methodology to use the current facial datasets by augmenting it with tools that enable masked faces to be recognized with low false-positive rates and high overall accuracy, without requiring the user dataset to be recreated by taking new pictures for authentication. We present an open-source tool, MaskTheFace to mask faces effectively creating a large dataset of masked faces. The dataset generated with this tool is then used towards training an effective facial recognition system with target accuracy for masked faces. We report an increase of 38% in the true positive rate for the Facenet system. We also test the accuracy of re-trained system on a custom real-world dataset MFR2 and report similar accuracy.

PAPER TITLE: Masked Face Recognition using ResNet-50

AUTHORS: Bishwas Mandal, Adaeze Okeukwu, Yihong Theis

ABSTRACT: Over the last twenty years, there have seen several outbreaks of different coronavirus diseases across the world. These outbreaks often led to respiratory tract diseases and have proved to be fatal sometimes. Currently, we are facing an elusive health crisis with the emergence of COVID-19 disease of the coronavirus family. One of the modes of transmission of COVID-19 is airborne transmission. This transmission occurs as humans breathe in the droplets released by an infected person through breathing, speaking, singing, coughing, or sneezing. Hence, public health officials have mandated the use of face masks which can reduce disease transmission by 65%. For face recognition programs, commonly used for security verification purposes, the use of face mask presents an arduous challenge since these programs were typically trained with human faces devoid of masks but now due to the onset of Covid-19 pandemic, they are forced to identify faces with masks. Hence, this paper investigates the same problem by developing a deep learning based model capable of accurately identifying people with face-masks. In this paper, the authors train a ResNet-50 based architecture that performs well at recognizing masked faces. The outcome of this study could be seamlessly integrated into existing face recognition programs that are designed to detect faces for security verification purposes.

PAPER TITLE:Deep Facial Expression Recognition

AUTHORS: Shan Li; Weihong Deng

ABSTRACT: With the transition of facial expression recognition (FER) from laboratory-controlled to challenging in-the-wild conditions and the recent success of deep learning techniques in various fields, deep neural networks have increasingly been leveraged to learn discriminative representations for automatic FER. Recent deep FER systems generally focus on two important issues: overfitting caused by a lack of sufficient training data and expression-unrelated variations, such as illumination, head pose, and identity bias. In this survey, we provide a comprehensive review of deep FER, including datasets and algorithms that provide insights into these intrinsic problems. First, we introduce the available datasets that are widely used in the literature and provide accepted data selection and evaluation principles for these datasets. We then describe the standard pipeline of a deep FER system with the related background knowledge and suggestions for applicable implementations for each stage. For the state-of-the-art in deep FER, we introduce existing novel deep neural networks and related training strategies that are designed for FER based on both static images and dynamic image sequences and discuss their advantages and limitations. Competitive performances and experimental comparisons on widely used benchmarks are also summarized. We then extend our survey to additional related issues and application scenarios. Finally, we review the remaining challenges and corresponding opportunities in this field as well as future directions for the design of robust deep FER systems.

PAPER TITLE: A Face Expression Recognition Using CNN & LBP

AUTHORS: Rahul Ravi; S.V Yadhukrishna; Rajalakshmi prithviraj

ABSTRACT: Visual interaction is an effective means of communication for human beings as social beings. Even a simple change in facial expression signifies happiness, sorrow, surprise and anxiety. The facial expressions of every person should vary in various contexts such as lighting, posture and even background. All these factors still remain an issue while recognizing facial expressions. This paper hopes to bring out a fair comparison between two of the most commonly used face expression recognition [FER] techniques and to shed some light on their precision. The methods being used here are Local binary patterns [LBP] and Convolution neural networks [CNN]. The LBP is meant as a method only for the purpose of extracting features so the Support vector machine [SVM] classifier is being utilized for classifying the extracted features from LBP. The dataset used for the purpose of testing and training in this paper are CK+, JAFFE and YALE FACE. About 70% is being utilized for training and the rest 30% is used in the testing. CNN gives better precision than any other dataset with 97.32% of recognition rate on CK+ dataset and also shows 31.82% of the YALE FACE dataset that is the least accurate we have.

2.2 REVIEWS ON CONCEPTS

The concept related to this project is data set collection, data training, CNN, OpenCV, Face detection, trigger mail with the desired message, and the photo of the unauthorized invader.

2.2.1.Data Set Collection and Training

A data set collection of related, discrete items of related data that may be accessed individually or in combination or managed as a whole entity. A data set is organized into some type of data structure. A learning dataset is a collection of data that is used to train the model. A dataset acts as an example to teach the machine learning algorithm how to make predictions. The common types of data include Text data, Image data. We have used two different datasets to train our convolutional neural network (CNN). The first dataset we used is one of the widely known public datasets called The Virat Dataset and it is taken from its official website. The Virat dataset is designed to be authentic and challenging for video surveillance domains in terms of its clarity, variety of footage, and background clutter. We have used 1700 images from the dataset which are frames of the captured surveillance footage to train our model for binary classification into human entities and non-human objects. The Virat Dataset distinguishes itself with some important characteristics. The major characteristic of the dataset is the surveillance footage is collected in natural scenes over different time spans. Another important feature of the Virat dataset is quantity and diversity. The dataset consists of captured scenes from diverse backgrounds which help to reduce the overfitting that occurs in Convolutional Neural Network (CNN) layers. The second dataset we have used to further train our Convolutional Neural Network is MPII human pose dataset. It is a state-of-the-art benchmark for the evaluation of articulated human pose estimation. The dataset consists of around 25K images containing frame captures of over 40K people. The proposed implementation has partially used this dataset

2.2.2.CNN(Convolutional Neural Network)

CNN or Convolutional Neural Network is one of the most effective machine learning algorithms for the purpose of achieving image processing. The image processing tasks performed by CNN are classification tasks. The CNN approach can effectively identify objects and other items and classify them accurately. The CNN classifier works by providing an effective label for the classified object in an image. CNN can be used to identify complex patterns in the data effectively. The CNN algorithm consists of various components, such as Tensors, neurons, layers, kernel weight, and biases along with a differentiable score function. These components are equally responsible for the execution of the CNN algorithm. The various layers of the CNN algorithm are defined below.

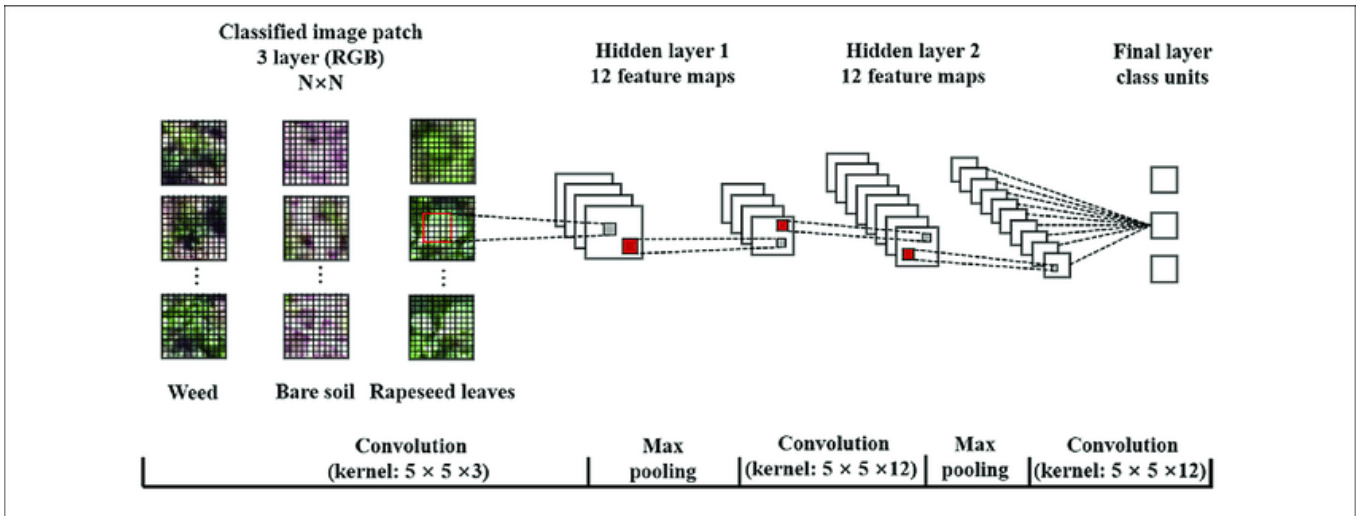

Fig.2.2.2 Convolutional Layer for RGB image

2.2.3.OPENCV(OPEN SOURCE COMPUTER VISION LIBRARY)

