MISSING CHILD DISTINGUSHING FRAMEWORK USING DEEP LEARNING AND SVM

An Mini Project Report Submitted
In partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering (Internet Of Things)

by

L.Vaishali	**20N31A6932**
M.Poojitha	**20N31A6934**
T.Sai Kumar Reddy	**20N31A6954**

Under the Guidance of

Mrs.P.Satyavathi
Assistant Professor
Department of Emerging Technologies
MRCET (Autonomous Institution, UGC Govt. of India)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-INTERNET OF THINGS
(EMERGING TECHNOLOGIES)

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY
(Autonomous Institution – UGC, Govt. of India)
(Affiliated to JNTU, Hyderabad, Approved by AICTE, Accredited by NBA & NAAC – 'A' Grade, ISO 9001:2015 Certified)
Maisammaguda (v), Near Dullapally, Via: Kompally, Hyderabad – 500 100, Telangana State, India.

2023-2024

DECLARATION

We here by declare that the project entitled “**MISSING CHILD DISTINGUISHING FRAMEWORK USING DEEP LEARNING AND SVM”** submitted to **Malla Reddy College of Engineering and Technology**, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) as part of IV Year B.Tech – I Semester and for the partial fulfillment of the requirement for the award of **Bachelor of Technology** in **Computer Science and Engineering (Internet of Things)** is a result of original research work done by us.

It is further declared that the project report or any part there of has not been previously submitted to any University or Institute for the award of degree or diploma.

1. L.Vaishali(20N31A6932)
2.M.Poojitha(20N31A6934)
3. T.Sai Kumar Reddy(21N35A6954)

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

(Sponsored by CMR Educational Society)

Estd : 2004

Recognized under 2(f) and 12 (B) of UGC ACT 1956

(Affiliated to JNTUH,Hyderabad, Approved by AICTE- Accredited by NBA & NAAC– 'A' Grade - ISO 9001:2015 Certified)

CERTIFICATE

This is to certify that this is the bonafide record of the project titled **" Missing Child Distinguishing Framework Using Deep Learnig and SVM "** submitted by **L.Vaishali, M.Poojitha, T.Sai Kumar Reddy,** bearing ***Roll No 20N31A6932, Roll No 20N31A6934, Roll No 20N31A6954*** of **B.Tech IV Year – I Semester** in the partial fulfillment of the requirements for the degree of **Bachelor of Technology** in **Computer Science and Engineering (Internet Of Things)**, Dept. of CSE (Emerging Technologies) during the year 2023-2024. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Mrs.P.Satyavathi
Project Guide
Department of CSE (ET)

Mrs.V.Bala
Project Coordinator
Department of CSE (ET)

Dr.P.Dileep
OVERALL COORDINATOR

EXTERNAL
EXAMINER

Dr.M.V.Kamal
HEAD
OF THE DEPARTMENT

Date of Viva-Voce Examination held on:____________________

ACKNOWLEDGEMENT

We feel ourselves honoured and privileged to place our warm salutation to our college "Malla Reddy College of Engineering and Technology (Autonomous Institution – UGC Govt. of India) and our Principal **Dr. S Srinivasa Rao,** Professor who gave us the opportunity to do the Mini Project (Project) during my IV Year B.Tech I Semester and profound the technical skills.

We express our heartiest thanks to our Director **Dr. V S K Reddy**, Professor for encouraging us in every aspect of our project and helping us realize us full potential.

We are thankful to our Head of the Department **Dr. M V Kamal**, Professor for providing training and guidance,excellent infrastructure and a nice atmosphere for completing this project successfully.

We would like to express our sincere gratitude and indebtedness to our project guide **Mrs.P.Satyavathi**, Assistant Professor for her valuable suggestions and interest throughout the course of this project.

We convey our heartfelt thanks to our Project Coordinator **Dr. P Dileep**, Professor for allowing for their regular guidance and constant encouragement during us dissertation work.

We would like to thank all our supporting **staff** of the Department of CSE (Emerging Technologies) and even all other department who have been helpful directly and in-directly in making our project a success.

Finally, we would like to take this opportunity to thank our **family** for their support and blessings for completion of our project that gave us the strength to do our project.

L.Vaishali-20N31A6932
M.Poojitha-20N31A6934
T.Sai Kumar Reddy -20N31A6954

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.

Keywords- Missing kid identification, face recognition, deep learning, CNN, VGG-Face, Multi class SVM

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
CNN	Convolutional Neural Network
VGG	Visual Geometry Group

1.Introduction

This chapter gives an overview about the purpose, aim, objectives, background and operation environment of the system.

1.1 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, 20-032018), 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 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.

1.2 Motivation:

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 childrenis 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 themremained untraced till the end of the year. Many NGOs claim that estimates of missing childrenare much higher than reported.

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 supportsthe 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 childcan vary due to changes in pose, orientation, illumination, occlusions, noise in background etc. The image taken by public may not be of good quality, assome of them may be captured from a distance without the knowledge of the child.

1.3 Literature Review:

1.3.1 FACE RECOGNITION USING HISTOGRAMS OF ORIENTED GRADIENTS **AUTHORS:**O. Deniz, G. Bueno, J. Salido, and F. D. la Torre

Abstract: 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 atdifferent 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.

1.3.2 FACE RECOGNITION USING SIFT FEATURES

AUTHORS: C. Geng and X. Jiang

Abstract: 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: Fisher face (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 NLDA.

1.3.3 MISSING CHILD IDENTIFICATION USING FACE RECOGNITION SYSTEM

AUTHORS: Rohit Satle, Vishnuprasad Poojary, John Abraham and Shilpa Wakode.

Abstract: The human face plays an important role in our social interaction, conveying people's identity. Face recognition is a task that humans perform routinely and effortlessly in their daily lives. This paper addresses the building of face recognition system by using Principal Component Analysis (PCA)method. The PCA has been extensively employed for face recognition algorithms. It not only reduces the dimensionality of the image, but also retains some of the variations in the image data. The significant features are known as "Eigen faces", because they are the eigenvectors (Principal Component) of the set of faces they do not necessarily correspond to the features such as eyes, ears, and noses. The projection operation characterize an individual face by a weighted sum of the Eigen faces features and so to recognize a particular face it is necessary only to compare these weights to those individuals.

1.4 Problem Definition:

The problem at hand is to create a Missing Child Distinguishing Framework that leverages the power of Deep Learning and Support Vector Machine (SVM) techniques. This framework aims to automate and improve the identification of missing children from images and videos, assisting law enforcement agencies and child protection organizations in their crucial task. The project encompasses data collection, preprocessing, the development of a Convolutional Neural Network (CNN) for feature extraction, and classification, as well as the integration of SVM for refining results. Challenges include limited and imbalanced data, privacy concerns, and real-time processing optimization. The potential benefits include faster and more accurate identification, improved collaboration, and increased chances of safely reuniting missing children with their families. However, it must be executed with a strong ethical framework, respecting privacy and addressing potential bias in model predictions.

Every year, thousands of children go missing globally, leading to unimaginable distress for their families and communities. Rapid and accurate identification of missing children is crucial in ensuring their safe return. Technology can play a pivotal role in addressing this problem by developing a Missing Child Distinguishing Framework.

Problem Statement:
The problem at hand is to create an advanced framework that leverages deep learning and Support Vector Machine (SVM) techniques to distinguish missing children from non-missing individuals based on visual data, primarily images. This framework aims to streamline the identification process and enhance the efficiency of law enforcement agencies and volunteers in locating missing children.

Key Objectives:
1. Binary Classification:Develop a system capable of accurately classifying images as either depicting missing children or non-missing individuals. This binary classification is the core objective of the framework.

2. Real-time Capability: Create a real-time application or platform that allows users to submit images for classification, enabling quick responses and action.

3. High Accuracy: Ensure that the framework achieves a high degree of accuracy, precision, recall, and F1-score, minimizing false positives and false negatives.

4. Data Privacy and Ethics: Implement stringent data privacy measures to protect sensitive information and adhere to ethical standards throughout the project.

5. Scalability: Design the framework to handle a growing dataset, accommodating new missing child cases as they emerge.

Project Scope:

- The project will focus on the development and implementation of a Missing Child Distinguishing Framework using deep learning and SVM techniques.
- Data sources will include images of missing children and non-missing individuals from law enforcement agencies and non-profit organizations.
- The framework will involve the entire software development life cycle, including data collection, preprocessing, model development, feature extraction, and real-time implementation.
- Collaboration with law enforcement agencies and organizations will be part of the project to provide insights and obtain feedback.

Deliverables:

1. A fully functional Missing Child Distinguishing Framework that integrates deep learning and SVM models.
2. A user-friendly interface (web application, mobile app, or API) for real-time image classification.
3. Regular updates to the dataset, model, and system to maintain its effectiveness.
4. Documentation on the architecture, design, and ethical considerations.
5. Reports on the system's performance and impact in assisting in the search for missing children.

Success Criteria:

- Achieving a high accuracy rate in distinguishing missing children from non-missing individuals.
- Receiving positive feedback and collaboration from law enforcement agencies and non-profit organizations.
- Raising awareness about missing children issues and the use of technology to address them.

Constraints:

- Legal and ethical considerations related to data usage and privacy must be strictly adhered to.
- Availability of funding, resources, and data sources.

1.5Objective of the Project:

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 supportsthe 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 childcan vary due to changes in pose, orientation, illumination, occlusions, noise in background etc. The image taken by public may not be of good quality, assome 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.

Creating a Missing Child Distinguishing Framework using Deep Learning and SVM is an important and noble endeavor. It has the potential to make a significant impact on society by helping to reunite missing children with their families. Here are some motivating factors to drive your efforts:

1. **Humanitarian Impact:** Our framework can help bring missing children back to their families and loved ones, providing hope and relief to countless families who have been living in uncertainty and despair.

