Bachelor of Science in Computer Science & Engineering

Finding Missing Persons using Face Recognition System Based on K-Nearest Neighbor Algorithm

by

Md. Talat Morshed

ID: 1504017

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

April, 2021

Finding Missing Persons using Face Recognition System Based on K-Nearest Neighbor Algorithm

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Md. Talat Morshed
ID: 1504017

Supervised by

Muhammad Kamal Hossen
Associate Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled '**Finding Missing Persons using Face Recognition System Based on K-Nearest Neighbor Algorithm'** submitted by ID: 1504017, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Muhammad Kamal Hossen
Associate Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Abu Hasant Mohammad Ashfak Habib
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

The success and final finish of this project necessitated a great deal of direction and assistance from many people, and I consider myself quite fortunate to have received this during my project's completion.

I respect and thank my project supervisor Muhammad Kamal Hossen, Associate Professor, Department of CSE, CUET for providing me an opportunity to do the project work and giving me all support and guidance. I am grateful to him for offering such excellent assistance and advice.

Dr. Md. Mokammel Haque, our department's head of department, took a genuine interest in our project work and directed us through it till it was completed by providing all of the essential information for establishing a good system.

I'd want to express my gratitude to Dr. Abu Hasant Mohammad Ashfak Habib, my external project adviser, for his advice and ideas during this project.

I would not forget to remember our honorable teachers, of Department of CSE, CUET for their encouragement and more over for their timely support and guidance till the completion of my project work.

I am grateful for and blessed to receive continual encouragement, assistance, and guidance from all members of the teaching staff of Department of CSE, CUET which helped my in successfully completing our project work.

Abstract

Around 20,000 children from Bangladesh go missing every year. A system for recognizing a person from an image or video is called facial recognition. This technology has been present for decades, but in recent years, its use has grown in prominence and accessibility as it now powers creative solutions, such as Criminal Detection using face data, Finding missing person etc. Missing Person identification is a project that can help to detect a missing person based on their face data using K-Nearest Neighbors(KNN) algorithm. The project is implemented as a website so that people find it easy to use. It is a platform based on interaction of Civilians and Police to find missing persons using image data. Images are uploaded from both sides. Image of a person is converted to gray scale image, processed with Histogram Equalization to adjust contrast. Haar Feature-based Cascade Classifiers is used to detect face in the image. Face features are extracted from the detected faces. The features are saved in CSV file. K-Nearest Neighbors algorithm is used to match face data and identify a person.

Keywords: Missing Person identification, KNN Algorithm, Haar Feature-based Cascade Classifiers

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Face recognition is a system that uses an image or video to recognize a person. This technology has been present for decades, but in recent years, it has grown more visible and accessible as it now fuels creative solutions, such as Criminal Detection using face data, Finding missing person etc. Missing Person identification is a project that can help to detect a missing person based on their face data using K Nearest Neighbor(KNN) algorithm. The project is implemented on a website so that people find it easy to use. It is a platform based on interaction of Civilians and Police to find missing persons using image data. The website also provides records of Missing Persons.
In this chapter, we shall discuss the framework, objective and motivation of the project and the challenges we may face while implementing the project.

1.2 Framework/Design Overview

Our developed framework has 3 components. They are given below

- **Phone Verification:** The phone numbers of the people that are uploading images are verified via One-Time Password verification. A random number is generated and sent to user so that he can verify his/her phone number.
- **Client Application:** General users can upload details of missing persons. Police has to login to file a missing person case. Police can also view found person records. Face features are extracted from uploaded images and saved in CSV files.

- **Face Recognition:** In this part, uploaded images are used to find the missing persons. There is also an option for uploading videos. From the video, if a person is found there is a message sending system to notify the authority about the found person.

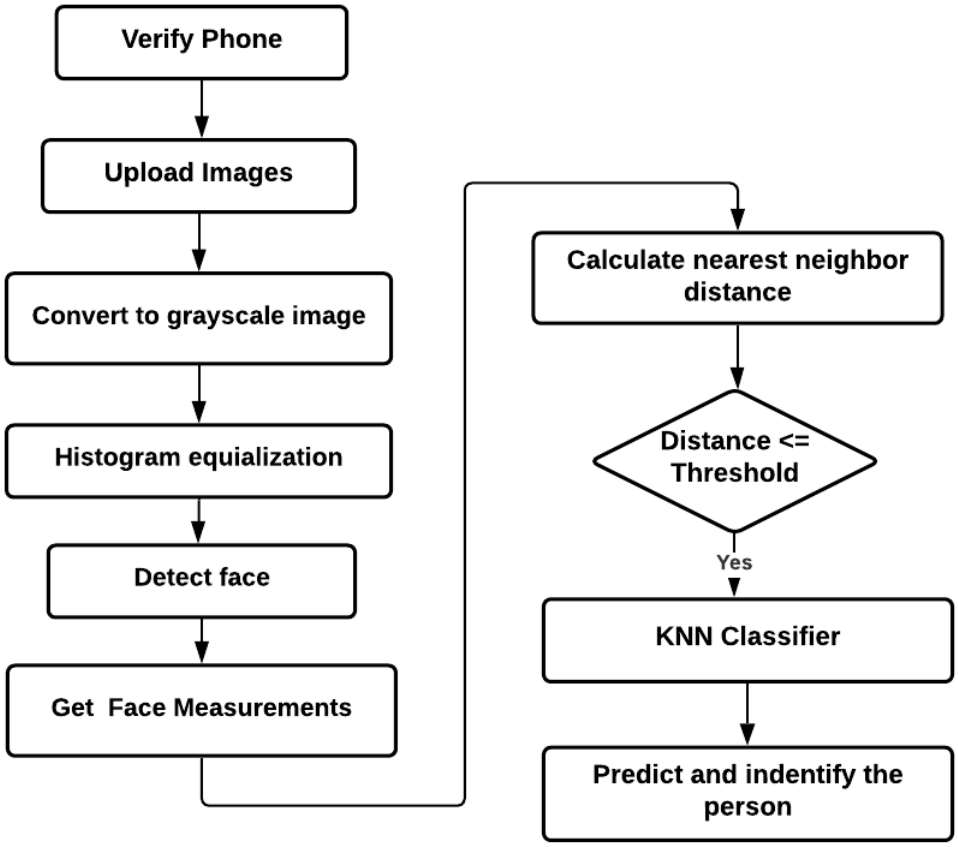


Figure 1.1: Overview of the framework

1.3 Difficulties

There were some difficulties while implementing the project.

Storing huge amounts of images is a big problem. We need a lot of storage. Maintaining them are also difficult. Storing images efficiently is a major challenge. Image pre-processing is also a big challenge. Images are needed to be processed so that in challenging scenarios faces can be detected successfully.

Choosing proper model for prediction is also a difficulty. Many algorithms are needed to be implemented to choose the right prediction model.

1.4 Applications

Applications of our work are given below:

- Identify people in photos or videos.
- Finding missing persons and wanted persons.
- Phone verification system using One Time Password.
- Dataset of missing persons and wanted persons can be used in future.
- During police stops, officers can also use mobile devices to identify individuals.

1.5 Motivation

Motivations for our work are:

- Around 20,000 children from Bangladesh go missing every year.
- Lack of a platform to find missing persons.
- Searching for a person in social media is not effective and time consuming.
- There is not automated finding system for missing peoples.

1.6 Contribution of the thesis

The goals for the work are given below:

- Face recognition is an old technology, but it's uses are limited in our country. We have built a web project that can help people in finding missing persons.
- Finding efficient way to store huge amount of images. Face features are saved in a CSV file. So, recognition takes less time.
- Implementing phone verification system so that people can not misuse the project.
- Identifying missing persons from uploaded video

- Implementing face recognition system using KNN algorithm

1.7 Thesis Organization

The structure of the report can be described as follows:

- Chapter 1 has the overview of the project. It explains challenges, applications and motivations of the project.
- Chapter 2 has the overview of technologies that are used in this project. It also compares our project with some related projects.
- Chapter 3 has the methodology and implementation details of the project.
- Chapter 4 provides experimental results and performance evaluation of the project
- Chapter 5 represents overall summary of this project and recommendation for future work related to this work.

1.8 Conclusion

An overview of the project is given in this chapter. It provides information about applications, motivations and challenges of the project. It also shows contribution of the project.

Chapter 2

Literature Review

2.1 Introduction

We will discuss about the terms and technologies used in our project in this chapter. We will also discuss about the implementation challenges that we have faced during implementing the project in this chapter.