Open Source Computer Vision is a free and open-source library of programming functions that are broadly used in computer vision and image processing tasks. It includes colorful algorithms and tools for image and videotape processing, like point

discovery, object recognition, and face recognition. In the environment of face recognition, OpenCV provides several pre-trained face discovery models that can be used to descry faces in images or videotape aqueducts. Once the faces are detected, OpenCV can be used to prize the features of the detected faces and also compares them with a database of known faces to identify or corroborate individualities. OpenCV also provides several pre-trained models for facial recognition, similar to Eigenfaces, Fisherfaces, and Original Double Patterns Histograms(LBPH). These models use machine literacy algorithms to fete faces grounded on the uprooted features, and they can be trained on a dataset of faces to ease their delicacy. Overall, OpenCV provides an essential set of tools and algorithms for face recognition, which can be used to develop robust and accurate face recognition systems for a wide range of operations.In the field of Artificial Intelligence, Computer Vision is one of the most interesting and Challenging tasks. Computer Vision acts like a bridge between Computer Software and visualizations around us. It allows computer software to understand and learn about the visualizations in the surroundings. For Example: Based on the color, shape, and size determining the fruit. This task can be very easy for the human brain however in the Computer Vision pipeline, first we gather the data, then we perform the data processing activities and then we train and teach the model to understand how to distinguish between the fruits based on size, shape, and color of the fruit.

Currently, various packages are present to perform machine learning, deep learning, and computer vision tasks. By far, computer vision is the best module for such complex activities. OpenCV is an open-source library. It is supported by various programming languages such as R, and Python. It runs on most of the platforms such as Windows, Linux, and MacOS.

Advantages of OpenCV:

- OpenCV is an open-source library and is free of cost.
- As compared to other libraries, it is fast since it is written in C/C++.
- It works better on Systems with lesser **RAM.**
- It supports most of the Operating Systems such as Windows, Linux, and MacOS.

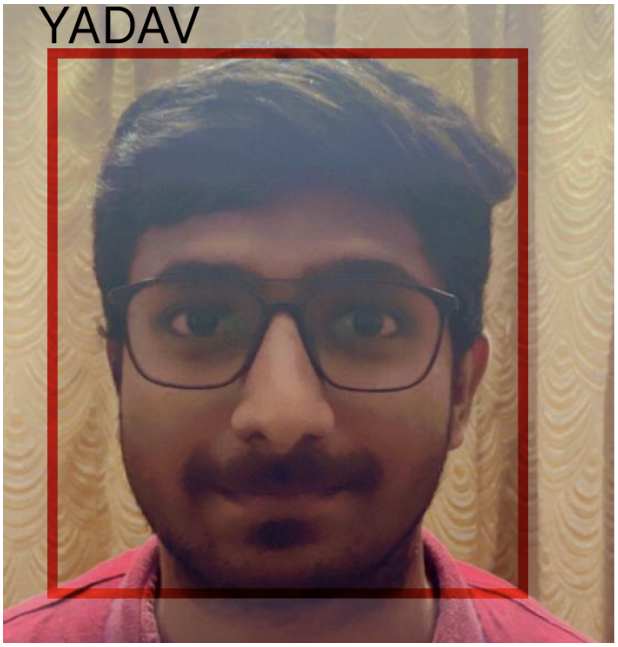

Fig 2.2.3 OpevCV Facial Recognition

2.2.4.AI Mail Triggering Process for an unauthorized invasion

The Method of an AI mail-triggering process for unauthorized invaders in restricted areas may include the following elements:

1. Objectives: The primary goal of the project is to develop an AI mail-triggering process that can identify unauthorized invaders in restricted areas and trigger an alert to designated individuals or groups.
2. Data Collection: The project will involve collecting data from sensors, cameras,

or other sources to detect when an unauthorized invader enters a restricted area. This may include installing new hardware or integrating with existing security systems.

3. AI Model Development: The project will involve developing an AI model that can analyze the data collected from the sensors and cameras to determine whether an unauthorized invader is present. This may involve selecting an appropriate AI algorithm, testing and refining the model, and integrating it into a larger system.

4. Alert Triggering: The project will involve developing a process for triggering an alert when an unauthorized invader is detected. This may include sending an email to designated individuals or groups, activating an audible alarm, or other forms of communication.

5. Testing and Validation: The project will involve testing the AI mail triggering process to ensure that it accurately detects unauthorized invaders and triggers alerts in a timely manner. This may involve testing the system with various scenarios, such as different lighting conditions or environmental factors.

6. Deployment and Maintenance: Once the AI mail triggering process is developed and tested, it will need to be deployed and maintained over time. This may involve installing the system on hardware or cloud servers, updating the system as needed, and providing ongoing technical support.

7. Legal and Ethical Considerations: The project will need to consider legal and ethical considerations around privacy and data protection, as well as the potential for false positives or other issues that could impact the effectiveness of the system.

