Major Project Report (IT454)

Finding Missing People Using AI

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by:	**Submitted to:**
Esshaan Mahajan	**Dr. Sartaj Singh Sodhi**
Enroll no: 01916403219	**Associate Professor**
B.Tech. CSE 8th Sem	**USICT, GGSIPU**

UNIVERSITY SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, New Delhi-110078

(2019-2023)

ACKNOWLEDGEMENT

I thank the University School of Information Communication and technology for giving us the courage and perseverance in completing the project. This project itself is acknowledgements for me, I have given this project tremendous efforts in making this project a grand success.

I am highly indebted to Prof. Sartaj Singh Sodhi for his guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project. I am extremely thankful to him for providing such a nice support and guidance. I am thankful to our dean prof. Pravin Chandra for providing the necessary infrastructure and labs and also permitting to carry out this project.

I have taken efforts in this project, however it would not have been possible without the kind support and help of my friends. I would like to extend my sincere thanks to all of them.

Last but not least, I would like to express deep sense of gratitude and earnest thanks to my dear parents for their moral support and heartfelt cooperation in doing the project.

SUPERVISOR'S CERTIFICATE

This is to certify that Mr. Esshaan Mahajan of B.Tech (CSE, 8th Semester) of University School of Information, Communication and Technology, GGSIPU has successfully completed his project work entitled, "Finding Missing People Using AI", which is being Submitted to the university in the fulfilment of the requirement for the award of the degree of Bachelor of Technology.

This project report is record of authentic work carried out by him under my guidance.

Dr. Sartaj Singh Sodhi

Associate Professor

DECLARATION BY THE CANDIDATE

I declare that this project report titled 'Finding Missing People Using AI' submitted in partial fulfillment of the degree of B. Tech in (Computer Science and Engineering) is a record of original work carried out by me under the supervision of Dr. Sartaj Singh Sodhi and has not formed the basis for the award of any other degree or diploma, in this orany other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Esshaan Mahajan

01916403219

B. Tech CSE

ABSTRACT

Face recognition is a widely used technique for identifying and verifying human faces in images. However, its application to the problem of finding missing people is challenging due to the lack of reliable and updated face images of the missing persons, as well as the possible changes in their appearance over time.

In this project, we propose a novel approach to match the images of potential missing people from a reported cases database using face recognition. Our approach consists of three main steps:

(1) detecting the face in the image; (2) extracting robust and discriminative face features using a deep neural network; and (3) comparing the face features of the query image with those of the reported cases.

TABLE OF CONTENTES

List of Figures

1.Introduction

1.1 Problem statement

The existing problem of people, especially children going missing in India is a serious and alarming issue that needs urgent attention and action. According to the National Crime Records Bureau (NCRB), more than **73 thousand children** in India were reported missing in 2019. This number has increased significantly over the last three years, indicating a worsening situation. Many of these cases are likely to have close links to “human trafficking and child labour”, which are grave violations of human rights and dignity. The lack of resources and coordination among the authorities, civil society and families to find and rescue the missing children is a major challenge that hampers the efforts to address this problem. There is a need for more awareness, data, technology and collaboration to prevent and combat this problem effectively.

This project aims to develop a face recognition system that can match the images of potential missing people from a reported cases database. The system will use an optimised face recognition algorithm based on KNN to compare the facial features and contours of the probe images with the stored images in the database. The system will provide a list of confirmed matches with their corresponding information and location. The system will help to locate missing people faster and more accurately, and to assist law enforcement agencies and families in finding their loved ones.

1.2 Objective

The objective of this project is to provide an innovative solution to the problem of finding people, especially children, who go missing. Police come across several challenges while investigating a missing person case. This project can be used as an important tool by the police and cyber cell to locate a missing person. Furthermore, this application allows the general public to help in the investigation by uploading image which can then be matched in the existing database

The application can be used to report missing people, and find potential missing person and their details from the missing person database.

The agenda is to match potential missing people from a database is to identify and locate persons who have been reported as lost, kidnapped, trafficked, or otherwise unaccounted for. The system uses a biometric software application that compares and analyzes the facial features and contours of a person in an image with the profiles stored in the database. The system also performs a manual verification process by qualified and experienced officers to confirm the results of the automated system. The system aims to reduce the time and effort required to find missing people and to provide accurate and reliable information to the authorities and the public.

1.2 Advantages of Project

The use of face recognition technology in the context of finding missing individuals has gained traction in recent years, and for good reason. This technology offers several advantages for matching potential missing persons with the reported missing people database, which could greatly facilitate finding the individuals in question.

One key advantage of a project that uses face recognition technology to match potential missing individuals with a database of reported missing people is its ability to quickly and accurately identify individuals. This is because the technology relies on algorithms that can quickly compare facial features and match them against a large database. As a result, it can save significant time and effort on the part of law enforcement and other organizations that are involved in the search for missing individuals.

In addition, face recognition technology offers a high degree of accuracy. The algorithms used in this technology are designed to analyze a variety of facial features, such as the distance between the eyes, the shape of the nose, and the contours of the face. This level of detail can help to eliminate false positives, which can help prevent confusion and ensure that the right person is identified.

Another advantage of using face recognition technology in a project to match potential missing persons with reported missing people is the potential to save lives. When time is of the essence, as it often is in missing persons cases, the ability to quickly and accurately identify individuals can be critical. Face recognition technology can help to facilitate this identification process, potentially saving lives in the process.

Furthermore, face recognition technology can provide a significant boost to existing missing persons databases. By integrating this technology into existing databases, the process of matching potential missing persons with reported missing people can become more efficient and accurate. This, in turn, can lead to more successful matches and a higher rate of reunification between missing individuals and their loved ones.

2. System Requirements:

There are a lot of technical specifications for the system to work the manner in which it is intended to work. These technical specifications include software requirements as well as hardware requirements.

Hardware Requirements: As the project is related to the development of an application, the only two major hardware devices that were required were:

a) Mobile phone with good camera quality to capture the photo of missing person
b) Computer system to report or help find the missing person.

Software Requirements:

- Python: Python is a versatile and powerful programming language that can be used for various purposes, such as web development, data analysis, machine learning, and automation. It was created by Guido van Rossum in 1991 and has since become one of the most popular languages in the world. Python has a clear and concise syntax that makes it easy to read and write code. It also supports multiple paradigms, such as object-oriented, functional, and procedural programming. Python has a large and active community that contributes to its rich set of libraries and frameworks, which enable developers to create applications with less code and more functionality.
- Postgres: PostgreSQL, also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley. In 1996, the project was renamed to PostgreSQL to reflect its support for SQL. After a review in 2007, the development team decided to keep the name PostgreSQL and the alias Postgres.PostgreSQL features

transactions with atomicity, consistency, isolation, durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures. It is designed to handle a range of workloads, from single machines to data warehouses or web services with many concurrent users. It was the default database for macOS Server and is also available for Linux, FreeBSD, OpenBSD, and Windows.

- PyQT5: PyQt5 is a Python binding of the cross-platform GUI toolkit Qt v5. It is a free and open-source library that allows you to create graphical user interfaces (GUIs) for your Python applications. PyQt5 is based on the Qt framework, which is a mature and well-established GUI toolkit that is used by a wide range of applications.PyQt5 provides a wide range of features for creating GUIs, including:
  - A large library of widgets and controls
  - Support for various platforms, including Windows, macOS, and Linux
  - A powerful event system
  - A well-documented API

- Sklearn: Sklearn is a Python library that provides various tools for machine learning and data analysis. It is built on top of SciPy, NumPy, and Matplotlib, and it offers a consistent and simple interface for applying different algorithms to various datasets. Sklearn was started in 2007 as a Google Summer of Code project by David Cournapeau, and it has since grown into a popular and widely used open source project with many contributors. Sklearn supports various tasks such as classification, regression, clustering, dimensionality reduction, model selection, and preprocessing.
- Pandas: Pandas is a popular Python library for data analysis and manipulation. It provides a fast and flexible way to work with structured and unstructured data. Pandas offers various tools and methods to perform common tasks such as reading and writing data, filtering and sorting, aggregating and summarizing, merging and joining, reshaping and pivoting, and visualizing and exploring data.

Pandas is built on top of NumPy, another Python library for numerical computing, and integrates well with other libraries such as SciPy, Matplotlib, and Scikit-learn.

- Numpy : NumPy is a Python library that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It is the fundamental package for scientific computing in Python.NumPy arrays are similar to Python lists, but they are much faster and more efficient. They are also of a fixed size, which means that they cannot be resized after they are created. This makes them ideal for storing large amounts of data that does not change often.NumPy provides a wide range of mathematical functions for operating on arrays, including:
  - Mathematical operations, such as addition, subtraction, multiplication, and division
  - Statistical operations, such as mean, standard deviation, and correlation
  - Linear algebra operations, such as matrix multiplication and inverse
  - Fourier transform operations

- Pillow : Pillow is a Python library that provides image processing capabilities. It supports various image formats, such as JPEG, PNG, BMP, GIF, and TIFF. Pillow also offers features such as cropping, resizing, rotating, filtering, drawing, and text rendering. Pillow is a fork of the PIL (Python Imaging Library) project, which is no longer maintained. Pillow aims to be compatible with PIL and to provide bug fixes and new features.
- Uvicorn: Uvicorn is an ASGI (Asynchronous Server Gateway Interface) web server implementation for Python. It is a lightweight and high-performance server that is well-suited for hosting asynchronous web applications. If one is looking for a high-performance ASGI web server for Python, Uvicorn is a great option. It is used by a wide range of projects, including FastAPI, Starlette, and Quart.Uvicorn is based on the uvloop event loop, which provides superior performance

compared to the standard asyncio event loop. It also supports HTTP/1.1 and WebSockets, making it a versatile choice for a wide range of web applications.