2.2 Python

Python is an interpreted high-level general-purpose programming language. The readability of the code is a priority in its design principles. Python is very popular now. It's object-oriented methods and programming concepts are intended to help programmers for small and large projects writing clear, logical code.
Python is a dynamically typed programming language. Python supports a variety of design patterns. As python has very extensive standard libraries to work with, for this reason python is sometimes referred to as a language with "batteries included" . The design patterns include object-oriented, structured (especially procedural) and functional programming.
Python's advantages are:

- It's simple to read, learn, and write
- Productivity Gains
- Interpreted Language
- Typed Dynamically
- Support from a Large Library

2.3 Django

Django is an MVT web platform for developing web applications. Youtube, Google, NASA and various giant companies use django to build their website. Unlike other systems, the models are all contained in a single file, models.py, which can make larger projects feel crowded.
Django's model makes use of a powerful ORM layer. The ORM layer makes database and data management easy to use and speeds up the process of development. Django's features are given below:

- Documentations are Excellent
- Rapid Development Support
- Framework is based on Python
- It is Highly Scalable

2.4 HTML

HTML means HyperText Markup Language. HTML is a markup language and it combines hypertext and markup. It is a markup language used to build web pages. The connection between web pages is referred to as "hypertext".
It's features are:

- It's easy to understand and use
- A web page can include images, videos, and audio
- Text can be enhanced with hypertext
- It is employed in the creation of a website.

2.5 CSS

The language that we use to style a Web page is CSS.

- CSS means Cascading Style Sheets

- CSS describes the way of displaying HTML elements on digital devices like computer or in various medias
- CSS helps us to save time and our effort. It can keep track of the layout of multiple web pages at the same time.

2.6 Javascript

JavaScript is a lighthearted and adaptable programming language. It's one of the most important web development tools, and it can be used on both the front-end and back-end. It is:

- Adaptable and durable
- Pleasant to the front-end
- Pleasant to the back-end
- Allows web applications to function

2.7 SQLite

SQLite is a C library that contains a relational database management system (RDBMS). SQLite is not a client–server database engine, unlike many other database management systems. Rather, it is incorporated into the final product.
SQLite is ACID-compliant. SQLite implements the SQL format to a great measurement. For local/client storage, SQLite is a popular alternative for embedded database software in application software such as web browsers. It is arguably the most commonly used database engine, as it is currently used by a number of embedded systems, operating systems, and popular browsers.

2.8 Face Recognition

Over the years, human face detection and recognition techniques have gotten a lot of attention, and a lot of algorithms have been developed. Face recognition

provides a number of benefits in society, including enhanced safety and protection, crime prevention, and reduced human contact. Face recognition has many potential applications in computer vision, surveillance system, semantic video analysis and automatic access control systems [1]. It may also help pay medical efforts in some circumstances. It has a lot of benefits. Some are given below:

- Helps finding missing people
- Defends companies from fraud
- Enhances the security measures
- Organizes photos better

2.9 Haar Feature-Based Cascade Classifiers

In their paper "Rapid Object Detection using a Boosted Cascade of Simple Features," Paul Viola and Michael Jones propose an efficient object detection approach based on Haar feature-based cascade classifiers. It's a machine-learning technique in which a cascade function is trained using a large number of positive and negative images. After that, it's used to find objects in photographs.
To train the classifier, the algorithm requires a large number of positive images (images of faces) and negative images (images without faces). We will have to get rid of some of the features then. This is accomplished through the application of Haar characteristics. They look a lot like our convolutional kernel. The number of pixels beneath the white rectangle is subtracted from the total number of pixels beneath the black rectangle to yield a single integer for each feature.
The Haar Cascade classifier analyzes pixels in an image and divides them into squares based on their purpose, using the Haar Wavelet approach. The "features" detected are computed using "integral picture" concepts. The In Haar Cascades, the Ada-boost learning technique is used to select a small number of significant characteristics from a vast collection in order to produce a useful result. Classifiers then use cascading techniques to detect the face in an image.
Based on the Viola-Jones recognition, the Haar cascade classifier algorithm, which is trained using a collection of input faces and non-faces to train a classifier that

recognizes faces.

2.10 K-Nearest Neighbor Algorithm

KNN may be used to tackle both classification and regression predicting issues. But, it is more typically utilized in categorization difficulties on the market. When analyzing any approach, we look at three main factors:

- Predictive Accuracy
- Time for Calculation
- Outputs are simple to understand

The KNN algorithm performs well across all parameters. It is widely used as KNN is simple to comprehend and calculate in a short amount of time. The algorithm saves all available data and classifies new data points based on their similarity to previous data points. This indicates that the K-NN algorithm can swiftly classify new data into a well-defined category. KNN algorithm is known as lazy learner algorithm because it doesn't learn from the training set straight away. The algorithm stores the information to use it to classify it later.

Working procedure of KNN algorithm:

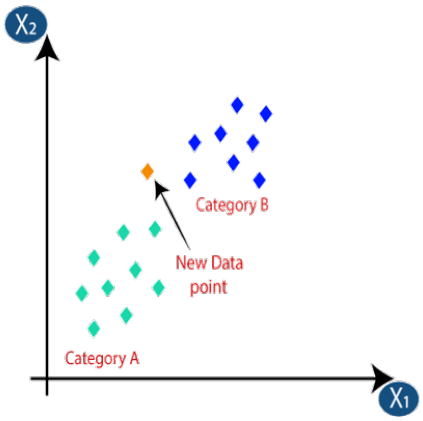


Figure 2.1: Choosing a data point

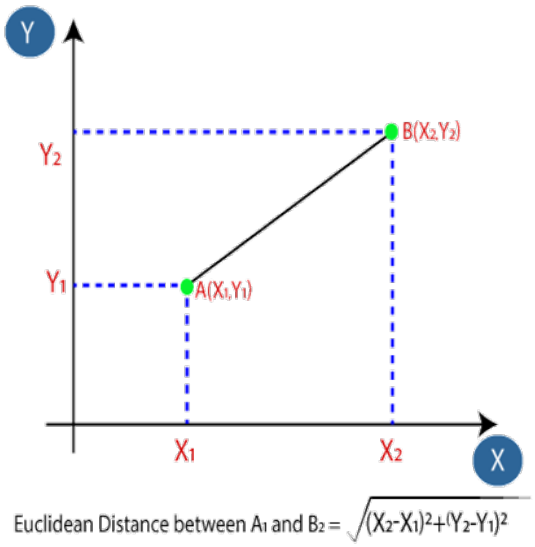


Figure 2.2: Calculating euclidean distance

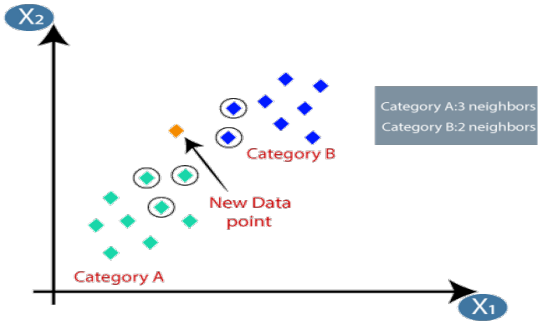


Figure 2.3: Categorizing based on nearest neighbors

2.11 SIFT Feature Detection Algorithm

Scale invariant feature transform (SIFT) has shown to be very powerful for general object detection/recognition. And recently, it has been applied in face recognition [2]. After being collected from a collection of reference images, object SIFT keypoints are extracted and in database they are saved. By comparing each feature in the new image to this database and determining candidate matching features based on the Euclidean distance between their feature vectors, an object in a new image is recognized.

To filter out good matches, keypoints subsets that agree on the item and its position, size, and orientation in the new image are chosen from the whole set of matches. Right object matches are those that pass all of these conditions.

2.12 Face_Recognition Framework

Deep learning and dlib's state-of-the-art face recognition were used to create this. The model achieves a 99.38 percent accuracy on the Labeled Faces in the Wild benchmark.
From Python or the command line, we can control faces as well asrecognize with one of world's easiest face recognition library.

2.13 Related Literature Review

Many work have been done regarding Face Recognition. There are many methods for face detection, face feature extraction, face prediction model etc. Researchers have been trying to improve face recognition system for years.

In [3], Face recognition could be broken down into three stages, according to the writers. For first phase, the supplied color image is translated into multiple color space models. The eigen values and eigen vectors of each color space model are recovered in the second step. For final phase, a closest neighbor classifier is developed to classify the face images using the retrieved features.

In [4],For consumer applications, the authors proposed a near-real-time efficient face recognition device. They looked at support vector machines (SVM), linear discriminant analysis (LDA), and K nearest neighbor classification techniques (KNN).

The authors in [1] proposeda novel technique for Face recognition using a hybrid SPCA-KNN approach (SIFT-PCAKNN). Three steps make up the suggested process. components The first section is focused on face image preprocessing. Graph Based Algorithm and SIFT (Scale Invariant Function Transformation) Feature Transform) descriptor is a descriptor that describes how a feature is transformed.

Researchers in [5] created an enhanced strategy to improve human face recognition by employing a back-propagation neural network (BPNN) and features

extraction based on the correlation between the training images. A crucial addition of this research is the creation of a new collection termed the T-Dataset from the original training data set, which is utilized to train the BPNN.