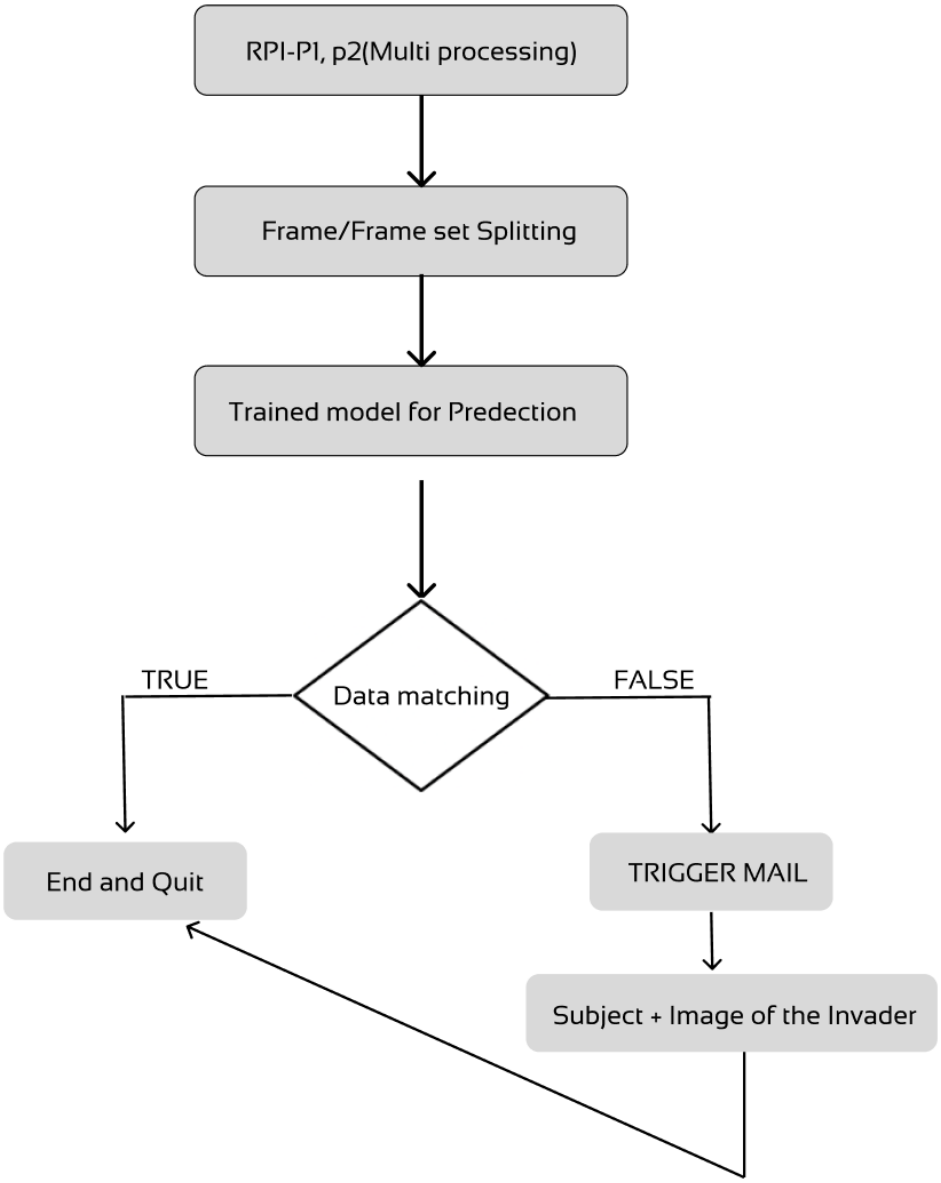

Fig 2.2.4 Trigger Mail process for Unauthorized Invader

CHAPTER 3
SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

There are several existing systems for AI facial recognition that are widely used in various industries, some of which are:

1. Amazon Rekognition: Amazon Rekognition is a cloud-based facial recognition service that can recognize, identify, and verify faces in images and video footage. It can be used in various industries, such as security, retail, and entertainment.
2. Microsoft Azure Face: Microsoft Azure Face is a facial recognition API that can detect, analyze, and identify faces in images and video. It can be used for identity verification, access control, and customer engagement.
3. Google Cloud Vision API: Google Cloud Vision API provides facial recognition capabilities that can detect faces, recognize facial expressions, and verify identities. It can be used in various industries, including healthcare, finance, and transportation.
4. FaceFirst: FaceFirst is a facial recognition platform that can recognize faces in real-time and match them against a watchlist of known individuals. It is used in various industries, such as retail, transportation, and law enforcement.
5. Kairos: Kairos is a facial recognition API that can detect, verify, and analyze faces in images and video. It is used in various industries, including healthcare, finance, and security.
6. NEC NeoFace: NEC NeoFace is a facial recognition technology that can match faces against a large database of known individuals. It is used in various industries, such as law enforcement, border control, and aviation security.

3.1.1. LIMITATIONS:

- Low System efficiency.
- Platform Dependent.

3.2. PROPOSED SYSTEM:

This paper presents a new AI trespasser system that uses breach mail technology to detect and prevent potential privacy violations by AI systems. The system operates by continuously monitoring AI systems for any signs of data breaches and sends out alerts to the relevant parties in the form of breach emails. These breach emails contain detailed information about the nature of the breach, its severity, and any necessary actions that need to be taken to resolve the issue. This system provides a proactive and effective solution to the problem of AI privacy breaches and will be valuable to organizations that rely on AI systems to manage sensitive information. Here the Authorized Facial features of the existing members have been captured and trained using OpenCV, and the generated model dataset will be deployed to detect the unauthorized trespasser which in turn triggers a breach mail if the detected facial features are not found in the generated/trained dataset model.

3.1.1. ADVANTAGES:

- Greater system efficiency.
- Platform Independent.
- Better security, Easy integration, and automated identification.

3.3.REQUIREMENTS ANALYSIS:

3.3.1. HARDWARE REQUIREMENTS

The Hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete specification of the whole system. They are used by the software engineers as the starting point for the system design. It shows what the system does not and how it should be implemented.

- 4 GB RAM (Minimum)
- 80 GB HDD
- Dual Core processor
- CDROM (installation only). VGA resolution monitors
- Microsoft Windows 98/2000/NT with service pack 6 / XP with service pack 2/ Windows 7 with service pack 2
- SQL Server 2008 R2

3.1.1. SOFTWARE REQUIREMENTS

The software specification are the specification of the system. It should include both the specification and a definition of the requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide the basis for creating the software requirement specification. It is useful in estimating costs, planning team activities, performing tasks, and tracking the team's progress throughout the development activity.

- **DATASET:** Image dataset.
- **TOOLS:** Visual Studio Code, Cmake.
- **LIBRARIES**: CNN, OpenCV, and Dlib

CHAPTER 4
SOFTWARE DESCRIPTION

4.1 VISUAL STUDIO CODE

Visual Studie Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with Electron Framework, for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting code completion, snippets, code refactoring, and embedded Git. Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

4.2 Cmake:

CMake is an extensible, open-source system that manages the build process in an operating system and in a compiler-independent manner. Unlike many cross-platform systems, CMake is designed to be used in conjunction with native-built environments. The CMake Tools extension integrates Visual Studio Code and CMake to make it easy to configure, build, and debug your C++ project.CMake handles the difficult aspects of building software such as cross-platform builds, system introspection, and user-customized builds, in a simple manner that allows users to easily tailor builds for complex hardware and software systems.CMake is driven by the CMakeLists. txt files are written for a software project. The CMakeLists files determine everything from which options to present to users, to which source files to compile.

The common steps to build, test, and install software from source code based on CMake are as follows:

- Extract source files.
- Create a build directory and change it.
- Run CMake to configure the build tree.
- Build the software using the selected build tool.
- Test the built software.
- Install the built files.

