MISSING CHILD IDENTIFICATION SYSTEM

A Mini Project Report

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

In partial fulfillment of the requirements for the

award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

VEMPATI ASHRITH SHARMA - (20VE1A05P8)

Under the Guidance

of

Dr. U.M.FERNANSES DIMLO

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Affiliated to JNTUH, Approved by A.I.C.T.E and Accredited by NAAC, New Delhi)

Bandlaguda, Beside Indu Aranya, Nagole,

Hyderabad-500068, Ranga Reddy Dist.

(2020-2024)

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini Project Report on ***"MISSING CHILD IDENTIFICATION SYSTEM"*** submitted by **Vempati Ashrith Sharma** bearing Hall ticket No. **20VE1A05P8** in partial fulfillment of the requirements for theaward of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** from Jawaharlal Nehru Technological University, Kukatpally, Hyderabad for the academic year 2023-24 is a record of bonafide workcarried out by them under our guidance and Supervision.

Project Coordinator
Mrs. Srilatha Puli
Assoc. Professor

Head of the Department
Dr. U.M.Fernandes Dimlo
Professor

Internal Guide
Dr. U.M.Fernandes Dimlo
Professor

External Examiner

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I, **Vempati Ashrith Sharma** bearing **Roll No 20VE1A05P8** hereby declare that the Mini Project titled **MISSING CHILD IDENTIFICATION SYSTEM** done by us under the guidance of **Dr. U.M.Fernandes Dimlo , Professor** which is submitted in the partial fulfillment of the requirement for the award of the B. Tech degree in **Computer Science and Engineering** at **Sreyas Institute of Engineering & Technology** for Jawaharlal Nehru Technological University, Hyderabad is our original work.

Vempati Asrith Sharma - 20VE1A05P8

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without mention of the people who made it possible through their guidance and encouragement crowns all the efforts with success.

We take this opportunity to acknowledge with thanks and deep sense of gratitude to **Mrs. Srilatha Puli, Assistant Professor, Department of Computer Science** for her constant encouragement and valuable guidance during the Project work.

A Special vote of Thanks to **Dr. U.M.Fernandes Dimlo , Head of the Department** and **Dr. U.M.Fernandes Dimlo, Project Co-Ordinator** who has been a source of Continuous motivation and support. They had taken time and effort to guide andcorrect me all through the span of this work.

We owe very much to the **Department Faculty, Principal** and the **Management** who made my term at Sreyas a stepping stone for my career. We treasure every moment we had spent in the college.

Last but not the least, our heartiest gratitude to our parents and friends for their continuous encouragement and blessings. Without their support this work would not have been possible.

Vempati Asrith Sharma - 20VE1A05P8

ABSTRACT

In India a countless number of children are reported missing every year. Among the missing child cases a large percentage of children remain untraced. This paper presents a novel use of deep learning methodology for identifying the reported missing child from the photos of multitude of children available, with the help of face recognition. The public can upload photographs of suspicious child into a common portal with landmarks and remarks. The photo will be automatically compared with the registered photos of the missing child from the repository. Classification of the input child image is performed and photo with best match will be selected from the database of missing children. For this, a deep learning model is trained to correctly identify the missing child from the missing child image database provided, using the facial image uploaded by the public. The Convolutional Neural Network (CNN), a highly effective deep learning technique for image based applications is adopted here for face recognition. Face descriptors are extracted from the images using a pre-trained CNN model VGG-Face deep architecture. Compared with normal deep learning applications, our algorithm uses convolution network only as a high level feature extractor and the child recognition is done by the trained SVM classifier. Choosing the best performing CNN model for face recognition, VGG-Face and proper training of it results in a deep learning model invariant to noise, illumination, contrast, occlusion, image pose and age of the child and it outperforms earlier methods in face recognition based missing child identification. The classification performance achieved for child identification system is 99.41%. It was evaluated on 43 Child cases.

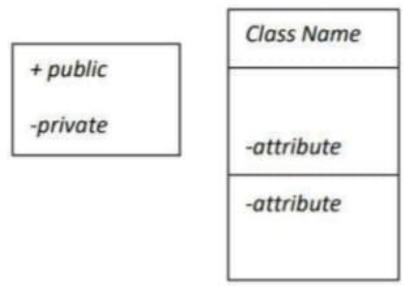
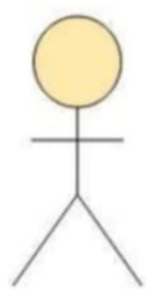
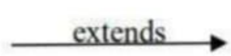
<u>Keywords</u>: convolutional neural network(CNN) ,Visual geometry group(VGG), support vector machine(SVM).

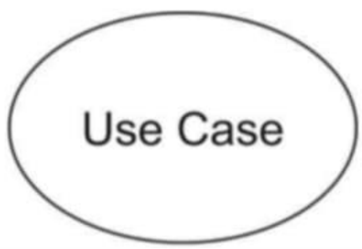
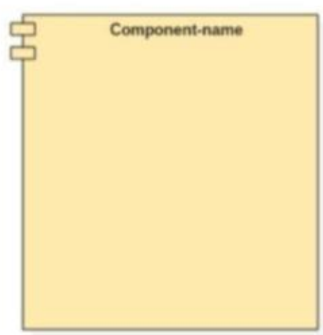
TABLE OF CONTENTS

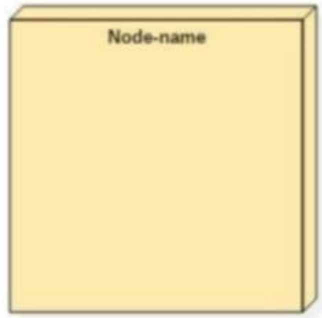
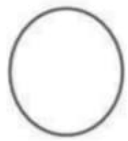
LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF SYMBOLS

SNO.	NAME	NOTATION	DESCRIPTION
1	CLASS		Represents a collection of similar entities grouped together.
2	ASSOCIATION		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3	ACTOR		It aggregates several classes into a single classes.
5	RELATION (uses)	*Use* *s*	Used for additional process communication.
6	RELATION (extents)		Extends relationship is used when one use case is similar to another use case but does a bit more.

7	COMMUNICATION		Communication between various use cases.
8	STATE		State of the process
9	INITIAL STATE		Initial state of the object
10	FINAL STATE		Final state of the object
11	CONTROL FLOW		Represents various control flow between the states.
12	DECISION BOX		Represents decision making process from a Constraint
13	USE CASE		Interact ion between the system and external environment.
14	COMPONENT		Represents physical modules which is a collection of components.

15	NODE		Represents physical modules which are a collection of components.
16	DATA PROCESS/ STATE		[illegible] ent [illegible] has [illegible] triggered due to some event [illegible] action.
17	EXTERNAL ENTITY		Represents external entities such as keyboard, sensors, etc
18	TRANSITION		Represents communication that occurs between processes.
19	OBJECT LIFELINE		Represents the vertical dimensions that the object communications.
20	MESSAGE		Represents the message exchanged.

CHAPTER 1

INTRODUCTION

Children are the greatest asset of each nation. The future of any country depends upon the right upbringing of its children. India is the second populous country in the world and children represent a significant percentage of total population. But unfortunately a large number of children go missing every year in India due to various reasons including abduction or kidnapping, run-away children, trafficked children and lost children. A deeply disturbing fact about India's missing children is that while on an average 174 children go missing every day, half of them remain untraced. Children who go missing may be exploited and abused for various purposes. As per the National Crime Records Bureau (NCRB) report which was cited by the Ministry of Home Affairs (MHA) in the Parliament (LS Q no. 3928, 20-03- 2018), more than one lakh children (1,11,569 in actual numbers) were reported to have gone missing till 2016, and 55,625 of them remained untraced till the end of the year. Many NGOs claim that estimates of missing children are much higher than reported Mostly missing child cases are reported to the police. The child missing from one region may be found in another region or another state, for various reasons. So even if a child is found, it is difficult to identify him/her from the reported missing cases. A framework and methodology for developing an assistive tool for tracing missing child is described in this paper. An idea for maintaining a virtual space is proposed, such that the recent photographs of children given by parents at the time of reporting missing cases is saved in a repository. The public is given provision to voluntarily take photographs of children in suspected situations and uploaded in that portal. Automatic searching of this photo among the missing child case images will be provided in the application. This supports the police officials to locate the child anywhere in India. When a child is found, the photograph at that time is matched against the images uploaded by the Police/guardian at the time of missing.

Sometimes the child has been missing for a long time. This age gap reflects in the images since aging affects the shape of the face and texture of the skin. The feature discriminator invariant to aging effects has to be derived. This is the challenge in missing child identification compared to the other face recognition systems. Also facial appearance of child can vary due to changes in pose, orientation, illumination, occlusions, noise in background etc. The image taken by public may not be of good quality, as some of them may be captured from a distance without the knowledge of the child. A deep learning [1] architecture considering all these constrain is designed here. The proposed system is comparatively an easy, inexpensive and reliable method compared to other biometrics like finger print and iris recognition systems.

MOTIVATION

A missing person can be characterized as the one who can be a child or an adult -- who is lost, voluntarily or involuntarily. There are various categories of missing cases of which only 43% of missing cases' reasons are known, 99% are juvenile runways, 2500 cases are due to family problems and around 500 cases are kidnapped by strangers (which include both teens and adults). Women add about 52% of missing cases and males 48%. "In India, there are no budgets allocated to finding missing people", claimed by an official source. A missing person faces many obstacles; few are subjected to death (murder), rape or abusement. People concerned with the missing person such as parents, friends, relatives and guardians are exposed to stress and worries from not knowing whether the missing person is alive or dead. In our system, the image of the person given by the guardian at the time of missing is stored in the database. The public is given authority to upload photographs of any person in uncertain situations.