2.14 Conclusion

In this chapter we have given a literature review of our work in details. We also have discussed some related literature review. Implementations challenges are also discussed in this chapter.

2.14.1 Implementation Challenges

We have to make a website using Django for the project. Learning Django framework, it's ORM, using machine learning in Django are very tough to learn and takes so much time.

Users have to upload many images, storing that many images efficiently is also a challenge.

Extracting facial features from images is also a challenge. There are several methods to do it. We are trying to compare them and use a suitable a one in our project.

Choosing the right model for prediction is also a challenge. We have to try various models, compare them and select the one with best results.

Chapter 3

Methodology

3.1 Introduction

The detailed methodology for our work will be discussed in this chapter. In this chapter, the overview of our project is given in the first portion. Later, detailed methodology of our work is discussed. Then implementation of our work is shown. Finally we have given the conclusion at the end of this chapter.

3.2 Diagram/Overview of Framework

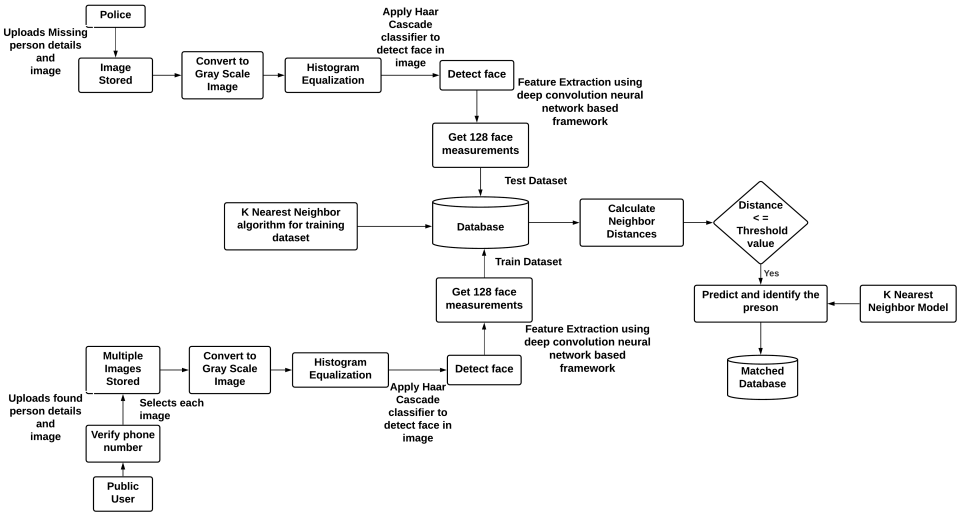


Figure 3.1: Full overview of our system

There can be 3 scenarios:

- General people find a lost person and upload his photos with details. Police files a missing person case with image, the image gets matched and police finds details of the person.

- Police files a missing person case with image. General people find a lost person and upload his photos with details. Later police search for new found persons and find out the people whose images get matched with police's database images.
- Police files a missing person case with image. General people find a lost person and upload his video. If a person gets matched, an sms is sent to police with details.

3.3 Detailed Explanation

One-Time Password Verification Process: General users in our system has to be verified. The verification process ensures that people can not misuse our project. People are verified with their phone number. If any people misuse our project, police can track them via their phone number. National ID number would be more suitable. But because of privacy of national id data, we could not use their data. So we used phone verification system in our project.

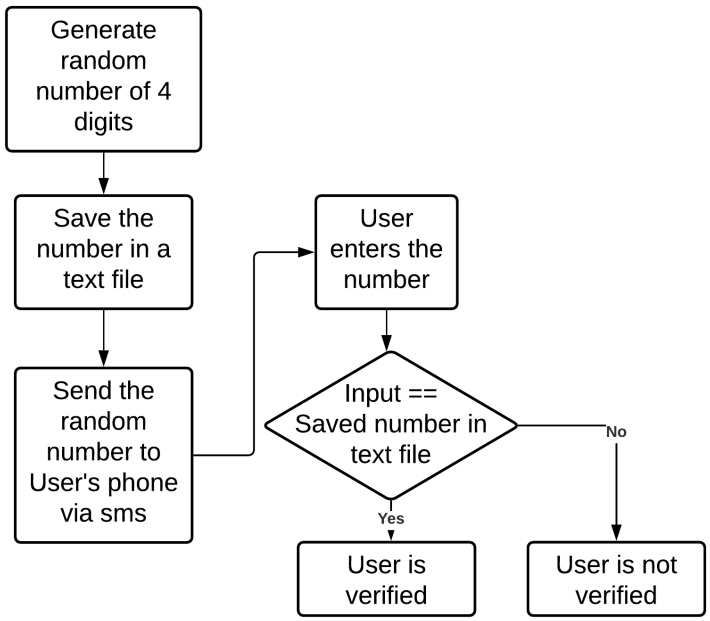


Figure 3.2: One-Time Password verification process

Users of the project:

- **General People:** General people helps in finding missing persons. When

a missing person is found, people need to take pictures of that people. The user has to be verified using his/her phone number. Then user uploads the details and pictures of the missing person in the website. The images are preprocessed and details are saved in found person database.

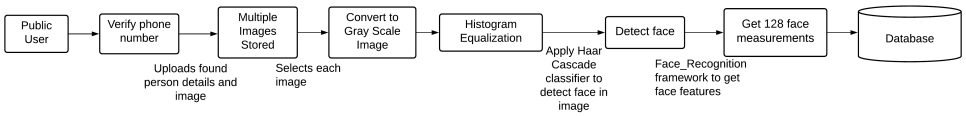


Figure 3.3: Public user uploading details of a found person

General people can also upload video of the found person. If the person gets matched, details are sent to police via sms.

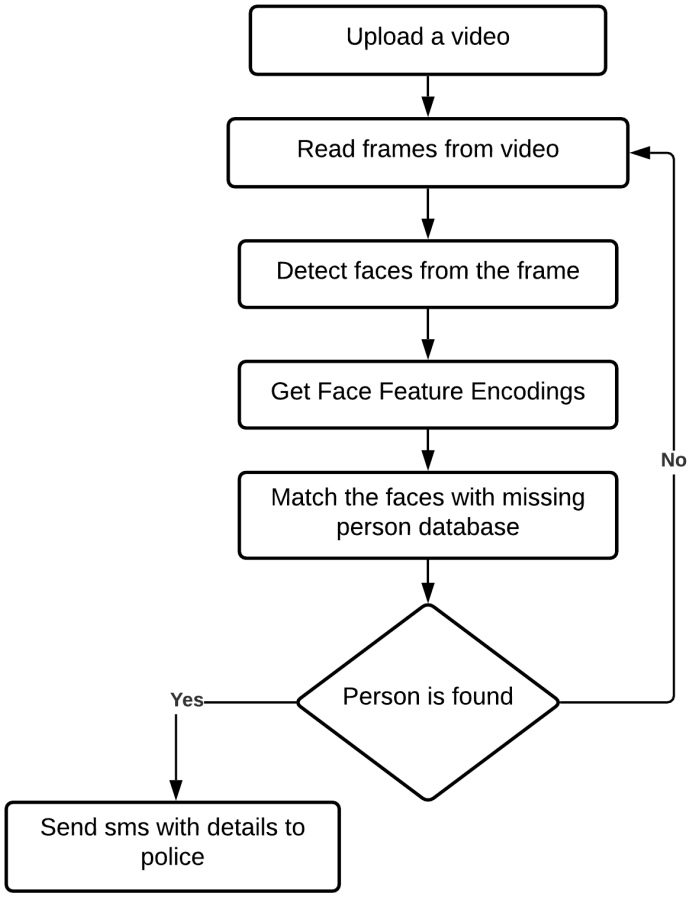


Figure 3.4: Public user uploading video of a found person

- **Police:** Police files missing person cases. Police needs to log in the website. New accounts can be created by the admin of the website. When police files a missing person case, 2 scenarios can happen.

1. The person is matched with found people database, police gets the details of the person.

2. The person is not found. But the details are saved in the missing person database. Later, if the person is found somewhere, police can track them using the database.