CHAPTER 5

SYSTEM DESIGN

5.1. ARCHITECTURE DIAGRAM:

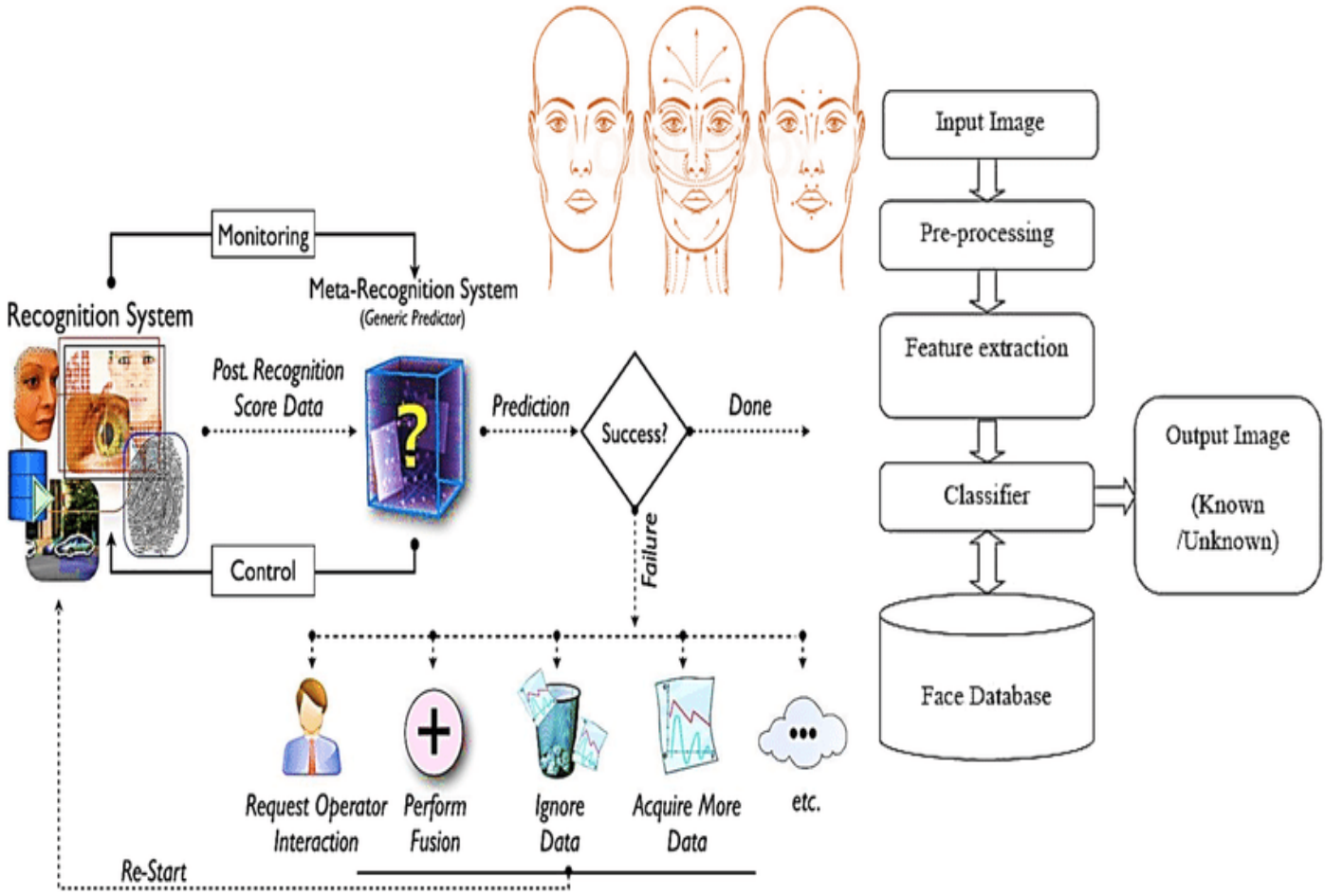

Fig 5.1.The Architecture of the system

5.2 DATAFLOW DIAGRAMS:

5.2.1 LEVEL 0 DFD:

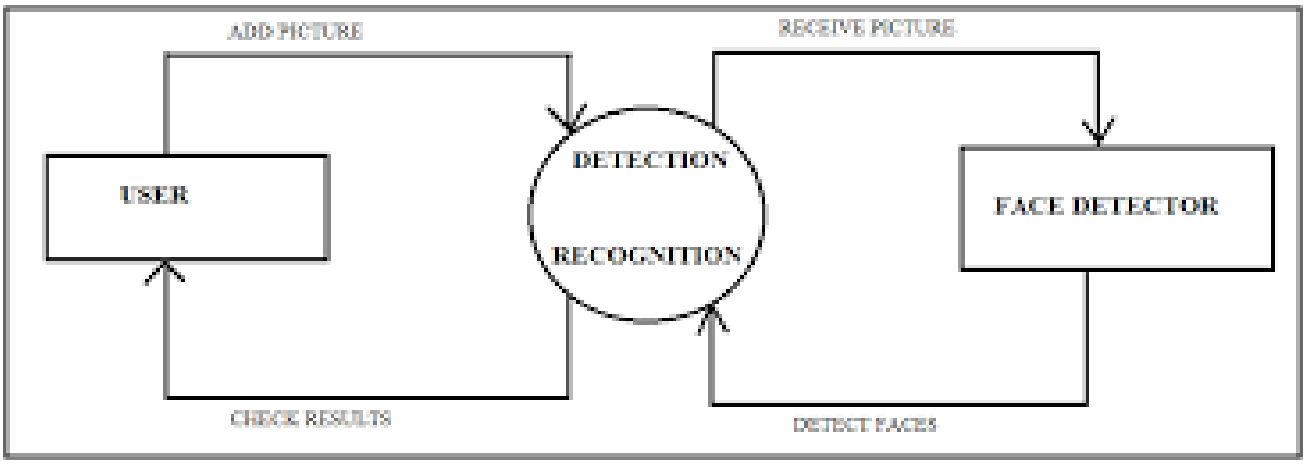

Fig 5.2.1 Level 0 DFD

5.2.2 LEVEL 1:

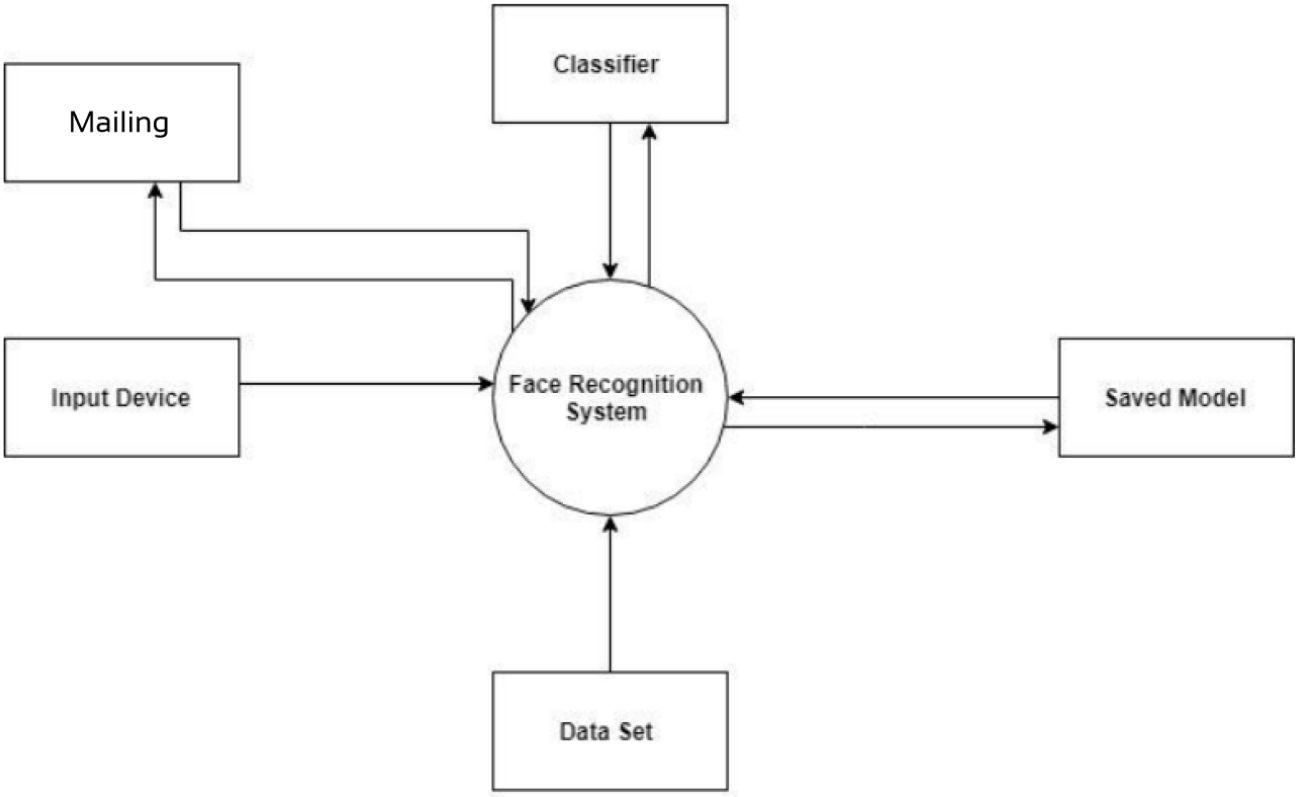

Fig 5.2.2-Level 1 DFD

5.2.1 LEVEL 2:

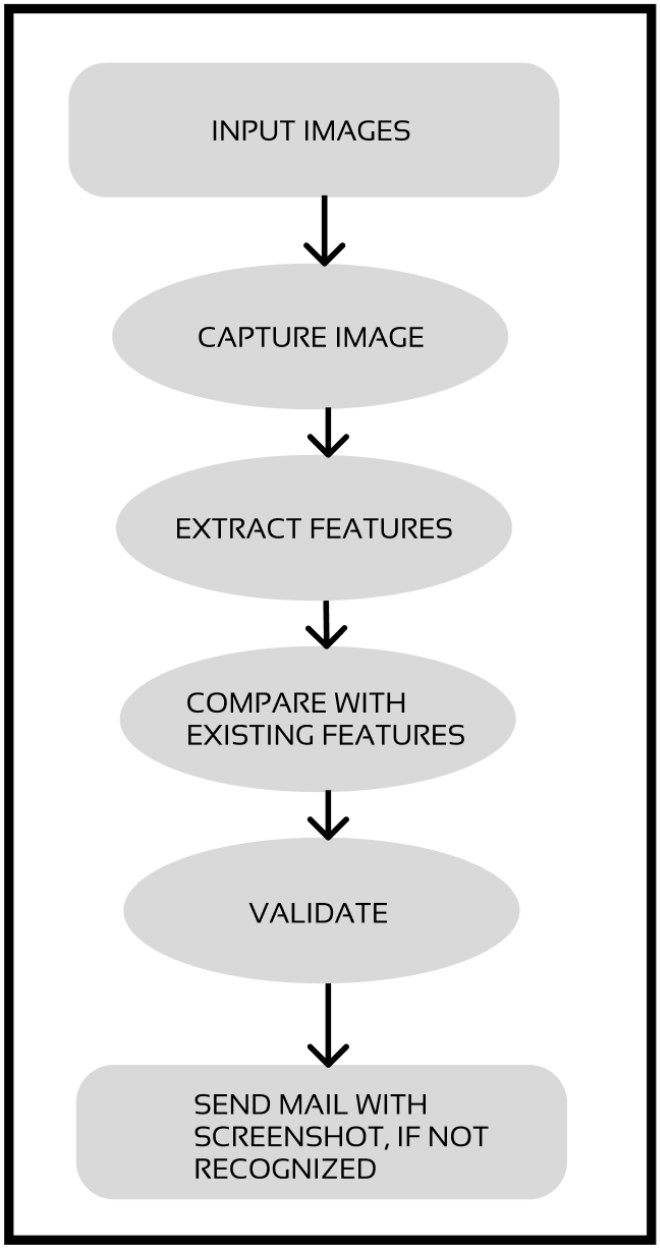

Fig 5.2.3- Level 2 DFD

CHAPTER-6 MODULES

6.1 MODULE DESCRIPTION:

AI face recognition is a module that uses artificial intelligence (AI) algorithms to identify and verify individuals by analyzing their facial features. It is commonly used in security and surveillance systems, as well as in applications such as digital identity verification, access control, and attendance tracking. The AI face recognition module typically consists of several sub-modules, including face detection, face alignment, feature extraction, and matching.

- Face detection: This module identifies the location and size of faces in an image or video stream. It is often based on deep learning techniques that can accurately detect faces in various lighting conditions, orientations, and poses.

- Face alignment: This module corrects the orientation and position of detected faces to ensure that they are aligned in a standard format for further processing. This step can help improve the accuracy of feature extraction and matching.

- Feature extraction: This module extracts a set of unique features from each detected face, such as the distance between the eyes, the shape of the nose, and the contour of the jawline. These features are used to create a mathematical representation, or face template, of each individual's face.

- Matching: This module compares the face template of a person to a database of known face templates to determine if there is a match. The matching process can be

based on various algorithms, such as cosine similarity or neural networks, that compute a similarity score between two face templates. The threshold for accepting a match can be adjusted to control the false positive and false negative rates of the system.

Overall, the AI face recognition module can provide fast and accurate identification of individuals, even in complex and challenging scenarios. However, it also raises concerns about privacy, bias, and potential misuse, which need to be carefully addressed and regulated.

CHAPTER 7

CODING

SAMPLE SOURCE CODE:

```
import face_recognition
import pickle
import mail
import numpy as np
#import serial
import time
import sys
import cv2

#use pickle to mine through dataset
mob=mail.mclass()

def compare_face():

    with open('dataset_faces.dat','rb') as f:
        all_face_encodings=pickle.load(f)

    #grab list of names and list of encodings

    face_names = list(all_face_encodings.keys())