- OpenCV : OpenCV is a popular open-source library for computer vision, machine learning and image processing. It supports various platforms and languages, such as Python, C++, Java and more. OpenCV can be used for a wide range of applications, such as face recognition, object detection, video analysis, optical character recognition and more. OpenCV is also optimized for real-time performance and has a large community of developers and users.

- Psycopg2 : Psycopg2 is a Python module that provides a comprehensive and powerful interface to the PostgreSQL database system. It is the most popular PostgreSQL adapter for the Python programming language. Psycopg2 is a fully featured DB API 2.0 compliant database adapter.

- Dlib: Dlib is a cross-platform software library that provides a range of algorithms and tools for machine learning, computer vision, image processing, and numerical optimization. Dlib is written in C++ and can be used as a standalone library or integrated with other frameworks such as TensorFlow, PyTorch, and OpenCV. Dlib is open source and licensed under the Boost Software License, which allows for both commercial and non-commercial use.

3. PHASE DEVELOPMENT

3.1 Project Layout:

The aim of this project is to develop a web-based application that can help find missing persons using artificial intelligence. The application will have the following features and functionalities:

- The application will run on a web server using uvicorn, which is a fast and lightweight ASGI server for Python.

- The application will have a graphical user interface (GUI) that will allow users to register new cases of missing persons. The GUI will be built using PyQT5, which is a cross-platform framework for creating desktop applications with Python.

- The users will be able to enter all the relevant information about the missing person, such as their name, age, contact number, and other details. They will also be able to upload an image of the missing person, which will be used for matching purposes.

- The information and the image of the missing person will be stored in a Postgres database, which is a powerful and reliable relational database system.

- The application will also allow users to submit an image of a potential missing person that they have encountered or seen somewhere. This could be done by any responsible citizen or police personnel who wants to help find missing persons.

- The submitted image will then be compared with all the images of missing persons in the database using an artificial intelligence

technique. The technique will use face recognition algorithms to find the best match among the available images.

- If a match is found, the application will display all the information about the missing person, as well as the contact information of the person who registered the case to the police. This way, the police can contact the person who is looking for the missing person and can investigate about their whereabouts.

- The application will also keep track of the successfully matched cases and update the database accordingly. This will help keep the database up-to-date.

3.2 About AI technique

Face recognition:

Face recognition technology is becoming increasingly popular across the globe, with its applications ranging from security and surveillance to marketing and advertising. Despite its popularity, there are valid concerns surrounding its use, including issues of privacy, accuracy, and bias.

At its core, face recognition technology works by using algorithms to analyze and identify unique features of a person's face, such as the distance between their eyes or the shape of their nose. This data is then compared to a database of pre-existing images to determine a match. While this technology has the potential to be incredibly powerful and effective, there are still a number of challenges that must be addressed before it can be widely adopted.

One of the most pressing concerns surrounding face recognition technology is its potential for erroneous identification. Studies have shown that these systems can be prone to errors, particularly when dealing with people of different races or ages. In addition, there have been numerous reports of innocent people being falsely accused or arrested based on flawed face recognition matches. To address these issues, there are ongoing efforts to improve the accuracy of these systems through better training data and more robust algorithms.

Another major concern relates to the privacy implications of face recognition technology. As these systems become more widespread, there are fears that they may be used to track individuals without their knowledge or consent. In addition, there are concerns that these systems may be used to discriminate against certain groups, such as minorities, based on the data they generate. To address these concerns, there are ongoing debates about the need for clear regulations and guidelines around the use of these systems, as well as the importance of informed consent and transparency.

Despite these challenges, there are also many potential benefits to be gained from the use of face recognition technology. For example, it has already been used to great effect in the field of security, helping to prevent crimes and identify suspects. It has also been used in healthcare to help diagnose and treat diseases, and in marketing to identify consumer preferences and behavior. However, it is important that these benefits are carefully weighed against the potential risks and that responsible use of this technology is prioritized.

In conclusion, face recognition technology is a powerful tool with a broad range of applications and potential benefits, but one that is accompanied by significant challenges and concerns. Addressing these challenges will require ongoing research and development, as well as careful consideration of the ethical and legal implications of its use. Ultimately, if these issues can be successfully addressed, face recognition technology has the potential to transform the way we live and work in the 21st century.

Face detection:

Face detection is an emerging technology that has gained widespread attention in recent years. It is the process of detecting and locating human faces within an image or video frame. The application of this technology has proven to be useful in various fields such as security, surveillance, and entertainment. Face detection relies on various techniques that enable the detection of faces with varying degrees of accuracy and reliability.

One of the most commonly used approaches for face detection is the Viola-Jones algorithm, which is based on Haar features and AdaBoost. This algorithm works well for detecting frontal faces but can be less accurate for detecting faces in profile or other orientations. Other techniques include the use of convolutional neural networks (CNNs), which have proven to be very effective in detecting faces in different orientations and lighting conditions.

In recent years, deep learning models have become increasingly popular due to their ability to learn from large data sets and perform well in various applications. FaceNet, for example, is a deep learning model that uses a triplet loss function to learn to map faces into a high-dimensional feature space where faces from the same person are closer together than faces from different people. This technique can be used for face verification and identification, which has many applications in security and surveillance.

One of the main challenges in face detection is dealing with variations in lighting conditions, poses, and facial expressions. These variations can make it difficult to accurately detect faces, especially in real-world settings. To address this challenge, researchers have developed techniques such as illumination normalization, pose normalization, and facial expression recognition.

Another important consideration in face detection is privacy. The use of face detection technology raises concerns about the collection and use of personal data. Therefore, it is essential to develop ethical and legal frameworks to protect the privacy of individuals while still allowing for the use of this technology in appropriate settings.

Hence, face detection is a rapidly developing technology with many applications in various fields. Advancements in algorithms, deep learning models, and techniques for handling variations and privacy concerns are making face detection more accurate and reliable. As this technology continues to evolve, it is important to carefully consider its ethical and legal implications to ensure that it is used appropriately and for the benefit of society.

In this project, we use the get_frontal_face_detector method in library dlib to find human faces that are looking more or less towards the camera.
The get_frontal_face_detector method in the dlib library is used to get a pre-trained face detector. This detector is based on a technique called Histogram of Oriented Gradients (HOG), which is a feature descriptor that is used to extract features from images. The HOG face detector is very fast and efficient, and it can be used to detect faces in real time.

The get_frontal_face_detector method does not accept any parameters. It simply returns the pre-trained face detector. The returned detector can then be used to detect faces in images.

Here is an example of how to use the get_frontal_face_detector method:

```
import dlib

detector = dlib.get_frontal_face_detector()

image = dlib.load_image("my_image.jpg")

faces = detector(image)

for face in faces:
    print("Found a face!")
    print(face)
```

This code will load the image "my_image.jpg" and then use the get_frontal_face_detector method to detect faces in the image. If any faces are found, they will be printed to the console.

The get_frontal_face_detector method is a powerful tool that can be used to detect faces in images. It is fast, efficient, and easy to use.

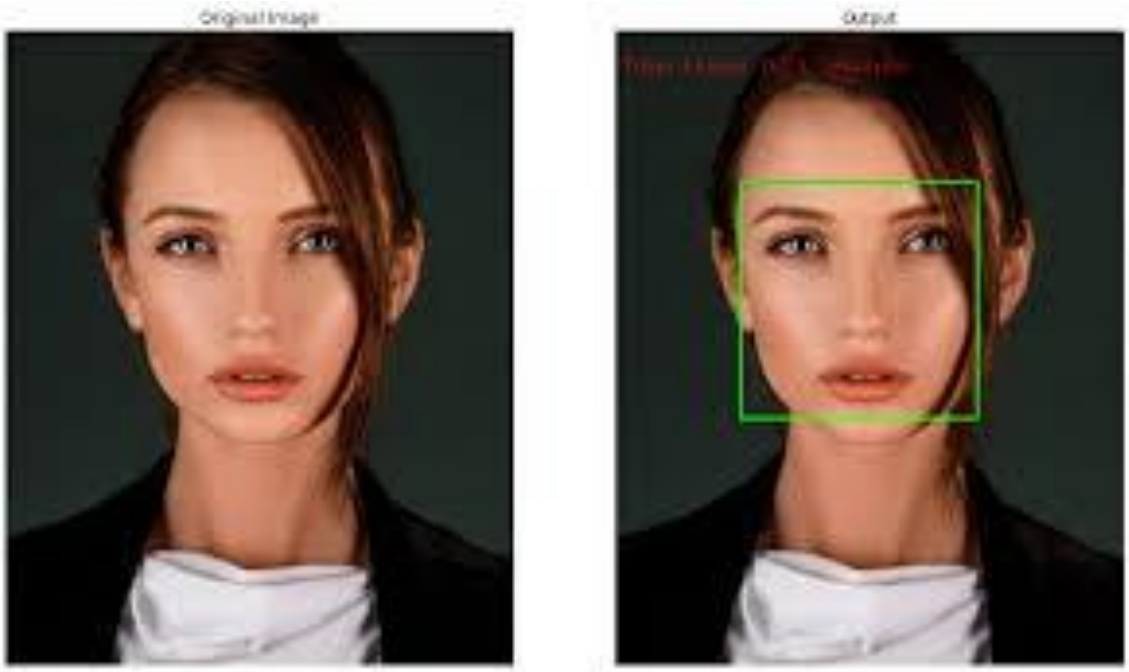

Fig 3.1. Face detection

Face encodings:

Face recognition technology has become increasingly popular in recent years due to its vast array of applications in security, social media, and e-commerce platforms. One of the essential components of face recognition technology is face encodings. Face encodings can be defined as numerical representations of faces that capture unique facial features and characteristics of an individual. In this essay, we will discuss the concept of face encodings in detail, including how they are generated, how they are used in face recognition, and their limitations.