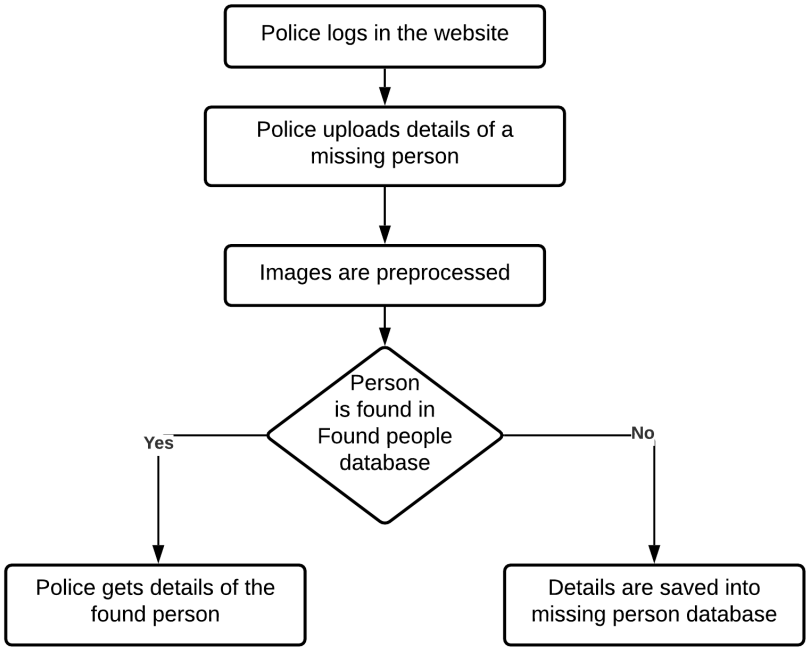


Figure 3.5: Police uploading details of a missing person

3.3.1 Face Recognition System

3.3.1.1 Convert to Grayscale Image

Converting the input image to gray scale is a standard technique in image processing. Since detecting luminance rather than color yields better results in object detection, this is the case.

The explanation for the distinction between such images and every other type of color image is that each pixel requires less details. In reality, In RGB space, a gray color is one in which the red, green, and blue components are all equal in intensity, unlike a full color picture, which requires three intensities for each pixel, a single intensity value is required for each pixel.

3.3.1.2 Apply Histogram Equalization

The histogram of an image is a graphical representation of the pixel intensity levels in the image. A data structure that maintains track of all of the image's pixel intensity levels and their frequency.

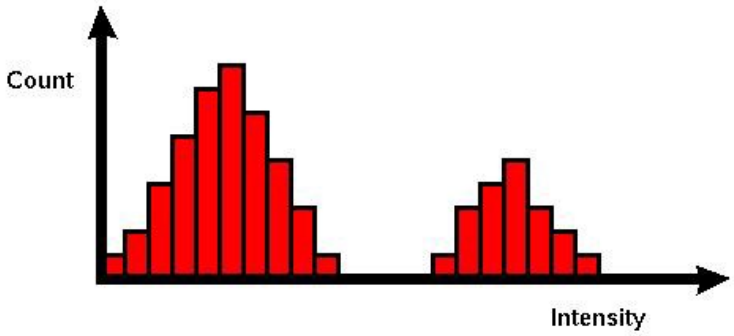


Figure 3.6: Histogram of an Image

The pixel intensity levels of the image are represented by the X-axis in the image above. The strength level is normally between 0 and 255. There is only one histogram for a gray-scale picture.

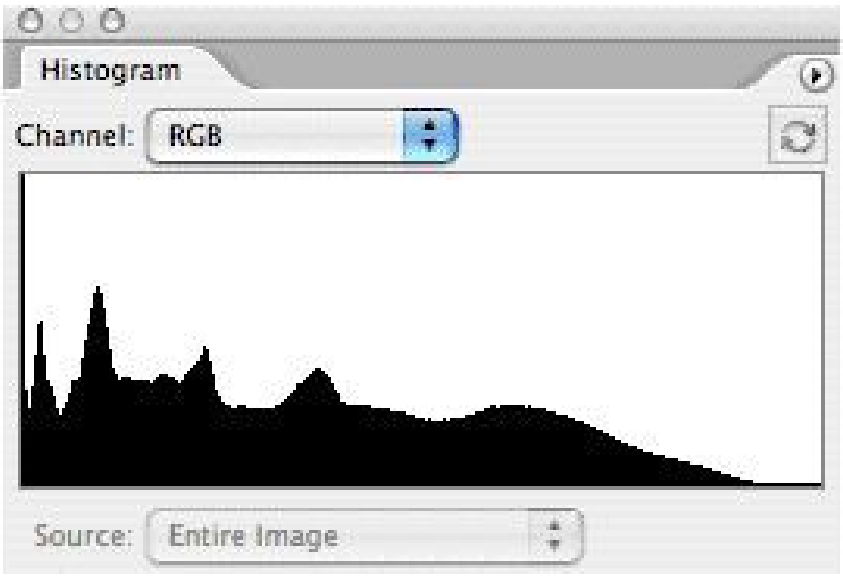


Figure 3.7: Histogram of a Gray-Scale Image

Histogram Equalization is an image processing technique that uses the histogram to change the contrast of an image. It spreads out the most common pixel intensity values or extends out the image's intensity spectrum to improve the

image's contrast. Histogram equalization does this by allowing the image's low-contrast areas to obtain more contrast.

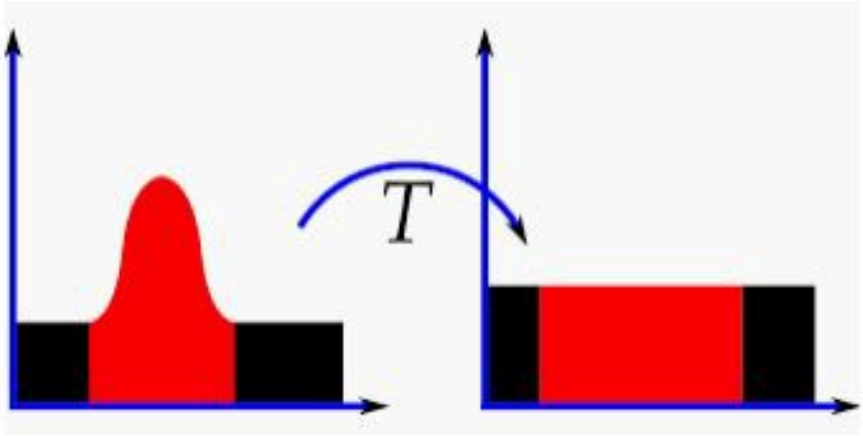


Figure 3.8: Graphical Representation of Histogram Equalization

3.3.1.3 Detect Face in Image using Haar Cascade Classifier

The Haar Cascade classifier analyzes pixels in an image and divides them into squares based on their function using the Haar Wavelet approach. "Integral image" concepts are used to compute the "features" that are detected. The Adaboost learning algorithm is used in Haar Cascades to pick a small number of important features from a large collection to produce an effective result. Classifiers then use cascading techniques to detect the face in an image.

The Haar cascade classifier is based on the Viola-Jones detection algorithm, which is trained by feeding a set of input faces and non-faces into a classifier that can identify a face.

Rather than adding all 6000 features to a single window, divide them into different stages of classifiers and add them one by one. (In most cases, the first few stages would have a small number of features.) If a window fails the first test, it should be discarded. We don't think about the remaining features.

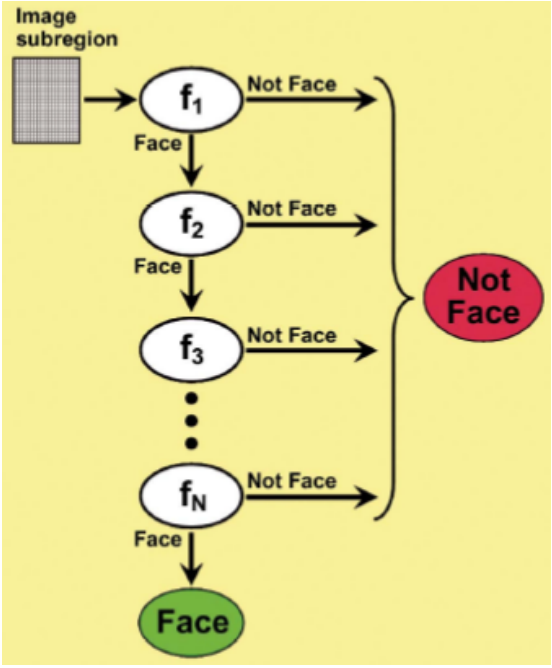


Figure 3.9: Face Detection using Haar Cascade Classifier

Pre-trained classifiers are available in OpenCV. The parameters needed for the classifier are:

- Image: A CV_8U matrix containing an image in which objects are detected.
- ScaleFactor: At each image scale, this parameter specifies how much the image size is reduced. As shown in the picture, this scale factor is used to

Figure 3.10: Viola-Jones Face Detection

make the scale pyramid in the picture. If the scale factor is 1.03, it means we're resizing with a small measure, i.e. shrinking the size by 3%. This increases the chances of finding a matching size with the detection model, but it's expensive.

- MinNeighbors: The number of neighbors each candidate rectangle should have in order for it to be retained is a parameter. The quality of the detected faces is affected by this parameter: a higher value means less detections but better quality.
- Flags: In the function cvHaarDetectObjects, this parameter has the same value for an old cascade. It is not included in the development of a new cascade.
- MinSize: The tiniest object possible. Objects smaller than that are ignored.
- MaxSize: The largest potential object size. Objects that are larger than that are ignored.

