

Security Technologies Using Facial Recognition

Ramiz Salama^{a*}, Computer Engineering Department, Near East University, Nicosia, Cyprus.

Mohamed Nour^b, Computer Engineering Department, Near East University, Nicosia, Cyprus.

Suggested Citation:

Ramiz Salama & Mohamed Nour (2023). Security Technologies Using Facial Recognition. *Global Journal of Computer Sciences: Theory and Research*. 13(1), 01-27. <https://doi.org/10.18844/gjcs.v13i1.8294>

Received from; October 30, 2022, revised from; December 22, 2022 and accepted from February 05.

Selection and peer review under responsibility of Assoc. Prof. Dr. Ezgi Pelin YILDIZ, Kafkas University, Turkey.

©2023 United World Center of Research Innovation and Publication. All rights reserved.

Abstract

Faces are one of the simplest methods to determine a person's identity. Face recognition is a unique identifying method that uses an individual's traits to determine the identity of that individual. The proposed recognition process is divided into two stages: face recognition and object recognition. Unless the item is very close, this procedure is very rapid for humans. The recognition of human faces is introduced next. The stage is then reproduced and used as a model for facial image recognition (face recognition). That's one of the professionally created and well-researched biometrics procedures. The eigenface approach and the Fisher face method are two common face recognition pattern algorithms that have been developed. Recognition of facial images The Eigenface approach is based on the reduction of face dimensional space for facial traits using Principal Component Analysis (PCA). The major goal of applying PCA on face recognition was to generate Eigen faces (face space) by identifying the eigenvector corresponding to the face image's biggest eigenvalue. Image processing and security systems are areas of interest in this research face recognition integrated into a security system.

Keywords: face recognition, security systems, camera, python;

Introduction

In today's technological world, discovering and identifying a person has become an essential duty for a variety of security reasons. Since the dawn of technology, advancements in the field of facial recognition have taken an active essential role in today's world. A Face Recognition Security System, which can identify intruders to restricted areas, has been geometrically rising in present era. As an individual arrives the zone in question, the camera takes a series of photos, which are then uploaded to the program, where they are processed and linked to a database of trusted people.

Home security systems in the modern era provide multiple servers for the owner, such as crime motion detection, which alerts the owner to any unusual activities, natural disasters that may occur, such as a fire breaking out, in which case the system will notify the competent authorities in case of

* ADDRESS FOR CORRESPONDENCE: Ramiz Salama*, Computer Engineering Department, Near East University, Nicosia, Cyprus.
E-mail address: ramiz.salama@neu.edu.tr / Tel.: +9-533-8418142

emergency, allows easy and safe access to the owner's house, and keeps the owner informed of everything that is happening around him from his environment. Furthermore, technology is fast expanding, forcing more people to rely on it, giving a range of services for the everyday barriers encountered by individuals, which will assist more people in overcoming their daily life challenges. Winter is approaching, according to the government, which means the sun is sinking sooner and the night hours are becoming longer, and what it entails. According to the UK Daily Mail, police data show that about 31 percent of house break-ins occurs between 1:00 AM and the late sunrise, dubbed "the hour of thieves." (Satari et al., 2014), That is why steps have been made to reduce such occurrences, but first, allow me to inform you [1-5].

What's up? To assist a person in recognizing who is visiting their houses and to alert authorities if a suspicious person comes by. Or, even better, to assist those who are visually handicapped or who suffered from amnesia in earlier events. A billion-dollar industry is built around the concept of "security," a primary word that has represented one of the major issues since the ancient times when cavemen were trying to figure out how to secure their food to our current time where security intervenes on a daily basis. According to a Statista research paper, this industry cost 12.1 billion u.s. dollars in 2020 and is projected to increase to nearly billion us dollars by 2025 (Norval, & Prasopoulou, 2017).

Our fear of crime has been and continues to be a key issue for many countries attempting to combat it. Because of this anxiety, an increase in the purchase of home security systems has demonstrated a readiness to prevent crime and violence. Over the last two decades, researchers have focused on the elements that have a significant impact on the type of crime. These studies include (Hale, 1996; Lupton & Tulloch, 1999; Farrall & Gadd, 2004; Moore & Shepherd, 2007), among others. Despite the fact that such studies have provided insight on the nature of crime, they have disregarded the importance of home security systems in deterring crime [6-10]. As a consequence, I placed a premium on such systems in my study (Satari et al. 2014; Pawar et al., 2018).

As previously said, various hypotheses have been advanced on how crime was reduced in the recent decade due to improving elements such as economic, social, and environmental. The graphic above shows how the growing use of home security systems has certainly reduced the ratio of crimes. This project is about home - or organization - security using facial recognition. According to the strategy, this system was separated into five tasks in this project:

- The first duty is to get an image of whoever is standing at the door or waiting for permission.
- The second goal is to use face recognition to identify the individual.
- Third, make a judgment on whether or not to allow such individual authorization.
- Fourth, notify the homeowner - or the person in control - whether the individual is recognized or not, as well as the authorization status.
- Fifth task: Allow the homeowner - or the person in control - to manually grant authorization to unidentified persons.

Aims and Goals

The purpose of this project that I'm presenting is to design a home security system that includes a CCTV camera that captures a picture of the person at the door so that they may be identified facially afterwards. This will provide the necessary protection for the home owner and keep him awake in the event of a burglary or other undesired activity. A remote access system will give the homeowner with

a regulated environment that will improve his level of life, protect him, and aid him in his everyday activities.

Many flaws in previous systems have been found, such as problems detecting and categorizing people in low or dim lighting, poor image capture quality, the cost of training the database, and so on. The bulk of the flaws in the previous systems, however, are addressed in this research. The ultimate aim is to minimize the quantity of photographs required for training and to notify the administrator as soon as the individual is detected (Satari, 2014; McClellan, 2019).

Difficulties

The main issue that faced me was the lack of experience in this project field due to my lack of interest and lack of information, which resulted in increasing the scale of researches and information gathering about each of the specific tasks. Furthermore, one of the challenges encountered is a lack of practical experiences since, as previously stated, I focused more on the theoretical side of my studies in most of my studies, thus I had limited access to the practical field, which was done in a small number of courses throughout time. The technical issues were identifying and selecting the proper hardware and software that would be suited for the homeowner and whether it would fulfill his demands or not, as it is not a standard home security system, but rather a more advanced system that performs a more complex duty.

Moreover, technical difficulties with the facial identification system are that the system requires a high level of accuracy, which can be affected by a variety of conditions, such as the individual's pose difference, which can affect the face scan, and the expression of the human's face, which usually changes due to the state of emotions, which can affect the scan. Several difficulties were uncovered during the examination of past studies. overcoming standardized facial image classification difficulties, incorporating face pattern harvesting and similarity analysis, recovering nearby facial images that genuinely align its query, harvesting precise and reliable or right characteristics from low quality or blurred images, even if images captured in poor lighting are only a small portion of them (Nigam et al. 2021; Kostka et al. 2021).

Regarding the notification part the possible difficulties were:

1. False alarms

Erroneous alarms arise as a result of incorrect system operations; they occur less commonly when the system has a minor malfunction, leading the message to be false.

2. System dysfunction

The system may shut down for unknown reasons, leading the notice to be deactivated, resulting in poor protection for the owner.

3. Owner notifications

The owner may encounter a circumstance in which the notifications do not reach; for example, if the owner's phone runs out of power or is in quiet mode, the incoming notice will not be read.

4. Maintenance

Such systems need constant maintenance in order for the system to be effective and error-free, as well as to avoid breakdowns that result in negative withdrawals.