Face encodings are generated through a process known as face embedding. This process involves the use of deep neural networks to extract facial features from an image, followed by encoding these features into a numerical vector. The numerical vector is usually a high-dimensional matrix that represents the unique facial features of a face. The deep neural networks used for face embedding are typically trained on a large dataset of faces to ensure accuracy in facial recognition.

Once the face encodings are generated, they are used in the face recognition process. Face recognition involves comparing the face encoding of a person captured by a camera to a database of face encodings. If a match is found, the system identifies the person in the image. Face encodings are used in several face recognition algorithms, including OpenFace, FaceNet, and DeepFace.

One of the significant advantages of using face encodings in face recognition is their ability to capture unique facial features that distinguish one person from another. Face encodings are robust to small changes in facial expression, lighting, and orientation, making them suitable for real-world applications. Additionally, face encodings are extremely efficient in terms of storage space since they are represented in numerical vectors.

Despite their advantages, face encodings have some limitations. One of the primary challenges with face encodings is their susceptibility to errors in face embedding. Errors in face embedding can result in false matches or mismatches, leading to inaccurate facial recognition. Furthermore, face encodings can be subject to bias, both at the data and algorithmic levels.

In conclusion, face encodings are an essential component of face recognition, enabling accurate identification of individuals based on unique facial features. They are robust, highly efficient, and suitable for real-world applications. However, they have limitations that must be addressed to ensure their accuracy and reliability. Future research should focus on developing improved face embedding techniques that are less susceptible to errors and bias.

We then use the pose_estimator method that takes in an image region containing some object and outputs a set of point locations that define the pose of the object. The classic example of this is human face pose prediction, where you take an image of a human face as input and are expected to identify the locations of important facial landmarks such as the corners of the mouth and eyes, tip of the nose, and so forth. The pose_estimator method in the dlib library is used to estimate the pose of a face in an image. The pose of a face is defined by the 6D position and orientation of the face in 3D space. The pose_estimator method uses a technique called facial landmark detection to estimate the pose of a face. Facial landmark detection is the process of finding the location of facial landmarks in an image. The facial landmarks that are used by the pose_estimator method are the eyes, nose, mouth, and chin.

The pose_estimator method takes two parameters: an image and a face detector. The image is the image that the pose of the face is being estimated from. The face detector is used to find the faces in the image.

The pose_estimator method returns a Pose object. The Pose object contains the 6D position and orientation of the face in 3D space.

Here is an example of how to use the pose_estimator method:

```
import dlib

detector = dlib.get_frontal_face_detector()

pose_estimator =
dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

image = dlib.load_image("my_image.jpg")

faces = detector(image)

for face in faces:
    pose = pose_estimator(image, face)
    print(pose)
```

This code will load the image "my_image.jpg" and then use the get_frontal_face_detector method to detect faces in the image. If any faces are found, the pose_estimator method will be used to estimate the pose of each face. The pose of each face will then be printed to the console.

The pose_estimator method is a powerful tool that can be used to estimate the pose of a face in an image. It is fast, efficient, and easy to use.

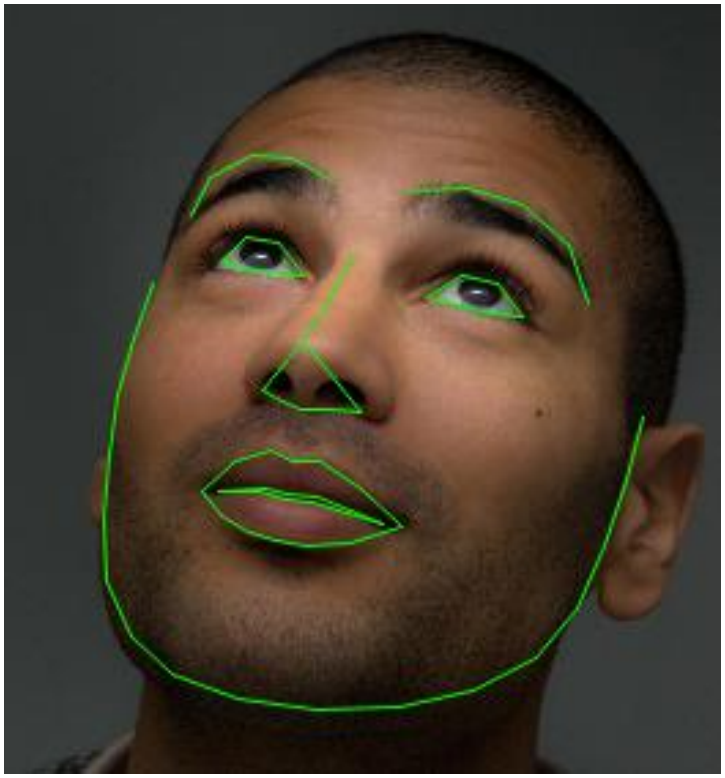

Fig 3.2. Pose Estimation

Furthermore, we use a model to covert this data into face encodings. The model used is a ResNet network with 29 conv layers. It's essentially a version of the ResNet-34 network from the paper Deep Residual Learning for Image Recognition. ResNet-34 is a 34-layer convolutional neural network (CNN) that was introduced in the paper *Deep Residual Learning for Image Recognition* by He et al. (2016). ResNet-34 is a residual network, which means that it uses shortcut connections to help the network learn more complex features.

The ResNet-34 architecture consists of 34 layers, which are divided into 4 blocks. Each block consists of a sequence of convolutional layers, followed by a shortcut connection. The shortcut connection adds the input of the block to the output of the block. This helps the network to learn more complex features, as it allows the network to learn the residual between the input and output of the block.

ResNet-34 has been shown to be very effective for image recognition tasks. It achieved a top-5 error rate of 21.2% on the ImageNet dataset, which is a benchmark dataset for image recognition.

Here is a diagram of the ResNet-34 architecture:

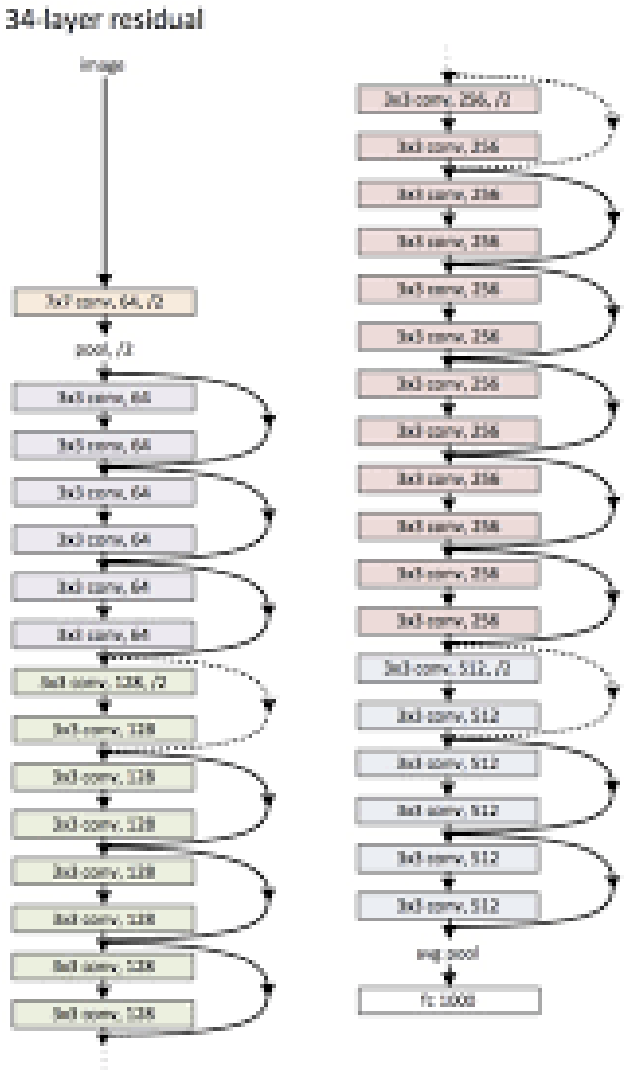

Fig 3.3. Resnet-34 Architecture

The ResNet-34 architecture is composed of the following layers:

- Conv2d: 2D convolutional layer
- BatchNorm2d: Batch normalization layer
- ReLU: Rectified linear unit activation function
- MaxPool2d: Max pooling layer
- Shortcut: Shortcut connection

The ResNet-34 architecture has been used for a variety of image recognition tasks, including object detection, image classification, and image segmentation. It is a powerful and effective CNN architecture that has been shown to be very successful for these tasks.

Here are some of the benefits of using ResNet-34:

- It is a very effective CNN architecture for image recognition tasks.
- It is relatively simple to implement.
- It is relatively efficient to train.

Overall, ResNet-34 is a powerful and effective CNN architecture that can be used for a variety of image recognition tasks. It is a good choice for applications where accuracy is important and computational resources are available.

The network used was a pretrained one on a dataset of about 3 million faces. This dataset is derived from a number of datasets available.