1.1PROBLEM STATEMENT

Mostly missing child cases are reported to the police. The child missing from one region may be found in another region or another state, for various reasons. So even if a child is found, it is difficult to identify him/her from the reported missing cases. A framework and methodology for developing an assistive tool for tracing missing child is described in this paper. An idea for maintaining a virtual space is proposed, such that the recent photographs of children given by parents at the time of reporting missing cases is saved in a repository. The public is given provision to voluntarily take photographs of children in suspected situations and uploaded in that portal. Automatic searching of this photo among the missing child case images will be provided in the application. This supports the police officials to locate the child anywhere in India.

CHAPTER 2
LITERATURE SURVEY

Earliest methods for face recognition commonly used computer vision features such as HOG, LBP, SIFT, or SURF. However, features extracted using a CNN network for getting facial representations gives better performance in face recognition than handcrafted features.

In, missing child identification is proposed which employees principal component analysis using Eigen vectors is used for face recognition system.

Find Face is a website that lets users search for members of the social network VK by uploading a photograph. Find Face employs a facial recognition neural network algorithm developed by N-Tech Lab to match faces in the photographs uploaded by its users against faces in photographs published on VK, with a reported accuracy of 70 percent.

The app has allowed police officers to share information and work together with public.

2.1 EXISTING SYSTEM

Mostly missing child cases are reported to the police. The child missing from one region may be found in another region or another state, for various reasons. So even if a child is found, it is difficult to identify him/her from the reported missing cases. A framework and methodology for developing an assistive tool for tracing missing child is described in this paper. An idea for maintaining a virtual space is proposed, such that the recent photographs of children given by parents at the time of reporting missing cases is saved in a repository. The public is given provision to voluntarily take photographs of children in suspected situations and uploaded in that portal. Automatic searching of this photo among the missing child case images will be provided in the application. This supports the police officials to locate the child anywhere in India.

2.2 PROPOSED SYSTEM

Here we propose a methodology for missing child identification which combines facial feature extraction based on deep learning and matching based on KNN. The proposed system utilizes face recognition for missing child identification. This is to help authorities and parents in missing child investigation. It consists of a national portal for storing details of missing child along with the photo. Whenever a child missing is reported, along with the FIR, the concerned officer uploads the photo of the missing child into the portal. Public can search for any matching child in the database for the images with them. The system will prompt the most matching cases. Once the matching is found, the officer can get the details of the child. Data processing Preprocessing input raw image in the context of face recognition involves acquiring the face region and standardizing images in a format compatible with the CNN architecture employed. Each CNN has a different input size requirement. The photographs of missing child acquired by a digital camera or mobile phone are taken and categorized into separate cases for creating the database of face recognition system. The face region in each image is identified and cropped for getting the input face images.

2.2.1 PROPOSED APPROACH OF THE WORK

A missing child identification system is proposed, which combines the powerful CNN based deep learning approach for feature extraction and support vector machine classifier for classification of different child categories. This system is evaluated with the deep learning model which is trained with feature representations of children faces. By discarding the soft max of the VGG-Face model and extracting CNN image features to train a multi class SVM, it was possible to achieve superior performance. Performance of the proposed system is tested using the photographs of children with different lighting conditions, noises and also images at different ages of children. The classification achieved a higher accuracy of 99.41% which shows that the proposed methodology of face recognition could be used for reliable missing children identification.

Modules

Image Acquisition

The first step of the system is image acquisition. It will be done by High-quality images of the person are obtained through digital cameras, cell phone cameras, or scanners.

Annotated Dataset Collection

A Knowledge-based database is created by collecting all the details of users.

Image Processing

The obtained images that will be engaged in a preprocessing step are further enhanced specifically for image features during processing. The segmentation process divides the images into several segments and utilized in the extraction of the person's face from the background.

Feature-Extraction

This section involves the convolutionary layers that obtain image features from the resize images and is also joined after each convolution with the ReLU. Max and average pooling of the feature extraction decreases the size. Ultimately, both the convolutional and the pooling layers act as purifiers to generate those image characteristics.

Classification

The final step is to classify images, to train deep learning models along with the labeled images to be trained on how to recognize and classify faces according to learned visual patterns. The authors used an open-source implementation via the TensorFlow module, using Python and OpenCV including the VGG-16 CNN model.

Output screen:

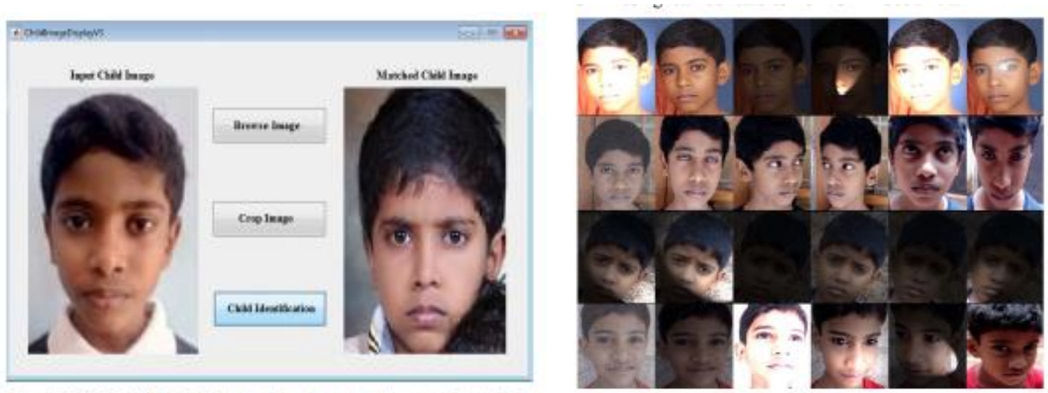


Fig. 4. GUI for child identification showing an input image and matched output image in the database

Fig. 5. Images with variations correctly classified by the system

Fig 2.2.1: output screen

ALGORITHM

Convolutional Neural Network (CNN)

The convolutional networks are currently used in visual recognition. There are number of convolutional layers in CNN. After these convolutional layers, next layers are fully connected layers as in multilayer neural network. The CNN is designed in such a way that the benefit of 2D structure of input image can be taken. This target is accomplish with the help of number of local connections and tied weights along with various pooling techniques which result in translation invariant features. The main advantages of using CNN are ease of training and possessing less parameter as compared to other networks with equal number of hidden states'. For this work, we are using Visual Group Geometry (VGG) network, which is Deep CNN for large scale image recognition. It is available in 16 layers as well as 19 layers. The classification error results for both 16 and 19 layers are almost same for validation set as well as test set, which is around 7.4% and 7.3%. This model gives the features of images which are used in further process of caption generation. B. Long Short-Term Memory (LSTM) the transitory dynamics in a set of things are modeled by using a recurrent neural network. It is very difficult for ordinary RNN to acquire long term dynamics as they get vanished and exploding weights or gradients. The memory cell is main block of LSTM. It stores the present value for long period of time. Gates are there for controlling update time of state of cell. The number of connections between memory cell and gates represent variants.

Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

- Upload Image
- View Matched Profiles

Non-functional requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

2.3 DEEP FACE RECOGNITION

The goal of this paper is face recognition – from either a single photograph or from a set of faces tracked in a video. Recent progress in this area has been due to two factors: (i) end to end learning for the task using a convolutional neural network (CNN), and (ii) the availability of very large scale training datasets. We make two contributions: first, we show how a very large scale dataset (2.6M images, over 2.6K people) can be assembled by a combination of automation and human in the loop, and discuss the trade off between data purity and time; second, we traverse through the complexities of deep network training and face recognition to present methods and procedures to achieve comparable state of the art results on the standard LFW and YTF face benchmarks.

Very Deep Convolutional Networks for Large-Scale Image Recognition

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3x3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16-19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing

ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision

Face Recognition Using Histograms Of Oriented Gradients

Face recognition has been a long standing problem in computer vision. Recently, Histograms of Oriented Gradients (HOGs) have proven to be an effective descriptor for object recognition in general and face recognition in particular. In this paper, we investigate a simple but powerful approach to make robust use of HOG features for face recognition. The three main contributions of this work are: First, in order to compensate for errors in facial feature detection due to occlusions, pose and illumination changes, we propose to extract HOG descriptors from a regular grid. Second, fusion of HOG descriptors at different scales allows to capture important structure for face recognition. Third, we identify the necessity of performing dimensionality reduction to remove noise and make the classification process less prone to overfitting. This is particularly important if HOG features are extracted from overlapping cells. Finally, experimental results on four databases illustrate the benefits of our approach.

Face Recognition Using Sift Features

Scale Invariant Feature Transform (SIFT) has shown to be a powerful technique for general object recognition/detection. In this paper, we propose two new approaches: Volume-SIFT (VSIFT) and Partial-Descriptor-SIFT (PDSIFT) for face recognition based on the original SIFT algorithm. We compare holistic approaches: Fisherface (FLDA), the null space approach (NLDA) and Eigenfeature Regularization and Extraction (ERE) with feature based approaches: SIFT and PDSIFT. Experiments on the ORL and AR databases show that the performance of PDSIFT is significantly better than the original SIFT approach. Moreover, PDSIFT can achieve comparable performance as the most successful holistic approach ERE and significantly outperforms FLDA and

Algorithm	Features
DCNN	Each voice is passed only once to the DCNN and feature maps are extracted. Selectiv to generate predictions. Combines all the three models used in DCNN together.

2.4 SOFTWARE REQUIREMENTS

- Programming Language / Platform : Python
- IDE : pycharm/jupyter

2.5 HARDWARE REQUIREMENTS

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum

DATASET

In developed countries, the government create dataset which is helpful for recognize the human face which compares the suspicious act with trained dataset and information stored in database.