2. **Social Responsibility:** Contributing to a project that focuses on the welfare of children is a powerful demonstration of social responsibility. It can bring communities together and raise awareness about the importance of child safety.

3. **Use of Cutting-Edge Technology**: By utilizing deep learning and SVM, our are leveraging the latest technological advancements to address a critical social issue. This can attract interest from the tech community and demonstrate the practical applications of AI.

4. **Global Reach:** Missing children is a worldwide issue, and our framework can be implemented globally, making a difference in the lives of children across borders.

5. **Community and Family Reunification:** There is no greater joy than reuniting a missing child with their family. Our framework can be a part of these heartwarming stories, and that emotional satisfaction is a powerful motivator.

6. **Learning Opportunity:** Developing this framework will be a tremendous learning experience. We'll have the opportunity to work with state-of-the-art technologies, gain valuable insights into deep learning and SVM, and contribute to the field of computer vision.

7. **Collaboration and Networking:** Our project can attract the attention of like-minded individuals, organizations, and researchers who are passionate about child safety. Collaborating with them can lead to a broader impact and new opportunities.

8. **Preventing Future Incidents:** By improving the ability to distinguish missing children, our contribute to the prevention of future child abductions and trafficking, which is crucial for long-term child safety.

9. **Recognition and Awards:** Successful development and implementation of our framework can lead to recognition and awards, which can further motivate you and your team.

10. **Inspiration for Others:** Our project can inspire others to use their skills and knowledge for social good, setting an example for how technology can be harnessed to address critical societal challenges.

Remember that creating such a framework requires dedication, time, and effort. However, the potential to make a profound impact on the lives of children and families, along with the opportunity to leverage advanced technology, should serve as powerful motivators to keep you inspired and committed to our mission.

2.System Analysis

In this chapter, we will discuss and analyze about the developing process of Audit Control including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs. proposed provides a view of how the proposed system will be more efficient than the existing one.

2.1 Existing System and Proposed System

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.

DISADVANTAGES OF EXISTING SYSTEM:

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.

Proposed System:

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.

Approch to build Missing Child Distinguishing Framework using Deep Learning and SVM

Creating a Missing Child Distinguishing Framework using Deep Learning and SVM is a complex project that involves several key components and steps. Here is a detailed explanation of how you can approach this project:

1. Data Collection:

The first step in building this framework is to gather a comprehensive dataset of images of missing children. These images should be obtained legally, typically from law enforcement agencies, non-profit organizations, and other reputable sources. It's crucial to ensure data privacy and ethical considerations.

2. Data Preprocessing:

Clean the dataset by removing duplicates, irrelevant images, and any sensitive information that should not be publicly shared.

Annotate the images with labels indicating whether a child is missing or not.

3. Feature Extraction:

For deep learning, you can use Convolutional Neural Networks (CNNs) to automatically extract relevant features from the images. CNNs are excellent at capturing spatial features in images.

Alternatively, you can use pre-trained CNN models (e.g., VGG, ResNet, Inception) and fine-tune them for your task, which can save time and resources.

4. Deep Learning Model:

Train a deep learning model using your preprocessed and annotated dataset. This model's objective is to learn to distinguish between images of missing children and non-missing children.

Implement a classification model that can handle binary classification (missing or not missing).

5. Hyperparameter Tuning:

Experiment with different hyperparameters, such as learning rate, batch size, and architecture, to optimize your deep learning model's performance. You can use techniques like cross-validation to ensure the model generalizes well.

6. Support Vector Machine (SVM):

After training the deep learning model, extract relevant features from the final layers (such as fully connected layers) for each image. These extracted features serve as the input to an SVM classifier.

Train an SVM classifier with these features to further improve the model's distinguishing ability. SVM is effective for binary classification tasks and can be used to provide an additional layer of decision-making.

7. Model Evaluation:

Split your dataset into training, validation, and test sets to assess the model's performance.

Use evaluation metrics like accuracy, precision, recall, F1-score, and ROC curves to measure the model's ability to distinguish missing children.

8. Model Optimization:

Continuously fine-tune and optimize your model by learning from the evaluation results. This might involve adjusting the architecture, acquiring more data, or using data augmentation techniques.

9. Real-time Implementation:

Once your model is performing well in offline scenarios, integrate it into a real-time framework or application. This could involve setting up a web interface, mobile application, or API for law enforcement agencies or volunteers to use.

10. Continuous Improvement:

Regularly update the dataset to include new missing children cases and non-missing children images. This ensures that your model remains effective as new data becomes available.

11. Collaboration:

Collaborate with law enforcement agencies, non-profit organizations, and other stakeholders who can provide insights and feedback on the framework's performance.

12. Ethical Considerations:

Ensure that your system respects privacy and confidentiality laws. Take necessary precautions to prevent misuse of the technology.

13. Awareness and Outreach:

Use the framework to raise awareness about missing children issues, share success stories, and encourage communities to actively participate in the search for missing children.

Building a Missing Child Distinguishing Framework using Deep Learning and SVM is a challenging but highly impactful project. It combines the power of deep learning to extract complex features from images and the interpretability of SVM for classification. Remember to maintain ethical standards, data privacy, and collaborate with relevant authorities throughout the process to ensure the best possible outcome.

SOFTWARE ENVIRONMENT

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 analysisthe 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 developedsystem 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.

2.2 Functional Requriments (Hardware and Software):

HARDWARE REQUIREMENTS:

•	System	:	Pentium Dual Core.
•	Hard Disk	:	120 GB.
•	Monitor	:	15’’ LED
•	Input Devices	:	Keyboard, Mouse
•	Ram	:	1 GB

SOFTWARE REQUIREMENTS:

•	Operating system	:	Windows 10
•	Coding Language	:	python
•	Tool	:	PyCharm
•	Database	:	MYSQL
•	Server	:	Flask

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Input Design:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processingcan be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES:

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

FUNCTIONAL REQUIREMENTS:

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform.

1. ADMIN

2.USER

NON –FUNCTIONAL REQUIREMENTS:

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, “how fast does the website load?” Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are> 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement

3.Software Environment

3.1 Software:

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Sources newsgroup in 1991, and version 1.0 was released in 1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date ofJanuary 1, 2020 has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python,it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

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**.**

WHY CHOOSE PYTHON

If you're going to write programs, there are literally dozens of commonly used languagesto choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

Python is Popular

Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day.

According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the Popularity of Programming Language Index Popularity of Programming Language Index.

Python is interpreted

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

Python is Simple

As programming languages go, Python is relatively uncluttered,developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionalityof the language.Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Some things that Python is often used for are:

- Web development

- Scientific programming
- Desktop GUIs
- Network programming
- Game programming

3.2 Modules used in the project

Preprocessing

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.

Upload Photo

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. The public can upload photo of any suspicious child at any time into the portal with details like place, time, landmarks and remarks. The photo uploaded by the users will be automatically compared with photos of the registered missing children and if a matching photo with sufficient score is found, then an alert email will be sent to the concerned officer. The message will also be visible in the message box of the concerned officer login screen.

Search

Whenever users uploads photo of a suspected child, the system generates template vector of the facial features from the uploaded photo. If a matching is found in the repository, the system displays the most matched photo and pushes a message to the concerned Officer portal or Email the alert message of matching child. Similarly the Officer can check for any matching with the database atany time using the proposed system.

MODULES DESCRIPTION:

1. To implements this project we have used FGNET missing child dataset and by this dataset we have build and saved CNN model is available inside model folder.
2. When user upload image used trained CNN model will be applied on test to check whether image is exists in missing child are not.
3. When user found any suspected child roaming on road then user will take image and upload here and then CNN trained model will apply to get missing result. if unidentified image also upload then application say not found.

FGNET :

The FGNet (Face and Gesture Recognition Network) dataset is a well-known facial image dataset that has been used in the field of computer vision for research and development of facial recognition and age estimation models. While FGNet is not specific to the context of a "Missing Child Distinguishing Framework," it can be a valuable resource for developing the facial recognition component of such a system. Here's more information about the FGNet dataset:

1. Dataset Composition:

The FGNet dataset primarily contains facial images of individuals at different ages, spanning from infancy to old age. It is designed for age progression and age-invariant face recognition research.

The dataset includes images of individuals at different ages, allowing researchers to examine how faces change with time. This can be valuable in a "missing child" context, as it helps in estimating how a child's face might change as they age.

2. Size and Variability:

The dataset comprises a substantial number of subjects and images, making it useful for training and testing facial recognition models.

Images are collected under different lighting conditions, with varying expressions, and different backgrounds, reflecting real-world variations.

3. Ground Truth Information:

FGNet provides ground truth information, which means it includes labels or annotations associated with the age of individuals in the images. This is crucial for training age estimation models.

4. Research Applications:

Researchers and practitioners use the FGNet dataset to develop and evaluate facial recognition models, especially age estimation models. It can be beneficial for age progression, face verification, and facial feature extraction tasks.

5. Utilization in Missing Child Distinguishing Framework:

While FGNet is not designed specifically for missing child identification, it can be used in combination with other datasets and techniques to build a robust missing child distinguishing framework.

FGNet's value lies in its ability to assist in age progression and estimation, which can be crucial in identifying missing children who may have aged since their disappearance.

In the context of a Missing Child Distinguishing Framework using Deep Learning and SVM, you could potentially incorporate the FGNet dataset for the following purposes:

Age Progression: Use the FGNet dataset to develop models that can estimate how a child's face might change as they age. This can be useful when identifying missing children who have been missing for an extended period.

Age Estimation: Leverage the age annotations in the FGNet dataset to build age estimation models, which can help determine the approximate age of a person in a given image. This information can be valuable in narrowing down potential matches.

Model Training: Utilize FGNet data, alongside missing child images, to train your deep learning model for facial recognition. This can enhance the model's ability to recognize faces and assist in distinguishing missing children.