Our aim is to map an image of a human face to a 128-dimensional vector space where images of the same person are near to each other and images from different people are far apart. Therefore, we can perform face recognition by mapping faces to the 128D space and then checking if their Euclidean distance is small enough. For classification we use KNN algorithm.

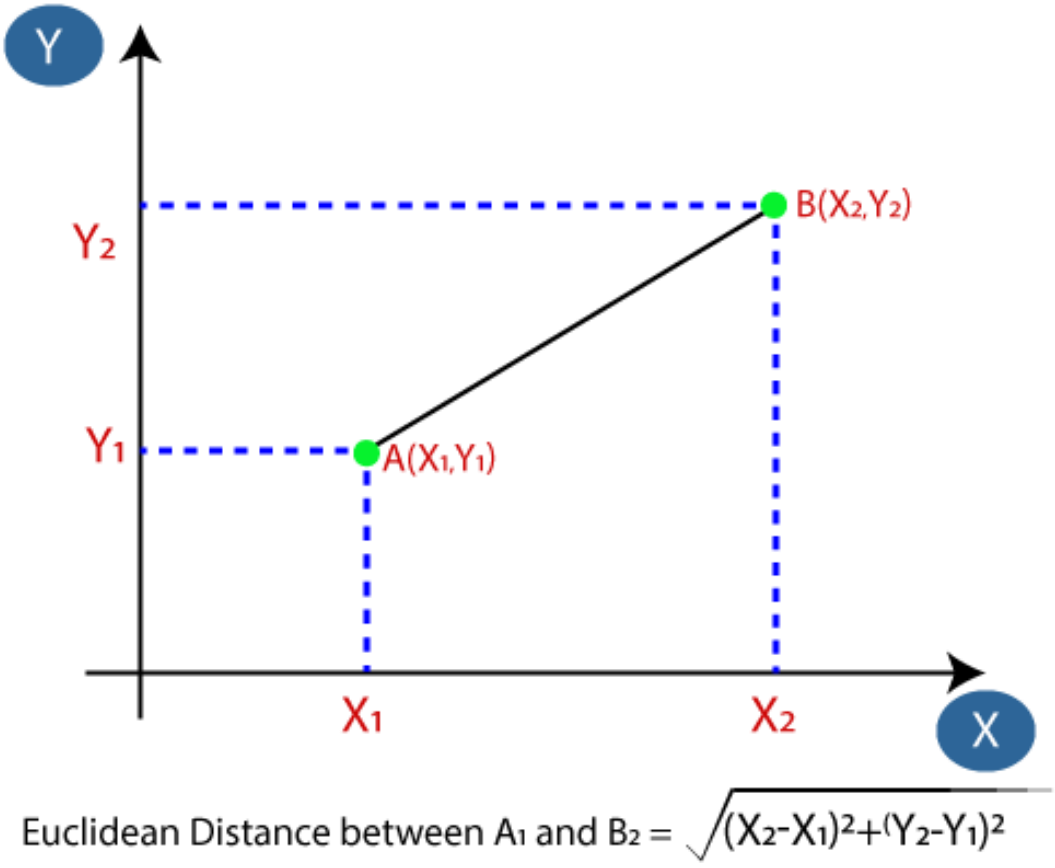

Fig 3.4. Euclidean distance

Classification:

Face recognition technology has become more prevalent in recent times, with its use in security systems, social media platforms, and facial biometric systems. Classification is an integral part of face recognition, as it is the process of identifying a face image and assigning it to a person or a group based on specific features and characteristics. In this essay, we will discuss the importance of classification in face recognition technology, its methods, challenges, and applications.

The primary goal of face recognition technology is to identify and verify an individual's identity based on facial features. One of the essential steps involved in this process is classification, which involves identifying and labeling a face image based on features like shape, texture, and color. There are many different methods of classification in face recognition, including neural networks, decision trees, support vector machines, and nearest neighbor classifiers.

Neural networks use a layered approach that mimics the functioning of the human brain. Decision trees are models that use a series of branching rules to classify images. Support vector machines use a linear or nonlinear boundary to separate images into different classes. Nearest neighbor classifiers use the distance between images to classify them.

All these methods have their advantages and disadvantages, and their effectiveness depends on the complexity of the task at hand. Neural networks are suitable for complex classification problems, but they require a large amount of data for training. Decision trees are simple to understand and implement, but they may not be able to classify complex images accurately. Support vector machines are suitable for binary classification tasks, but they may not work well for multi-class problems. Nearest neighbor classifiers have excellent accuracy, but they may be time-consuming and memory-intensive.

The most common types of classification in face recognition are:

- Identity classification: This is the most common type of classification in face recognition. It involves assigning a unique identity label to each face image. This is the type of classification used in security systems, where the goal is to identify individuals.
- Gender classification: This type of classification involves assigning a gender label to a face image. This type of classification is often used in marketing and advertising, where the goal is to target specific demographics.
- Age classification: This type of classification involves assigning an age label to a face image. This type of classification is often used in security systems, where the goal is to identify minors or elderly people.
- Emotion classification: This type of classification involves assigning an emotion label to a face image. This type of classification is often used in customer service applications, where the goal is to identify customer satisfaction.

For the purpose of our project, we use Identity classification.

One of the significant challenges faced in classification in face recognition is the presence of pose, illumination, and expression variations. These variations can significantly affect the accuracy of face recognition and classification algorithms. Techniques like image normalization, data augmentation, and feature selection can be used to reduce the impact of these variations. However, these techniques may not be effective in all cases, and research in this area is ongoing.

Classification in face recognition has many applications, including security and surveillance systems, social media platforms, and facial biometric

systems. Security and surveillance systems use face recognition to identify and track individuals in public spaces like airports, train stations, and shopping malls. Social media platforms use it to identify individuals in photos and suggest tags. Facial biometric systems use it for authentication and identification in various sectors like banking, healthcare, and government.

In conclusion, classification is an essential aspect of face recognition technology, as it plays a crucial role in identifying and verifying an individual's identity based on facial features. While there are different methods of classification available, each has its advantages and limitations. The challenges in classification, such as pose, illumination, and expression variation, need to be overcome to improve the accuracy of face recognition systems. The applications of classification in face recognition are vast, and the technology has the potential to revolutionize various sectors in the future.

KNN classifier:

A KNN classifier is a machine learning algorithm that uses the proximity of data points to assign them to a category. It is based on the idea that similar data points are likely to belong to the same class. The K-Nearest Neighbors (KNN) classifier distinguishes itself by being a simple yet powerful method for predicting a target variable based on the values of its neighboring data points. The KNN classifier is categorized as a non-parametric algorithm, meaning that it does not assume any distributional properties for the underlying data.

The KNN classifier works by finding the K nearest neighbours of a new data point and taking a majority vote among them. The fundamental premise behind KNN is to assume that similar things exist in close proximity to each

other. This assumption forms the basis of the KNN algorithm, which involves determining the K nearest points to a query point, and using the class of those nearest neighbors to predict the class of the query point. The value of K is chosen by the user, and it determines how many neighboring data points are taken into consideration when making a prediction.

The KNN classifier can be used for both classification and regression problems. In classification, KNN predicts the class membership of a new observation based on the majority class of its K closest neighbors. In regression, the predicted value is the average value of the K nearest data points.

One of the key advantages of the KNN algorithm is its simplicity and interpretability. It also does not require any training time, as the algorithm simply needs to store the training data in memory. Another advantage is its ability to handle multi-class problems, where the KNN algorithm assigns a class based on a plurality vote among the K nearest neighbors. Furthermore, the KNN classifier can be used for both classification and regression problems, but it is more commonly used for classification. Here are a few more advantages:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

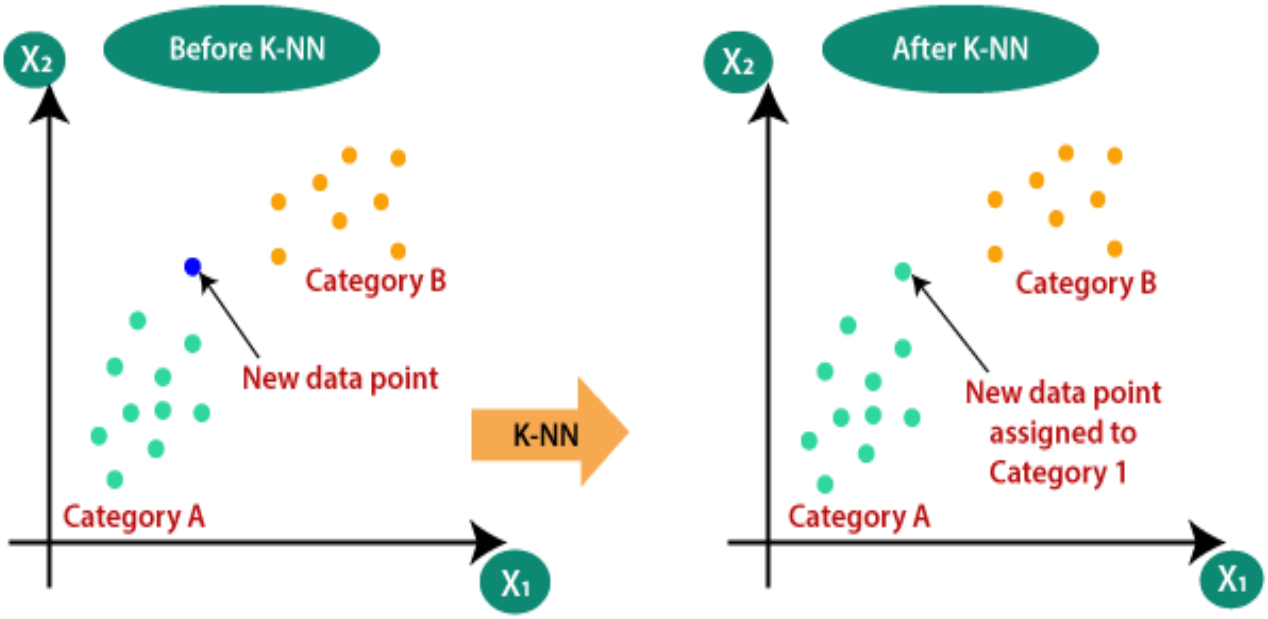

Fig 3.5. KNN algorithm

3.3 Project Specifications

This is a description of a software application that helps to find missing people by matching their images with the ones reported by responsible citizens. The application has three main components: a login page, an application window and UI for entering images.