3.3.1.4 Face Feature Extraction

We used two methods for feature extraction. Details of the methods are given below:

- SIFT (Scale-Invariant Feature Transform): In the SIFT algorithm, there are four key steps. We'll go through them one by one. 1. Scale-space Extrema Detection: The scale space of an image is a function L(x,y,σ) obtained by convolutioning a Gaussian kernel(Blurring) with the input image at various scales. The number of octaves and scale used in scale-space is determined by the size of the original picture. As a result, we create multiple octaves of the original picture. The image size of each octave is half that of the previous one. Using the Gaussian Blur operator, images are gradually blurred over an octave. Blurring is defined as the convolution of the Gaussian operator and the image in mathematics. Each pixel in a Gaussian blur has a unique expression or "operator" added to it. The effect is a smudged picture.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.1}$$

  Here, G is Gaussian Blur operator and I is an image. x,y are the location coordinates and σ is the "scale" parameter. Consider it the sum of blur. The blur is increased as the value increases.

$$G(x, y, \sigma) = 1/(2\pi\sigma^2)e^{-}(x^2 + y^2)/2\sigma^2 \tag{3.2}$$

After that We'll use those blurred images to create a new collection of images called the Difference of Gaussians (DoG). These DoG photos are excellent for identifying interesting keypoints in an image. The difference of Gaussian blurring of an image of two separate σ is obtained as the difference of Gaussian, let's say σ and kσ, yields the difference of Gaussian blurring. In the Gaussian Pyramid, this process is repeated for different octaves of the image. The following picture depicts it: We've created a scale space and

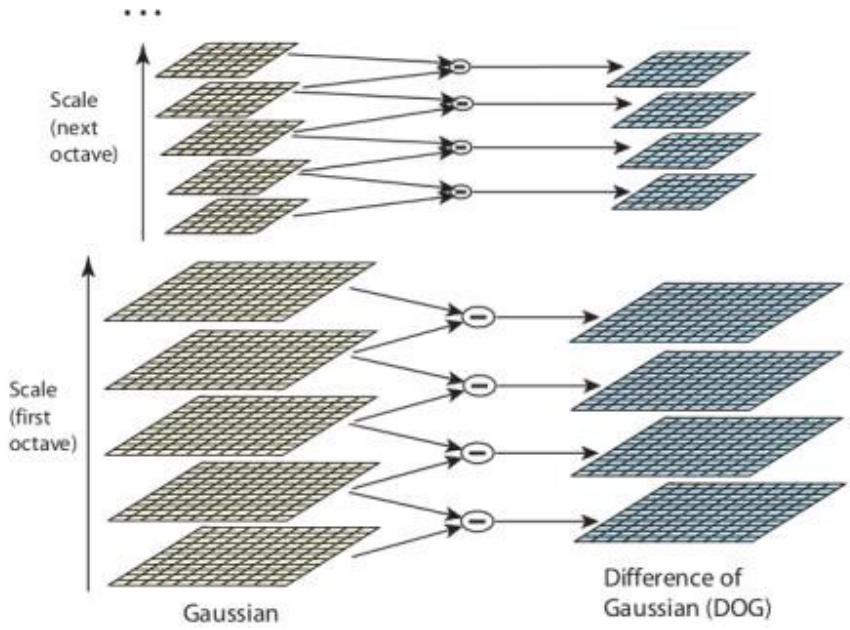


Figure 3.11: Difference of Gaussian kernel

used it to measure the Difference of Gaussians up to this point. These are then used to measure scale-invariant Laplacian of Gaussian approximations.

2. Keypoint Localization: The keypoints generated in the previous phase result in a large number of keypoints. Some of them are too close to the edge, or there isn't enough contrast. They aren't as useful as features in both cases. As a result, we get rid of them. The method is identical to that used to remove edge features in the Harris Corner Detector. We simply check the intensities of low contrast features.

They used a Taylor series expansion of scale space to get a more precise position of extrema. If the intensity at these extrema is less than a threshold value, they reject it. Edges have a higher response in DoG, so they must

be eliminated as well. The principal curvature was calculated using a 2x2 Hessian matrix (H).

3. Orientation Assignment: To achieve image rotation invariance, each keypoint now has an orientation assigned to it. A neighborhood is drawn around the keypoint place, depending on the scale, and the gradient magnitude and direction in that region are determined. The result is a 36-bin histogram with 360-degree orientation. The highest point in the histogram is taken, any peak above 80% of it is also taken into account when determining the orientation. It produces keypoints that are the same size and position, but face in different directions. It helps to keep the matching stable.

4. Keypoint Descriptor: A keypoint descriptor has now been developed. The keypoint is surrounded by a 16x16 neighborhood. It is made up of 16 4x4 sub-blocks. Each sub-block receives an 8-bin orientation histogram. As a result, there are 128 bin values to choose from. A vector is used to express a keypoint descriptor in order to shape it. In addition, several steps are taken to ensure robustness against changes in illumination, rotation, and other factors.

- Face Feature Extraction with face_recognition Framework based on ResNet-34 from the Deep Residual Learning for Image Recognition paper by He et al.: Davis King trained the network on a dataset of over 3 million images. The network outperforms other state-of-the-art approaches on the Labeled Faces in the Wild (LFW) dataset, achieving 99.38 percent accuracy on the Labeled Faces in the Wild benchmark. For each face, it generates 128-d face encodings. These encodings will be used to match faces.

3.3.1.5 Use KNN to Find the Prediction of Face

K-Nearest Neighbour algorithm is one of the most fundamental Machine Learning algorithms. It is based on the Supervised Learning technique. The K-NN algorithm assumes that the new case/data and old cases are identical, and it assigns the new case to the category that is closest to the current categories. The

K-NN algorithm stores all available data and classifies a new data point based on its similarity to the existing data. This implies that new data can be quickly categorized into a well-suited group using the K-NN algorithm as it appears. K-NN technique can be applied to both regression and classification problems. However it is more typically employed for classification. The K-NN algorithm is a non-parametric algorithm. It means that it doesn't make any assumptions about the data.

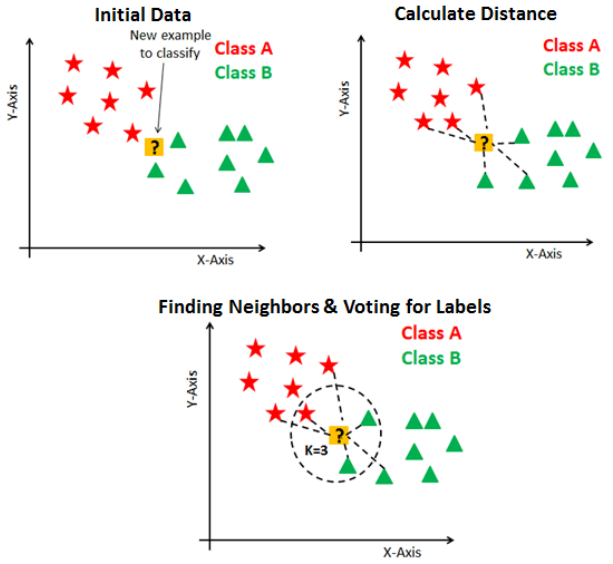


Figure 3.12: KNN Working Process

Steps of KNN Algorithm:

Step-1: Choose the Kth neighbor's number.

Step-2: Determine the Euclidean distance between K neighbors.

Step-3: To use the calculated Euclidean distance, need to find the K nearest neighbors.

Step-4: Calculate how many data points each of these k neighbors has in each category.

Step-5: The new data points are assigned to the groups with that has maximum neighbors number.

Step-6: Our model is complete and ready for use.

3.3.1.6 Use SVM to Find the Prediction of Face

A quick example is the best way to understand the fundamentals of Support Vector Machines and how they function. Let's imagine we have two characteristics in our data: x and y, as well as two tags: red and blue. We're seeking for a classifier that can tell if two (x, y) coordinates are red or blue. We plot our already labeled training data on a plane: A support vector machine uses these

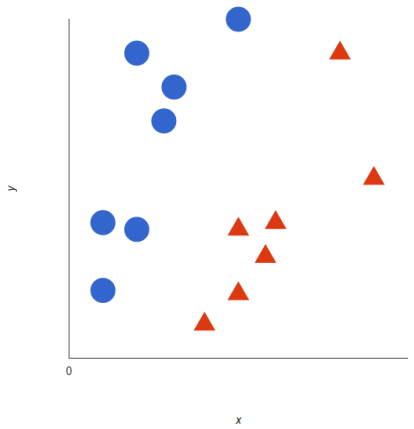


Figure 3.13: Labeled Data

data points to construct the hyperplane, which is a two-dimensional line that best separates the tags. For decision boundary the line is used: something which falls on one side is classified as blue, while something which falls on the other is categorized as red.