While FGNet is a valuable resource, it's essential to remember that missing child identification is a sensitive and ethically complex task. Ensure that you have the necessary permissions and legal agreements in place when using any facial image dataset and adhere to all privacy and data protection regulations. Additionally, consider combining FGNet with other missing child datasets for a comprehensive and accurate framework.

CNN(Convolutional Neural Network):

In a Missing Child Distinguishing Framework that utilizes Deep Learning and SVM, a Convolutional Neural Network (CNN) plays a crucial role in the initial stages of the project for feature extraction from images. Here's an explanation of how CNNs are used:

1. Image Feature Extraction:

CNNs are a class of deep neural networks particularly suited for processing and analyzing visual data, such as images. They are designed to automatically extract relevant features from images, making them a key component of your framework.

2. Convolutional Layers:

CNNs consist of multiple layers, including convolutional layers. These layers apply a set of filters (kernels) to the input image. Each filter extracts specific features like edges, corners, and textures. Through multiple convolutional layers, the network gradually learns to recognize more complex patterns.

3. Pooling Layers:

After convolution, pooling layers reduce the spatial dimensions of the feature maps while retaining the most important information. Max-pooling is a common pooling technique that selects the maximum value in a local region.

4. Fully Connected Layers:

The final layers of the CNN are typically fully connected layers. These layers take the high-level features extracted from the previous layers and combine them to make a prediction. In your case, the prediction would be whether the image contains a missing child or not.

5. Training the CNN:

To train the CNN, you would provide it with a labeled dataset of images, where each image is tagged as either "missing child" or "not missing child." The CNN learns to adjust its internal parameters (weights and biases) during training to make accurate predictions.

6. Backpropagation:

During training, the network uses a process called backpropagation to update its parameters based on the errors it makes when classifying images. This process continues iteratively until the network's performance improves.

7. Transfer Learning:

If you have limited data for missing child images, you can leverage transfer learning. This involves using a pre-trained CNN model on a large dataset (e.g., ImageNet) and fine-tuning it for your specific task. This approach often leads to faster convergence and better results.

8. Feature Extraction for SVM:

Once the CNN is trained, you can extract features from the final layers (usually fully connected layers) of the network for each image in your dataset. These extracted features serve as input to the SVM classifier.

9. Combining CNN and SVM:

The CNN extracts high-level features from the images, which are then used by the SVM for classification. The SVM, known for its effective binary classification capabilities, can further improve the distinguishing ability of your framework.

By using a CNN in your Missing Child Distinguishing Framework, you're harnessing the power of deep learning to automatically and adaptively learn features from images, making it more effective in recognizing the unique patterns and characteristics of missing children. The combination of a CNN for feature extraction and an SVM for classification is a powerful approach to address this critical problem.

SVM(Support Vector Manchine):

Support Vector Machine (SVM) is a crucial component in the Missing Child Distinguishing Framework that complements deep learning models. SVM is a supervised machine learning algorithm used for classification tasks, and it can enhance the accuracy and robustness of the system for distinguishing missing children. Here's an explanation of how SVM works in this context:

1. Feature Extraction:

- In the context of distinguishing missing children using deep learning and SVM, deep learning models, such as Convolutional Neural Networks (CNNs), are initially employed to extract relevant features from images. These features serve as input data for both the deep learning model and the SVM.

2. Dimensionality Reduction:

- The features extracted by the deep learning model may be high-dimensional and may contain a lot of information, some of which might not be relevant for the classification task.
- SVM can be particularly useful in reducing the dimensionality of the feature space by identifying a subset of the most informative features that contribute to the classification task. This simplification can improve the SVM's efficiency and classification performance.

3. Binary Classification:

- SVM is well-suited for binary classification problems, such as distinguishing between missing children (positive class) and non-missing children (negative class). It aims to find a decision boundary that best separates the two classes while maximizing the margin between them.

4. Margin Maximization:

- The SVM's primary objective is to find a hyperplane that maximizes the margin between the two classes. The margin is defined as the distance between the hyperplane and the nearest data points from each class.
- SVM seeks the hyperplane that not only separates the two classes but also provides the most robust classification by maximizing this margin. This margin maximization makes SVM resilient to noise and helps reduce overfitting.

5. Kernel Functions:

- SVM can handle non-linear separable data by using kernel functions. In the context of image data, a commonly used kernel is the Radial Basis Function (RBF) kernel. This kernel can transform the feature space to a higher-dimensional space where the data might become linearly separable.
- Using an appropriate kernel, SVM can learn complex decision boundaries, which can be beneficial when distinguishing missing children based on their image features.

6. Support Vectors:

- Support vectors are data points that are closest to the decision boundary. These are the most challenging instances for the classifier because they are near the margin. SVM learns from these support vectors and is less influenced by the other data points, making it robust to outliers.

7. Model Training and Testing:

- The SVM model is trained on the features extracted from images of missing and non-missing children. The training data is labeled, and the SVM learns to find the optimal decision boundary that separates the two classes.
- The trained SVM model can then be used to predict whether a given image of a child falls into the missing or non-missing category.

8. Model Evaluation:

- The performance of the SVM in the Missing Child Distinguishing Framework is assessed using evaluation metrics such as accuracy, precision, recall, F1-score, and ROC curves. This helps measure the effectiveness of the SVM in classifying images of children.

9. Ensemble with Deep Learning:

- The SVM can be used in conjunction with the deep learning model to create an ensemble. While the deep learning model excels at feature extraction, the SVM provides robust classification, and their combination can result in improved overall performance.

By integrating SVM into the Missing Child Distinguishing Framework, you can enhance the system's classification accuracy, especially when dealing with challenging and potentially noisy image data. SVM's ability to handle non-linear separable data and maximize the margin between classes makes it a valuable addition to the deep learning-based feature extraction process

4.System Design and UML Diagrams

System design is transition from a user-oriented document to programmers or data base personel.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:

In designing the software following principles are followed:

1.Modularity and partitioning:

Software is designed such that, each system should consist 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 provides.

4.1 Dataflow 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.
2. 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.
3. 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.
4. 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 detail.

There are four components in DFD:

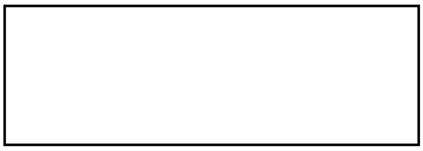
1.External Entity

2.Process

3.Data Flow

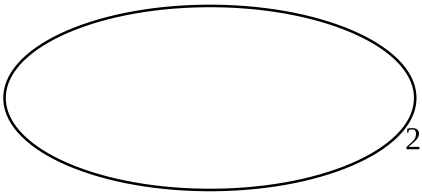
1)External Entity: It is an outside system that sends or receives data, communicating with the system. They are the sources and destinations of information entering and leaving the system. They might be an outside organization or person, a computer system or a business system. They are known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram. These are sources and destinations of the system's input and output.

Representation:



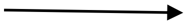
Process: It is just like a function that changes the data, producing an output. It might perform computations for sort data based on logic or direct the dataflow based on business rules.

Representation:



Data Flow: A dataflow represents a package of information flowing between two objects in the data-flow diagram, Data flows are used to model the flow of information into the system, out of the system and between the elements within the system.

Representation:



Data Store: These are the files or repositories that hold information for later use, such as adatabase table or a membership form. Each data store receives a simple label.

Representation:

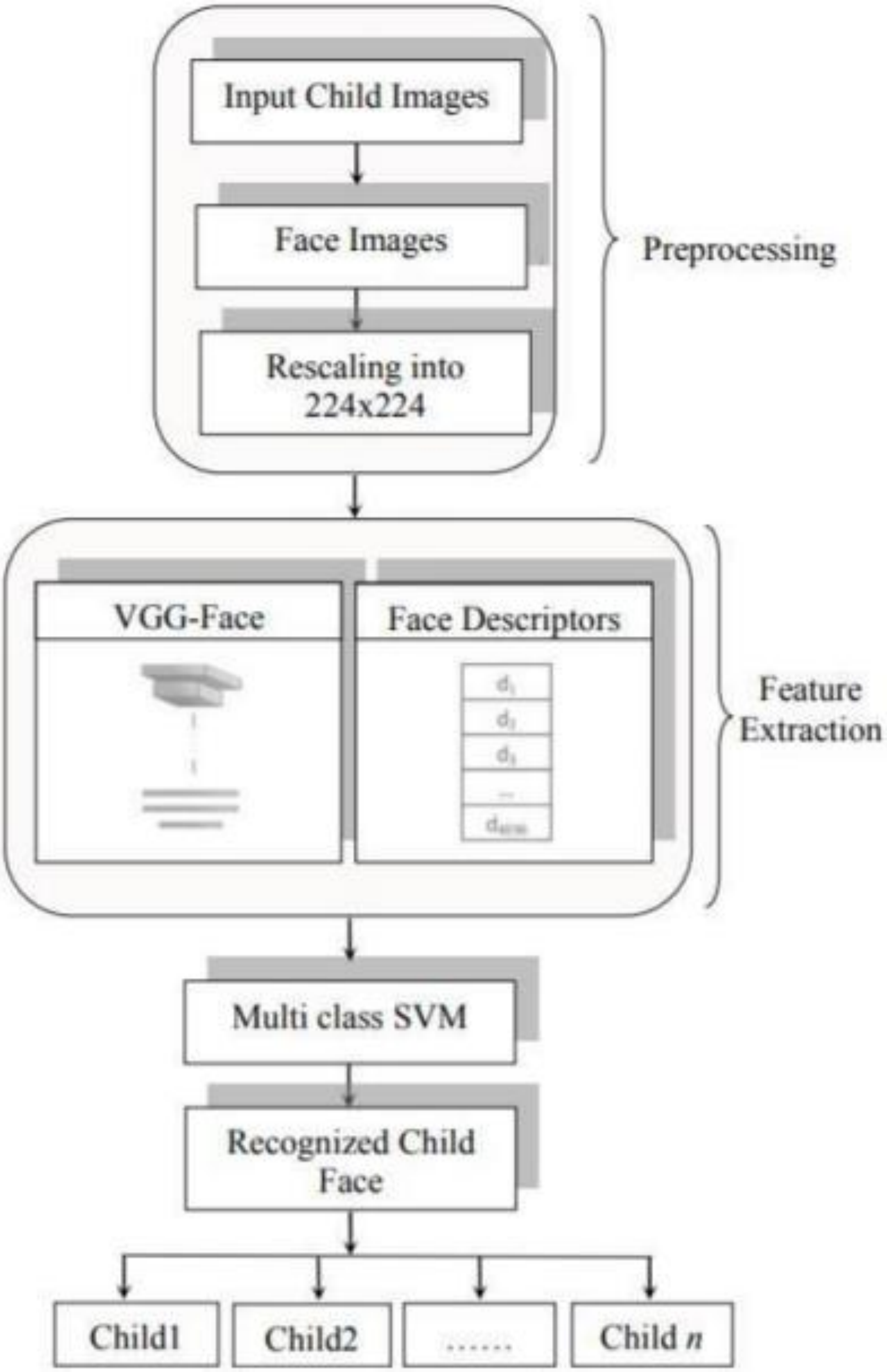


Fig 4.1 Data flow Diagram

4.2 ARCHITECTURE:

Here we propose a methodology for missing child identification which combines facial feature extraction based on deep learning and matching based on support vector machine. The proposed system utilizes face recognition for missing child identification. This is to help authorities and parents in missing child investigation. The architecture of the proposed frame work is given below.

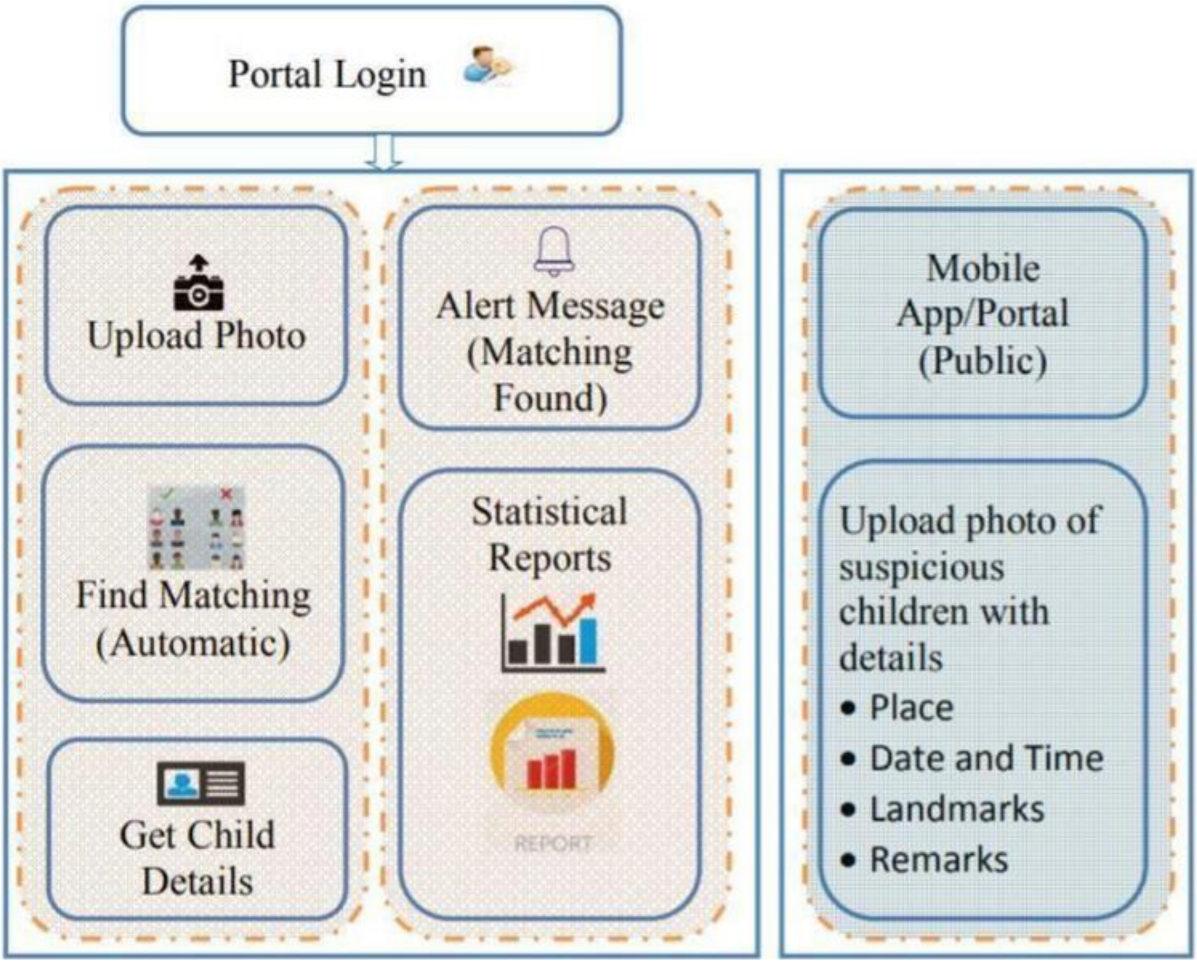

Portal Login
Upload Photo
Find Matching (Automatic)
Get Child Details
Alert Message (Matching Found)
Statistical Reports
REPORT
Mobile App/Portal (Public)
Upload photo of suspicious children with details
• Place
• Date and Time
• Landmarks
• Remarks

FIGURE 4.2 Architecture of proposed child identification system

Images of reported missing children are saved in a repository and the face area is selected for cropping to obtain input face images. Learned features from a Convolutional Neural Network (CNN), a specific type of deep learning algorithm, are used for training a multi class SVM classifier. This machine learning approach is used to correctly label the child using the name indicated in the database provided by the concerned authority.In the following sections the paper details the work flow for child matching methodology. The flow chart of the automatic child face identification methodology is as shown in Fig 4.2.

4.1UNIFIED MODELING LANGUAGE (UML) Daigrams:

UML stands for Unified Modelling Language. Taking SRS document of analysis as input to the design phase drawn UML diagrams. The UML is only language so is just one part of the software development method. The UML is process independent, although optimally it should be used in a process that should be driven, architecture-centric, iterative, and incremental. The UML is language for visualizing, specifying, constructing, documenting the articles in a software- intensive system. It is based on diagrammatic representations of software components. A modelling language is a language whose vocabulary and rules focus on the conceptual and physical representation of the system. A modelling language such as the UML is thus a standard language for software blueprints.

Unified Modeling Language (UML) diagrams play a crucial role in software development, offering a standardized and visual representation of different facets of a software system. Among the various types of UML diagrams, the Class Diagram stands out as a foundational tool for showcasing the static structure of a system. This diagram encapsulates classes, their attributes, methods, and the relationships that exist between them. Classes serve as blueprints for objects, with attributes representing their characteristics and methods indicating their functionalities. Associations delineate the connections between classes, offering a comprehensive view of the system's architecture.

In the realm of physical deployment, Component Diagrams serve as a means to represent the physical components of a system and their dependencies. Components, both physical and logical, are depicted along with dependency arrows that illustrate the relationships between them. Meanwhile, Deployment Diagrams focus on the physical deployment of software components on hardware nodes, providing a visual guide to the system's hardware architecture.

Finally, Package Diagrams aid in organizing and representing the structure of a system in terms of packages or namespaces. Packages group related elements, offering a means to manage the complexity of large systems.In essence, UML diagrams collectively form a comprehensive suite of tools that empower developers, designers, and stakeholders to visualize, understand, and communicate intricate details of software systems throughout their development lifecycle.

CLASS DIAGRAM:

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represents the whole system. The following points should be remembered while drawing a class diagram − The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance. Responsibility (attributes and methods) of each class should be clearly identified for each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram. At the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