5. Power

To work, the notification system requires constant electricity and internet access. The system will not work if any of the connections fails, the cameras will disconnect owing to the power connection

shutting down, and the notice will not reach the owner at the specified time due to the internet connection failing.

6. Other creatures disturbing the motion detector

Other organisms, such as animals, may interfere with and upset the motion detector, resulting in erroneous detection and incorrect warning to the homeowner.

Contribution

For centuries, our major concerns have been safety and security. Humans nowadays rely on technology in every part of their lives. Every day, technology is employed to help us in our daily lives. Technology has become so pervasive on a daily basis that it has become difficult not to utilize it. Mobile phones, laptop computers, and other forms of technology may have a significant impact on one's life. Home security systems are a new form of advanced technology that has just been produced that is dramatically lowering crime rates and giving a better degree of safety for homeowners. They may also perform a variety of activities like as providing protection, keeping residents informed about the environment, and notifying them in the event of an emergency. With the rise in crime, every homeowner should install a home security system. This will make the environment considerably safer. Engineers have recently developed home security systems, allowing the new systems to conduct more complex duties as needed rather than simply acting as an observation tool. They added additional features to the system, which aided in winning the trust and happiness of homeowners, resulting in increased demand for home security systems.

Any individual will be able to help with this project by utilizing different elements that will be provided by the security system. The security system will offer personal identity, resulting in a more secure environment for the owner. The system will alert the homeowner of all of the above-mentioned features and will maintain a secure database with all incoming notifications. The digital notification will inform the owner at any moment, allowing the owner to remain aware of various actions taking place around the property. During normal operations, the technology will automatically inform officials of any unexpected activities.

Organization of The Project

This project is organized into four sections. The first section provides an overview and information regarding picture processing and acquisition. The second chapter will go into further detail about how picture acquisition works, as well as the methods involved in processing an image, storing it in a database, and running it through a teaching machine. The third step is to alert the house owner, and the final step is to notify the police.

BACKGROUNDS/ RELATED WORK

Definitions of Artificial Intelligence

Artificial intelligence (AI) is the ability of a computing device or a computer-controlled robot to perform tasks commonly associated with intelligent beings. The term typically refers to the effort to create systems with human-like mental abilities, such as the ability to reason, uncover meaning, generalize, or learn from past experience. Since the advent of digital computer in the mid-1950s, it has been demonstrated that computers can be trained to execute highly complex tasks with great competence, such as finding proofs for logical theories or playing chess. Nonetheless, despite ongoing increases in the computer's speed of processing and the capacity of the memories, it is impossible for

a program to match the human flexibility across broader areas or even to compete in activities that requires a high level of everyday knowledge.

On the other hand, some programs have surpassed the performance levels of human experts and professionals in performing specific tasks, so artificial intelligence in this limited sense can be found in applications ranging from medical diagnosis to computer search engines to voice or handwriting recognition [11-15].

History of The Development of A.I

Alan Matheson Turing, a British logician and computer developer, pioneered the scope of A.I in the midmost of the twentieth decade. Turing created an abstracted computing machine around 1935, consisting of an endless memory and a scanner that scans the memory symbol by symbol, reading what it finds and writing new symbols. The scanner's operations are governed by an instruction program, which is also stored in memory in the form of symbols. Turing's stored-program notion implies that the machine may operate on its own program, changing or improving it. The universal Turing machine is the name given to Turing's notion. All current computers are, in essence, universal Turing machines. Turing worked as a cryptanalyst at the Code of Government and the School of Cypher in Bletchley Park, Buckingham, England, during WWII. Turing was unable to begin work on the notion of constructing a stored-program digital computing system until World War II ended in 1945. Despite this, he attached great importance to machine intelligence throughout the war. Donald Michie, Turing's colleague at Bletchley Park, recalls Turing constantly explaining how computers might acquire knowledge as well as solve new issues using guiding principles, a method now known as heuristic problem solving.

Turing presented what was most likely the first public presentation on computer intelligence in London in 1947, telling, "What we want is a machine that can learn from experience," and that the "possibility of allowing the machine change its very own commands gives the comprehensive framework for this." And at 1948, he defined a lot of the key ideas of Artificial Intelligence in a work called "Intelligent Machinery." on the other hand, he did not release this essay, and several of his insights were later reproduced by others. As mentioned in the section Connectionism, one of his early motifs was to intern a network of artificial neurons to perform certain functions.

Turing prophesied in 1945 that computers would one day be capable of playing extremely strong chess. Deep Blue, a Global Business Machines Corporation (IBM) chess computer, defeated the incumbent world grandmaster, Garry Kasparov, in a 6-game battle just over half a century later, in 1997. While Turing's forecast that chess scripting would help us learn how some people thought didn't even come true. Deep Blue's 256 simultaneous CPUs permitted it to assess 200 million alternative plays per sec and see up to Fourteen turns of play ahead. Many get along with MIT linguist Noam Chomsky. He claims that computers that beat the Grandmaster in chess are as amazing as bulldozers that win the Olympic weightlifting tournament.

The Turing Test

Around 1950, Turing avoided the conventional argument over the concept of intelligence by developing the Turing test, a practical evaluation of computer intelligence which is now known colloquially as that of the Turing test. A computer, a person interrogator, and a human foil are used in the Turing test. The interrogator seeks to discover which of the two people is the computer by interrogating the other two participants. The keypad and display screen are used for all communication. The interrogator may ask as many penetrating and broad-ranging inquiries as he or

she like, and the computer may do what it takes to generate a false identity. (For example, in response to the query, "Are you really a computer?" the computer may say, "No," and then react to a query to reduplicate a gigantic figure by another with a lengthy delay and false result.) Foil should help the questioner make an accurate identification. A variety of population perform the part of interrogator and foil, and if a large number of the interrogators are still unable to differentiate a computer from a normal human being, the computer is considered an intelligent, thinking organism (according to Turing's test supporters). Hugh Loebner, an American philanthropist, founded the yearly Loebner Prize competition in 1991, giving a prize of \$100,000 to the 1st computer to finish the Turing test and giving an amount of \$2,000 to the top attempt each year. Yet, no Artificial Intelligence software got close to finishing a pure test.

How Does A.I Imitates Humans?

Artificial Neural Networks (ANNs) are computing systems made up of linked components called artificial neurons that are modeled after human brain neurons. These neurons are built up of dozens, if not hundreds, of linked layers of algorithms.

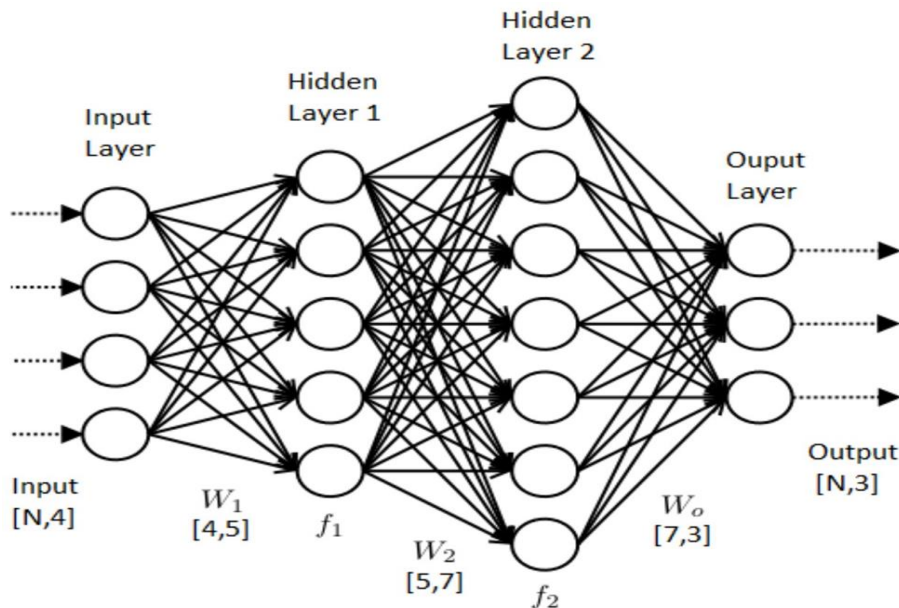


Figure 1. ANNs complex networks.