The missing children images are acquired from predefined datasets that is collected from local and international cop departments. where the dataset included 82 images – 41 pairs which included images of kids, teens, adults (male and female) with age gap, different hairstyles.

ARCHITECTURE

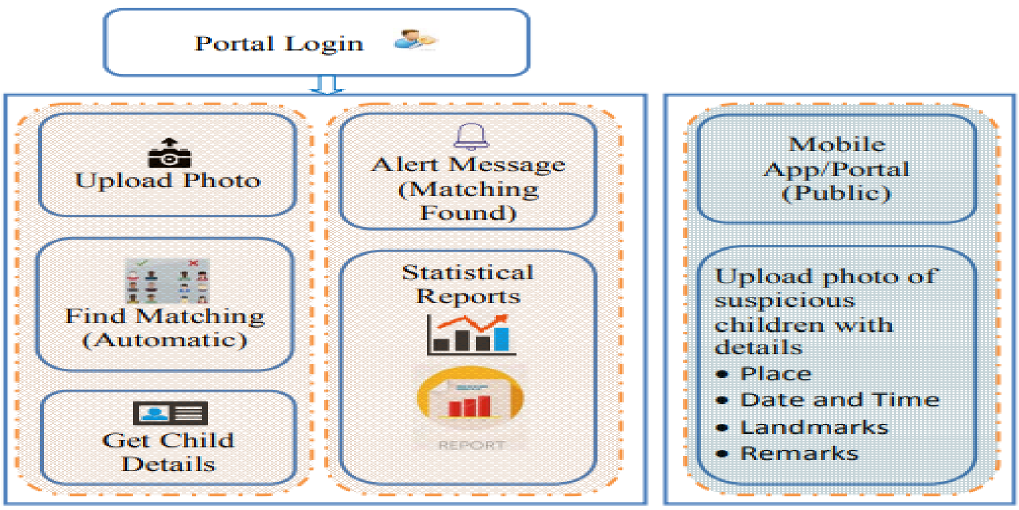


Fig2.5.1: Architecture of Proposed System

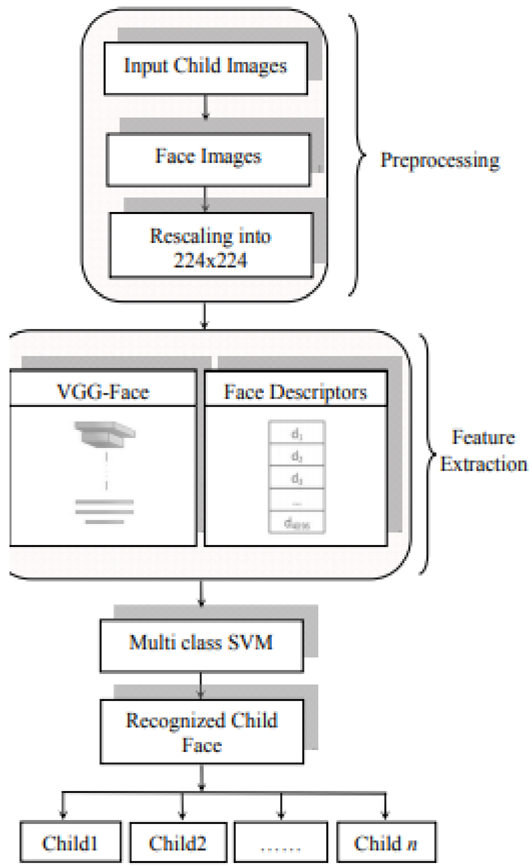


Fig2.5.2: Software Flow of face recognition system

Innovation in the project

In this work, we compare the various types of images and the accuracy level of results is very satisfying. It performs well with both images and videos. The results displayed are 90% accurate. This requires less memory space to implement and takes less time when compared with other approaches. By using this the missing children/person can be easily identifiable and it keeps on updating dynamically. The analysis process carried out with real criminal images in the web and it provides good results. We believe that, this application will decrease the crimes in our environment.

CHAPTER 3

SYSTEM DESIGN

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

Software Design Requirements

In designing the software following principles are followed:

1. **Modularity and partitioning**: software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.

2. **Coupling:** modules should have little dependence on other modules of a system.

3. **Cohesion:** modules should carry out in a single processing function.

4. **Shared use:** avoid duplication by allowing a single module be called by other that need the function it provide.

3.1 DATA FLOW DIAGRAM

1 The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

2 DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

3 DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional details. Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

Technical Architecture

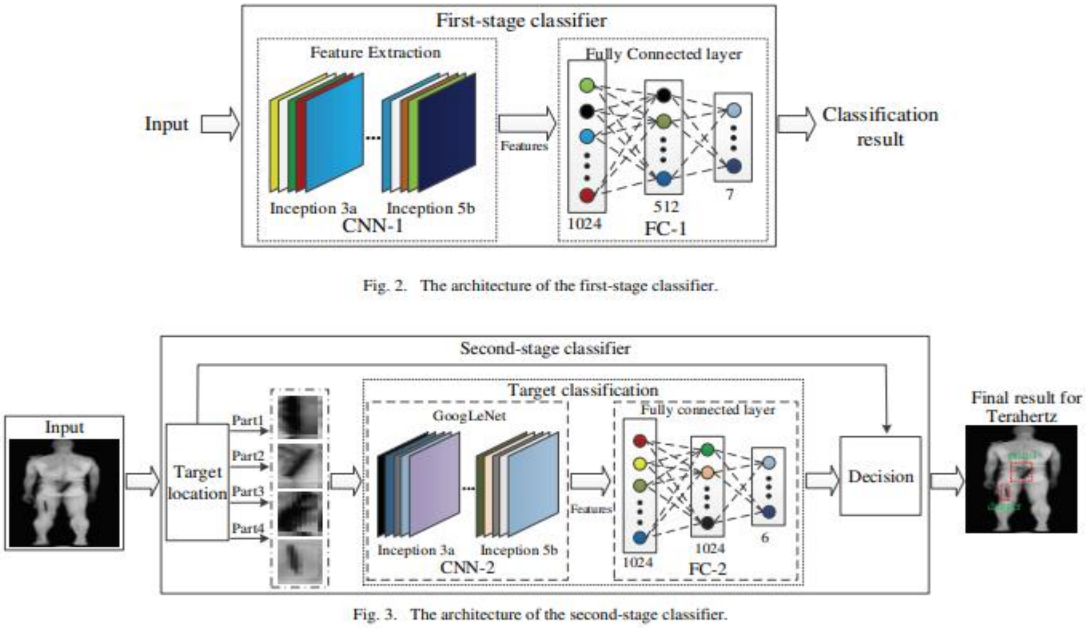


Fig3.1.1: technical Architecture

System Architecture

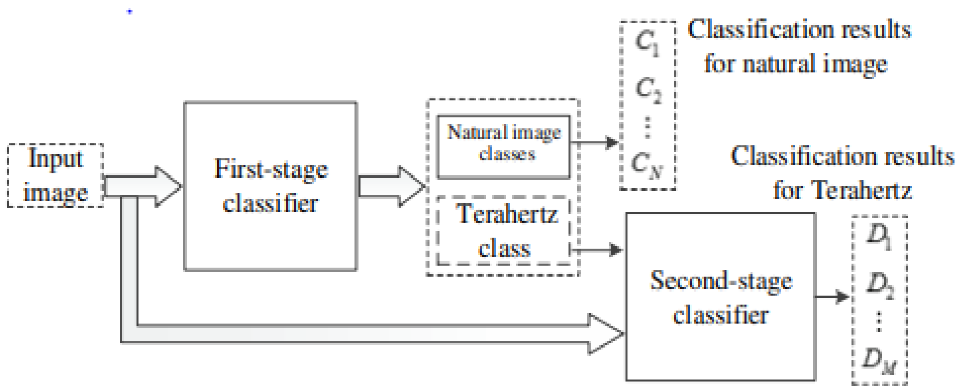


Fig3.1.2: System Architecture

Implement

The face identification algorithm is implemented using MATLAB a platform. The experiments are carried on Microsoft Windows 7, 64 bit Operating System with Intel core i7, 3.60GHz processors having 32GB RAM. For dealing with CNN architectures additional processing capability is needed. Use of GPU is recommended for training the models and Nvidia GeForce TitanX 12GB graphics card is used. The user defined database includes 846 child face images with 43 unique children cases. Training and test set is prepared by splitting the database images. 80% of images from each child category are selected for training and 20% for testing, resulting in 677 training set images and 169 test set images. The training set and validation set consists of images of each child in the earlier days and testing is done with images of children after an age gap to evaluate the system in all conditions. CNN implementation is based on MatConvNet package [9] with deep integration of CNN building blocks in MATLAB environment. Pre-trained VGG-Face CNN is also provided by MatConvNet. For the experiments here MatConvNet 1.0-beta25 version is downloaded and used. The training set images are preprocessed to the size specified by the CNN architecture before passing to the CNN model. The face region is cropped within a rectangular region from every image of the acquired input database. The images fed to VGG-Face are of fixed size by rescaling to 224x224. The activations to the input image produced by the first fully connected layer of the VGG-Face network architecture is taken as the CNN Feature descriptor. The normalized feature vector, each having a length of 4096, is used for training the SVM classifier for classifying the image of face and recognizes the child.

Implementation

The relevance of the application of the system is associated with the possible implementation of the following security threats:

1. The problem of "identity theft": Financial theft of personal data (using other people's personal data, you can purchase goods and services);

2. Identity theft with criminal intent, personal cloning, business and commercial identity theft.

3. Losses of companies as a result of theft of corporate (access to internal databases) data by employees of the companies themselves or by external intruders.

Prospects for further research are related to the expansion of the experimental base in the field of voice identification, for example, more in-depth studies on the application of the developed method and algorithm for forensic examination

3.2 SOFTWARE DEVELOPMENT LIFE CYCLE

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies use to develop

these systems.