    face_encodings = np.array(list(all_face_encodings.values()))


    #unlnown face encoding

    unknown_image = face_recognition.load_image_file("./still.jpeg")
    unknown_encoding = face_recognition.face_encodings(unknown_image)[0]
```

```
    #get result of compared faces with encoding
    results = face_recognition.compare_faces(face_encodings, unknown_encoding)
    print(results)
    if True not in results:
        print("Unknown face detected")
        mob.multimediamail("Tresspasser alert from Home !!")
    #print result with list of names with true or false
    names_with_result=list(zip(face_names,results))
    print(names_with_result)

def detect_face():
    face_cascade = cv2.CascadeClassifier("./haarcascade_frontalface_default.xml")
    print("iam here hello")
    #To capture the video stream from webcam.
    cap = cv2.VideoCapture(0)
    print("Getting camera image...")
    cock=0
    while 1:

        ret, img = cap.read()
        cv2.namedWindow('img', cv2.WINDOW_NORMAL)
        #cv2.resizeWindow('img', 500,500)
        cv2.line(img,(500,250),(0,250),(0,255,0),1)
        cv2.line(img,(250,0),(250,500),(0,255,0),1)
        cv2.circle(img, (250, 250), 5, (255, 255, 255), -1)
        gray  = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3)
        #fob=facerec.facedetect()

    #detect the face and make a rectangle around it.
        for (x,y,w,h) in faces:
            #print(faces)
            no_face=faces.shape[0]
            #print(" i found "+str(no_face)+" faces in the image")
```

```
            try:
                if(cock==50):
                    cv2.imwrite("still.jpeg", img)
                    compare_face()
                    break


            except NameError: pass
            cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),5)
            roi_gray  = gray[y:y+h, x:x+w]
            roi_color = img[y:y+h, x:x+w]
            cv2.imshow('img',img)

        #Hit 'Esc' to terminate execution
        if cock>50:
            break
        cock+=1



detect_face()
```

Trigger Mail():

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

class mclass:
    def __init__(self):
        print("iam inside")
```

```
def sendname(self,sname,to="yadavraj2705@gmail.com"):
    fromaddr = "ugendarraj2002@gmail.com"
    # creates SMTP session
    s = smtplib.SMTP('smtp.gmail.com', 587)

    # start TLS for security
    s.starttls()

    # Authentication
    s.login(fromaddr, "cxoepxjecfddhbxy")

    # message to be sent
    message = sname

    # sending the mail
    s.sendmail("ugendarraj2002@gmail.com", "yadavraj2705@gmail.com", message)

    # terminating the session
    s.quit()
def multimediamail(self,subj):

    body="Message from Headquaters"
    fromaddr = "ugendarraj2002@gmail.com"
    toaddr = "yadavraj2705@gmail.com"