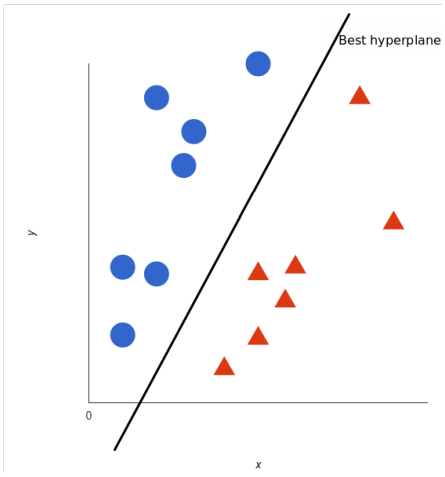


Figure 3.14: Decision Boundary for Separating Data

3.3.2 Implementation

We have implemented our project using Django framework. It is used for both front-end and back-end of our project. Necessary tools and frameworks for face recognition system are installed in python. We have used the frameworks to build the proper logics for implementing our project.

3.3.2.1 Implementation Tools

The implementation tools that are used in our project are categorized into two category, 1.Hardware, 2.Software. They are listed below:

- **Hardware Tools**

  Laptop

- **System Configuration**

  - Intel Core i3 CPU
  - Windows 10
  - 4.00GB RAM
  - 120GB SSD

- **Software Tools, Libraries and Frameworks**

  1. Windows
  2. VSCode
  3. Anaconda
  4. Google Chrome
  5. Python
  6. Django
  7. Scikit-learn
  8. OpenCV
  9. Matplotlib
  10. Jupyter Notebook
  11. Twillio
  12. Pandas
  13. Numpy

3.3.2.2 Homepage Without Authenticated User (General people)

This is the homepage for our website

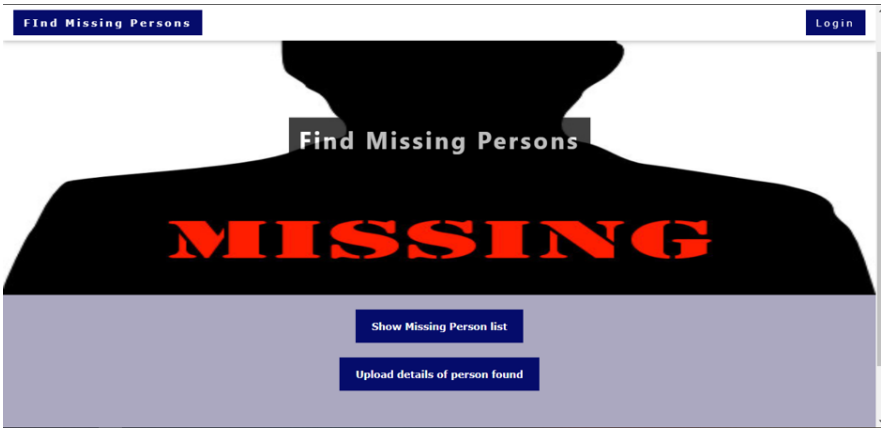


Figure 3.15: Homepage

3.3.2.3 Phone Verification Before Uploading Details of a Found Person

This form is for verifying a person via their phone.

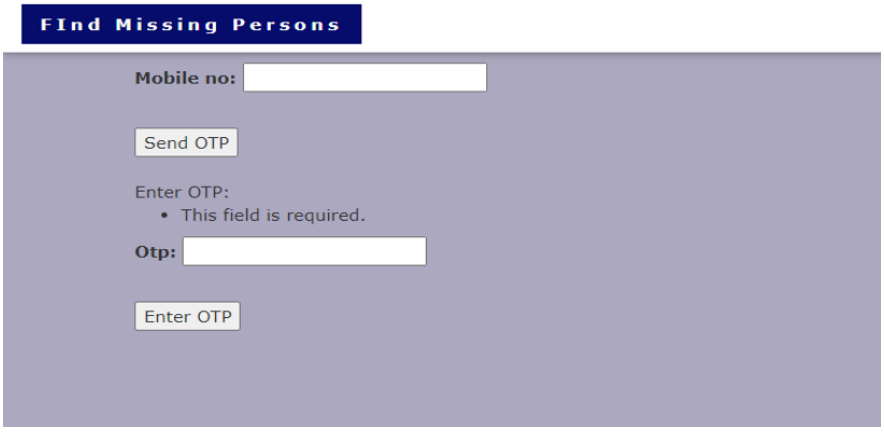


Figure 3.16: One-Time Password Verification

3.3.2.4 Uploading Details of a Found Person

This is the form for uploading details and pictures of a found person.

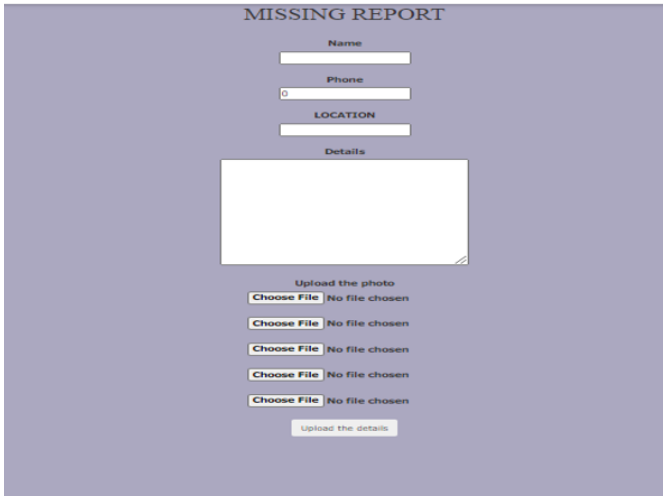


Figure 3.17: Details of Found Person

3.3.2.5 Converting to GrayScale Image and Detecting Face

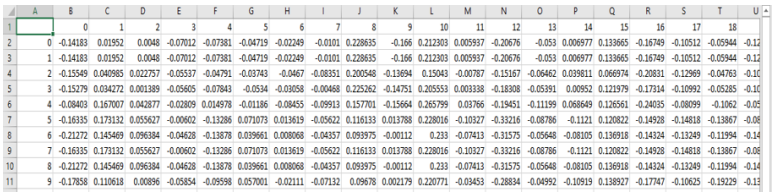
Uploaded pictures are converted to grayscale image. Then Histogram equalization is applied on the images. Faces are detected using Haar Cascade Classifier and cropped.

Figure 3.18: Input Picture of Found Person

Figure 3.19: Output Picture of Found Person

3.3.2.6 Feature Extraction

Face features are extracted from the image. 128d face data are extracted from face features. Then they are saved into a CSV file.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
2	0	-0.14183	0.01952	0.0048	-0.07012	-0.07381	-0.04719	-0.02249	-0.0101	0.228635	-0.166	0.212303	0.005937	-0.20676	-0.053	0.006977	0.133665	-0.16749	-0.10512	-0.05944	-0.12
3	1	-0.14183	0.01952	0.0048	-0.07012	-0.07381	-0.04719	-0.02249	-0.0101	0.228635	-0.166	0.212303	0.005937	-0.20676	-0.053	0.006977	0.133665	-0.16749	-0.10512	-0.05944	-0.12
4	2	-0.15549	0.040985	0.022757	-0.05537	-0.04791	-0.03743	-0.0467	-0.08351	0.200548	-0.13694	0.15043	-0.00787	-0.15167	-0.06462	0.039811	0.066974	-0.20831	-0.12969	-0.04763	-0.10
5	3	-0.15279	0.034272	0.001389	-0.05605	-0.07843	-0.0534	-0.03058	-0.00468	0.225262	-0.14751	0.205553	0.003338	-0.18308	-0.05391	0.00952	0.121979	-0.17314	-0.10992	-0.05285	-0.10
6	4	-0.08403	0.167007	0.042877	-0.02809	0.014978	-0.01186	-0.08455	-0.09913	0.157701	-0.15664	0.265799	0.03766	-0.19451	-0.11199	0.068649	0.126561	-0.24035	-0.08099	-0.1062	-0.05
7	5	-0.16335	0.173132	0.055627	-0.00602	-0.13286	0.071073	0.013619	-0.05622	0.116133	0.013788	0.228016	-0.10327	-0.33216	-0.08786	-0.1121	0.120822	-0.14928	-0.14818	-0.13867	-0.08
8	6	-0.21272	0.145469	0.096384	-0.04628	-0.13878	0.039661	0.008068	-0.04357	0.093975	-0.00112	0.233	-0.07413	-0.31575	-0.05648	-0.08105	0.136918	-0.14324	-0.13249	-0.11994	-0.14
9	7	-0.16335	0.173132	0.055627	-0.00602	-0.13286	0.071073	0.013619	-0.05622	0.116133	0.013788	0.228016	-0.10327	-0.33216	-0.08786	-0.1121	0.120822	-0.14928	-0.14818	-0.13867	-0.08
10	8	-0.21272	0.145469	0.096384	-0.04628	-0.13878	0.039661	0.008068	-0.04357	0.093975	-0.00112	0.233	-0.07413	-0.31575	-0.05648	-0.08105	0.136918	-0.14324	-0.13249	-0.11994	-0.14
11	9	-0.17858	0.110618	0.00896	-0.05854	-0.09598	0.057001	-0.02111	-0.07132	0.09678	0.002179	0.220771	-0.03453	-0.28834	-0.04992	-0.10919	0.138927	-0.17747	-0.10625	-0.19229	-0.13

Figure 3.20: CSV file for storing face features

3.3.2.7 Filing a Case for a Missing Person to Police

This form is for filling a missing person case.