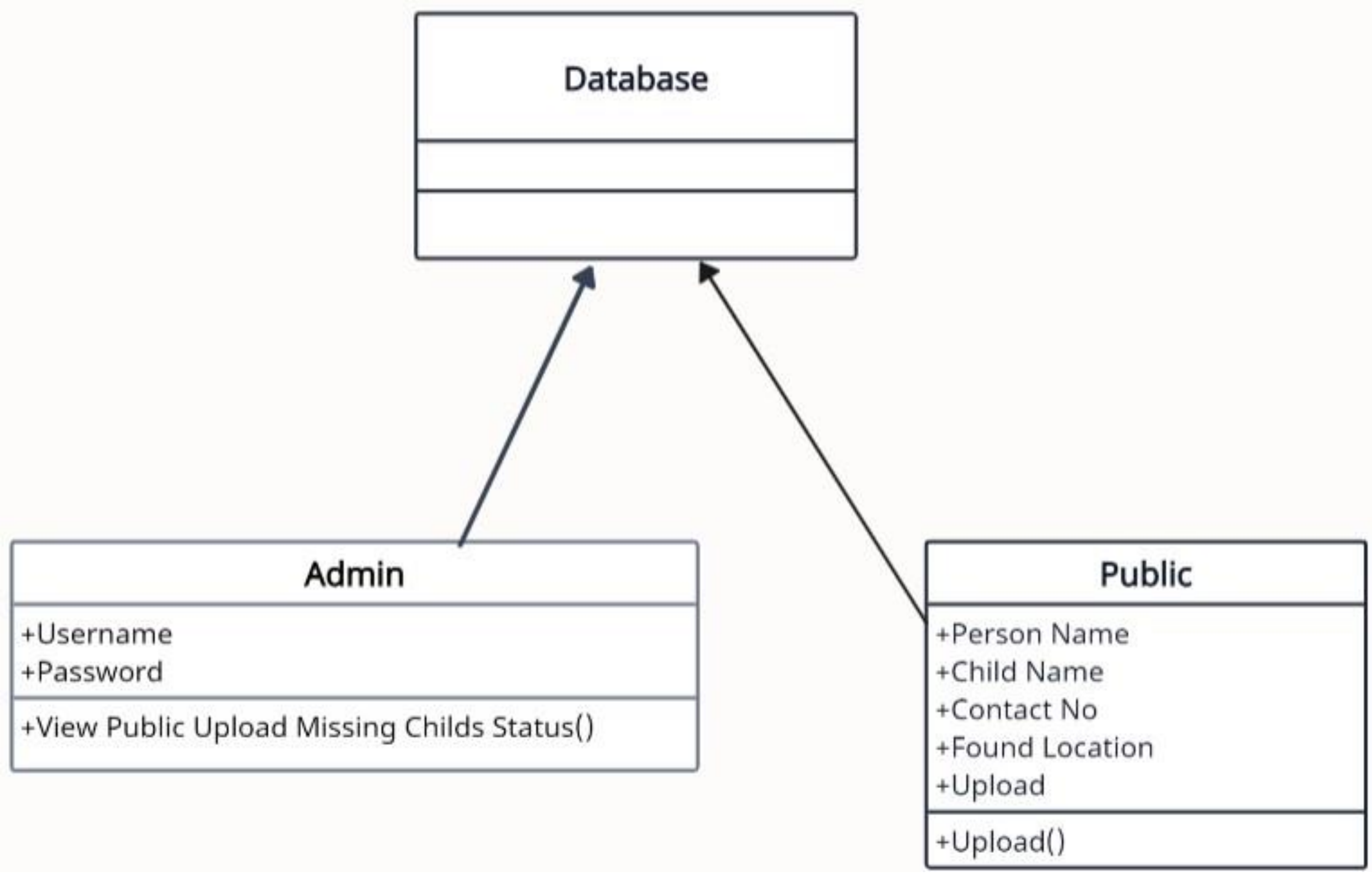


FIGURE 4.3.2.1 CLASS DIAGRAM

USE CASE DIAGRAM:

A use case diagram is a graph of actors set of use cases enclosed by a system boundary, communication associations between the actors and users and generalization among use cases. The use case model defines the outside (actors) and inside (use case) of the system's behavior. Use Case during requirement elicitation and analysis to represent the functionality ofthe system. Use case describes a function by the system that yields a visible result foran actor. The identification of actors and use cases results in the definitions of the boundary of the system i.e., differentiating the tasks accomplished by the system andthe tasks accomplished by its environment. The actors are outside the boundary of thesystem, whereas the use cases are inside the boundary of the system. Use case describes the behavior of the system as seen from the actor's point of view. It describes the functionn provided by the system as a set of events that yield a visible result for the actor.

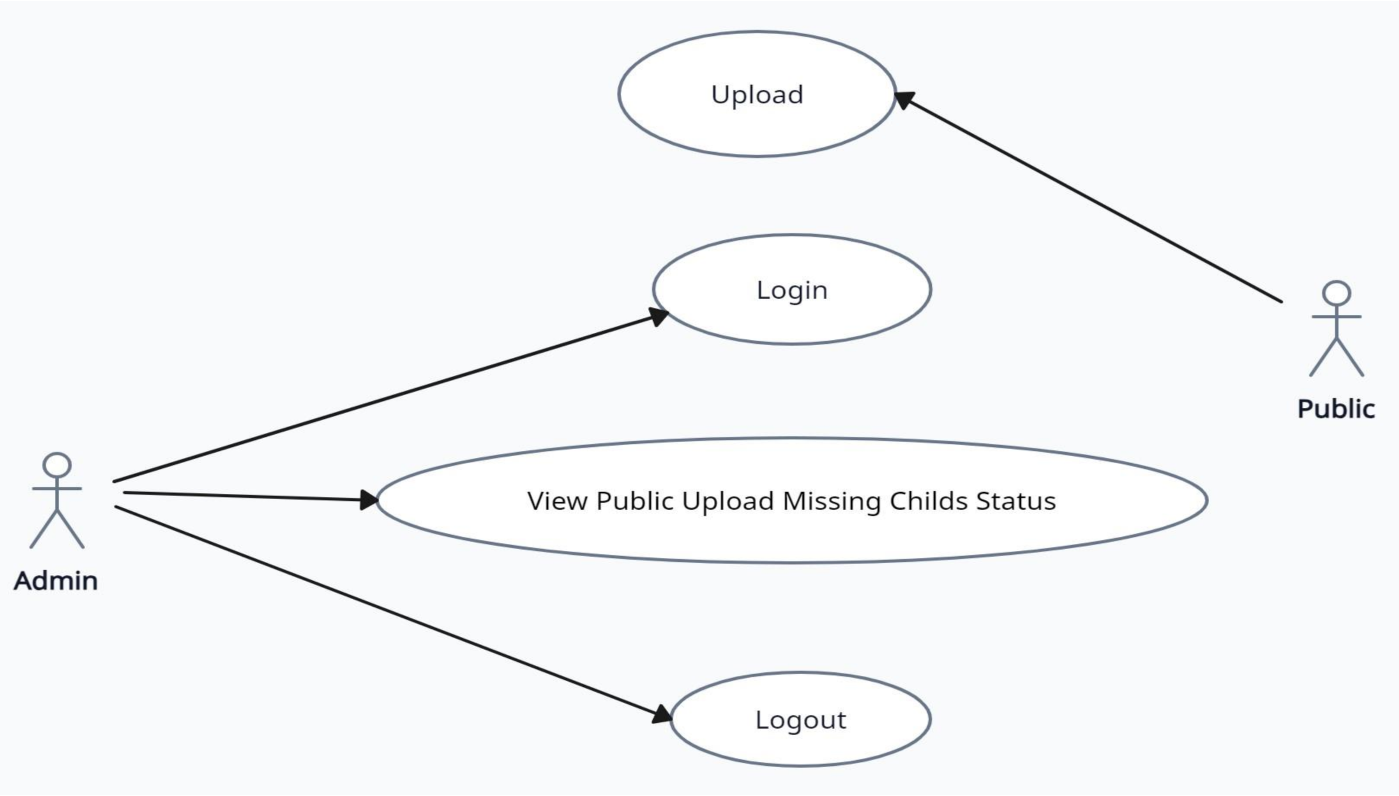


FIGURE 4.3.2.2 USE CASE DIAGRAM

SEQUENCE DIAGRAM:

Sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. If the lifeline is that of an object, it demonstrates a role. Leaving the instance name blank can represent anonymous and unnamed instances.

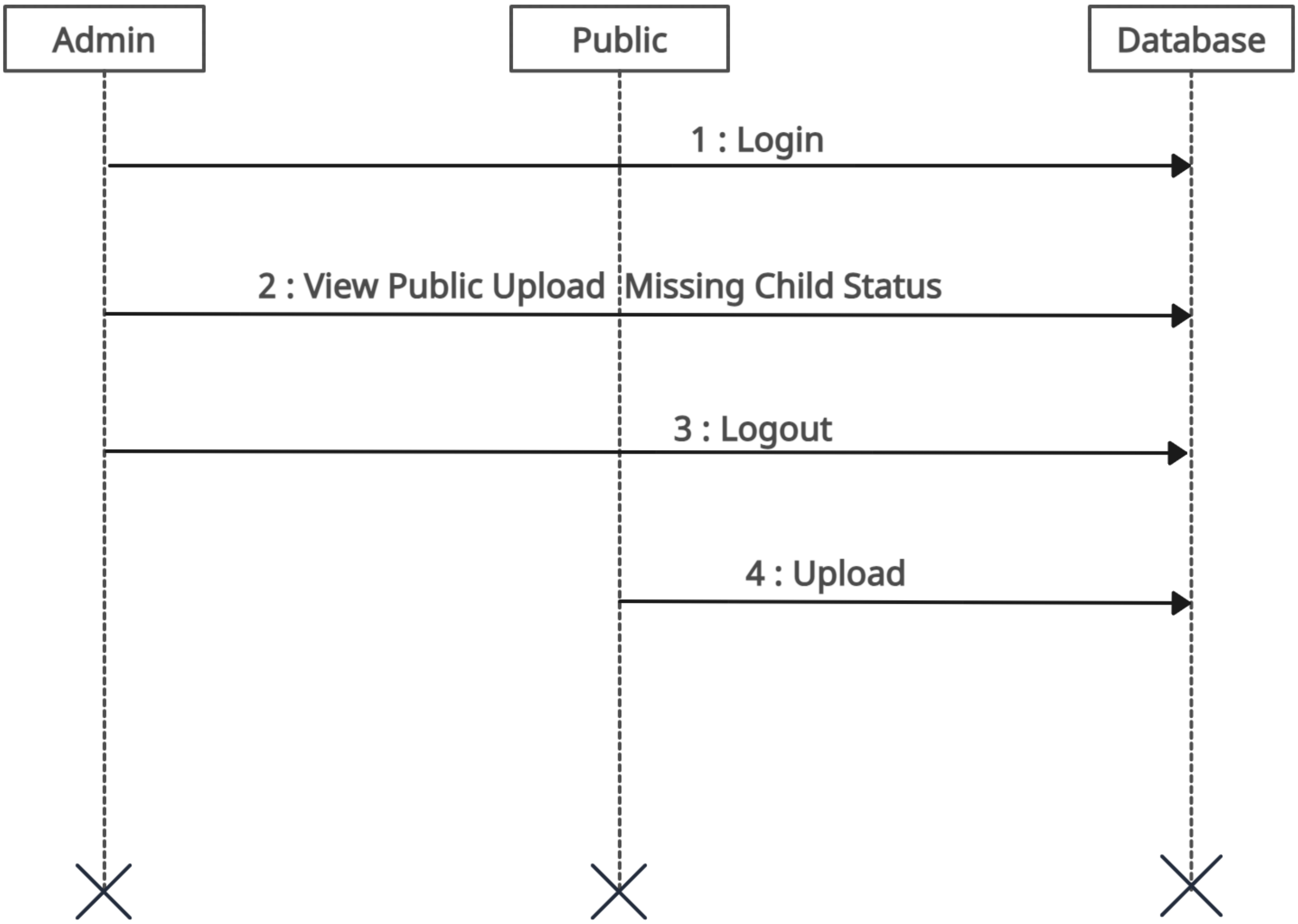


FIGURE 4.3.2.3 SEQUENCE DIAGRAM

ACTIVITY DIAGRAM:

Activity diagram represent the business and operational workflows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state.