Login page: The login page allows the user to authenticate themselves with their username and password. Only authorized users can access the application window, where they can perform various tasks related to the missing people cases.

Application window: This contains the main GUI for a user to submit a case, match the case, refresh the model or train it on the new data, view submitted cases and finally see all the confirmed cases. The application window has the following features:

- Submit New case: This feature enables the user to enter a new case of a missing person. A new window will open where the user can provide the details of the person, such as name, age, father's name, contact details, and an image. The image should be clear and recent, and preferably show the face of the person.

- View reported case: This feature allows the user to view the cases that they have submitted previously. The user can see the details and the status of each case, such as pending, matched, or confirmed.

- Match case: This feature initiates the process of matching the images of the submitted cases with the ones reported by responsible citizens. The application uses a deep learning model to compare the

facial features and find potential matches. The user can review the matches and confirm or reject them.

- Refresh model: This feature enables the user to train the model on the updated dataset. The dataset consists of the images and details of the missing people and the responsible citizens. The user can refresh the model periodically to improve its accuracy and performance.

- View confirmed cases: This feature allows the user to see the cases that have been confirmed by the system. The user can see the details of the missing person and the responsible citizen who reported them, as well as their location and contact information.

GUI for responsible citizens: The application also has a GUI for responsible citizens who want to help find missing people. The GUI allows them to enter the images, location, and contact details of any person they suspect to be missing. This data is then used by the application to match with the images of reported people.

4. IMPLEMENTATION

Login Page:

The figure shows a screenshot of the login window that appears when a user launches the application. The user has entered a valid username and password in the corresponding fields and is ready to click on the login button. The figure demonstrates how the user can access the main features of the application after providing the correct authentication information.

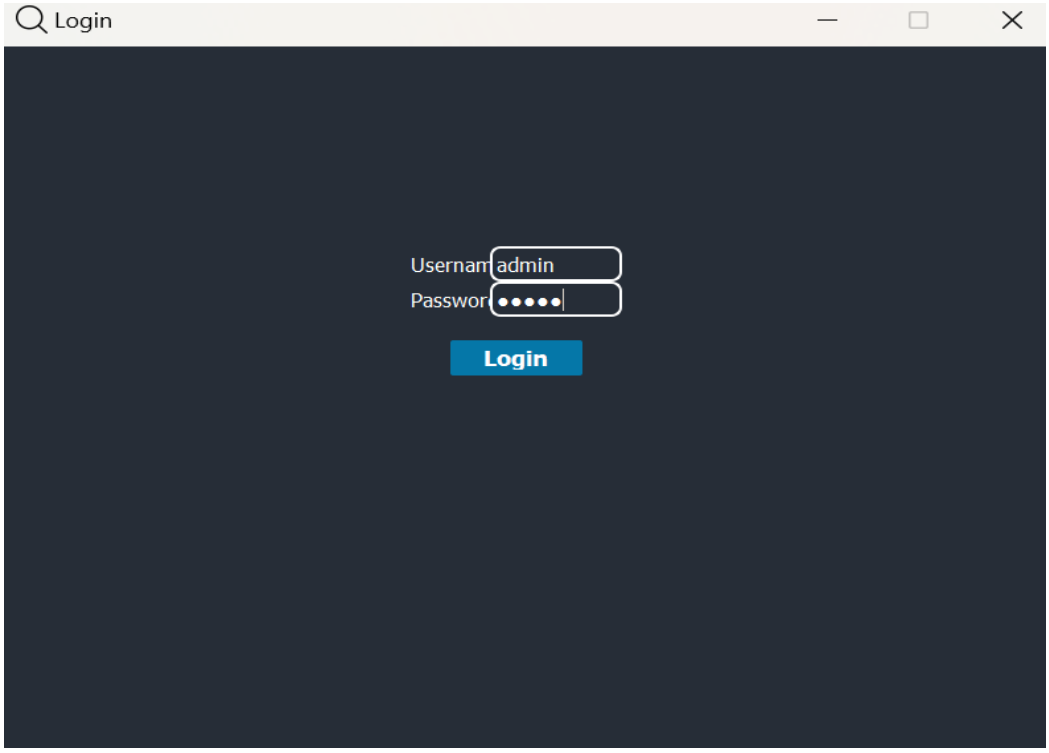

Fig 4.1. Start-up page and Successful login authentication

The login process was unsuccessful because one or more of the required fields were left blank. Please make sure to enter your username, password, before clicking the login button.

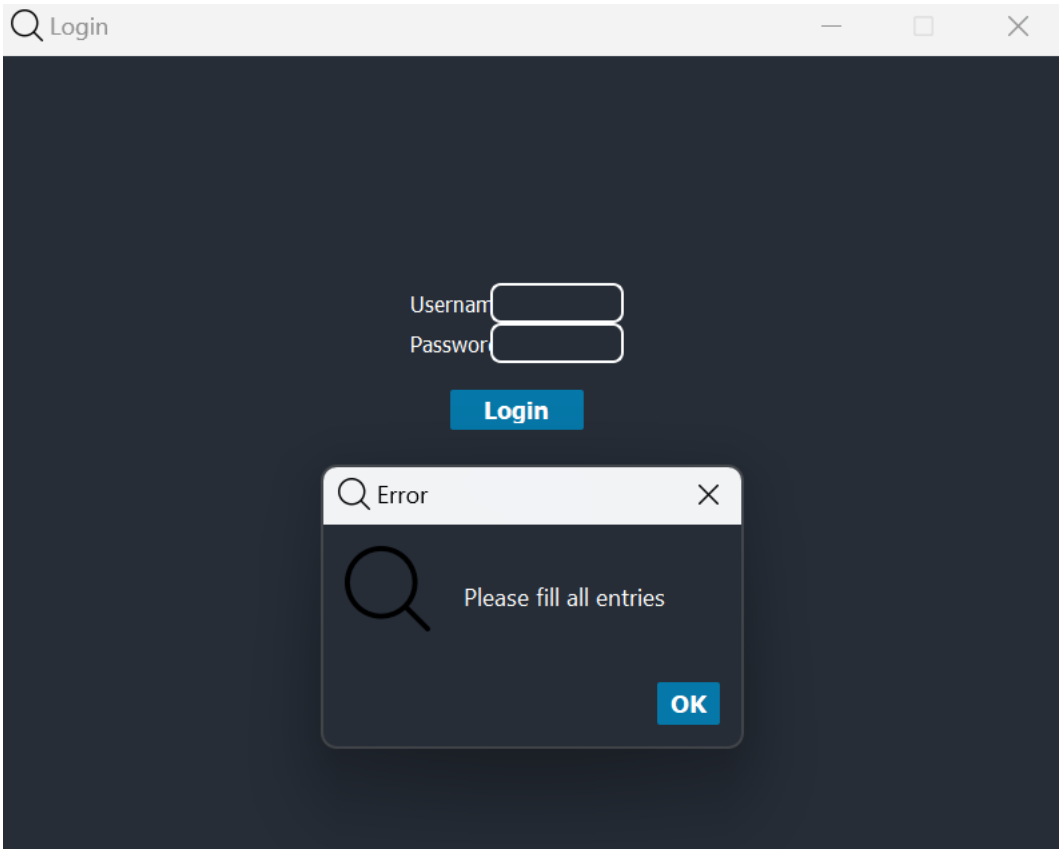

Fig 4.2. Login fail due to not filling all entries

One of the common reasons why users cannot access their accounts is that they have entered an incorrect username or password. This can happen due to various factors, such as typing errors, case sensitivity, forgotten credentials, or unauthorized changes. If you encounter a login failure message, please check that you have entered your username and password correctly