History of Facial Recognition Systems

A technology-based way of recognizing a human face is known as facial recognition. A face recognition system maps facial characteristics from an image or video using biometrics. It compares the information to a database of known faces to determine a match. Facial recognition can aid in the verification of a person's identification. The history of face recognition may be traced back to the 1960s. That's when mathematician and computer scientist Woodrow Wilson Bledsoe devised a measuring technique that could be used to categorize images of people's faces. Bledsoe is considered as the unofficial father of face recognition technology as a result of his work. Bledsoe's work piqued the curiosity of law enforcement officials. And, from the 1970s to the 1990s, government agencies

created their own facial recognition systems. These were primitive in comparison to today's technology, but the work on these systems paved the groundwork for contemporary facial recognition algorithms.

Many consider 2001 to be a watershed moment in the history of face recognition technology. That's when law enforcement officers at Super Bowl XXXV employed face recognition to assist identify persons in the crowd. The Pinellas County Sheriff's Office in Florida built its own face recognition database the same year. However, it wasn't until the 2010s that computers were sophisticated enough to make face recognition a more common function. In reality, face recognition software proved terrorist Osama bin Laden's identity in 2011. In 2015, the Baltimore Police Department utilized face recognition to identify protesters after Freddie Gray was deceased due to a spinal injury while being transferred in a police van.

Customers are actively using face recognition on their phones as well as other personal gadgets. In 2015, Windows Hello and Android's Trusted Face enabled users to log into their smartphones by merely pointing them at their faces. In 2017, Apple's iPhone X debuted its Face ID face recognition technology. This technology has sparked debate, with detractors claiming that it violates people's privacy. Cities like San Francisco, Oakland, and Boston has outlawed the use of face recognition by governments. And, following Black Lives Matter rallies against police brutality in the summer of 2020, numerous corporate behemoths, including Amazon, Microsoft, and IBM, stated that they would no longer sell facial recognition software to law enforcement agencies [16-20].

How Does It Work?

You might be adept at identifying people's faces. You probably have no trouble recognizing the face of a family member, friend, or acquaintance. You recognize their face characteristics — their eyes, nose, and mouth — and how they work together.

That is how a facial recognition system operates on a larger, algorithmic scale. Recognition technology sees data where you see a face. That information can be saved and retrieved. According to a Georgetown University research, half of all American adults have their photos recorded in one or more facial-recognition databases that law enforcement authorities can check.

So, how does face recognition function? Although technologies differ, the following are the basic steps:

A photo or video of your face is used to capture an image of your face. Your face might be seen alone or in a public. Your photograph might show you staring straight ahead or almost in profile.

The geometry of your face is read by facial recognition software. The distance across your eyes and the distance across your brow to your chin are important considerations. The program recognizes facial landmarks — one system recognizes 68 of them — that are important in differentiating your face. As a consequence, your face signature was created.

Your mathematical formula for a facial signature is matched to a database of known individuals. Consider the following: At least 117 million people in the United States have photos of their faces in one or more government databases. The FBI has accessibility to 412 million facial pictures for searches, as shown in a May 2018 report.

A decision has been made. Your faceprint could match one in a database of facial recognition systems.

How Accurate Is It?

Critics are concerned that face recognition might result in incorrect identifications. What if a police agency wrongly identifies someone smashing a store window throughout a riot as someone who was not near the event using face recognition technology? How probable is it that something will occur? That is conditional. According to the National institute of standards, the best face identification algorithm had an error rate of 0.08 percent as of April 2020. That's a significant improvement over 2014, when the highest accuracy had a 4.1 percent mistake rate.

According to a 2020 report by the Centre for Strategic & International Studies (CSI), accuracy is better when identification algorithms are used to match persons to clear, static photos, such as a passport photo or mugshot. According to the report, when applied in this manner, face recognition algorithms may get accuracy scores as high as 99.97 percent on the National Bureau of Standards and Technology's Facial Recognition Vendors Test. In practice, however, accuracy rates are typically lower. According to the CSI report, the Face Recognition Vendor Test discovered that the mistake rate for one algorithm increased from 0.1 percent when faces were matched to high-resolution mugshots to around 9.3% when faces were matched to the pictures of the people caught in public. Error rates increased when people were not looking straight at the camera or were partially covered by shadows or objects. Another issue is aging. According to the Facial Recognition Vendor Test, middle-tier recognition systems exhibited mistake rates that increased by roughly a factor of ten when attempting to match photographs of participants shot 18 years earlier.

Face Recognition Systems Challenges

As previously said, opportunities and challenges coexist. Rising business interest in face recognition is a positive thing, but it is also a challenging enterprise when it comes to the issues associated with it, which have continually limited its quality of service. These issues arise when the environment is unhelpful, leading in the numerous facial appearances/expressions. The obstacles that are restricting the ability of a Facial Recognition Program to access the additional distance are listed below [20-25].

Illumination

Light fluctuations are abbreviated as illumination. A little shift in terms of lighting is a huge obstacle to facial detection & recognition and can also give a noticeable impact on the outcomes. If the illumination got replaced and the said person is photographed by similar sensors and nearly same face expressions & attitude, the outcomes might appear very varied. The appearance of the face is considerably altered by illumination. It's been revealed that the difference in the two identical faces photographed under different illumination conditions is bigger than the difference of the two separate faces photographed under same lighting conditions.

Illumination is an abbreviation for light fluctuations. A little change in lighting conditions is a considerable barrier to automatic face recognition and can have a big influence on its outcomes. If the lighting changes and the same person is caught with the same sensor and virtually similar face expressions & stance, the outcome might seem considerably varied. The look of the face is radically

altered by illumination. It was discovered that the dissimilarity across 2 identical faces taken under multiple illuminations is exceeding the difference between 2 distinct faces taken within similar lighting