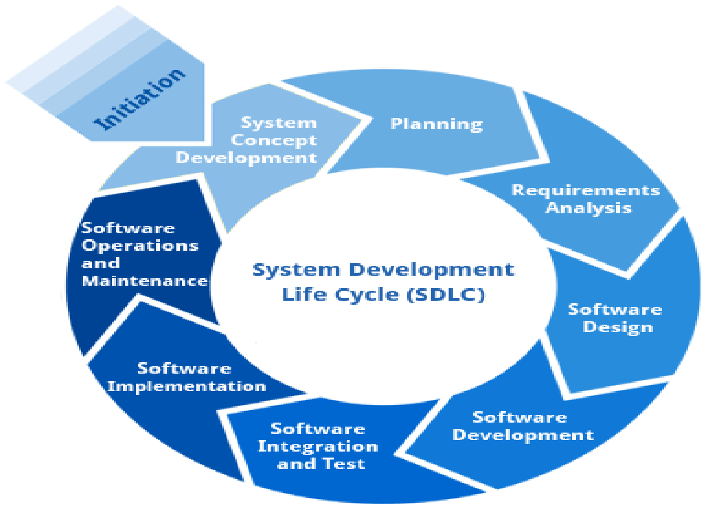


Fig3.2.1: System Development Life Cycle

Requirement Analysis and Design

Analysis gathers the requirements for the system. This stage includes a detailed study of the business needs of the organization. Options for changing the business process may be considered. Design focuses on high level design like, what programs are needed and how are they going to interact, low-level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

Implementation

In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like PYTHON 3.6, Anaconda Cloud are used for coding. With respect to the type of application, the right programming language is chosen.

Testing

In this phase the system is tested. Normally programs are written as a series of individual modules, this subject to separate and detailed test. The system is then tested as a whole. The separate modules are brought together and tested as a complete system. The system is tested to ensure that interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of data (volume testing) and that the system does what the user requires (acceptance/beta testing).

Maintenance

Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

3.2.1 SDLC METHDOLOGIES

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The following diagram shows how a spiral model acts like:

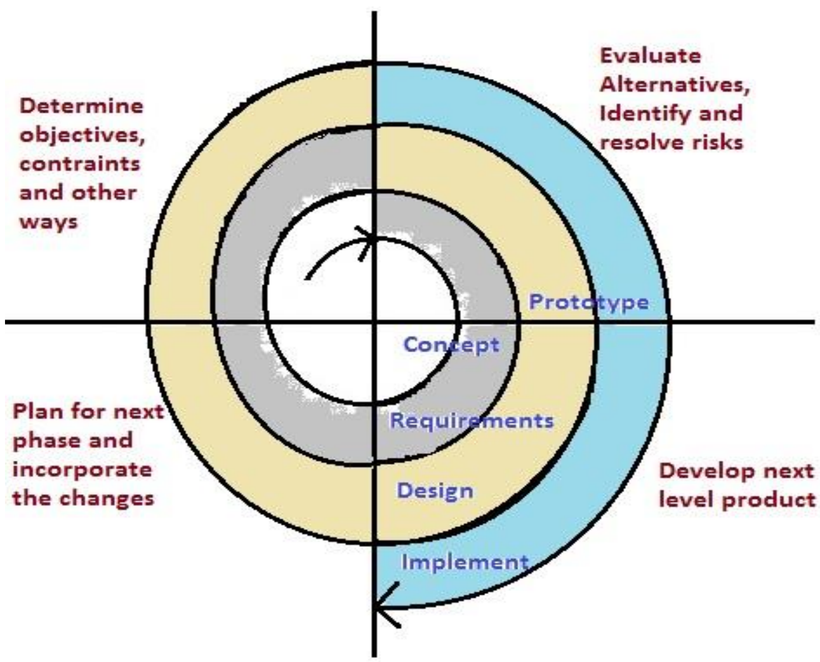


Fig3.2.1.1: Spiral Model

The steps for Spiral Model can be generalized as follows

- The new system requirements are defined in as much details as possible.

- This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
- Evaluating the first prototype in terms of its strengths, weakness, and risks.
- Defining the requirements of the second prototype.
- Planning a designing the second prototype.
- Constructing and testing the second prototype.
    - At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
    - The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
    - The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
    - The final system is constructed, based on the refined prototype.

- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

3.3 MACHINE LEARNING PROCESS

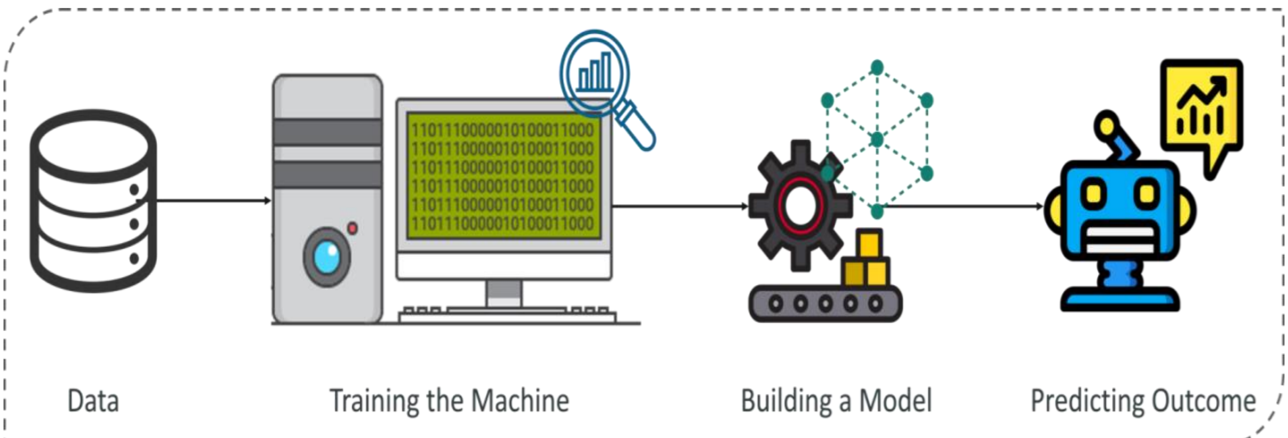
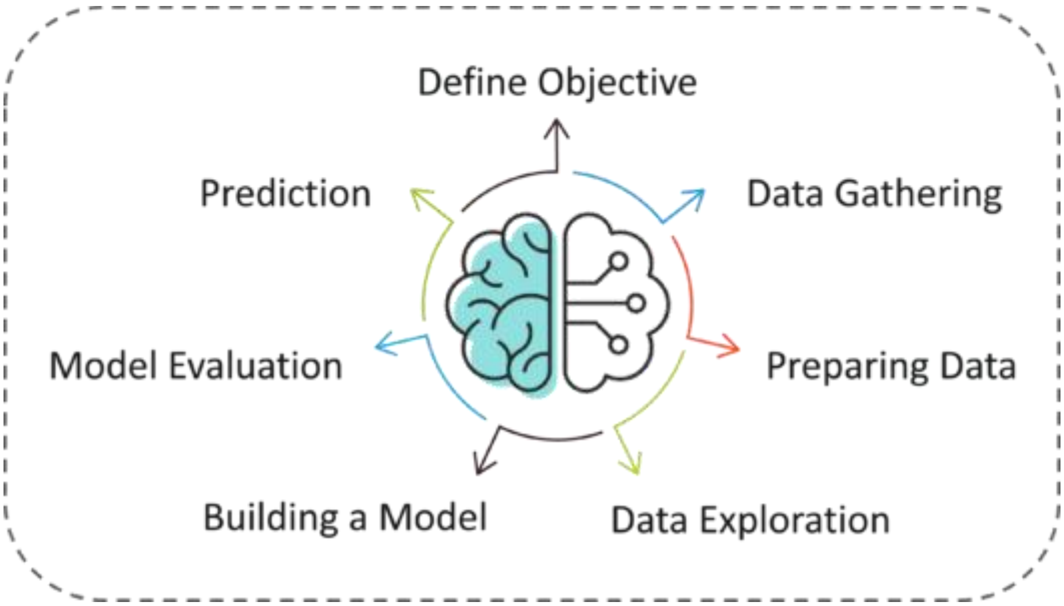


Fig3.3.1: Machine learning process

Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model.

The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set.

The Machine Learning process involves building a Predictive model that can be used to find a solution for a Problem Statement. To understand the Machine Learning process let's assume that you have been given a problem that needs to be solved by using Machine Learning.



The Steps of Machine Learning Process

Step 1: Define the objective of the Problem Statement

At this step, we must understand what exactly needs to be predicted. In our case, the objective is to predict the possibility of rain by studying weather conditions. At this stage, it is also essential to take mental notes on what kind of data can be used to solve this problem or the type of approach you must follow to get to the solution.

Step 2: Data Gathering

At this stage, you must be asking questions such as,

- What kind of data is needed to solve this problem?
- Is the data available?
- How can I get the data?

Once you know the types of data that is required, you must understand how you can derive this data. Data collection can be done manually or by web scraping. However, if you're a beginner and you're just looking to learn Machine Learning you don't have to worry about getting the data. There are 1000s of data resources on the web, you can just download the data set and get going.

Coming back to the problem at hand, the data needed for weather forecasting includes measures such as humidity level, temperature, pressure, locality, whether or not you live in a hill station, etc. Such data must be collected and stored for analysis.

Step 3: Data Preparation

The data you collected is almost never in the right format. You will encounter a lot of inconsistencies in the data set such as missing values, redundant variables, duplicate values, etc. Removing such inconsistencies is very essential because they might lead to wrongful computations and predictions. Therefore, at this stage, you scan the data set for any inconsistencies and you fix them then and there.