So, what is the importance of an Activity diagram, as opposed to a State diagram?

State diagram shows the different states an object is in during the lifecycle of its existence in the system, and the transitions in the states of the objects. These transitions depict the activities causing these transitions, shown by arrows.

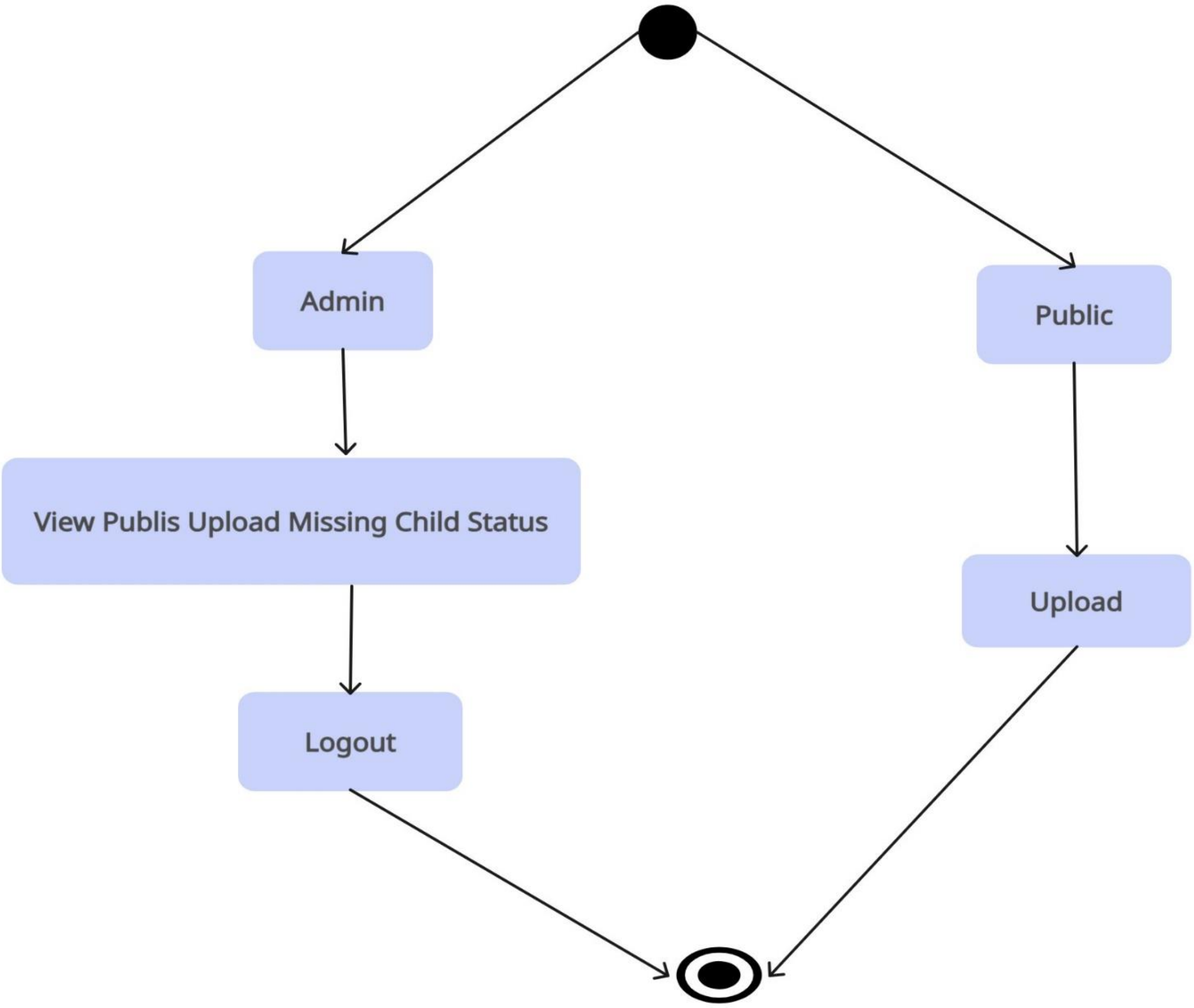


FIGURE 4.3.2.4 ACTIVITY DIAGRAM

5. Software Development Life Cycle

5.1 Phases of SDLC

The Software Development Life Cycle (SDLC) for a project like "Missing Child Distinguishing Framework using Deep Learning and SVM" involves a structured approach to the planning, design, development, testing, deployment, and maintenance of the system. Here's a detailed explanation of the SDLC phases for this project:

1. Requirement Analysis:

- In this phase, the primary focus is to understand the project's objectives and requirements. Detailed steps include:

  - Meeting with stakeholders, including law enforcement agencies, non-profit organizations, and potential users, to gather information about their needs and expectations.

  - Identifying the target audience and understanding the specific problem you aim to solve, which is distinguishing missing children from non-missing individuals.

  - Creating a detailed project scope that outlines the system's boundaries, features, and functionalities.

  - Documenting legal and ethical considerations related to data collection and usage, ensuring compliance with relevant laws and regulations, and addressing privacy concerns.

2. System Design:

- This phase involves creating a high-level design for the system. Key steps include:

  - Defining the overall architecture, outlining how data flows, and identifying the main components, including data storage, data preprocessing, deep learning model, SVM integration, and real-time implementation.

  - Specifying the technical requirements, such as the programming languages, frameworks, and libraries to be used.

  - Creating a system design document that provides a detailed blueprint for the project.

3. Data Collection and Preprocessing:

- Data is a critical component in this project. In this phase:

  - Gather images of missing children and non-missing individuals from reliable sources, such as law enforcement databases or non-profit organizations.

- Clean and format the data, removing duplicates and irrelevant images.

- Annotate the dataset, labeling images to distinguish between missing children and non-missing individuals.

- Apply data preprocessing techniques to standardize image sizes, improve image quality, and handle any missing or corrupted data.

4. Model Development:

- The deep learning model development is a core component. Steps include:

- Designing and implementing a Convolutional Neural Network (CNN) architecture, which can be based on well-known models or developed from scratch.

- Training the CNN model using the annotated dataset for binary classification (missing or not missing).

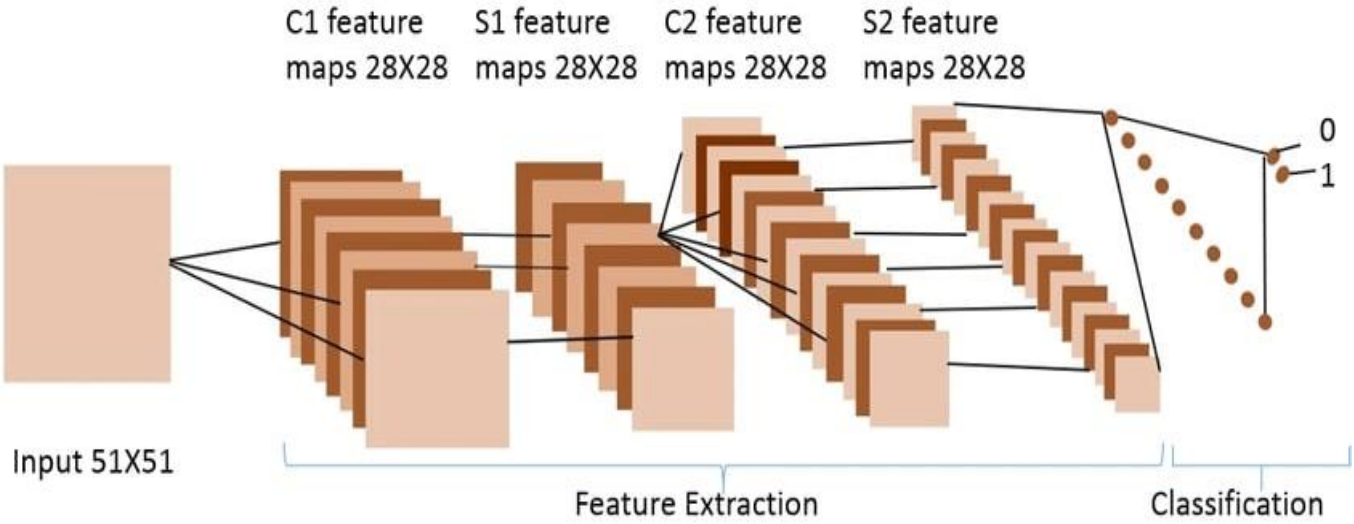
- Fine-tuning the model by experimenting with hyperparameters, optimizing loss functions, and applying data augmentation to improve performance.

5. Feature Extraction:

- After training the deep learning model, you need to extract features from it. Key steps are:

- Define a feature extraction pipeline that captures relevant information from the final layers of the CNN model.

- Extract features for all the images in the dataset, creating a feature dataset to be used as input for the SVM classifier.