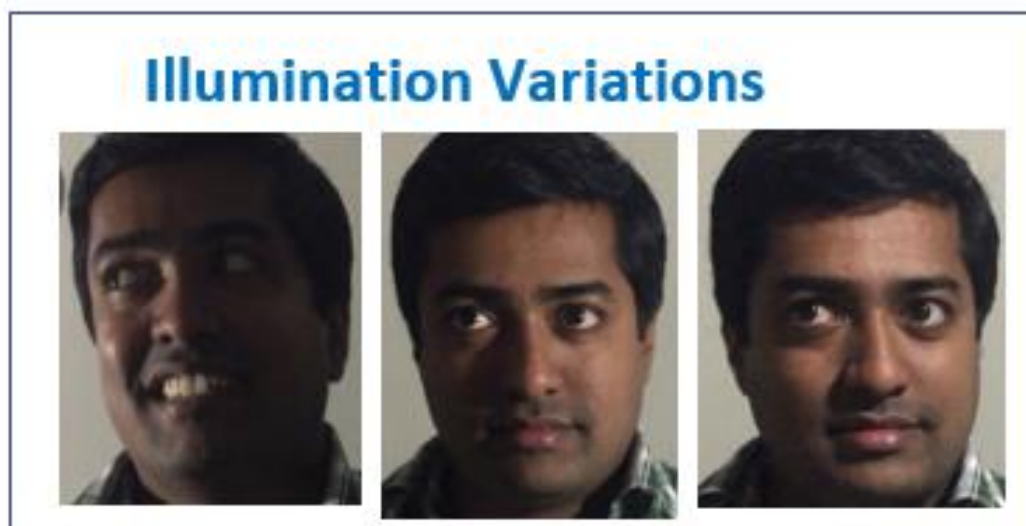


Figure 2

- **Pose**

Facial recognition frameworks are quite responsive to difference in posture. Persons face location differ as his or her head motion and viewing angle vary. Head motions or changing cam POVs will undoubtedly cause changes in facial look and intraclass variations, resulting in drastically lower automatic face recognition percentages. Recognizing the genuine face gets increasingly difficult as the angle of rotation rises. Errors or no recognition may occur if the dataset only contains a frontal shot of the face.

- **Occlusion**

Occlusion is described as an obstruction that occurs when one or more areas of the face are occluded, preventing the full face from being used as the input picture. one of the most major challenging struggles in facial recognition algorithms is Occlusion

- This occurs as a result of a beard, moustache, or accessories (protective glasses, cap, helmet, etcetera.), and it is common in real life situations. The inclusion of such features complicates the situation, rendering automatic facial recognition a difficult problem to solve.

- **Expressions**

Facial expression is one of the most important biometric data since this distinctive trait are important in determining individual identifying and feelings. Diverse situations elicit various expressions, resulting in a spectrum of emotional responses and, subsequently, a shift in facial movements. Additional factor worth examining is the many expressions of the very same person. Human expressions are instances of a person's reactions, particularly macro-expressions like as joy, grief, fury, disdain, fear, and surprise. Micro-expressions are quick facial patterns that arise unconsciously.

As a consequence of changes in one's emotional reaction, micro expressions and macro arise on a person's expression, and at the consequence of such sentiments, which are various, effective identification becomes difficult.



Figure 3

- **Low Resolution**

A common image must have a quality of at least 16*pixel blocks. A low-resolution picture is one that has a resolution less than 16*16. The low-quality photos may be acquired from standalone devices that are small-scaled such as city CCTV system, ATM cams, and supermarket security cameras. Because the camera is not particularly close to the face, such cams can only record a limited proportion of the physical facial area, and then only a face region with a pixel below 16*. Because most of the characteristics get destroyed in such a little shot, it does not convey much information. It is conceivable that identifying people will be difficult.

- **Aging**

The shape of the face changes over the years and shows as ageing, that is another issue for biometric authentication. As we age, our face traits, shapes/lines, and other characteristics change. It is performed after just a long period of inspection and image recovery.

For accuracy verification, the collection is produced for a distinct age category over a period of time. The recognition technique in this case is based on feature extraction, which comprises basic elements like as scars, marks, eyebrows, haircuts, and so forth.

- **Model Complexity**

Existing cutting-edge face - recognition systems are based on a 'too-deep' Convolutional Neural Network (CNN) topology that seems to be extremely complex and unsuitable for real time performance on microcontrollers. A successful facial recognition program should be able to adjust to changes in expression, illumination, posture, and occlusion. It should be expandable to a large group of people while requiring only a few photographs upon registration and avoiding complex design.

- **Photographs**

These tactics were susceptible to early facial recognition systems. Authorized hackers used images to fool the systems employed by Lenovo, Asus, and Toshiba laptops in 2009.

- **Stealing the numerical code**

A thief might steal the codes if each facial pattern is translated into a numerical code before matching.

Fortunately, technological advancements and a more intelligent approach to user interface design have made it more difficult for hackers. The greatest defense against photo spoofing is liveness detection. 3D scanning is beneficial (see the technology section), and some systems require individuals to blink during setup to confirm their 'aliveness.' Another prudent precaution is to provide multiple degrees of protection based on the use case.

- Facial recognition alone may be sufficient in low-risk settings.
- However, if the risk is severe, the system may need multi-factor authentication, such as a password and a fingerprint.

Theories Used

Tensor Flow

Google designed and distributed TensorFlow, a Python tool for fast data processing. It's a simple package for creating Deep Learning models, either directly or through wrapper libraries built on top of TensorFlow to make the process easier. TensorFlow is a free and open-source toolkit for doing rapid numerical computations. Google built and maintains it, and it is free software licensed under the Apache 2.0 license. While the true C++ API is available, it is ostensibly for the Scripting language TensorFlow, unlike other Deep Learning numerical libraries such as Theano, was created with the goal of being used in both development and production systems, such as Google's RankBrain and the entertaining DeepDream project. It can run on single-processor PCs, GPUs, mobile platforms, and decentralized systems with hundreds of workstations. Tensor flow is easy to set up since it is built with Python and the install tensorflow script.

Keras

Keras is a Python-based deep learning API that runs on top of TensorFlow, a deep learning framework. It was designed with the goal of allowing for quick experimentation. It is vital to be able to move quickly from concept to conclusion when conducting research.

- Simple to grasp, yet not simple. Keras reduces developer cognitive burden, allowing you to concentrate on the most critical components of the problem.
- Keras is versatile and adheres to the concept of incremental complexity disclosure: basic operations should be quick and easy, while indefinitely complicated processes should be feasible through a plausible route that builds on what you've learnt so far.
- Keras is an effective platform that provides industry-leading performance and scalability, and it is used by many businesses.

Layers and models are the two most important data types in Keras. The sequential model, which is a linear stack of levels, is the most fundamental sort of model. For more complicated designs, use the Keras's functional API, which lets you to create arbitrary layer structures or entire models from A to Z by subclassing. Keras's is included in TensorFlow 2 as TensorFlow. Eras. To get started with Keras's, just install TensorFlow 2.

TensorFlow/Keras are ideally suited to:

- Windows 10
- Python 3.8

Python Facial Recognition Library

Face recognition library created with the dlib library, which was created with the c++ programming language. As a result, the library aids in the correctness of the person's face by 99.38 percent. not to mention its ease of incorporation into the project Pip will install facial recognition, but first make sure that the dlib library is installed. pip3 install face_recognition. The face_location (image file) function from the face_recognition library is used to find a face in an image.

```
import face_recognition
image = face_recognition.load_image_file("your_file.jpg")
face_locations = face_recognition.face_locations(image)
```

When the aforementioned code is used, it acquires a picture that is stored by your file name; otherwise, the image name in the code should be changed to its directory name. To put it simply, if we want to acquire an image from a streaming video feed, OpenCV is the package we'll need for face detection. In the facial recognition library, the function compare_faces () is in charge of comparing the face to the data system and continue with the facial recognition. Compare_faces () returns a list of true and false values that suggest the correct known face encoding that fits the checked face's primary encoding.

```
import face_recognition
known_image = face_recognition.load_image_file("hemanth.jpg")
unknown_image = face_recognition.load_image_file("unknown.jpg")
hemanth_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]
results = face_recognition.compare_faces([hemanth_encoding], unknown_encoding)
```

According to the code above, there is one known picture named "Hemanth.jpg" and another called "unknown.jpg" that are being compared. Face encodings () method is also being constructed, and each face in the picture is assigned a Spatial dimension face encoding. OpenCV (Open-Source Computer Vision Library), a free artificial intelligence and image processing software library, is used. OpenCV was created to offer a fundamental substructure noun for computer. Vision operations and to speed up the incorporation of machine learning and artificial intelligence into commercial applications.