Step 4: Exploratory Data Analysis

Grab your detective glasses because this stage is all about diving deep into data and finding all the hidden data mysteries. EDA or Exploratory Data Analysis is the brainstorming stage of Machine Learning. Data Exploration involves understanding the patterns and trends in the data. At this stage, all the useful insights are drawn and correlations between the variables are understood.

For example, in the case of predicting rainfall, we know that there is a strong possibility of rain if the temperature has fallen low. Such correlations must be understood and mapped at this stage.

Step 5: Building a Machine Learning Model

All the insights and patterns derived during Data Exploration are used to build the Machine Learning Model. This stage always begins by splitting the data set into two parts, training data, and testing data. The training data will be used to build and analyze the model. The logic of the model is based on the Machine Learning Algorithm that is being implemented.

Choosing the right algorithm depends on the type of problem you're trying to solve, the data set and the level of complexity of the problem. In the upcoming sections, we will discuss the different types of problems that can be solved by using Machine Learning.

Step 6: Model Evaluation & Optimization

After building a model by using the training data set, it is finally time to put the model to a test. The testing data set is used to check the efficiency of the model and how accurately it can predict the outcome. Once the accuracy is calculated, any further improvements in the model can be implemented at this stage. Methods like parameter tuning and cross-validation can be used to improve the performance of the model.

Step 7: Predictions

Once the model is evaluated and improved, it is finally used to make predictions. The final output can be a Categorical variable (eg. True or False) or it can be a Continuous Quantity (eg. the predicted value of a stock).

In our case, for predicting the occurrence of rainfall, the output will be a categorical variable.

3.4 REQUIREMENT ANALYSIS

Functional Requirements:

1) Loading Dataset
2) Data cleaning
3) Data Transformation
4) Data Visualization
5) Choosing Algorithm
6) Train Dataset
7) Validate Dataset
8) Record Accuracy
9) Create Model
10) Predict Outcome

Nonfunctional requirement

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy (.e. how precise are the systems numerical answers.).

- Portability
- Reliability
- Usability
- Time Constraints
- Error messages
- Actions which cannot be undone should ask for confirmation
- Responsive design should be implemented
- Space Constraints
- Performance
- Standards
- Ethics
- Interoperability
- Security
- Privacy
- Scalability

3.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Software Environment

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

3.6 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS

The Primary goals in the design of the UML are as follows:
Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models. Provide extendibility and specialization mechanisms to extend the core concepts Be independent of particular programming languages and development process. Provide a formal basis for understanding the modeling language. Encourage the growth of OO tools market. Support higher level development concepts such as collaborations, frameworks, patterns and components. Integrate best practices.

3.6 USE CASE DIAGRAM

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

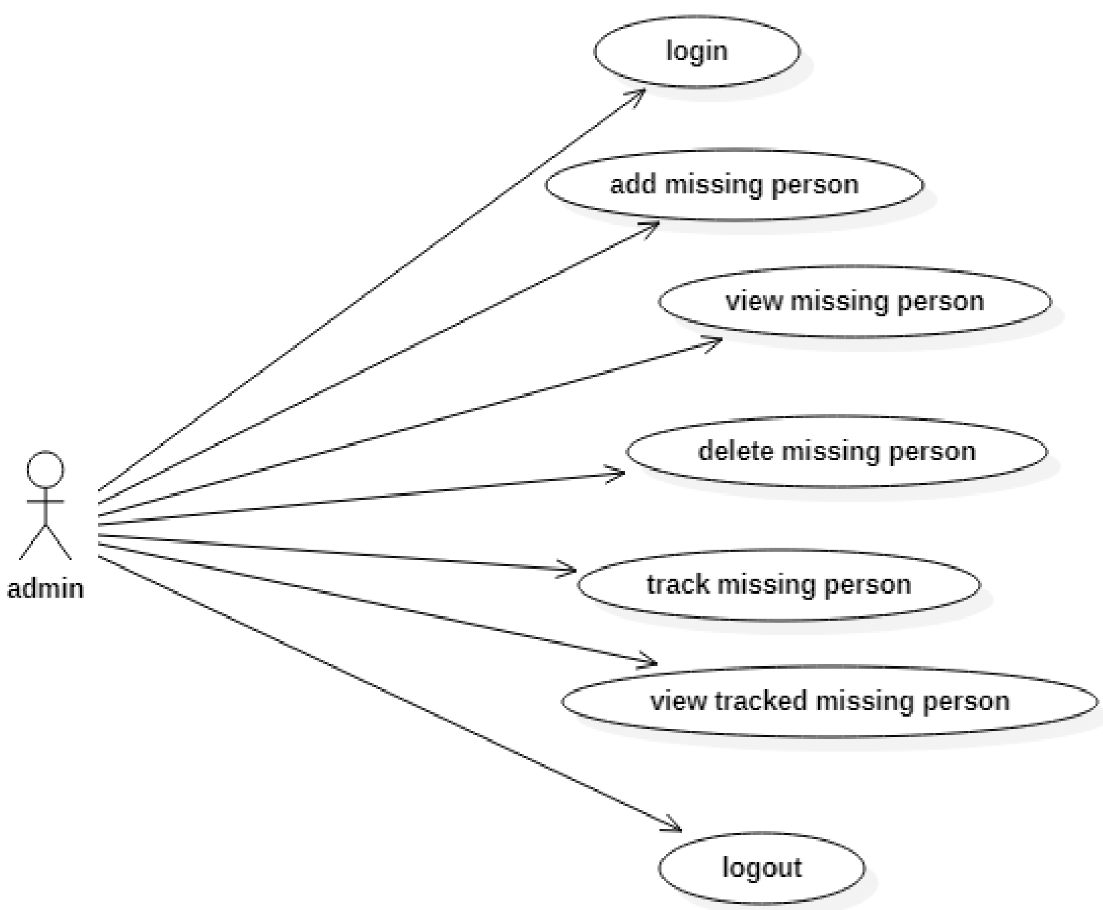


Fig3.6.1: Use case diagram

3.6.2 SEQUENCE DIAGRAM

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment.