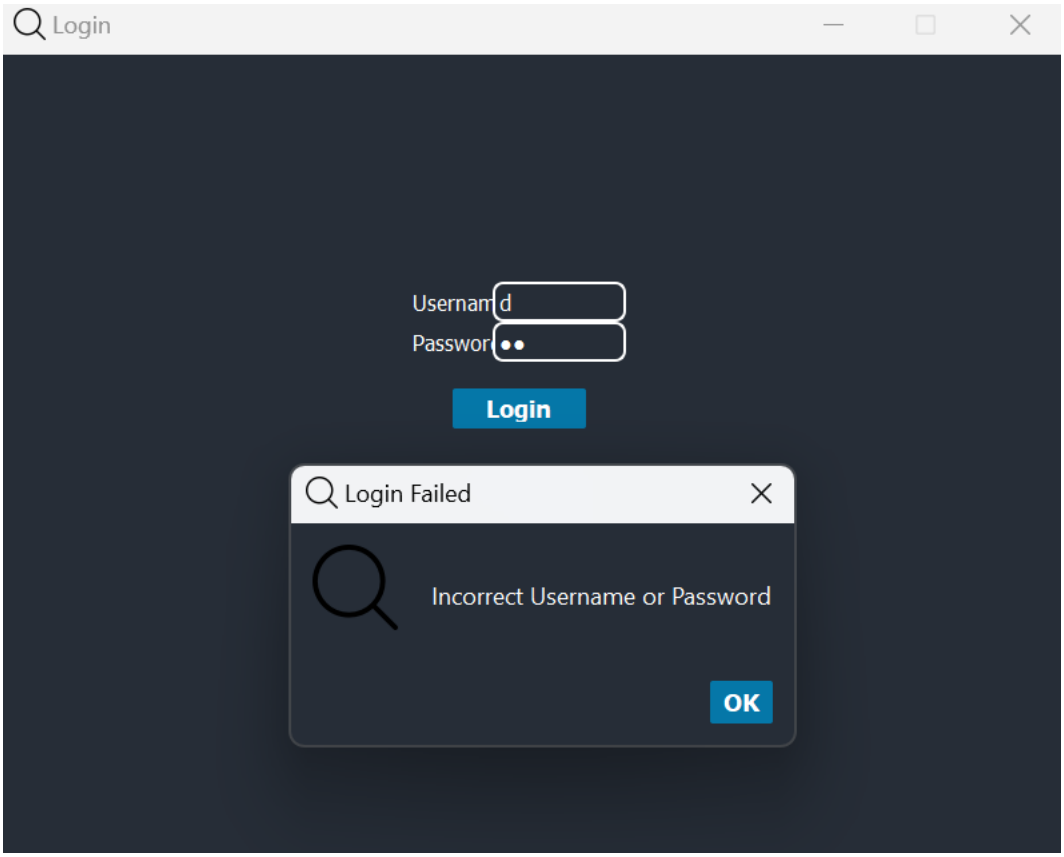

Fig 4.3. Login fail due to providing incorrect username or password

App window:

The application window start-up page is the first screen that the user sees when they launch the application. It provides an overview of the main features and functions of the application, as well as a navigation menu to access different sections.

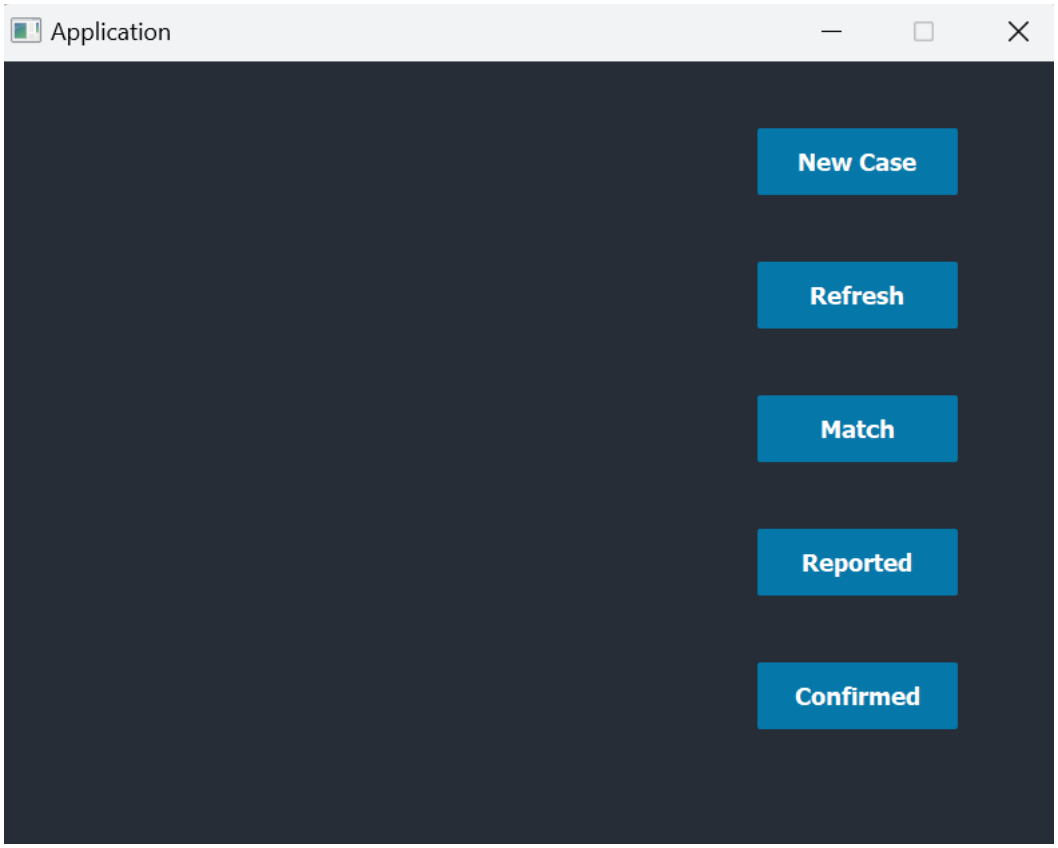

Fig 4.4. Application window start-up page

This feature enables the user to enter a new case of a missing person. A new window will open where the user can provide the details of the person, such as name, age, father's name, contact details, and an image. The image should be clear and recent, and preferably show the face of the person.

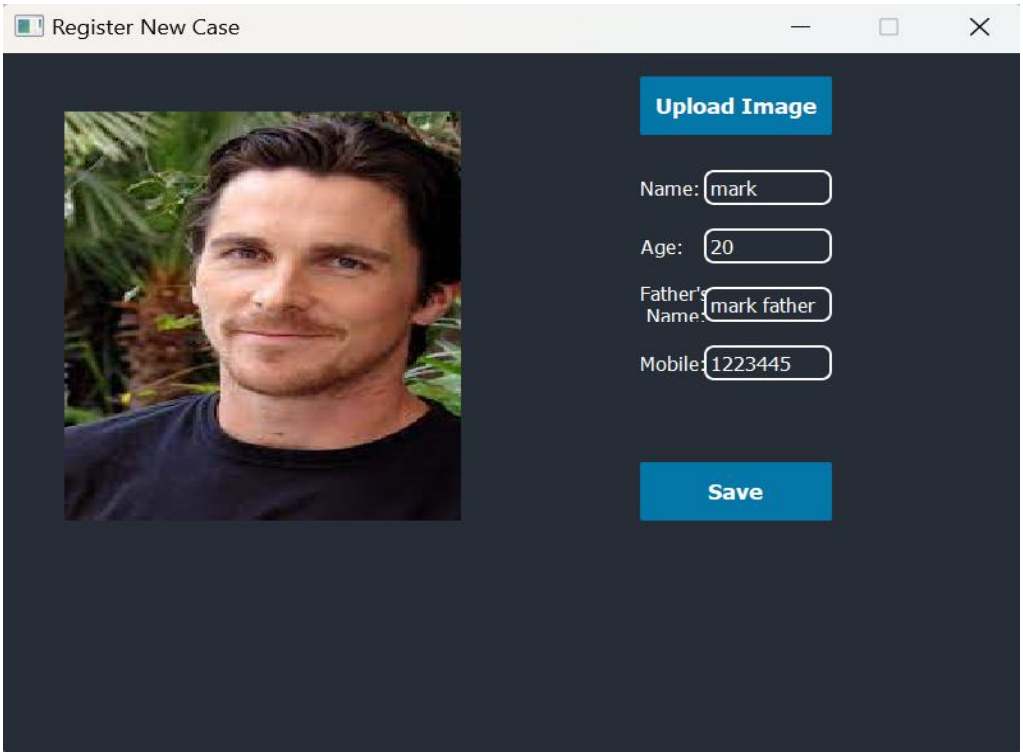

Fig 4.5. Registering new cases

This feature enables the user to train the model on the updated dataset. The user can refresh the model periodically to improve its accuracy and performance.

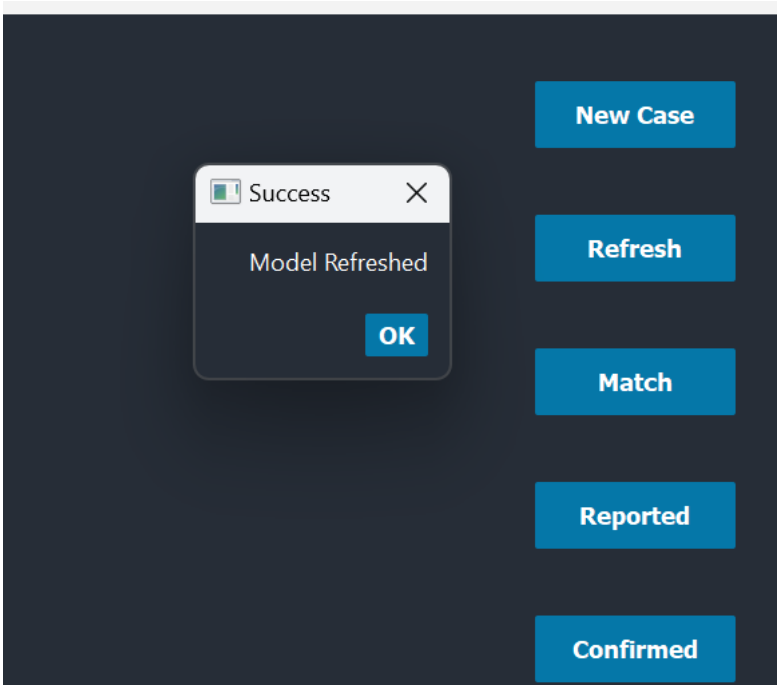

Fig 4.6. Message received after successful model refresh

To initiate and view the match of images from the database, click on the "Match" button to start the matching process.