SMTP

The SMTP library is used throughout the code to send emails for notification. The Simple Mail Transfer Protocol (SMTP) has provided a stable and effective basis for message transfer agents' relay capabilities.

Despite being a decade old, SMTP has shown to be quite resilient.

Despite this, it is evident that a number of protocol extensions are necessary. Rather than defining these extensions as separate and arbitrary entities, this paper clarifies SMTP by providing a framework within which any future expansions may be built in a single consistent manner.

SMTP mail transactions are divided into three stages. The transaction is started with an MAIL command that identifies the sender. A sequence of one or more RCPT commands that convey information to the receiver follows. A DATA command is then used to give the mail data. Finally, the transaction is confirmed by the end of mail data signal.

Google's SMTP server is smtp.google.com.
587 is the SMTP port.

CMAKE

CMake is a free multiplatform and libre software 'FLOSS' that uses a compiler-independent approach to automate software construction, testing, packaging, and installation. CMake is not a build system; instead, it creates build files for other applications.

It can handle directory hierarchies as well as programs that require numerous libraries.

It is compatible with native build environments like as Visual Studio.

It merely contains a few libraries and requires only a C++ expert and its own creation.

The dlib open-source library was used to measure the position of 68 cartesian coordinate systems that were used to create a person's face characteristics. It also employs pre-formation designs.

TKinter

Python is a graphical programming language with a user interface. Tkinter is a graphical program within the Python programming language, comparable to the Windows graphical program within the programming languages that come with Visual Studio. However, if programs with windows or graphical user interfaces must be constructed such that the user can interact with them in Python, the Tkinter library is required since it offers Python interface construction tools and pre-programming scripts. They are Python programming libraries that make it easier to construct graphical user interface apps. The following are the three most well-known graphical user interface libraries:

Python provides several possibilities for designing graphical user interfaces (GUIs). One of the most essential is Tkinter wxPython.

Widgets for TKinter

Widgets are buttons in a graphical user interface software. Tkinter now supports 15 distinct widget types. Widgets are included in the table below, along with a brief description.

Table 1. Tkinter widgets

operative	explanation
Toggle	appear buttons in the window
Canvas	Draw different shapes
Check button	Show different choices and allow the user choose between them
Entry	Line text field allow the accept values
Frame	Prepare different widgets.
Label	Generate line caption for different gadgets as well as show images.
List box	Show menu
Menu button	Give the user a variety of commands.
Menu	Provide multiple orders to the user
Message	Appear different texts
Radio button	Show variety of choices for buttons.
Scale	Produce Slippery contraption

Tkinter's most common features

Visual Studio tools or a programming language with a graphical computer program that all have the same characteristics but differ in how they are accessed, and a handful of them may have the same name.

JPython

Tkinter is Python's standard GUI library. When a programming language like Python is paired with a toolkit like Tkinter, creating a GUI is quick and simple. Tkinter is a fantastic OOUI that includes instrumentation.

This library makes it easy to create a graphical user interface (GUI) software. All that remains is to follow the steps outlined below:”

- Upload module
- Generate main page
- Include widgets
- Respond to every action that the visitor has generated by entering the main iterator.

OpenCV

Open-Source Computer Vision would be a prominent computer vision repository that was developed in 1999 by Intel. The cross-platform repo focuses on the real time image analysis but also provides patent-free applications and among the latest video processing technique. Willow Garage assumed over during 2008, and OpenCV 2.3.1 now provides a C++, C, Python, and Android development interface. Because OpenCV is released underneath a License agreement, it could be used in either commercial and academic settings. OpenCV 2.4 now contains the fresh Face Classifier module for biometrics, able to start experimenting with it right away. This is the escorting I wished for while practicing my image recognition abilities. It demonstrates the way to use Facial Recognizer in OpenCV to do face recognition (whole source code listing) and provides an overview of the systems involvedAs many people have requested; I also will teach how to make the representations featured in several books. You may hire freelancers to develop the best programming languages for Face Recognition.

C++

C families are an extra option for doing image processing, identification, and motion detection. There are two ways to utilize them: either you build your own functionality and processes from scratch (physically write them yourself), or you use certain libraries built for C families, such as OpenGL, OpenCV, EmguCV, and many more libraries. The most common standard libraries used by most C, C++, and C# languages is GDI/GDI++, an API for accessing and performing graphical processing, programming, and display on your monitor. I utilized this for undergraduate projects that required the use of C++ and C#. Face recognition is the initial stage of intelligent picture or video processing for face recognition in unrestricted scenery with compound backdrops (outside settings, airport, and train/bus station). The accuracy of the latter is highly dependent on the results of your facial recognition. Typically, professionals researching in that field do not have the time or ability to produce efficient C++ code suitable for commercial use, and instead limit itself towards the MATLAB development procedure.

SYSTEM MODULE& WORKDONE



Figure 5. window of GUI

- The program's name is Security System. The program's dimensions are 300*300 width*height.
- The level of security is initial security.
- This button will inform the user of any action in the vicinity of the residence.

Face Recognition

Brief face detection in python:

Face detection is a technique that trains a machine to recognize a face rather than just running a program or code. The distinctions between face detection and face recognition will be explained and highlighted in this section. Face detection is the process of detecting a face, in other words finding the face similarity from an image or a video since there is no difference where we acquire the picture from, but face recognition is a software that discovers faces, detects them, and then matches them. Where it specifies which face belongs to which person. To begin, a few libraries, such as OpenCV-python, must be installed and then loaded into our application with the line `import cv2`. Many pre-trained classifiers for facial traits such as the eyes, nose, grin, and the face itself are included in OpenCV. In addition, we must install the `dlib` library as well as the Face recognition library.

To download OpenCV, do the following on the command prompt:

- `pip install opencv-python`
- To download facial recognition, type the following on the command prompt:
- `pip install face recognition`

OpenCV

The majority of OpenCV input parameters are kept in their original condition. When images are imported, for example, they will be in the `cvMat` format, which is C++. Array operations, such as `cvSet2D`, `cvGet2D`, and so on, appear to play a variety of purposes. According to certain discussions, they appear to be worse. For `imageROI`, you'll need specific procedures like `cvSetImageROI`. When you find contours, you'll get `cvSeq` structures, which aren't as easy to work with as Local variables or NumPy arrays.

CV2

NumPy structures such as ndarray are supplied, as well as standard Python components such as lists, tuples, and dictionaries. In this environment, you may perform any NumPy function. as a result of such NumPy support NumPy is a fast and trustworthy array operations library.

When an image is loaded, for example, a ndarray is returned. `array[k,n]` returns the pixel value at (k,n). `ROI=array [c1:c2, r1: r2]` appears to be an example of imageROI array splitting. It is not necessary to have separate qualities. To merge two photographs, just use `res = img1+img2` instead of any functions. (However, NumPy addition is a modulo function for uint8 arrays like pictures.) You may read about the differences between OpenCV and Numpy here. I'll demonstrate how to accomplish it with Python terminal.