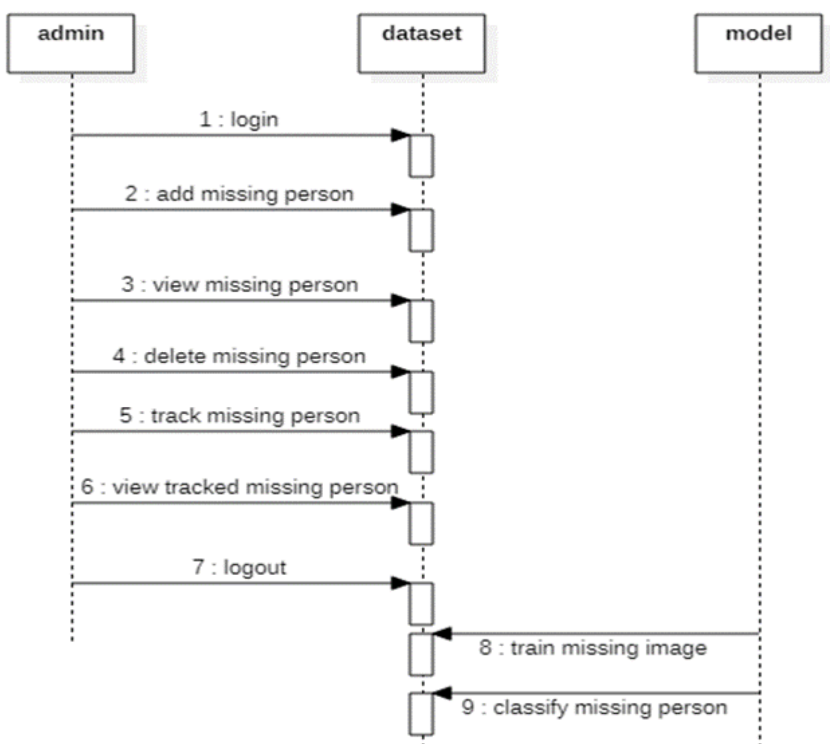


Fig3.6.2: Sequence diagram

3.6.3 ACTIVITY DIAGRAM

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

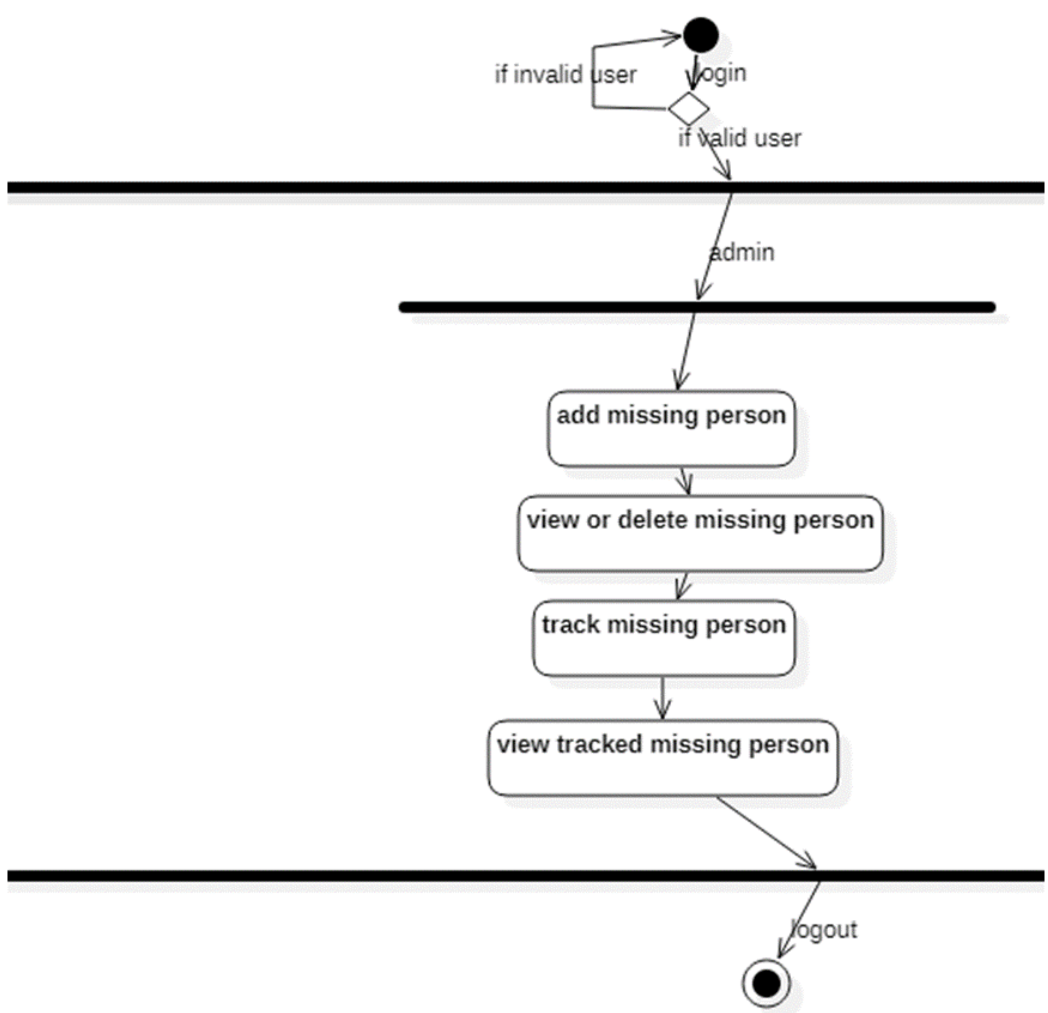


Fig3.6.3: Activity diagram

3.6.4 CLASS DIAGRAM:

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

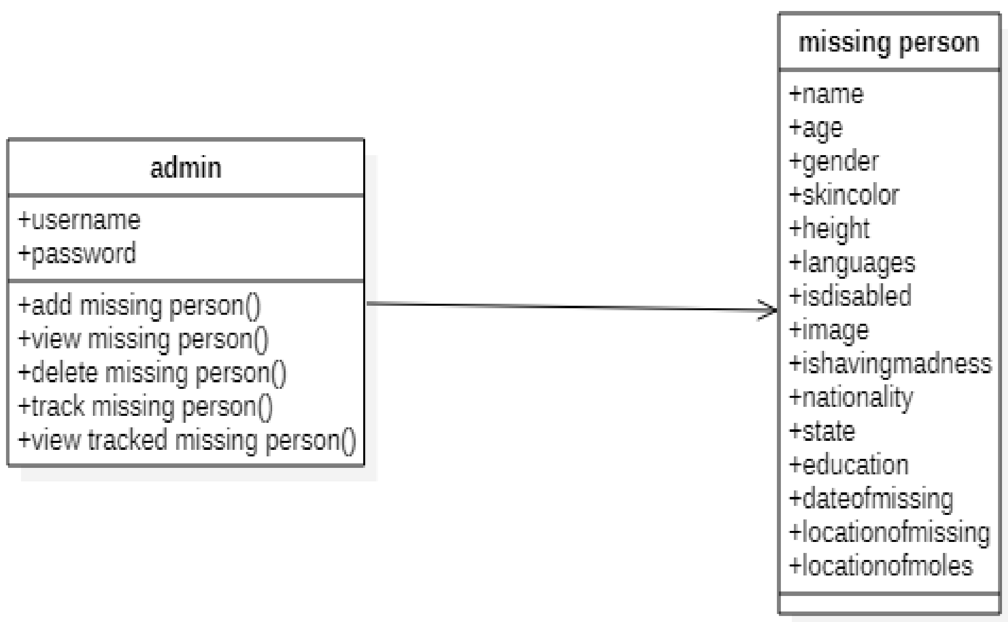


Fig3.6.4.: Class diagram

CHAPTER 4

IMPLEMENTATION

4.1 Sample Code

Missing child identification python code

```
import datetime

from missingchild.models import MissingPersonModel,RecordModel
from missingchild.forms import MissingPersons,LoginForm
from django.shortcuts import render
from django.utils import timezone

import cv2
import numpy as np
import face_recognition
import os

def addmissingchild(request):

    if request.method == "POST":

        missingchildForm = MissingPersons(request.POST,request.FILES)

        if missingchildForm.is_valid():

            missingModel = MissingPersonModel()
            missingModel.name = missingchildForm.cleaned_data["name"]
            missingModel.age = missingchildForm.cleaned_data["age"]
            missingModel.gender = missingchildForm.cleaned_data["gender"]
            missingModel.skincolor = missingchildForm.cleaned_data["skincolor"]
            missingModel.height = missingchildForm.cleaned_data["height"]
            missingModel.languages = missingchildForm.cleaned_data["languages"]
            missingModel.languages = missingchildForm.cleaned_data["languages"]
            missingModel.image = missingchildForm.cleaned_data["image"]
```

```
            missingModel.isdisabled = missingchildForm.cleaned_data["isdisabled"]
            missingModel.ishavingmadness =
missingchildForm.cleaned_data["ishavingmadness"]
            missingModel.nationality = missingchildForm.cleaned_data["nationality"]
            missingModel.state = missingchildForm.cleaned_data["state"]

            missingModel.education = missingchildForm.cleaned_data["education"]
            missingModel.dateofmissing =
missingchildForm.cleaned_data["dateofmissing"]
            missingModel.locationofmissing =
missingchildForm.cleaned_data["locationofmissing"]
            missingModel.locationofmoles =
missingchildForm.cleaned_data["locationofmoles"]

            missingModel.save()

            persons = []

            for person in MissingPersonModel.objects.all():
                person.image = str(person.image).split("/")[1]
                persons.append(person)

            return render(request, "missingpersons.html", {"persons": persons})

        else:
            return render(request, 'addmissingperson.html', {"message": "Please Fill Form
Data"})
    else:
        return render(request, 'addmissingperson.html', {"message": "Invalid Request"})

def login(request):

    if request.method == "GET":

        loginForm = LoginForm(request.GET)

        if loginForm.is_valid():

            uname = loginForm.cleaned_data["username"]
            upass = loginForm.cleaned_data["password"]
```

```
            print(uname,upass)

            if uname == "admin" and upass == "admin":
                print("in if")
                request.session['username'] = "admin"
                request.session['role'] = "admin"

                persons = []

                for person in MissingPersonModel.objects.all():
                    person.image = str(person.image).split("/")[1]
                    persons.append(person)

                return render(request, "missingpersons.html", {"persons": persons})

            else:
                return render(request, 'index.html', {"message": "Invalid username or
Password"})
        else:
            return render(request, 'index.html', {"message": "Please Enter Username and
Password"})
    else:
        return render(request, 'index.html', {"message": "Invalid Request"})

def viewmissingperson(request):

    persons = []

    for person in MissingPersonModel.objects.all():
        person.image = str(person.image).split("/")[1]
        persons.append(person)

    return render(request, "missingpersons.html", {"persons":persons})

def viewuserprofile(request):

    persons = []
    for person in MissingPersonModel.objects.filter(name=request.GET['username']):
        person.image = str(person.image).split("/")[1]
```

```
        persons.append(person)

    return render(request, "missingpersons.html", {"persons":persons})

def deletemissingperson(request):
    personid=request.GET['personid']
    MissingPersonModel.objects.get(id=personid).delete()

    persons = []

    for person in MissingPersonModel.objects.all():
        person.image = str(person.image).split("/")[1]
        persons.append(person)

    return render(request, "missingpersons.html", {"persons": persons})

def logout(request):
    try:
        del request.session['username']
    except:
        pass
    return render(request, 'index.html', {})

#============================================================
=================

def viewrecordedpersons(request):
    return render(request, "viewrecordedpersons.html",
{"recordedpersons":RecordModel.objects.all()})

def findMissingPerson(request):

    path =
"C:\\Users\\91800\\OneDrive\\Desktop\\project\\MissingChildIdentification\\images"
    images = []
    classNames = []
    myList = os.listdir(path)
    print(myList)
    for cl in myList:
        curImg = cv2.imread(f'{path}/{cl}')
```

```
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)

def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
print("images:",images, "type:",type(images))
print("Classes:",classNames,"type:",type(classNames))
encodeListKnown = findEncodings(images)
print('Encoding Complete')

cap = cv2.VideoCapture(0)

while True:
    success, img = cap.read()
    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)

    for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
        matchIndex = np.argmin(faceDis)
        print(matches,matchIndex)
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1 + 6, y2 - 6),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
```

```
            isRecorded = RecordModel.objects.filter(name=name.lower()).count()
            print("count",isRecorded)
            if isRecorded==1:
               print("in if")

RecordModel.objects.filter(name=name.lower()).update(recordeddate=timezone.now(
))
            else:

RecordModel(name=name.lower(),recordeddate=datetime.datetime.now()).save()

      cv2.imshow('Webcam', img)
      cv2.waitKey(1)
```