    # instance of MIMEMultipart
    msg = MIMEMultipart()

    # storing the senders email address
    msg['From'] = fromaddr

    # storing the receivers email address
    msg['To'] = toaddr

    # storing the subject
```

```
msg['Subject'] = subj

# attach the body with the msg instance
msg.attach(MIMEText(body, 'plain'))

# open the file to be sent
filename = "still.jpeg"
attachment = open("./still.jpeg", "rb")

# instance of MIMEBase and named as p
p = MIMEBase('application', 'octet-stream')

# To change the payload into encoded form
p.set_payload((attachment).read())

# encode into base64
encoders.encode_base64(p)

p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

# attach the instance 'p' to instance 'msg'
msg.attach(p)

# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login(fromaddr, "cxoepxjecfddhbxy")

# Converts the Multipart msg into a string
text = msg.as_string()
```

```
        # sending the mail
        s.sendmail(fromaddr, toaddr, text)

        # terminating the session
        print("mail sent")
        s.quit()


#ob=mclass()
#ob.newsbbc()
# ob.weather()
#ob.multimediamail("Tresspasser allert from Lucy !!")
#ob.sendname(sname="Hello UGENDAR RAJ")
```

Main.app()

```
import comparison_of_image_with_encoding1 as facerec
import numpy as np
#import serial
import time
import sys
import cv2

#exe_time=time.time()+40
#sys.path.append('/usr/local/lib/python2.7/site-packages')

#Setup Communication path for arduino (In place of 'COM5' (windows) or
"/dev/tty.usbmodemxxx' (mac) put the port to which your arduino is connected)
#arduino = serial.Serial(port='/dev/ttyACM0', baudrate=9600)
#time.sleep(2)
print("Connected to Arduino...")

#importing the Haarcascade for face detection
face_cascade = cv2.CascadeClassifier("./haarcascade_frontalface_default.xml")
print("iam here hello")
```

```
#To capture the video stream from webcam.
cap = cv2.VideoCapture(0)
print("Getting camera image...")
#Read the captured image, convert it to Gray image and find faces
global cock
global a

a=0
#path="./images/pic1.jpg"
cock=0
fob=facerec.facedetect()
while 1:

    ret, img = cap.read()
    cv2.namedWindow('img', cv2.WINDOW_NORMAL)
    #cv2.resizeWindow('img', 500,500)
    cv2.line(img,(500,250),(0,250),(0,255,0),1)
    cv2.line(img,(250,0),(250,500),(0,255,0),1)
    cv2.circle(img, (250, 250), 5, (255, 255, 255), -1)
    gray  = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3)
    #fob=facerec.facedetect()

#detect the face and make a rectangle around it.
    for (x,y,w,h) in faces:
        #print(faces)
        no_face=faces.shape[0]
        #print(" i found "+str(no_face)+" faces in the image")
        try:
            if(cock==50):
                #cock=1
                fob.unknownface(img)
                fob.printname(no_face)
                del cock
                #clock=int(input())
```

```
            #cv2.imshow("image",img)
            #cv2.waitKey(0)
         cock+=1#(cock+1)%100
      except NameError: pass
#        if(time.time()>=exe_time):
#            fob.unknownface(img)
#            fob.printname()
#            exe_time=time.time()+120
#
      cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),5)
      roi_gray  = gray[y:y+h, x:x+w]
      roi_color = img[y:y+h, x:x+w]


      arr = {y:y+h, x:x+w}
#        print (arr)

#        print ('X :',x)
#        print ('Y :',y)
#        print ('x+w :' ,(x+w))
#        print ('y+h :' ,(y+h))

#  #Center of roi (Rectangle)
#        xx = int(x+(x+h)/2)
#        yy = int(y+(y+w)/2)
#        cv2.circle(img, (xx, yy), 5, (0, 255, 255), 2)
#        print (xx)
#        print (yy)
#        center = (xx,yy)

 # sending data to arduino
#        print("Center of Rectangle is :", center)
#        data = "X{0:d}Y{1:d}Z".format(xx, yy)
#        print(type(data))
#        print(type(data.encode()))
```

```
#         print ("output = '" +data+ "'")
#         arduino.write(data.encode())

#Display the stream.
    cv2.imshow('img',img)

#Hit 'Esc' to terminate execution
    k = cv2.waitKey(30) & 0xff
    if k == 27:
       break