NumPy

NumPy is the most important Python library for machine learning. It is a Python library that includes a multidimensional array object, derived objects (such as masked arrays and matrices), and a number of routines for performing fast array operations such as arithmetic, rational, shape subterfuge, grouping, selecting, I/O, based on fourier converts, basic linear programming, standard mathematical operations, random computation, and more. The NumPy package is centered around the ndarray object. Many operations are performed in compiled code, which encapsulates n-dimensional arrays of homogenous data types for efficiency. There are some important differences between NumPy arrays and conventional Python sequences:

- With the exception of Python lists, NumPy arrays have a fixed size (which can grow dynamically). When the size of a ndarray is changed, a new array is created, and the existing one is destroyed.
- The members of a NumPy array must all be of the same type of data and, as a result, consume the same amount of memory. As an exception, components of (Py, including NumPy) items can be utilized, allowing for arrays of variable sizes.
- NumPy arrays simplify sophisticated mathematical and other operations on vast quantities of data. Such tasks are frequently completed faster and with less code when using Python's built-in sequences.
- A growing number of scientific and mathematical Python-based programs use NumPy arrays; while they typically accept Python sequences as input, they convert them to NumPy arrays before processing, and they return NumPy arrays on a regular basis. To put it another way, knowing how to utilize Python's built-in sequence types isn't enough to efficiently use much (if not all) of today's scientific/mathematical Python-based software; you also need to grasp when and how to use NumPy arrays.

Numpy is fast because it vectorizes, which means there are no explicit loops, indexing, or other activities in the code; these operations, of course, take place "in the background" in efficient, already compiled C language code. Vectorized code offers several benefits, including:

- Vectorized codes are denser and easier to understand
- When there are less computer language codes to implement, there are fewer defects.
- The coding is more comparable to traditional number theory (making it easier, typically, to correctly code arithmetic operations)

- The consequence of vectorization is more "Pythonic" code. We would not have utilized factorization if we had not used vectorization.
- Greater "Pythonic" script is generated as a result of vectorization. If we hadn't used vectorization, our script would have been plagued with wasteful and complicated for loops.

The word "streaming" refers to the implicit component behaviour of processes. In NumPy, all operations, including arithmetic, rational, bit-wise, multipurpose, and so on, work implicitly element-by-element, i.e., they broadcast. Furthermore, in the foregoing example, a and b might be bidirectional arrays of the same form, a scalar and an array, or even two arrays of different forms, as long as the shorter array is "infinitely adjustable" to the form of the bigger array in such a manner that the broadcast is comprehensible. Go to fundamentals broadcasting for further information on the "rules" of broadcasting. But what does this signify in terms of image processing? When a value of '1' is applied to a white pixel, a pure black pixel is formed, which is not suitable for image processing. Multiplying a = 127 by b = 128 yields a dark pixel. The previous cv was twelve units quicker than cv2. The product photos are comparable. cv2.imread() returns the image as a NumPy array object. Thus, NumPy's functions may be used to speed up calculations. The following example demonstrates how to leverage the ndarray object's item () and itemset () methods to speed up your loop version.

In a word, cv2 makes everything easier and faster.



$[-0.23, -0.54, \dots, 0.27]$

The network encodes a picture to 128 D encodings on a dataset of three million images, as seen in the figure below, and is represented as one matrix.

Object Detection

NumPy is a Python array manipulation toolkit that may be used to perform a wide range of tasks. It includes a high-performance multidimensional array object as well as tools for manipulating it.

It is by far the most important Python module for scientific computing. It has several unique features, including:

- Advanced (broadcasting) techniques
- A strong array of space objects
- C++ merging libraries
- In addition to its obvious scientific uses, NumPy might be used as a multidimensional store for general data. Numpy allows you to construct any form of data, allowing you to rapidly and easily interface NumPy with a broad range of databases.
- The practice of employing a computer system to process digital photographs using an algorithm is known as image processing.
- Using picture capturing software to import the image.
- Analysation and modification.
- The output might be an altered image or a report based on that image.
- PIL is an open-source tool for computer languages like Python that allows you to view, alter, and save a variety of image file formats.

`from tkinter.constants import CENTER, LEFT, RIGHT, W` #Symbolic constants can be used instead of strings when passing arguments to Tkinter calls. It is automatically imported by the main tkinter module.

`import tensorflow.keras` #The Keras API specification has been designed exclusively for Tensorflow. It extends the framework's support for a number of Tensorflow-specific capabilities.

`import numpy as np` #The np alias is commonly used to import NumPy. In Python, an alias is a different way of defining the same item. The NumPy package is now abbreviated as np rather than numpy.

`from tkinter import *` #As we discussed previously regarding the tkinter library, we import the hole library in this line.

`from PIL import Image, ImageOps` #import this line This module includes a number of 'ready-made' image processing techniques. This module is currently in development, and the most of the operators only work with L and RGB pictures from the previously described PIL library.

`TM_DATA = None` #the initialization of object detection by default.

`model = None` #define the model as null or an empty string.

`cap = None` #The Cumulative Accuracy Profile is used to assess the performance of the classification model. It aids us in grasping and generating conclusions about the robustness of the classification model, and as previously said, we set it to null.

`ret = None` #This boolean variable returns true if the frame is accessible.

`frame = None` #Create an image array vector depending on the default frame rate, which can be provided explicitly or implicitly as a null value.

`image_file = None` #the image's file path.

```
PredictionVariable = None
```

```
key = None # The indexes of a list are identical to keys. The index is used to find elements in a list. Dictionary keys are used to access values. The keys can be of any data type (int, float, string, and even tuple).
```

```
# To improve readability, disable scientific notation.
```

```
np.set_printoptions(suppress=True) # Decide on your printing options. These options govern how floating-point integers, arrays, and other NumPy objects are shown. To prevent the publication of tiny floating-point numbers, use suppress Scientific notation.
```

```
# Load the model
```

```
model = tensorflow.keras.models.load_model('keras_model.h5')
```

```
# Make an array of the appropriate shape to pass into the Keras model.
```

```
# The 'length,' or number of photos that may be placed in the array
```

```
#is defined by the first position in the shape tuple, which in this case is 1.
```

```
TM_DATA = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
```

```
window = Tk() #create a tkinter window
```

```
gui = GUI(window)
```

```
cap = cv.VideoCapture(0) # This class handles video capture via video files, picture sequences, or cameras. The class has a C++ API for accessing video files and picture frames, as well as for acquiring video from cams.
```

```
def process():
```

```
    global ret # permits you to update a parameter that's not in the current stage in the above-mentioned case ret.
```

```
    global count # allows you to alter a variable that's not in the current scope; in such instance, it will return the whole value in a string for a specified element.
```

```
    #Infinite loop for obtaining real-time video.
```

```
    while(True):
```

```
        #ret would return true or false. True indicates that there is a frame.
```

```
        ret, frame = cap.read()
```

```
        origFrame = frame
```

```
        frame = cv.resize(frame, (224, 224))
```

```
        image_array = np.asarray(frame) # will turn the provided picture into an array
```

```
        # Image normalization
```

```
        normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1
```

```
        # Add the picture into the array.
```

```
TM_DATA[0] = normalized_image_array
```

```
PredictionVariable = model.predict(TM_DATA)
```

```
if PredictionVariable[0][0] < 0.9999: # comparing prediction values to one
```

```
    count = 0
```

```
    #Detect the faces
```

```
face_locations, face_names = sfr.detect_known_faces(origFrame)
faceDetection(face_locations, face_names, origFrame)
```

```
cv.imshow("Detection", origFrame) # show the image in a new window
key = cv.waitKey(1)# enable the user to wait a certain amount of time before pushing a key on the
keyboard
if key == (ord('q')):
    break

cap.release()
cv.destroyAllWindows() #end the simulation
```