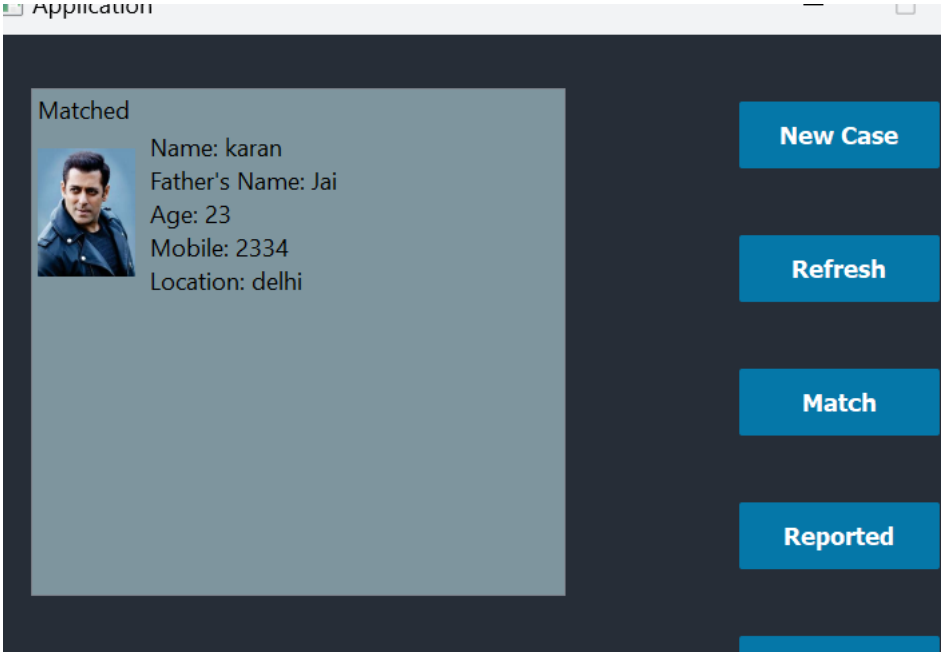

Fig 4.7. Initiating and viewing the match

To view the reported cases with their details, one needs to log in to the system with their username and password and click on the "Reported" tab on the right sidebar. One can see the status of the cases along with their details.

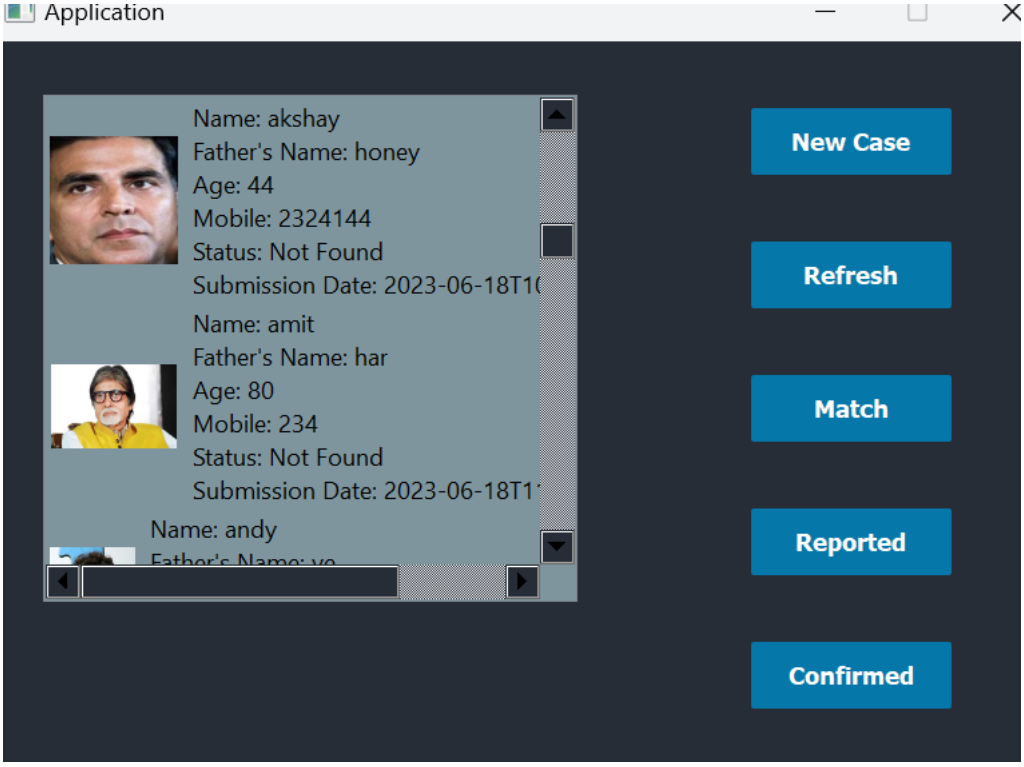

Fig 4.8. Viewing reported cases with their details

One can see the confirmed cases by clicking the confirmed button.

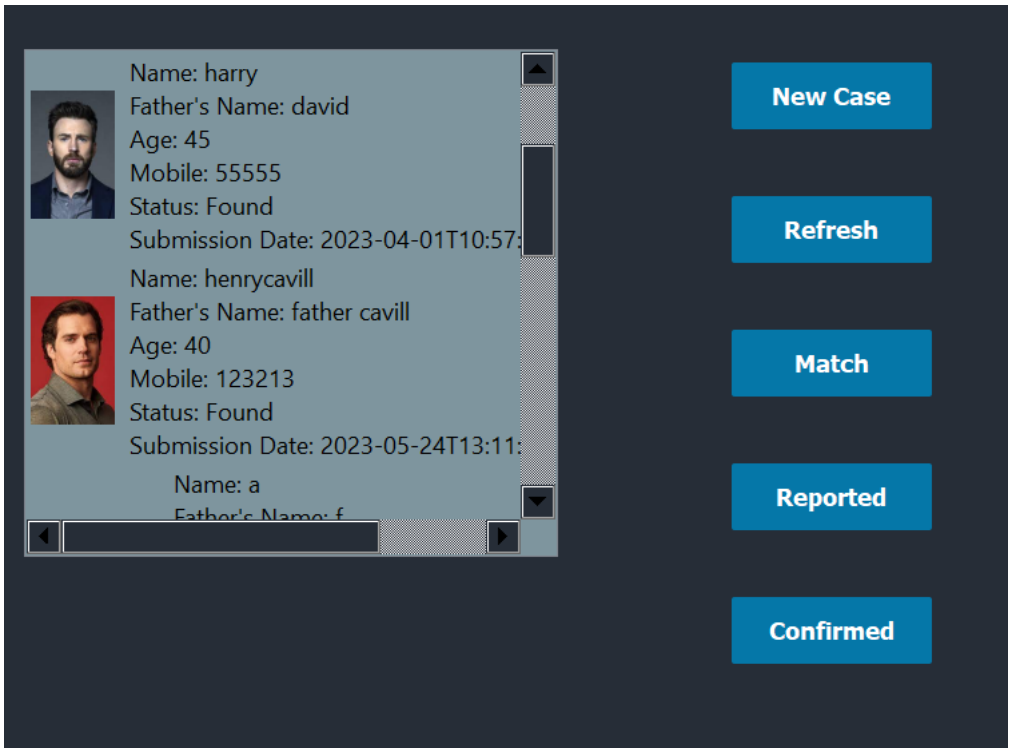

Fig 4.9. Confirmed cases

GUI:

The application also has a GUI for responsible citizens who want to help find missing people. The GUI allows them to enter the images, location, and contact details of any person they suspect to be missing. This data is then used by the application to match with the images of reported people.

This shows the start up page to submit the image of a potential missing person.

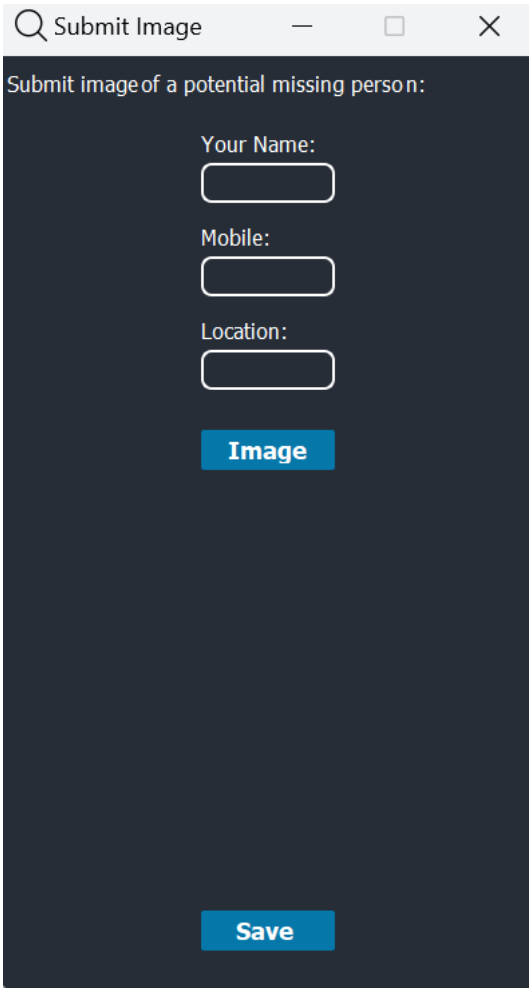

Fig 4.10. Start-up page of GUI

When the user tries to submit without providing all the necessary details, the application prompts the user to enter all the entries.