#face.py
#Displaying face.py.
```

CHAPTER-8

SCREENSHOTS

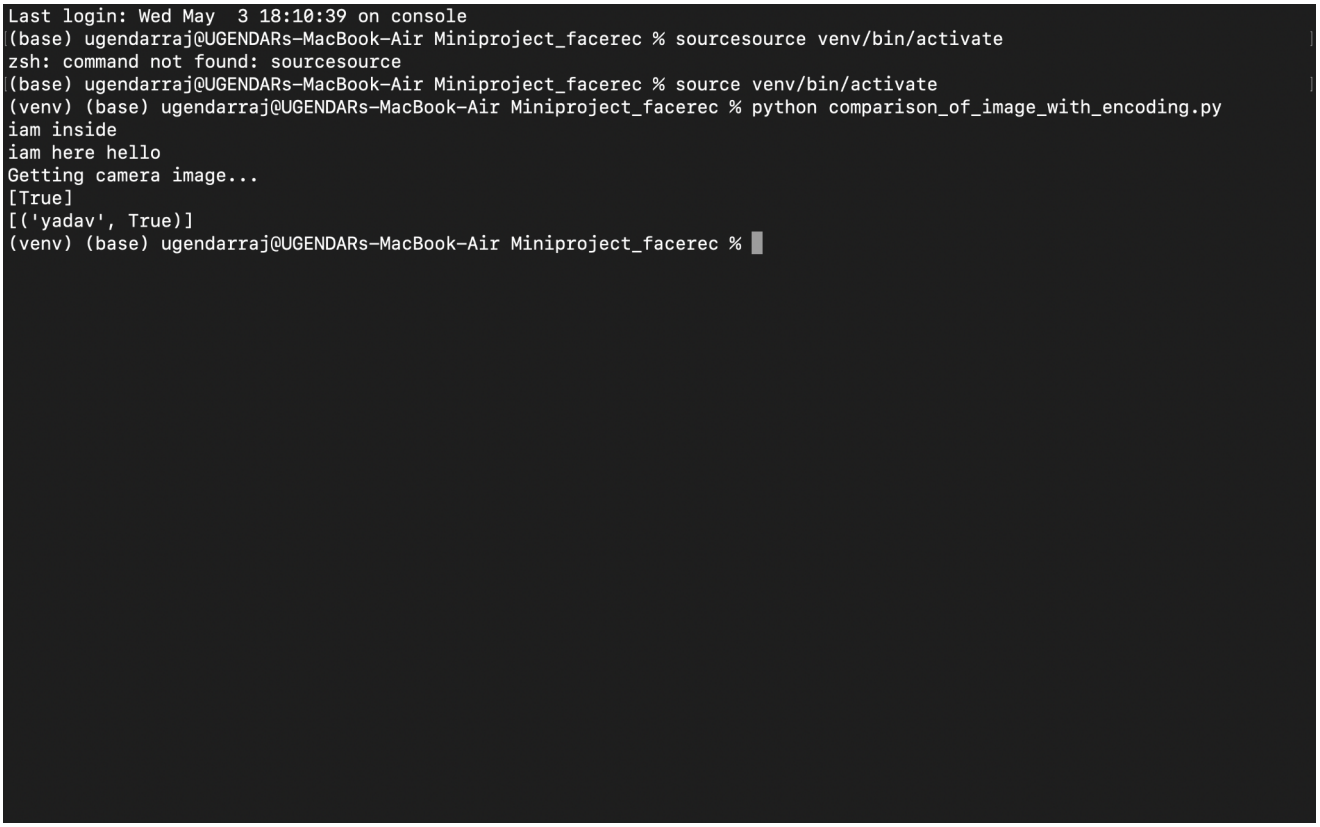

Figure.8 Terminal code to get into the Environment.

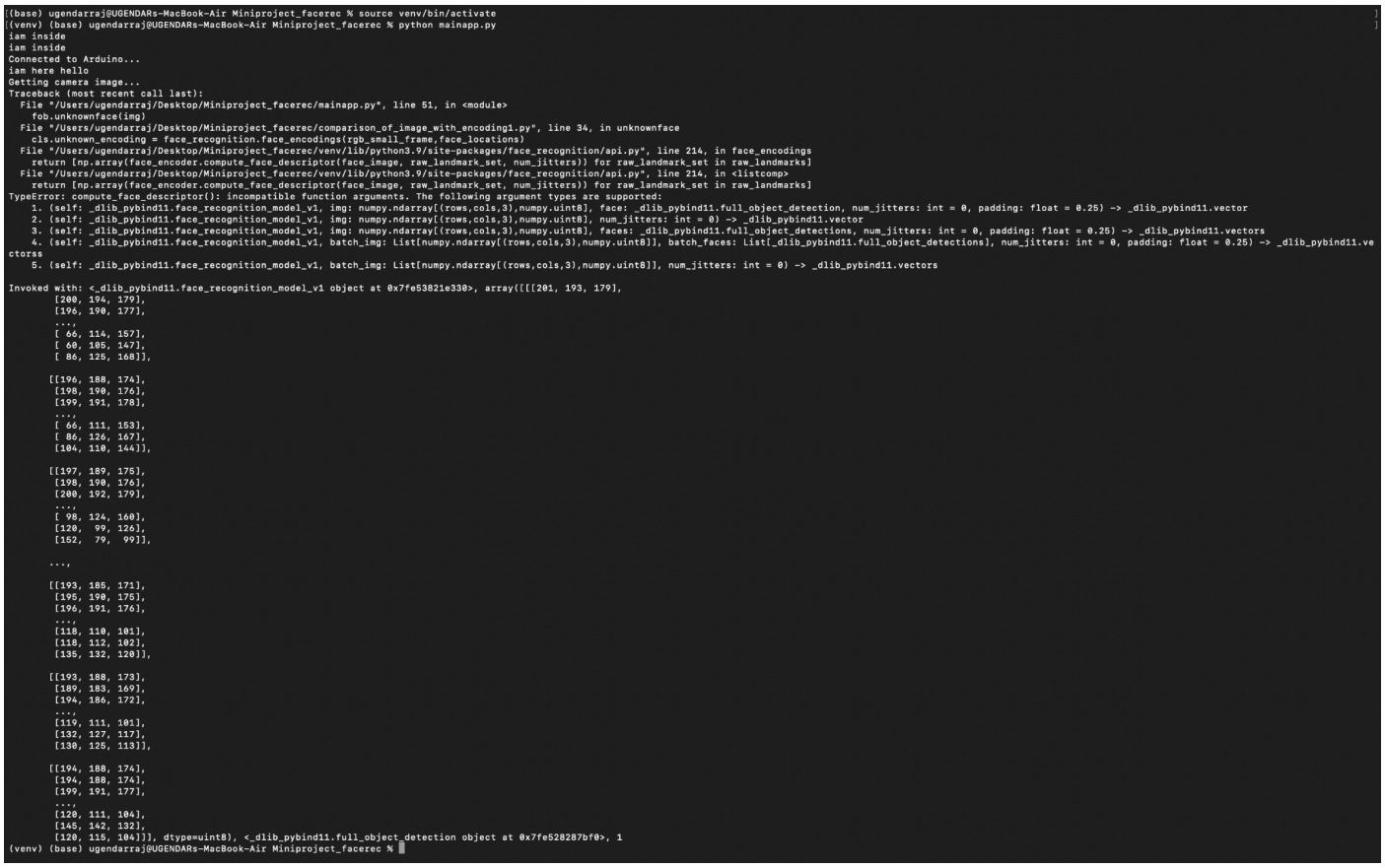

Figure 8.2 Pixels of array for the Images

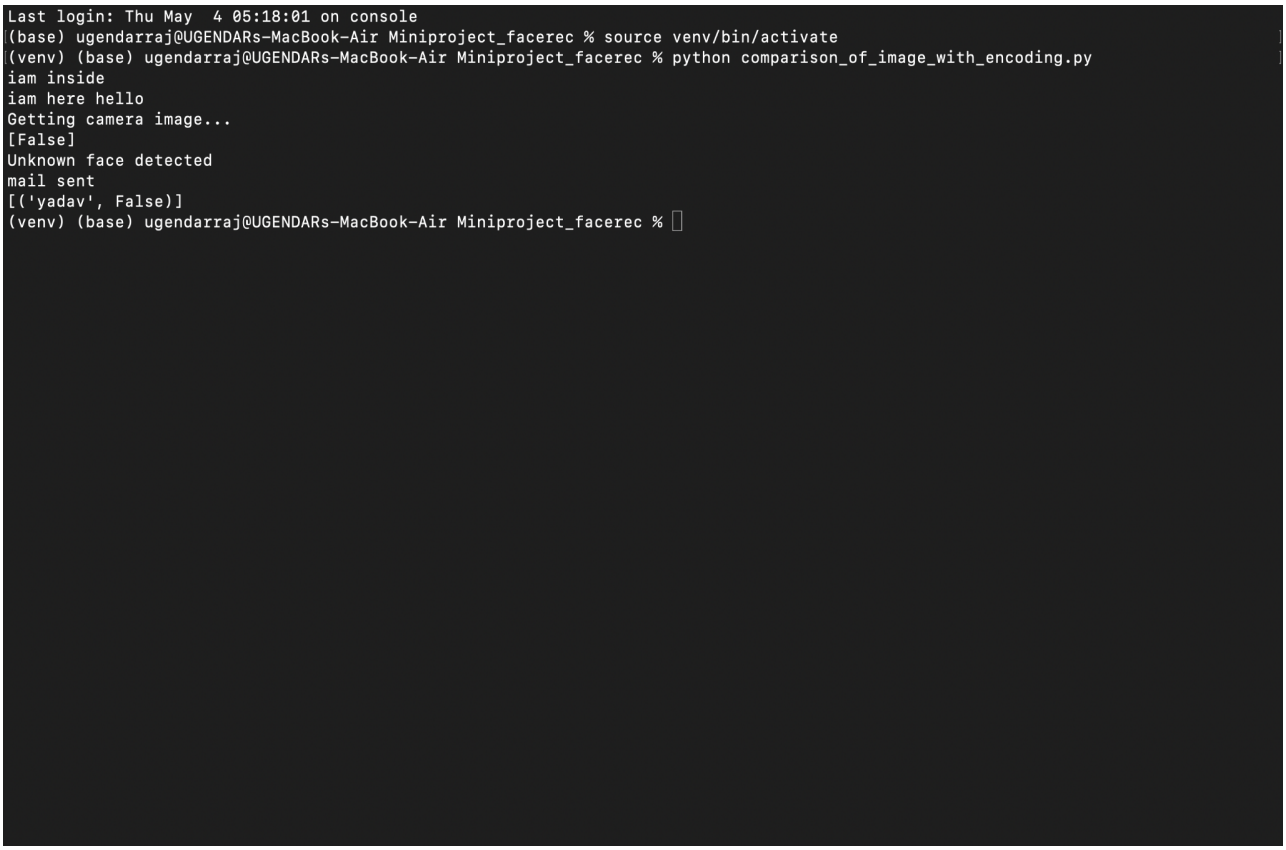

Figure 8.3 Sample output of unauthorized Faces.

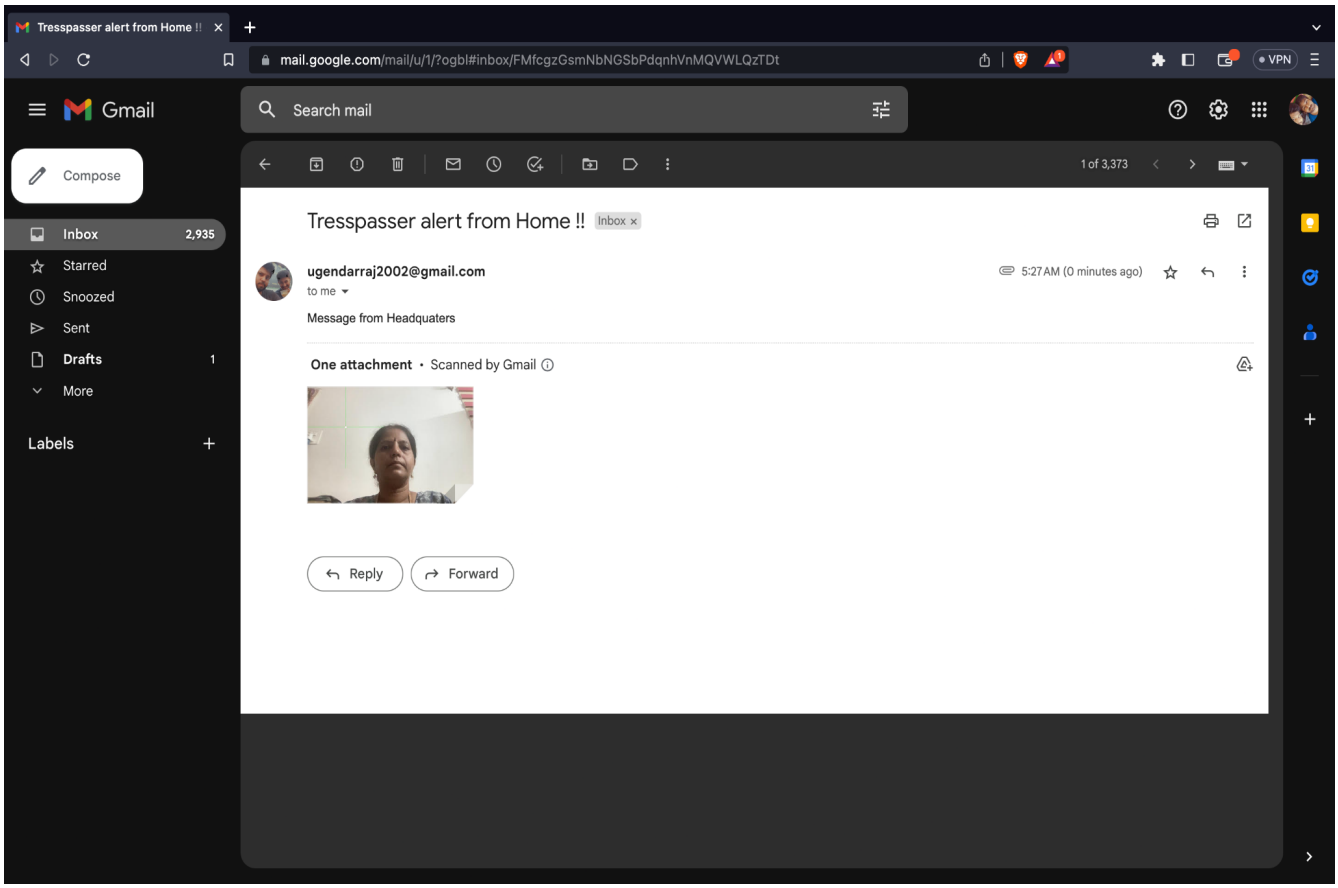

Figure 8.4 Sample Output for Trigger Mail Feature.

CHAPTER 9

CONCLUSION & FUTURE WORK

9.1 CONCLUSION

In conclusion, our study has shown that Trespasser can be easily identified and take appropriate action over them. These results have important implications for the future development of AI technology.Overall, our research has contributed to the growing body of knowledge in the field of AI. By Triggering the mail, we have advanced our Exploration of the usage of AI for Unauthorized Trespassers.

9.2 FUTURE WORK

This model can be enhanced better by using a High-end AI camera and using that in highly secured placed like Army and Defence force sector. Various kind of Airborne drones can be created from this model to deploy them into highly secured areas.

There are currently no regulations in the United States expressly covering the biometric data of a person. Facial recognition devices are already being tested or implemented for airport protection, and it is reported that their faceprint has now been produced by more than half the United States populace. Information may be collected and processed by a facial recognition program, and a person does not even recognize it. Then, a hacker might reach the details, and the knowledge of a person would propagate without even realizing it. Government entities or marketers may use this data to monitor individuals too. Worse still, a false positive may include a person for a crime they are not.

CHAPTER - 10

REFERENCES

The below-mentioned References are used as a base paper and reference website to explore and acquire prerequisite knowledge for our research paper.

[1] Ullah, Naeem, et al. ”A novel DeepMaskNet model for face mask detection and masked facial recognition.” Journal of King Saud University- Computer and Information Sciences (2022).