Notifications

```
import smtplib mail transfer library
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

def __init__(self, window):
self.window = window
self.user = 'notificationtest180@gmail.com'

self.password = 'vhcqfsoitizkgufii'
self.to = "hamodq6r@gmail.com" # In the GUI area where the instructions are specified and
configured, code is placed.

def createEmail(self, topic, message):
msg = MIMEMultipart('alternative') The message is made up of numerous sections.
msg['Subject'] = topic topic of notice
msg['From'] = self.user The sender's username has been entered into the GUI.
msg['To'] = self.to The receiver's username has been entered into the GUI.
html = '<html><body><p>Warning, unknown person comes to your door!</p></body></html>'
part2 = MIMEText(html, 'html')
msg.attach(part2) Text-sending command from stmplb
s = smtplib.SMTP_SSL('smtp.gmail.com') Google's local SMTP server
s.login(self.user, self.password) The sender's login and 16-digit passcode have been entered into the
GUI.
s.sendmail(self.user, self.to, msg.as_string())
s.quit() command to send email

def notifyUserSecurity(self, message):
try:
self.createEmail("Alert!" , "An unknown person seen at your door.") Message of security alert
except Exception as e:
print("Problem with notify security mail", e)
self.securityMessage.config(text= message, fg="#008000")
```

CODE STEPS:

- 1- importing the libraries that will be used for mail notification transfer
- 2- Setting up a Gmail account: The second step is to establish up a Google Mail account for development purposes. This will aid in the development of the code by giving access for the Python code and avoiding the exchange of personal log in information when programming the code. The Python code will be able to access the Google account security for Mail. On the other side, this will make it easy to categorize and organize the mail, allowing the account to be used just for development purposes.
- 3- Changing the settings for the newly formed Google account: In order for the code to operate, the personal mail account settings must be updated for third-party app use. Allowing third-party applications to utilize the mail and setting an APP Password that will be used in the code instead of the original user's password can be used to achieve this strategy. After activating 2-Step Verification, Google will automatically produce a 16-digit App password that will allow the user to utilize the Email for third-party app use by enabling less secure apps to use the Email.
- 4- In the GUI area, where it will be predefined and configured as commands, establish and specify the usernames for receiving and sending emails.
- 5- adding the mail transfer path, which includes the message's topic, the sender, the receiver and the message that is being send.
- 6- Adding the SMTP server: Because the mail accounts used for receiving and sending are Google mails, the Google SMTP server will be utilized.
- 7- Adding SMTP Port: 587 (the Google port).
- 8- The security notification command has been added.

`self.window.geometry("500x500")` the width and also the height of window

`#Security part`

`self.securityLabel = Label(window, text = "Secure Level:")` In this tkinter, a display box with a label will be implemented; the label will be secure level.

`self.securityMessage = Label(window, text="")`

`self.securityMessage.config(text="INITIAL SECURITY", fg="#008000")` In this tkinter, a display box with text will be implemented, and the label will be initial security with a green layout.

`self.notifySecurity = Button(window, text="Click to notify with email", command= lambda: gui.notifyUserSecurity("Secure"))` The lambda function will display a window with some buttons. Each button sends a text message to a common callback function as an argument.

`self.securityLabel.place(relx=0.3 , rely=0, anchor=NW)` It displays the (security level) position on the window.

`self.securityMessage.place(relx=0.6, rely=0, anchor=NW)` It displays the (initial security) message on the window.

`self.notifySecurity.place(relx=0.3, rely=0.1, anchor=NW)` It displays the (click to inform via email) button, which is located on the window.

EVALUATION/IMPLEMENTATION

Introduction to The Tools & Simulator Used:

For this project, the team used Python, which is an interpreted reasonably high software program. Its design approach increases code readability by utilizing a large amount of gap. Its language components and object-oriented philosophy are designed to help programmers produce clear, sensible code for both small and large projects. This is because the 3.8 version is compatible with various libraries needed to execute this project, including "Keras and Tensorflow."

Another program used in this project is visual studio c++, a software application developed by Microsoft that is widely used in many applications such as software development, machine learning AR/VR applications, and support for various libraries such as "Dlib and Numpy," both of which are required for this project.

Finally, visual studio code was utilized to run the code, which is an integrated programming environment built by Microsoft.

Scenarios

Among the circumstances encountered when testing the code were:

- Because the keras model was not recognized, we had to create a new path for it in the code in order for it to be recognized.
- If more than one person is in front of the camera, they will not be identified.
- The notifying component occasionally delivers the incorrect message.

RESULTS

The project's ultimate result was positive, since the code was debugged and tested. The coded code showed the Graphical user interface and activated the camera for detection as the code ran. The security system's graphical user interface provided the option for object notification. The camera identifies faces for identification, and if the individual is recognized, the system displays the person's name around the detected face. The camera also detects objects. When the user picks the notification option, an email is instantly sent to the house owner's account to alert him.

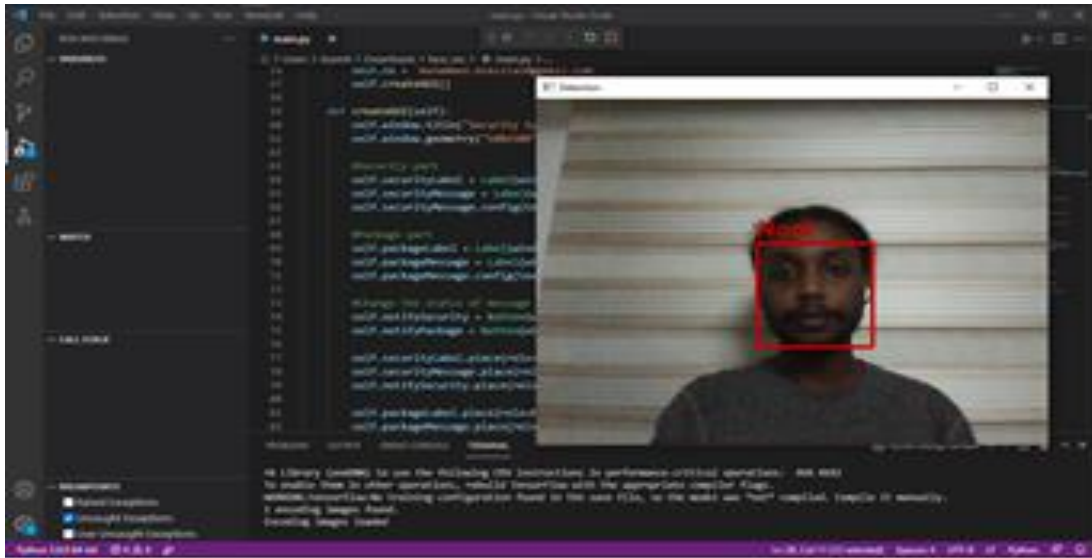


Figure 1. Identified & verified person.