Missing child identification webpage code

```
<!DOCTYPE HTML>
<html>

<head>
<title>Voidmain</title>
<link rel="stylesheet" type="text/css" href="{% static 'style/style.css'%}" title="style"
/>
</head>

<body>
	<div id="main">
		<div id="header">
			<div id="logo">
				<div id="logo_text">
					<!-- class="logo_colour", allows you to change the
colour of the text -->
					<h3>
						<a href="#"><font color="white">Missing
Child Identification</font></a>
					</h3>
					<br/><br/>
				</div>
```

```
                    </div>
                    <div id="menubar">
                              <ul id="menu">
          <li><a href="/addmissingchild">Add Missing Person</a>
          <li><a href="/viewmissingperson">View Missing Persons</a></li>
          <li><a href="/findMissingPerson">Track Missing Persons</a></li>
          <li><a href="/viewrecordedpersons">View Recorded Persons</a></li>
          <li><a href="/logout">Logout</a></li>
                              </ul>
                    </div>
          </div>
          <div id="content_header"></div>
          <div id="site_content">
                    <div id="content">

      <h1>Add Missing Person{{message}} </h1>

      <form action="/addmissingchildaction/" method="post" name="appform"
enctype="multipart/form-data">

          {% csrf_token %}

          <div class="form_settings">

            <p>
              <span>Name*</span><input class="contact" type="text"
name="name" required="required"/>
            </p>
            <p>
              <span>Age*</span><input class="contact" type="number"
name="age" required="required"/>
            </p>
            <p>
              <span>Gender</span>
              <select name="gender" required="required">
                <option value="">--select</option>
                <option value="male">Male</option>
                <option value="female">Female</option>
              </select>
            </p>
```

```
            <p>
              <span>Skin Color*</span><input class="contact" type="text"
name="skincolor" required="required"/>
            </p>
            <p>
              <span>Height*</span><input class="contact" type="text"
name="height" required="required"/>
            </p>
            <p>
              <span>Languages Speak*</span><input class="contact" type="text"
name="languages" required="required"/>
            </p>
            <p>
              <span>Photo*</span><input class="contact" type="file"
name="image" required="required"/>
            </p>
            <p>
              <span>Is Disabled</span>
              <select name="isdisabled" required="required">
                <option value="">--select</option>
                <option value="no">No</option>
                <option value="yes">Yes</option>
              </select>
            </p>
            <p>
              <span>Is Havind Madness</span>
              <select name="ishavingmadness" required="required">
                <option value="">--select</option>
                <option value="no">No</option>
                <option value="yes">Yes</option>
              </select>
            </p>
            <p>
              <span>nationality*</span><input class="contact" type="text"
name="nationality" required="required"/>
            </p>
            <p>
              <span>State*</span><input class="contact" type="text"
name="state" required="required"/>
            </p>
```

```
                    <p>
                        <span>Education*</span><input class="contact" type="text"
name="education" required="required"/>
                    </p>
                    <p>
                        <span>Date of Missing*</span><input class="contact" type="date"
name="dateofmissing" required="required"/>
                    </p>
                    <p>
                        <span>Location of Missing*</span><input class="contact"
type="text" name="locationofmissing" required="required"/>
                    </p>
                    <p>
                        <span>Location of Moles*</span><input class="contact"
type="text" name="locationofmoles" required="required"/>
                    </p>
                    <p style="padding-top: 15px">
                        <span> </span><input class="submit" type="submit"
                            name="contact_submitted" value="Add MissingPerson"
                            onclick="return validate()" />
                    </p>
                </div>
            </form>
                        </div>
                </div>
        </div>
</body>
</html>
```

CHAPTER 5

TESTING

5.1 IMPORTANCE

Testing is a process, which reveals error in the program. It is the major quality measure employee during software development during software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for we planned, through testing. Testing is the process of executing a program with the intent of finding an error. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself.

5.2 TYPES OF TESTING

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at different phases of software development are:

Unit Testing

Unit testing is done on individual modules as a computer and become executable. It is confined only to the designer’s requirements. Each module can betested using the following strategies.

Black Box Testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This Testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors

In this testing the output is checked for correctness. The logical flow of the data is not checked.

White Box Testing

In test cases are generated on the logic of each module by drawing flow graphs of that module and logical decision are tested on all the cases. It has been uses to generates the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false slides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to ensure their validity.

Integrating Testing

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user system meets all requirements of the client's specifications.

Acceptance Testing

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

Testing can be done in two ways:

- Bottom up Approach
- Top down Approach

Bottom up Approach

Testing can be performed starting form smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lowerlevel ones they are tested individually and then linked with the previously examined lower level modules.

Begins construction and testing with atomic modules. As modules are integrated from the bottom up, processing requirement for modules subordinate to a given level is always available and need for stubs is eliminated. The following steps implements this strategy.

- Low-level modules are combined in to clusters that perform a specific software sub function.

- A driver is written to coordinate test case input and output.
- Cluster is tested.
- Drivers are removed and moving upward in program structure combines clusters.

Integration moves upward, the need for separate test driver’s lesions.

If the top levels of program structures are integrated top down, the number of drivers can be reduced substantially and integration of clusters is greatly simplified.

Top down Approach

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interacting occurred. No attempt is made to verify the correctness of the lower level module.

Modules are integrated by moving downwards through the control hierarchy beginning with main program. The subordinate modules are incorporated into structure in either a breadth first manner or depth first manner. This process is done in five steps:

- Main control module is used as a test driver and steps are substituted or all modules directly to main program.
- Depending on the integration approach selected subordinate is replaced at a time with actual modules.
- Tests are conducted.
- On completion of each set of tests another stub is replaced with the real module

- Regression testing may be conducted to ensure trha5t new errors have not been introduced.

This process continuous from step 2 until entire program structure is reached. In top down integration strategy decision making occurs at upper levels in the hierarchy and is encountered first. If major control problems do exists early recognitions is essential. If depth first integration is selected a complete function ofthe software may be implemented and demonstrated.

Some problems occur when processing at low levels in hierarchy is required to adequately test upper level steps to replace low-level modules at the beginning of the top down testing. So no data flows upward in the program structure.

Validation

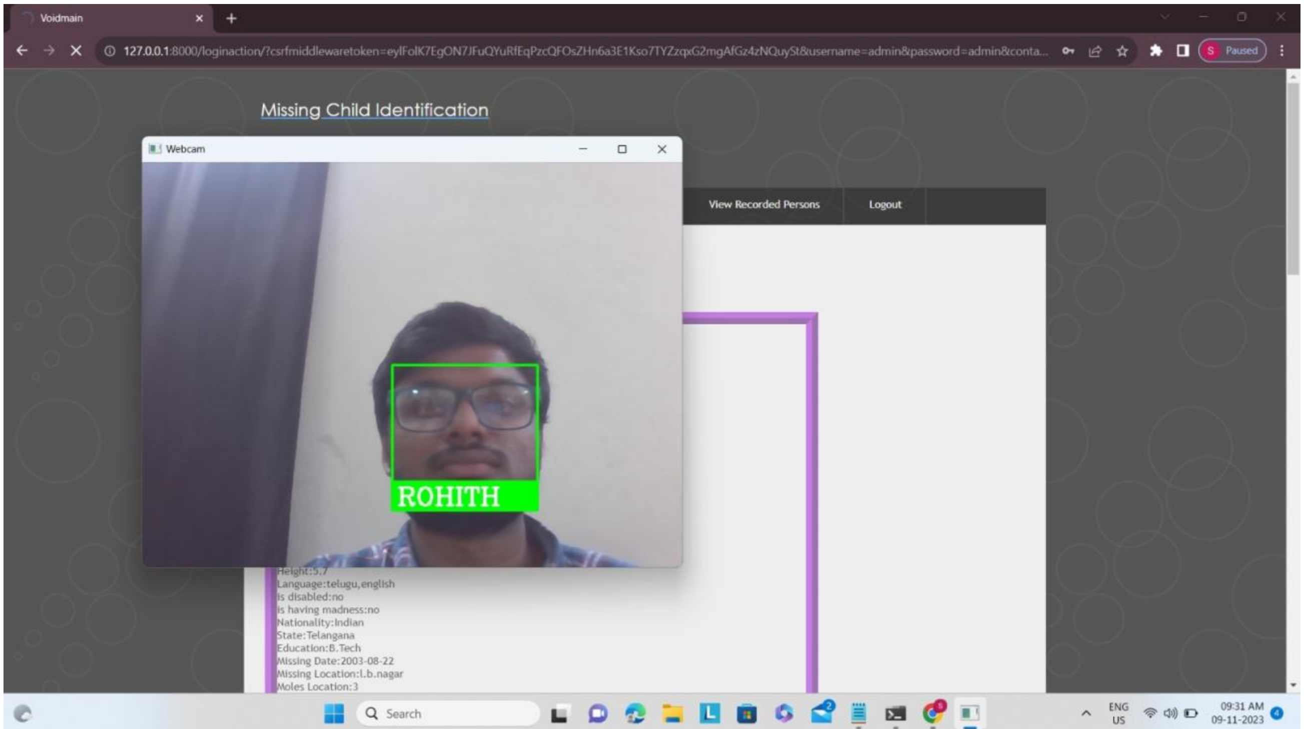
The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error message are displayed.