[2] Anwar, Aqeel, and Arijit Raychowdhury. ”Masked face recognition for secure authentication.” arXiv preprint arXiv:2008.11104 (2020).

[3] Mandal, Bishwas, Adaeze Nwokeukwu, and Yihong Theis. ”Masked face recognition using ResNet-50.” arXiv preprint arXiv:2104.08997 (2021).

[4] Mundial, Imran Qayyum, et al. ”Towards facial recognition problem in COVID-19 pandemic.” 2020 4th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM). IEEE, 2020.

[5] Golwalkar, Rucha, and Ninad Mehendale. ”Masked-face recognition using deep metric learning and FaceMaskNet-21.” Applied Intelligence (2022): 1-12.

[6] Wang, Zhongyuan, et al. ”Masked face recognition dataset and application.” arXiv preprint arXiv:2003.09093 (2020).

[7] Dlib, https://github.com/davisking/dlib.

[8] OpenCV, https://github.com/opencv/opencv .

[9] C Qinghua. Analysis of face recognition technology based on deep learning [J]. Computer products and circulation, 2020(05): 136.

[10] R. Gupta, S. Tanwar, F. Al-Turjman, P. Italiya, A. Nauman, and S. W. Kim, “Smart contract privacy protection using ai in cyber-physical systems: Tools, techniques, and challenges,” IEEE Access, pp. 1–1, 2020.

[11] S.Li and W. Deng, “Deep facial expression recognition: A survey,” IEEE Transactions on Affective Computing, pp. 1–1, 2020.

[12] A. Fathima and K. Vaidehi, “Review on facial expression recognition system using machine learning techniques,” in Advances in Decision Sciences, Image Processing, Security and Computer Vision, pp. 608–618, Springer, 2020.

[13] R. Ravi, S. Yadhukrishna, et al., “A face expression recognition using cnn & lbp,” in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 684–689, IEEE, 2020.

[14] A. L. A. Ramos, B. G. Dadiz, and A. B. G. Santos, “Classifying emotion based on facial expression analysis using Gabor filter: A basis for adaptive effective teaching strategy,” in Computational Science and Technology, pp. 469–479, Springer, 2020.

[15] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, “Machine learning models for secure data analytics: A taxonomy and threat model,” Computer Communications, vol. 153, pp. 406 – 440, 2020.