Figure 3.21: Missing Person Case Form

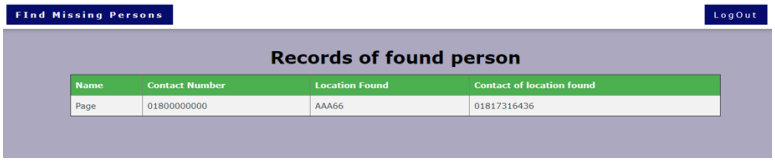


Figure 3.22: Output When a Person is Found

Figure 3.23: Found Person List

3.3.2.8 Searching Missing Person using Video

This system is for uploading video and finding missing people from the video

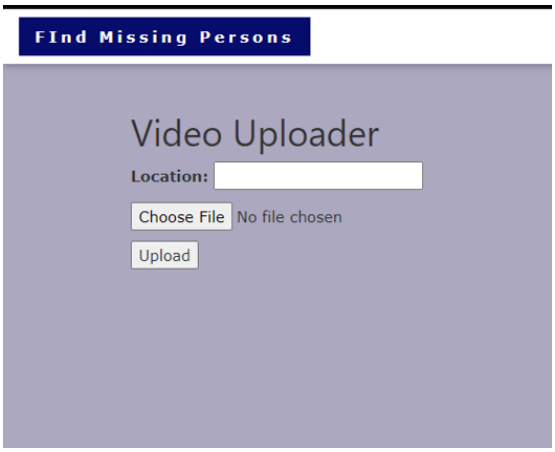


Figure 3.24: Video Upload Form

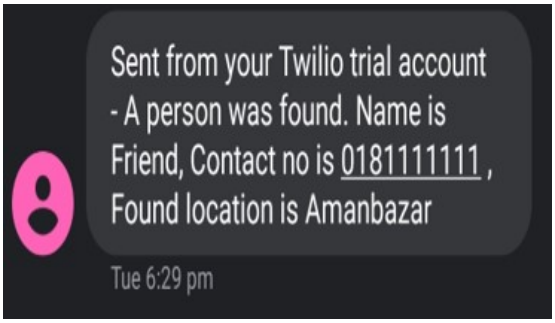


Figure 3.25: Output When a Person is Found in the Video

3.4 Conclusion

In this chapter, detailed methodology for our project is discussed. We have given overview of our methodology, tools used for our project and implementations of our project.

Chapter 4

Results and Discussions

4.1 Introduction

In this chapter, we will discuss about the results of our project. We will discuss about dataset description of our project. Then we will discuss impact of our project on society and environment. We will also discuss about ethical impacts. Then finally we will discuss evaluation of framework used in our project.

4.2 Dataset Description

In this project, we are using 360 pictures of different peoples. Pictures per person are random. For some people we used 5 pictures, for some we used only 1, for some people we used 2 to 3 pictures on average. The pictures are collected from google.

4.3 Impact Analysis

Impact analysis is the study of how a project affects the environment it is used. A project might have positive impacts as well as negative impacts on our society and environment. We will discuss the impacts of our project in this chapter. At first we will discuss social and environmental impacts, then we will discuss the ethical impacts of our project.

4.3.1 Social and Environmental Impact

Our project has positive impacts on society. In our country, many children and aged people are lost everyday. Finding the missing persons are tough job as

children and in most of the case aged people are unable to give their identities and details. Finding these peoples and bring them to their family takes too much time. With our project, it will be easier to find the missing persons as people from anywhere in the country that finds out a missing person, can upload the details immediately in our website. Police can track them easily. With the co operation of people and police, missing persons can be brought back to their family. So, our project can have very important impact on our society and environment.

4.3.2 Ethical Impact

We are using phone verification system in our project. So people can not misuse our project. This system can improve ethical impact.

4.4 Evaluation of Framework

In our face recognition system, we are saving face features extracted from images to CSV file. We are then applying Machine Learning Models to predict the faces. The models that we used in our project are: 1. K-Nearest Neighbor(KNN), 2. Support Vector Machine(SVM), 3. Naive Bayes.

Table 4.1: Accuracies of different models used in our project.

No	Model	Accuracy%
1	KNN	99
2	SVM	62
3	Naive Bayes	85

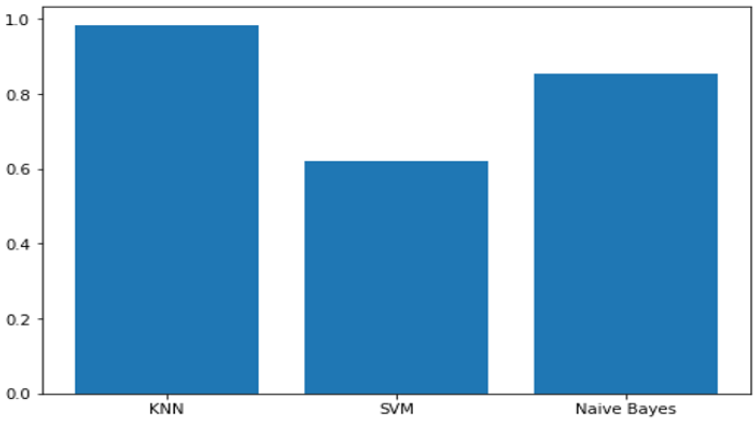


Figure 4.1: Accuracy comparison

From the results, we found that our proposed model K-Nearest Neighbor gives the best result. So we implemented our project with K-Nearest Neighbor algorithm as it gives 99% accuracy result.

4.5 Evaluation of Performance

We used confusion matrix for performance evaluation of our framework.

Confusion Matrix

The use of a confusion matrix to summarize the output of classification algorithms is a common practice. The name comes from the fact that it's simple to see if the machine is mixing up two groups. It is divided into four parts. They are as follows:

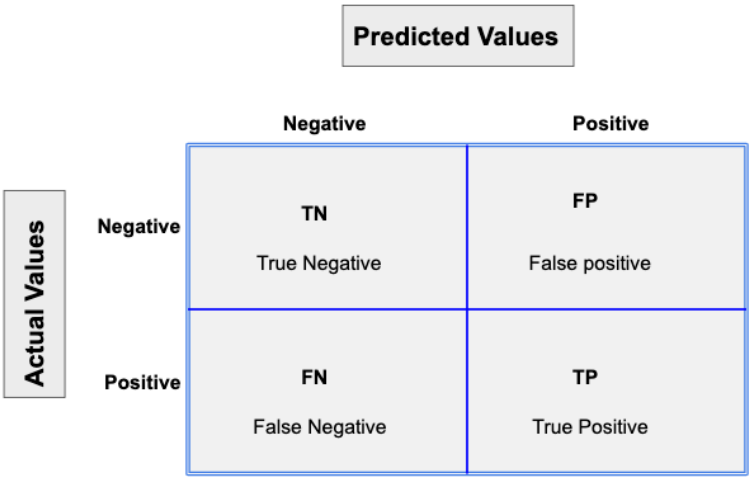


Figure 4.2: Confusion Matrix

- **True Positives**

  A A true positive is when our model correctly predicts the positive class.

- **True Negatives**

  A true negative is when our model correctly predicts the negative class.

- **False Positives**

  A false positive is when our model incorrectly predicts the positive class.

- **False Negatives**

  A false negative is when our model incorrectly predicts the negative class .