5.3 TEST CASES

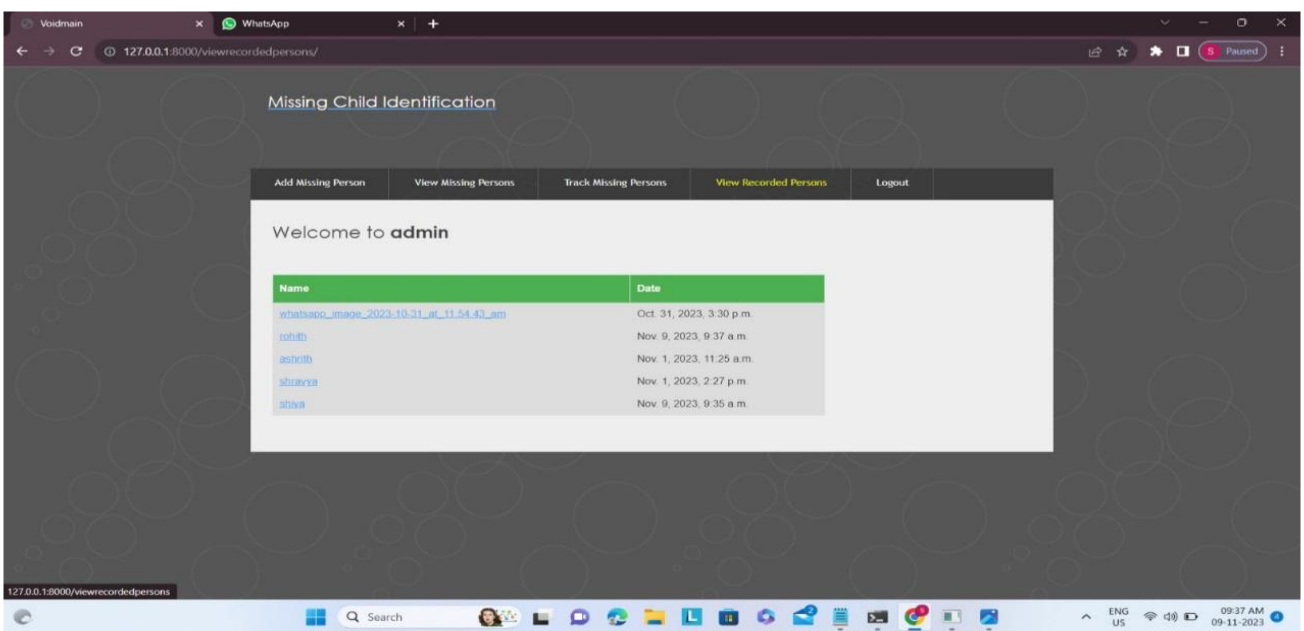
Tested	Test name	Inputs	Expected output	Actual Output	status
1	Load Dataset	Csv file	Read dataset	Load dataset	success
2	Split dataset	Train80% and test20%	Divide the training set and Testing set	Split train and Test	success
	Train Model	Train dataset, random value, predicted class	Train with best accuracy	Train with best accuracy	success
4	Validate Model	No .of Epochs	Validate the Model with best fit	Model Generated	success
5	Predict accuracy and Error Rate	Accuracy	Plot expected accuracy and predicted accuracy	Plot expected predicted accuracy	success
6	Test Data	Test column	Predicted accuracy	Predicted accuracy	success

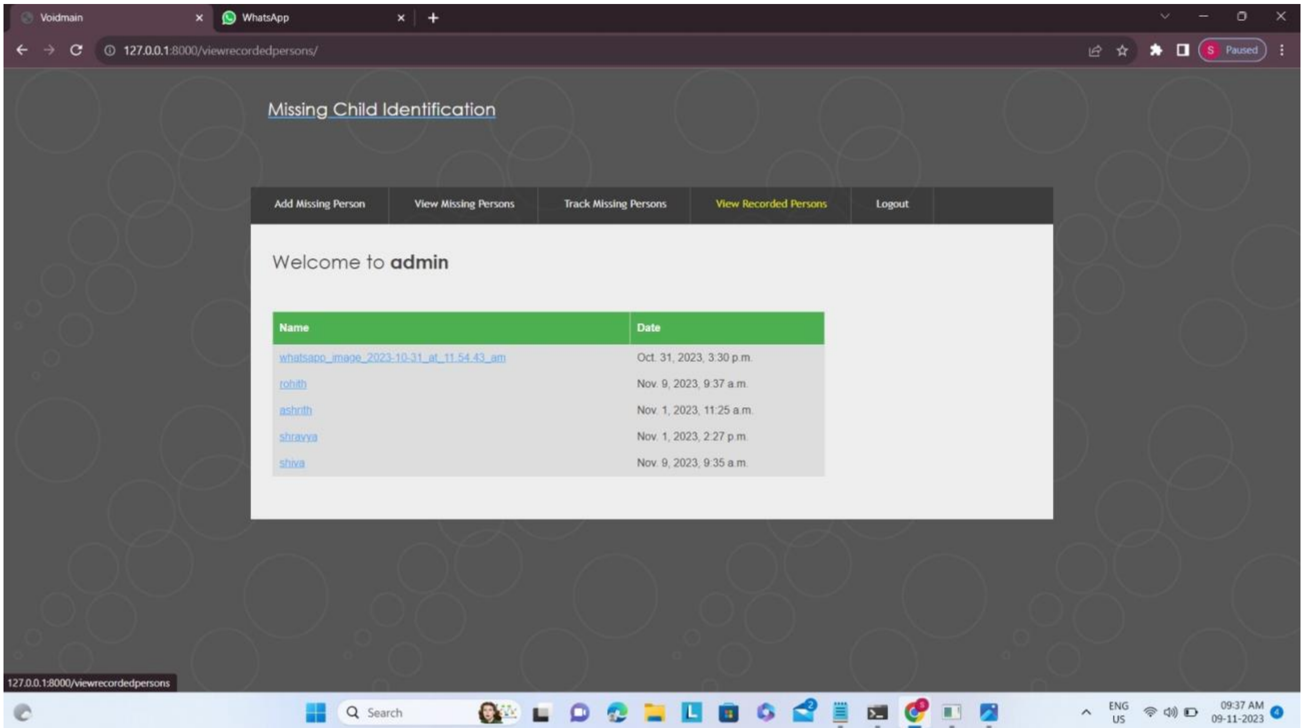
CHAPTER 6

RESULTS



Recorded Patterns




Voidmain
WhatsApp
127.0.0.1:8000/viewrecordedpersons/
Missing Child Identification
Add Missing Person
View Missing Persons
Track Missing Persons
View Recorded Persons
Logout
Welcome to admin
Name
Date
whatsapp_image_2023-10-31_at_11.54.43_am
Oct. 31, 2023, 3:30 p.m.
rohith
Nov. 9, 2023, 9:37 a.m.
ashrith
Nov. 1, 2023, 11:25 a.m.
shrayya
Nov. 1, 2023, 2:27 p.m.
shiva
Nov. 9, 2023, 9:35 a.m.
127.0.0.1:8000/viewrecordedpersons
Search
09:37 AM
09-11-2023

CHAPTER – 7
CONCLUSION AND FUTURE SCOPE

A missing child identification system is proposed, which combines the powerful CNN based deep learning approach for feature extraction and support vector machine classifier for classification of different child categories. This system is evaluated with the deep learning model which is trained with feature representations of children faces. By discarding the softmax of the VGG-Face model and extracting CNN image features to train a multi class SVM, it was possible to achieve superior performance. Performance of the proposed system is tested using the photographs of children with different lighting conditions, noises and also images at different ages of children. The classification achieved a higher accuracy of 99.41% which shows that the proposed methodology of face recognition could be used for reliable missing children identification.

FUTURE SCOPE

The future scope of a missing child identification system holds great potential as technology and society continue to evolve. Here are some areas where it can continue to develop and make a positive impact:

Advanced Facial Recognition:As facial recognition technology improves, it can become a valuable tool for identifying missing children. This could involve real-time monitoring of public spaces and comparing the images to a database of missing children.

Biometric Data Integration: In addition to facial recognition, integrating other biometric data like fingerprints, iris scans, or DNA profiles could enhance identification accuracy.

Machine Learning and AI:These technologies can be used to predict the most likely locations where missing children might be found based on historical data and patterns. Machine learning can also assist in image and voice recognition for more accurate identification.

IoT and Wearable Devices:The use of IoT devices, such as smart clothing or wearables, can help in tracking the location of children in real-time and notifying authorities or parents when a child goes missing.

Blockchain for Data Security:To ensure the security and integrity of the data in a missing child identification system, blockchain technology can be used to create tamper-proof records and maintain privacy.

Cross-border Collaboration: Missing children cases often cross borders, and international collaboration is crucial. Future systems should be designed with interoperability in mind, enabling different countries to share data and coordinate efforts effectively.

Community Engagement:Leveraging social media and community involvement is critical for spreading the word about missing children. Future systems should provide tools to engage communities and enlist their help in locating missing children.

Educational Initiatives:Implementing educational programs to teach children about safety and what to do in case they get lost or are in danger can help reduce the number of missing children.

Humanitarian and NGO Partnerships: Collaborating with organizations focused on child welfare and human rights can help missing child identification systems access additional resources and expertise.

Policy and Legal Frameworks: Developing and refining legal frameworks to protect the privacy of individuals and establish the responsibilities of various stakeholders in the identification and recovery process.

CHAPTER – 8

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", Nature, 521(7553):436–444, 2015.

[2] O. Deniz, G. Bueno, J. Salido, and F. D. la Torre, "Face recognition using histograms of oriented
gradients", Pattern Recognition Letters, 32(12):1598–1603, 2011.

[3] C. Geng and X. Jiang, "Face recognition using sift features", IEEE International Conference on
Image Processing(ICIP), 2009.

[4] RohitSatle, VishnuprasadPoojary, John Abraham, ShilpaWakode, "Missing child identification
using face recognition system", International Journal of Advanced Engineering and Innovative
Technology (IJAEIT), Volume 3 Issue 1 July - August 2016.

[5] https://en.wikipedia.org/wiki/FindFace

[6] https://www.reuters.com/article/us-china-trafficking-apps/mobileapp-helps-china-recoverhundreds-of-missing-childrenidUSKBN15J0GU

[7] Simonyan, Karen and Andrew Zisserman, "Very deep convolutional networks for large-scale
image recognition", International Conference on Learning Representations (ICLR), April 2015.

[8] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in British Machine
Vision Conference, vol. 1, no. 3, pp. 1-12, 2015.

[9] A. Vedaldi, and K. Lenc, "MatConvNet: Convolutional Neural Networks for MATLAB", ACM
International Conference on Multimedia, Brisbane, October 2015