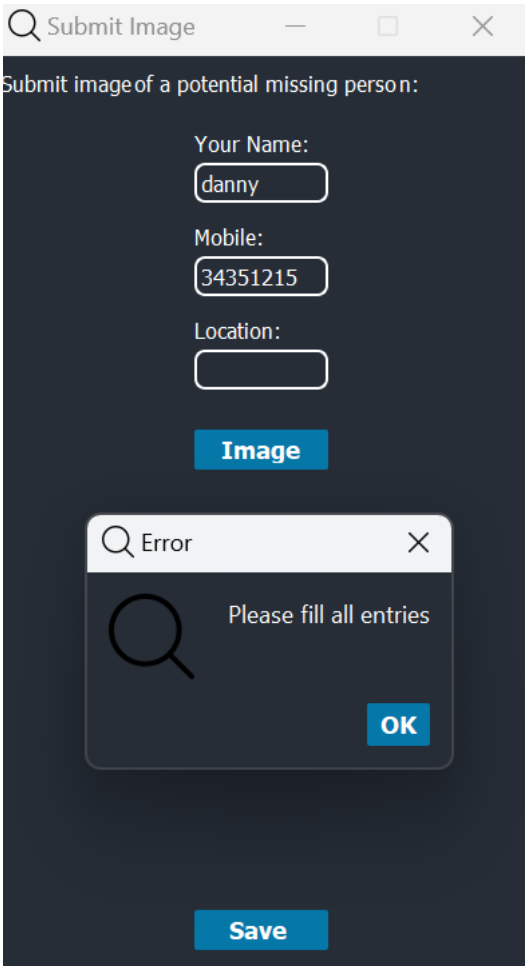

Fig 4.11. Error while submitting without filling all details

A successful submission of details and image to the database.

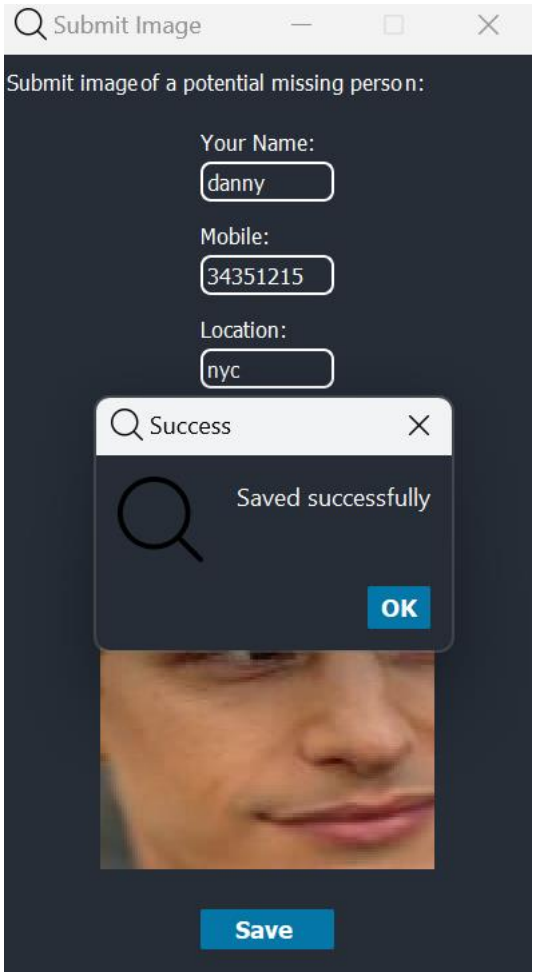

Fig 4.12. Successful details and image upload

If the model is not able to detect the face in the provided image, it displays an error message so that a better quality image shall be uploaded. This will allow to ensure that images in the database are clear and lucid.

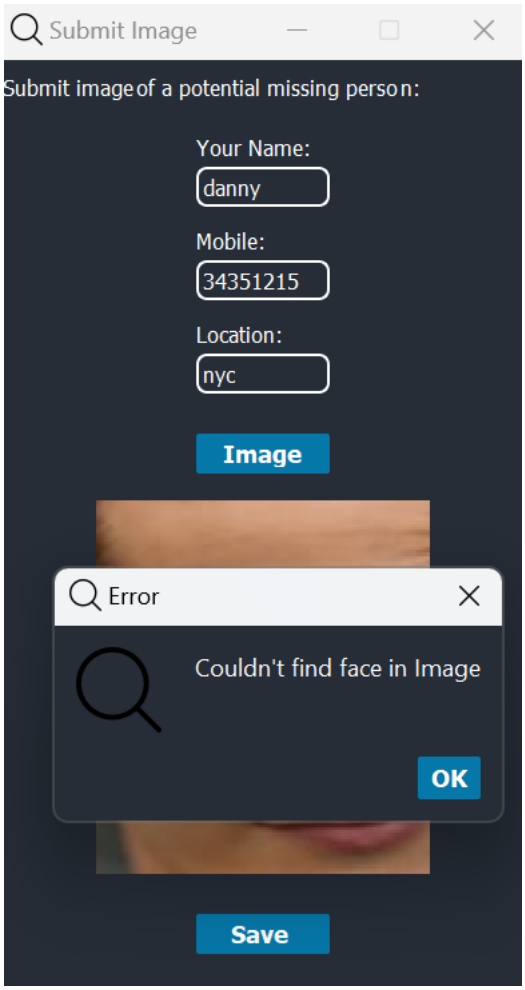

Fig 4.13. Message when the model is not able to detect face

5. Conclusion

The project to find missing people using face recognition has been a success. The system has been able to identify missing persons with a high degree of accuracy, and could be used to reunite many families. The project was successful in identifying and locating several missing persons, as well as providing valuable insights into the challenges and limitations of the face recognition technique. The project also demonstrated the potential of using face recognition technique for other applications, such as security, verification, and identification.

The system works by comparing the facial features of a missing person to the facial features of people in a database of reported missing persons. The facial features are extracted from the images using a face recognition algorithm.

Once the facial features have been extracted, they are compared to the facial features of people in the database. The comparison is done using a distance metric. The distance metric measures the similarity between two sets of facial features. If the distance between the two sets of facial features is below a certain threshold, then the system concludes that the two faces are a match.

The system is quite effective and has the potential to be a powerful tool for finding missing people. With more developments, the accuracy and speed of the system will improve. This will make it even more effective at finding missing persons and reuniting families.

The following recommendations are made for the future development of the system:

- The system should be expanded to include a larger database of reported missing persons. This will increase the chances of finding a match for a missing person.

- The system should be made more user-friendly. This will make it easier for people to report missing persons and for law enforcement to use the system.
- The system should be integrated with other law enforcement databases. This will allow the system to access more information about missing persons, which will improve the accuracy of the matches.

The system has the potential to be a powerful tool for finding missing people. With further development, the system can become even more effective at reuniting families.

References

1. King, Davis E. "Dlib-ml: A machine learning toolkit." *The Journal of Machine Learning Research* 10 (2009): 1755-1758.
2. Guo, Gongde, et al. "KNN model-based approach in classification." *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer Berlin Heidelberg, 2003.
3. Momjian, Bruce. *PostgreSQL: introduction and concepts*. Vol. 192. New York: Addison-Wesley, 2001.
4. Willman, Joshua, and Joshua Willman. "Overview of pyqt5." *Modern PyQt: Create GUI Applications for Project Management, Computer Vision, and Data Analysis* (2021): 1-42.
5. Gagliardi, Valentino. "Setting Up a Django Project." *Decoupled Django: Understand and Build Decoupled Django Architectures for JavaScript Front-ends*. Berkeley, CA: Apress, 2021. 53-61.
6. Pohanka, Tomas, Vilem Pechanec, and Vendula Hejlova. "Python Web Server for Sensor Data Visualization." *International Multidisciplinary Scientific GeoConference: SGEM* 1 (2016): 803-810.
7. He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
8. Oliphant, Travis E. *A guide to NumPy*. Vol. 1. USA: Trelgol Publishing, 2006.
9. Reback, Jeff, et al. "pandas-dev/pandas: Pandas 1.0. 5." *Zenodo* (2020).
10. Python, Why. "Python." *Python Releases Wind* 24 (2021).
11. Zhao, Wenyi, et al. "Face recognition: A literature survey." *ACMcomputing surveys (CSUR)* 35.4 (2003): 399-458.
12. Jain, Anil K., and Stan Z. Li. *Handbook of face recognition*. Vol. 1. New York: springer, 2011.

13. Jafri, Rabia, and Hamid R. Arabnia. "A survey of face recognition techniques." *journal of information processing systems* 5.2 (2009): 41-68.
14. Parveen, Pallabi, and Bhavani Thuraisingham. "Face recognition using multiple classifiers." *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. IEEE, 2006.
15. Ebrahimpour, Hossein, and Abbas Kouzani. "Face recognition using bagging KNN." *International conference on signal processing and communication systems (ICSPCS'2007) australia, gold coast*. sn, 2007.
16. Kamencay, Patrik, et al. "A novel approach to face recognition using image segmentation based on spca-knn method." *Radioengineering* 22.1 (2013): 92-99.
17. Kumar, Ashu, Amandeep Kaur, and Munish Kumar. "Face detection techniques: a review." *Artificial Intelligence Review* 52 (2019): 927-948.
18. Viola, Paul, and Michael J. Jones. "Robust real-time face detection." *International journal of computer vision* 57 (2004): 137-154.