The confusion matrices for the models that we used in our project are given below:

Confusion Matrix for K-Nearest Neighbor Algorithm

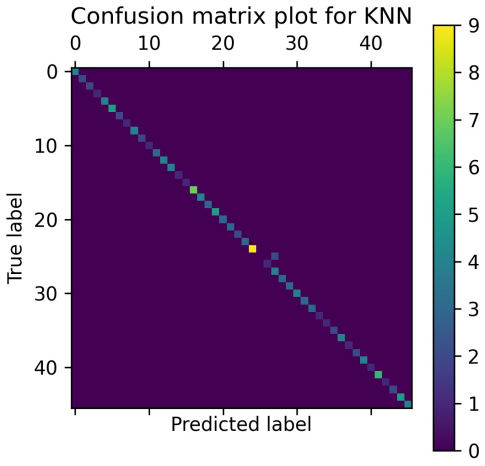


Figure 4.3: K-Nearest Neighbor Algorithm Confusion Matrix

Confusion Matrix for Support Vector Machine Algorithm

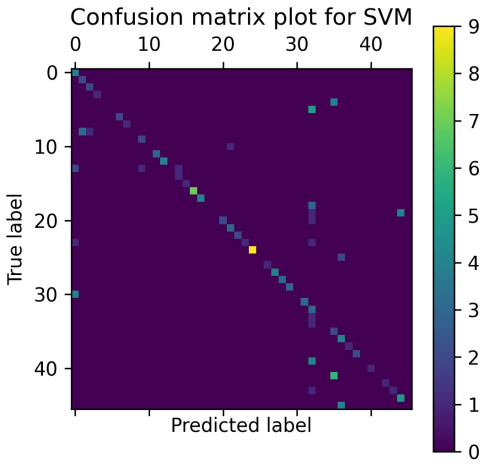


Figure 4.4: Support Vector Machine Algorithm Confusion Matrix

Confusion Matrix for Naive Bayes Algorithm

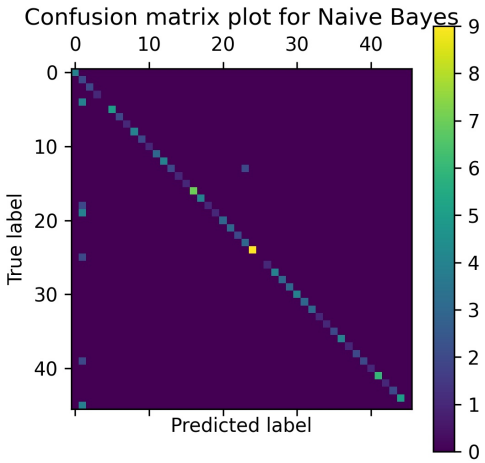


Figure 4.5: Naive Bayes Algorithm Confusion Matrix

For performance evaluation, we also used accuracy, precision, recall and f1 score. Details are given here:

Accuracy

Accuracy is defined as the ability to calculate an exact result. In other words, accuracy refers to how similar a calculated value is to a norm or true value. Accuracy is calculated using TP, TN, FP and FN values. Equation for calculating accuracy is given below:

$$Accuracy = (TP + TN)/(TP + FP + TN + FN)$$

We got 99% accuracy using KNN model, 62% accuracy using SVM and 85% accuracy using Naive Bayes model.

Precision

Precision is a quality indicator. When an algorithm has a higher precision, it returns more important results than trivial ones. Precision is calculated using TP and FP values. The equation to calculate precision is given below:

$$Precision = TP/(TP + FP)$$

Precision results for our used models are given below:

Table 4.2: Precision values for our used models

No	Model	Precision(Macro Average)%	Precision(Weighted Average)%
1	KNN	99	99
2	SVM	88	89
3	Naive Bayes	97	98

Recall

Total number of positive class predictions made from all of the dataset's positive cases is measured by recall. Recall is calculated using TP and FN values. Equation for calculating precision is given below:

$$Recall = TP/(TP + FN)$$

Recall results for our used models are given below:

Table 4.3: Recall values for our used models

No	Model	Recall(Macro Average)%	Recall(Weighted Average)%
1	KNN	98	99
2	SVM	66	62
3	Naive Bayes	88	85

F1 Score

The F1 Score is calculated by averaging Precision and Recall. The F1 score takes both values of false positives and false negatives. F1 Score is calculated using Recall and Precision values. Equation for calculating F1 Score is given below:

$$F1Score = 2 * (Recall * Precision)/(Recall + Precision)$$

F1 Scores for our used models are given below:

Table 4.4: F1 Scores for our used models

No	Model	F1 Score(Macro Average)%	F1 Score(Weighted Average)%
1	KNN	97	99
2	SVM	59	55
3	Naive Bayes	87	85

4.6 Conclusion

In this chapter, we analysed the results of our project. We discussed about the impacts of our project. We described the dataset for our project. Then we evaluated our framework and also evaluated performance. We calculated accuracy, recall, precision, f1 score for performance evaluation.

Chapter 5

Conclusion

5.1 Conclusion

Our work will be helpful for our country as many people are being lost every year. The project can help to find them quickly. It will create an opportunity to increase the use of image recognition in security and other fields.

The face recognition system of this project can be used in future works. The project can generate missing person database and later that database can be used if needed in future.

The system is more time efficient than manual process of finding people. We have used various machine learning algorithms and compared their results to choose the best model for our project. We have achieved very good accuracy result.

We also build verification system in our project. This project can help in finding missing people easily and efficiently. Our project can help people in our society.

5.2 Future Work

There are lots of scope to make various projects using Face Recognition system. Researchers in the world are trying to improve the field more and more.

In our work, we have tried to build a project using face recognition system that can help people greatly. It can be improved more.

We will do further research in the future to will the function vector so that high-dimensional data can be managed more easily. We can test the project with huge amount of data and try to improve the results in future.

Building an android or ios app can make the project more reachable to people.

This system can be implemented for real-time searching using very high quality CCTV camera and high computational system.

References

[1] P. Kamencay, M. Zachariasova, R. Hudec, R. Jarina, M. Benco and J. Hlubik, 'A novel approach to face recognition using image segmentation based on spca-knn method,' *Radioengineering*, vol. 22, no. 1, pp. 92–99, 2013 (cit. on pp. 8, 11).

[2] C. Geng and X. Jiang, 'Sift features for face recognition,' in *2009 2nd IEEE international conference on computer science and information technology*, IEEE, 2009, pp. 598–602 (cit. on p. 10).

[3] J. P. Jose, P. Poornima and K. M. Kumar, 'A novel method for color face recognition using knn classifier,' in *2012 International Conference on Computing, Communication and Applications*, IEEE, 2012, pp. 1–3 (cit. on p. 11).

[4] P. Parveen and B. Thuraisingham, 'Face recognition using multiple classifiers,' in *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, IEEE, 2006, pp. 179–186 (cit. on p. 11).

[5] M. A. Abuzneid and A. Mahmood, 'Enhanced human face recognition using lbph descriptor, multi-knn, and back-propagation neural network,' *IEEE access*, vol. 6, pp. 20 641–20 651, 2018 (cit. on p. 11).

[6] K. Sasirekha and K. Thangavel, 'Optimization of k-nearest neighbor using particle swarm optimization for face recognition,' *Neural Computing and Applications*, vol. 31, no. 11, pp. 7935–7944, 2019.

[7] M. S. Sarma, Y. Srinivas, M. Abhiram, L. Ullala, M. S. Prasanthi and J. R. Rao, 'Insider threat detection with face recognition and knn user classification,' in *2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, IEEE, 2017, pp. 39–44.

[8] M. Z. N. Al-Dabagh, M. M. Alhabib and F. Al-Mukhtar, 'Face recognition system based on kernel discriminant analysis, k-nearest neighbor and support vector machine,' *International Journal of Research and Engineering*, vol. 5, no. 3, pp. 335–338, 2018.

[9] S. Akbar, A. Ahmad, M. Hayat and F. Ali, 'Face recognition using hybrid feature space in conjunction with support vector machine,' *J. Appl. Environ. Biol. Sci*, vol. 5, no. 7, pp. 28–36, 2015.

[10] N. Stekas and D. Van Den Heuvel, 'Face recognition using local binary patterns histograms (lbph) on an fpga-based system on chip (soc),' in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, 2016, pp. 300–304.

[11] S. Suma and S. Raga, 'Real time face recognition of human faces by using lbph and viola jones algorithm,' *International Journal of Scientific Research in Computer Science and Engineering*, vol. 6, no. 5, pp. 6–10, 2018.

[12] V. B. T. Shoba and I. S. Sam, 'Face recognition using lbph descriptor and convolution neural network,' in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2018, pp. 1439–1444.

[13] A. Jagtap, V. Kangale, K. Unune and P. Gosavi, 'A study of lbph, eigenface, fisherface and haar-like features for face recognition using opencv,' in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, 2019, pp. 219–224.

[14] M. Aly, 'Face recognition using sift features,' *CNS/Bi/EE report*, vol. 186, 2006.

[15] H. Yanbin, Y. Jianqin and L. Jinping, 'Human face feature extraction and recognition base on sift,' in *2008 International Symposium on Computer Science and Computational Technology*, IEEE, vol. 1, 2008, pp. 719–722.

[16] S. Lawrence, C. L. Giles, A. C. Tsoi and A. D. Back, 'Face recognition: A convolutional neural-network approach,' *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[17] A. Sharifara, M. S. M. Rahim and Y. Anisi, 'A general review of human face detection including a study of neural networks and haar feature-based cascade classifier in face detection,' in *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, IEEE, 2014, pp. 73–78.

[18] T. Mantoro, M. A. Ayu *et al.*, 'Multi-faces recognition process using haar cascades and eigenface methods,' in *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, IEEE, 2018, pp. 1–5.