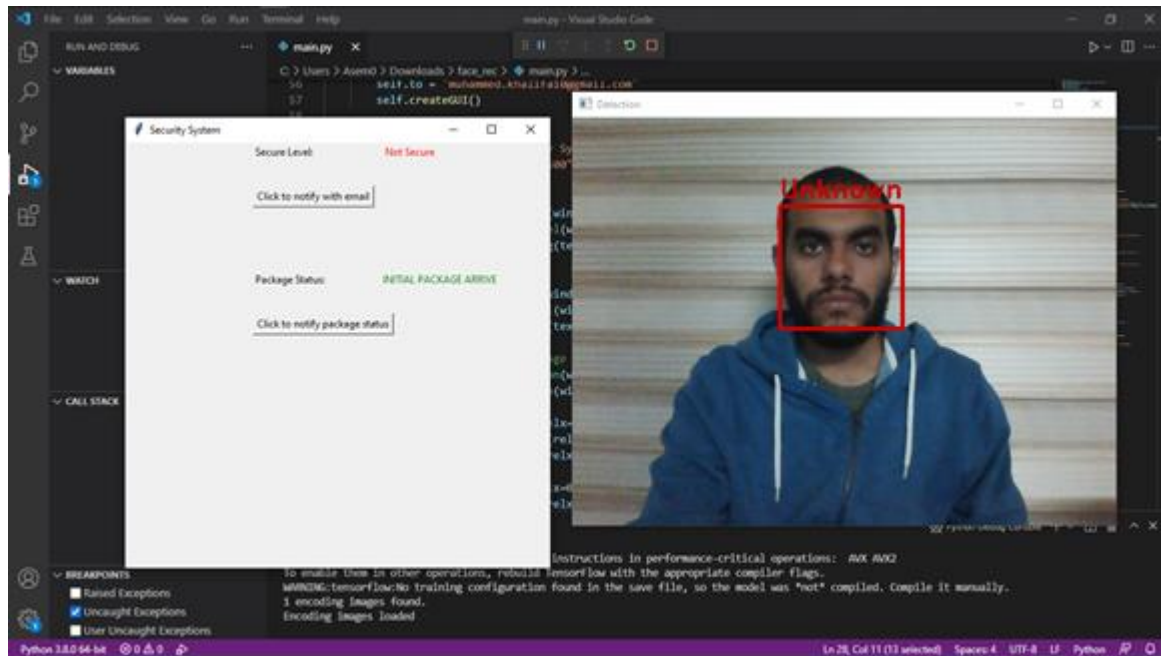


Figure 2. Unidentified person. angles.

CONCLUSION

To summarize what has been stated thus far, this project is about a security system powered by facial recognition. The project was divided into three parts: the first is face recognition, where I was able to detect both familiar and unfamiliar faces, the second is notification, where I would notify the homeowner if a known face or has arrived, and the third is alerting officials to unusual activity. Finally, I utilized a graphical user interface (GUI) to control everything. I was able to develop and run the necessary code for this project using the Python programming language and several of its libraries, and the anticipated results were obtained.

References

- [1] AlBdairi, A. J. A., Xiao, Z., Alkhayyat, A., Humaidi, A. J., Fadhel, M. A., Taher, B. H., ... & Al-Shamma, O. (2022). Face Recognition Based on Deep Learning and FPGA for Ethnicity Identification. *Applied Sciences*, 12(5), 2605.
- [2] Almeida, D., Shmarko, K., & Lomas, E. (2022). The ethics of facial recognition technologies, surveillance, and accountability in an age of artificial intelligence: a comparative analysis of US, EU, and UK regulatory frameworks. *AI and Ethics*, 2(3), 377-387.
- [3] Bickley, S. J., Chan, H. F., & Torgler, B. (2022). Artificial intelligence in the field of economics. *Scientometrics*, 127(4), 2055-2084.
- [4] Bouras, C., & Michos, E. (2022, January). An online real-time face recognition system for police purposes. In 2022 International Conference on Information Networking (ICOIN) (pp. 62-67). IEEE.
- [5] Bucher, T. (2022). Facing AI: conceptualizing 'face communication' as the modus operandi of facial recognition systems. *Media, Culture & Society*, 01634437211036975.
- [6] Chaabane, S. B., Hijji, M., Harrabi, R., & Seddik, H. (2022). Face recognition based on statistical features and SVM classifier. *Multimedia Tools and Applications*, 81(6), 8767-8784.
- [7] Černý, M. (2022). The history of chatbots: the journey from psychological experiment to educational object. *Journal of Applied Technical and Educational Sciences*, 12(3), 322-322.
- [8] Cimmino, L., Nappi, M., Narducci, F., & Pero, C. (2022). M2FRED: Mobile masked face REcognition through periocular dynamics analysis. *IEEE Access*, 10, 94388-94402.
- [9] Du, H., Shi, H., Zeng, D., Zhang, X. P., & Mei, T. (2022). The elements of end-to-end deep face recognition: A survey of recent advances. *ACM Computing Surveys (CSUR)*, 54(10s), 1-42.
- [10] Ge, H., Zhu, Z., Dai, Y., Wang, B., & Wu, X. (2022). Facial expression recognition based on deep learning. *Computer Methods and Programs in Biomedicine*, 215, 106621.
- [11] Hemathilaka, S., & Aponso, A. (2022). A Comprehensive Study on Occlusion Invariant Face Recognition under Face Mask Occlusion. *arXiv preprint arXiv:2201.09089*.
- [12] Hosgurmah, S., Mallappa, V. V., Patil, N. B., & Petli, V. (2022). A face recognition system using convolutional feature extraction with linear collaborative discriminant regression classification. *International Journal of Electrical and Computer Engineering*, 12(2), 1468.
- [13] Ibsen, M., Rathgeb, C., Fischer, D., Drozdowski, P., & Busch, C. (2022). Digital Face Manipulation in Biometric Systems. In *Handbook of Digital Face Manipulation and Detection* (pp. 27-43). Springer, Cham.
- [14] Jeevan, G., Zacharias, G. C., Nair, M. S., & Rajan, J. (2022). An empirical study of the impact of masks on face recognition. *Pattern Recognition*, 122, 108308.
- [15] Kaur, G., Sinha, R., Tiwari, P. K., Yadav, S. K., Pandey, P., Raj, R., ... & Rakhra, M. (2022). Face mask recognition system using CNN model. *Neuroscience Informatics*, 2(3), 100035.
- [16] Koley, S., Roy, H., Dhar, S., & Bhattacharjee, D. (2022). Illumination invariant face recognition using Fused Cross Lattice Pattern of Phase Congruency (FCLPPC). *Information Sciences*, 584, 633-648.

- [17] Lu, Y., Khan, M., & Ansari, M. D. (2022). Face recognition algorithm based on stack denoising and self-encoding LBP. *Journal of Intelligent Systems*, 31(1), 501-510.
- [18] Pawar, A. B., Gawali, P., Gite, M., Jawale, M. A., & William, P. (2022, April). Challenges for Hate Speech Recognition System: Approach based on Solution. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)* (pp. 699-704). IEEE.
- [19] Raharja, N. M., Fathansyah, M. A., & Chamim, A. N. N. (2022). Vehicle parking security system with face recognition detection based on eigenface algorithm. *Journal of Robotics and Control (JRC)*, 3(1), 78-85.
- [20] Smith, M., & Miller, S. (2022). The ethical application of biometric facial recognition technology. *Ai & Society*, 37(1), 167-175.
- [21] Takata, T., Sasaki, H., Yamano, H., Honma, M., & Shikano, M. (2022). Study on Horizon Scanning with a Focus on the Development of AI-Based Medical Products: Citation Network Analysis. *Therapeutic innovation & regulatory science*, 56(2), 263-275.
- [22] Zainal, M. M., & Hamdan, A. (2022). Artificial intelligence in healthcare and medical imaging: Role in fighting the spread of COVID-19. In *Advances in Data Science and Intelligent Data Communication Technologies for COVID-19* (pp. 173-193). Springer, Cham.
- [23] Zeng, X., & Long, L. (2022). Introduction to Artificial Intelligence. In *Beginning Deep Learning with TensorFlow* (pp. 1-45). Apress, Berkeley, CA.