6. SVM Classifier Development:

- Building and training the SVM classifier is a crucial step. Detailed steps include:

  - Implementing the SVM classifier to work with the extracted features.

  - Training the SVM model with the labeled dataset, optimizing hyperparameters, and fine-tuning to enhance its ability to distinguish between missing children and non-missing individuals.

7. Model Testing:

- In this phase, you evaluate the performance of the combined deep learning and SVM model. Key steps include:

  - Splitting the dataset into training, validation, and test sets to measure the model's performance.

  - Using evaluation metrics such as accuracy, precision, recall, F1-score, and ROC curves to assess the model's ability to distinguish missing children.

  - Using testing results to fine-tune the model further, iterating through the development phases if necessary to improve performance.

These main seven steps are pivotal in the SDLC for the "Missing Child Distinguishing Framework using Deep Learning and SVM." They lay the foundation for the successful development, training, and testing of the model, ultimately enabling it to effectively distinguish between missing children and non-missing individuals.

6. IMPLEMENTATION

6.1 Sample Code

Admin.py:

```
from django.contrib
import admin
# Register your models here.
```

Apps.py:

```
from django.apps
import AppConfig
class MissingChildappConfig(AppConfig):
        name = 'MissingChildApp'
```

Models.py

```
from django.db
import models
# Create your models here.
```

Tests.py

```
from django.test
import TestCase
# Create your tests here.
```

Urls.py

```
from django.urls
import path from .
import views urlpatterns = [path("index.html", views.index, name="index"),
          path('Login.html', views.Login, name="Login"),
          path('Upload.html', views.Upload, name="Upload"),
          path('OfficialLogin', views.OfficialLogin, name="OfficialLogin"),
          path('UploadAction', views.UploadAction, name="UploadAction"),
         path('ViewUpload', views.ViewUpload, name="ViewUpload"),
]
```

Views.py

```
from django.shortcuts import render

from django.template import RequestContext

import pymysql

from django.http import HttpResponse

from django.conf import settings

from django.core.files.storage import FileSystemStorage

import datetime

import os

import cv2

import numpy as np

from keras.utils.np_utils import to_categorical

from keras.layers import  MaxPooling2D

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Convolution2D

from keras.models import Sequential

from keras.models import model_from_json


global index

index = 0

global missing_child_classifier

global cascPath

global faceCascade


def index(request):

    if request.method == 'GET':

       return render(request, 'index.html', {})


def Login(request):

    if request.method == 'GET':

       return render(request, 'Login.html', {})
```

```
def Upload(request):
    if request.method == 'GET':
       return render(request, 'Upload.html', {})


def OfficialLogin(request):
    if request.method == 'POST':
      username = request.POST.get('t1', False)
      password = request.POST.get('t2', False)
      if username == 'admin' and password == 'admin':
       context= {'data':'welcome '+username}
       return render(request, 'OfficialScreen.html', context)
      else:
       context= {'data':'login failed'}
       return render(request, 'Login.html', context)


def ViewUpload(request):
    if request.method == 'GET':
        strdata = '<table border=1 align=center width=100%><tr><th>Upload Person Name</th><th>Child Name</th><th>Contact No</th><th>Found Location</th><th>Child Image <th>Uploaded Date</th><th>Status</th></tr><tr>'
         con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'missingchilddb',charset='utf8')
       with con:
          cur = con.cursor()
          cur.execute("select * FROM missing")
          rows = cur.fetchall()
          for row in rows:
             strdata+='<td>'+row[0]+'</td><td>'+str(row[1])+'</td><td>'+row[2]+'</td><td>'+row[3]+'</td><td><img src="/static/'+row[4]+'" width="200" height="200"></img></td><td>'
            strdata+=str(row[5])+'</td><td>'+str(row[6])+'</td></tr>'
    context= {'data':strdata}
    return render(request, 'ViewUpload.html', context)
```

```
def UploadAction(request):
    global index
    global missing_child_classifier
    global cascPath
    global faceCascade
    if request.method == 'POST' and request.FILES['t5']:
        output = ''
        person_name = request.POST.get('t1', False)
        child_name = request.POST.get('t2', False)
        contact_no = request.POST.get('t3', False)
        location = request.POST.get('t4', False)
        myfile = request.FILES['t5']
        fs = FileSystemStorage()
        filename = fs.save('MissingChildApp\static\\'+child_name+'.png', myfile)
        #if index == 0:
        cascPath = "haarcascade_frontalface_default.xml"
        faceCascade = cv2.CascadeClassifier(cascPath)
        #index = 1
        option = 0
        frame = cv2.imread(filename)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray,1.3,5)
        print("Found {0} faces!".format(len(faces)))
        img = ''
        status = 'Child not found in missing database'
        if len(faces) > 0:
            for (x, y, w, h) in faces:
                img = frame[y:y + h, x:x + w]
                option = 1
        if option == 1:
            with open('model/model.json', "r") as json_file:
                loaded_model_json = json_file.read()
                missing_child_classifier = model_from_json(loaded_model_json)
            missing_child_classifier.load_weights("model/model_weights.h5")
            missing_child_classifier._make_predict_function()
            img = cv2.resize(img, (64,64))
            im2arr = np.array(img)
            im2arr = im2arr.reshape(1,64,64,3)
            img = np.asarray(im2arr)
            img = img.astype('float32')
            img = img/255
            preds = missing_child_classifier.predict(img)
            if(np.amax(preds) > 0.60):
                status = 'Child found in missing database'
        now = datetime.datetime.now()
        current_time = now.strftime("%Y-%m-%d %H:%M:%S")
        filename = os.path.basename(filename)
        db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database
= 'missingchilddb',charset='utf8')
        db_cursor = db_connection.cursor()
    query = "INSERT INTO missing(person_name,child_name,contact_no,location,image,upload_date,status)
VALUES('"+person_name+"','"+child_name+"','"+contact_no+"','"+location+"','"+filename+"','"+str(current
_time)+"','"+status+"')"
        db_cursor.execute(query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        context= {'data':'Thank you for uploading. '+status}
        return render(request, 'Upload.html', context)
```

7.TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

7.1 TESTING OBJECTIVES:

- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation .
- To make sure that during the operation, incorrect input, processing and output will be detected .
- To see that when correct inputs are fed to the system the outputs are correct .
- To verify that the controls incorporated in the same system as intended .
- Testing is a process of executing a program with the intent of finding an error .
- A good test case is one that has a high probability of finding an as yet undiscovered error .

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

TESTING METHODOLOGIES:

- White box testing.
- Black box testing.
- Unit testing.
- Integration testing.
- User acceptance testing.
- Output testing.
- Validation testing.

White Box Testing:

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity.

Black Box Testing:

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top Down Integration:

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module.

2. Bottom Up Integration:

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated.

5) User acceptance Testing:

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

6) Output Testing:

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

7) Validation Testing:

Validation testing is generally performed on the following fields:

→**Text Field:** The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

→**Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform.

→**Preparation of Test Data:**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

→**Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

→**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

Functional testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2 USER TRAINING:

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

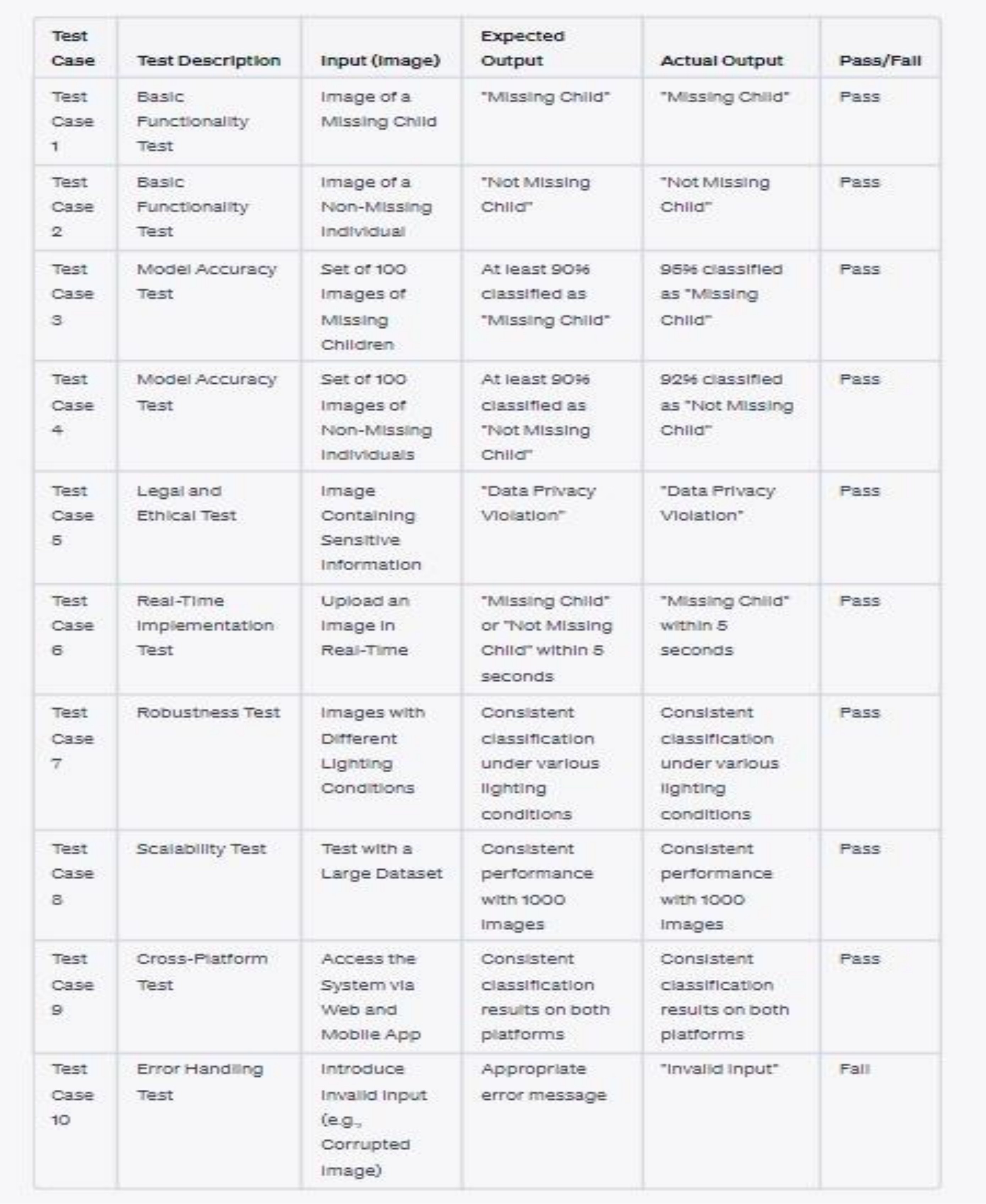
MAINTAINENCE:

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

7.3 TESTING STRATEGY :

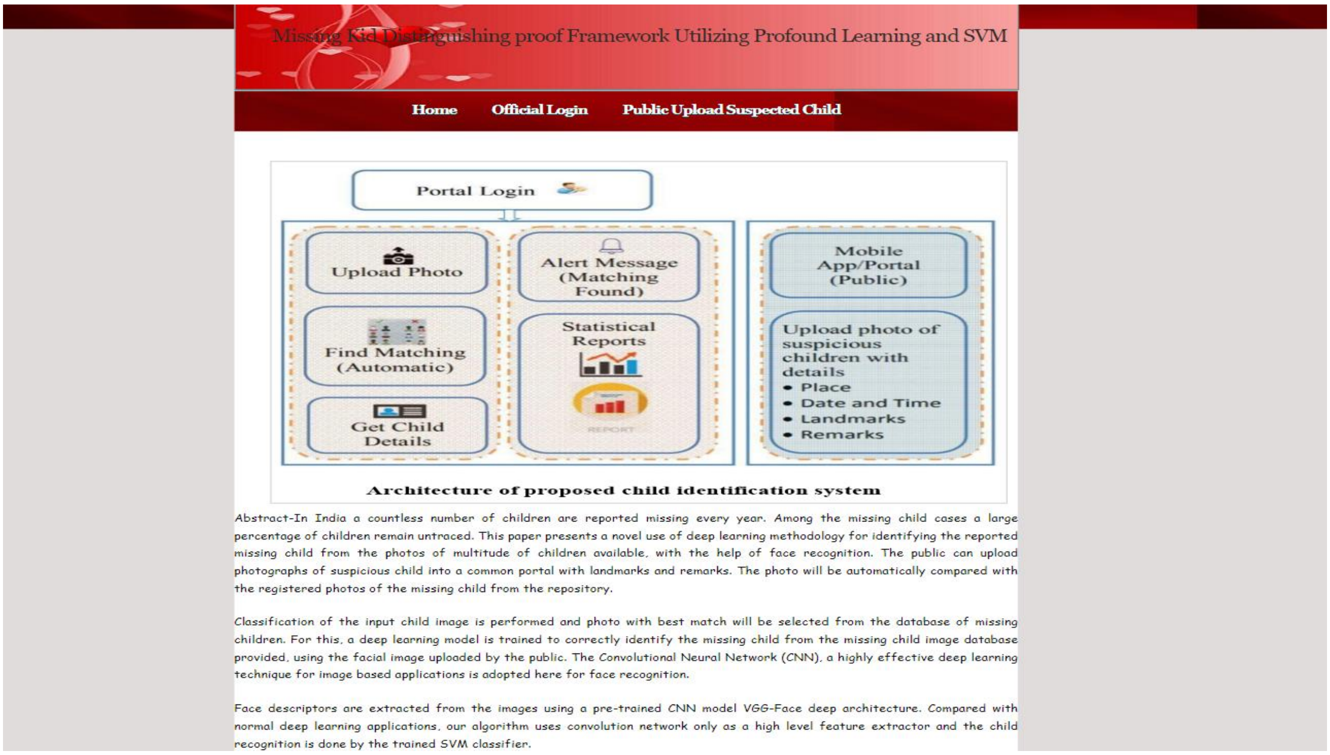
A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Two strategies are present unit testing and system testing.

TEST CASES:



Test Case	Test Description	Input (Image)	Expected Output	Actual Output	Pass/Fail
Test Case 1	Basic Functionality Test	Image of a Missing Child	"Missing Child"	"Missing Child"	Pass
Test Case 2	Basic Functionality Test	Image of a Non-Missing Individual	"Not Missing Child"	"Not Missing Child"	Pass
Test Case 3	Model Accuracy Test	Set of 100 Images of Missing Children	At least 90% classified as "Missing Child"	95% classified as "Missing Child"	Pass
Test Case 4	Model Accuracy Test	Set of 100 Images of Non-Missing Individuals	At least 90% classified as "Not Missing Child"	92% classified as "Not Missing Child"	Pass
Test Case 5	Legal and Ethical Test	Image Containing Sensitive Information	"Data Privacy Violation"	"Data Privacy Violation"	Pass
Test Case 6	Real-Time Implementation Test	Upload an Image in Real-Time	"Missing Child" or "Not Missing Child" within 5 seconds	"Missing Child" within 5 seconds	Pass
Test Case 7	Robustness Test	Images with Different Lighting Conditions	Consistent classification under various lighting conditions	Consistent classification under various lighting conditions	Pass
Test Case 8	Scalability Test	Test with a Large Dataset	Consistent performance with 1000 images	Consistent performance with 1000 images	Pass
Test Case 9	Cross-Platform Test	Access the System via Web and Mobile App	Consistent classification results on both platforms	Consistent classification results on both platforms	Pass
Test Case 10	Error Handling Test	Introduce Invalid Input (e.g., Corrupted Image)	Appropriate error message	"Invalid Input"	Fail

8.OUTPUT SCREENS


Missing Kid Distinguishing proof Framework Utilizing Profound Learning and SVM
Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Alert Message (Matching Found)
Mobile App/Portal (Public)
Find Matching (Automatic)
Statistical Reports
Upload photo of suspicious children with details
Place
Date and Time
Landmarks
Remarks
Get Child Details
REPORT

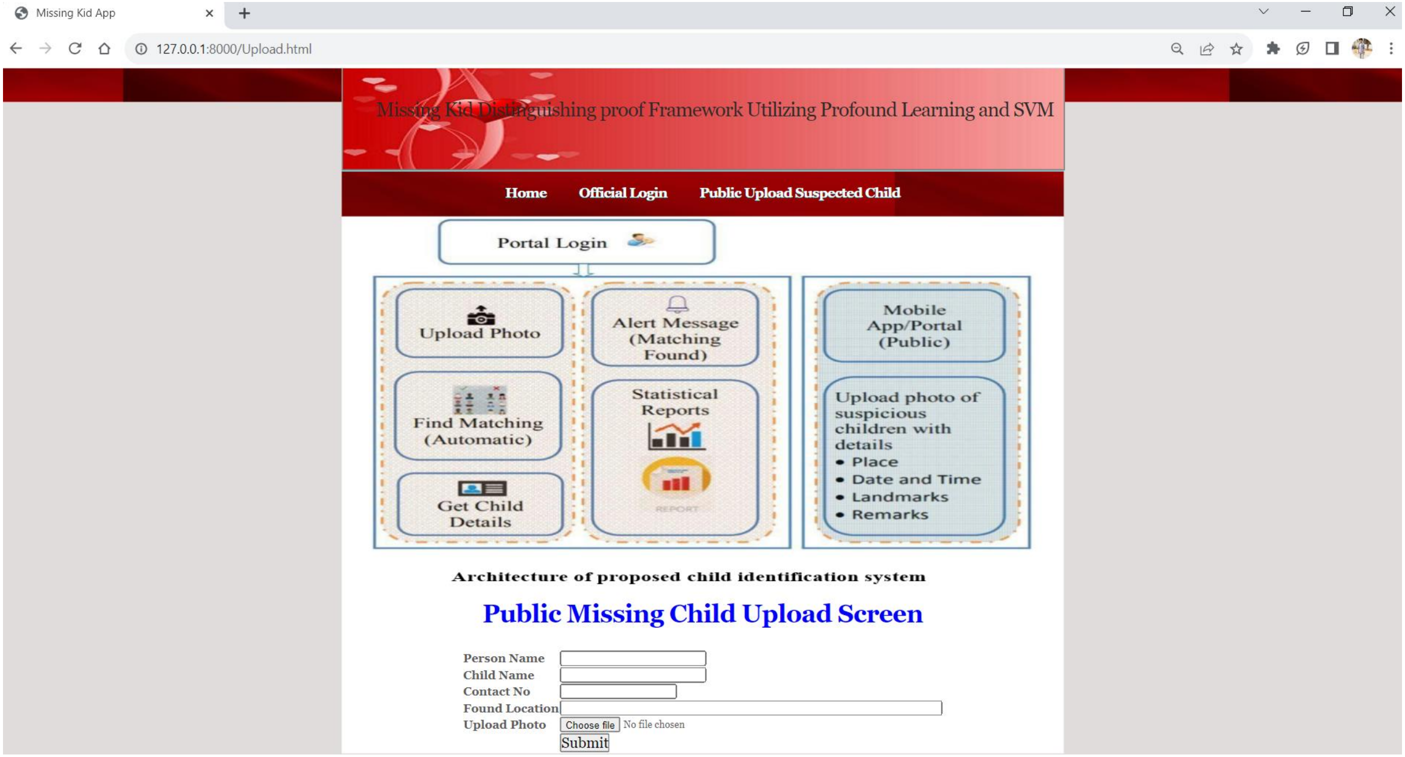
Architecture of proposed child identification system

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.

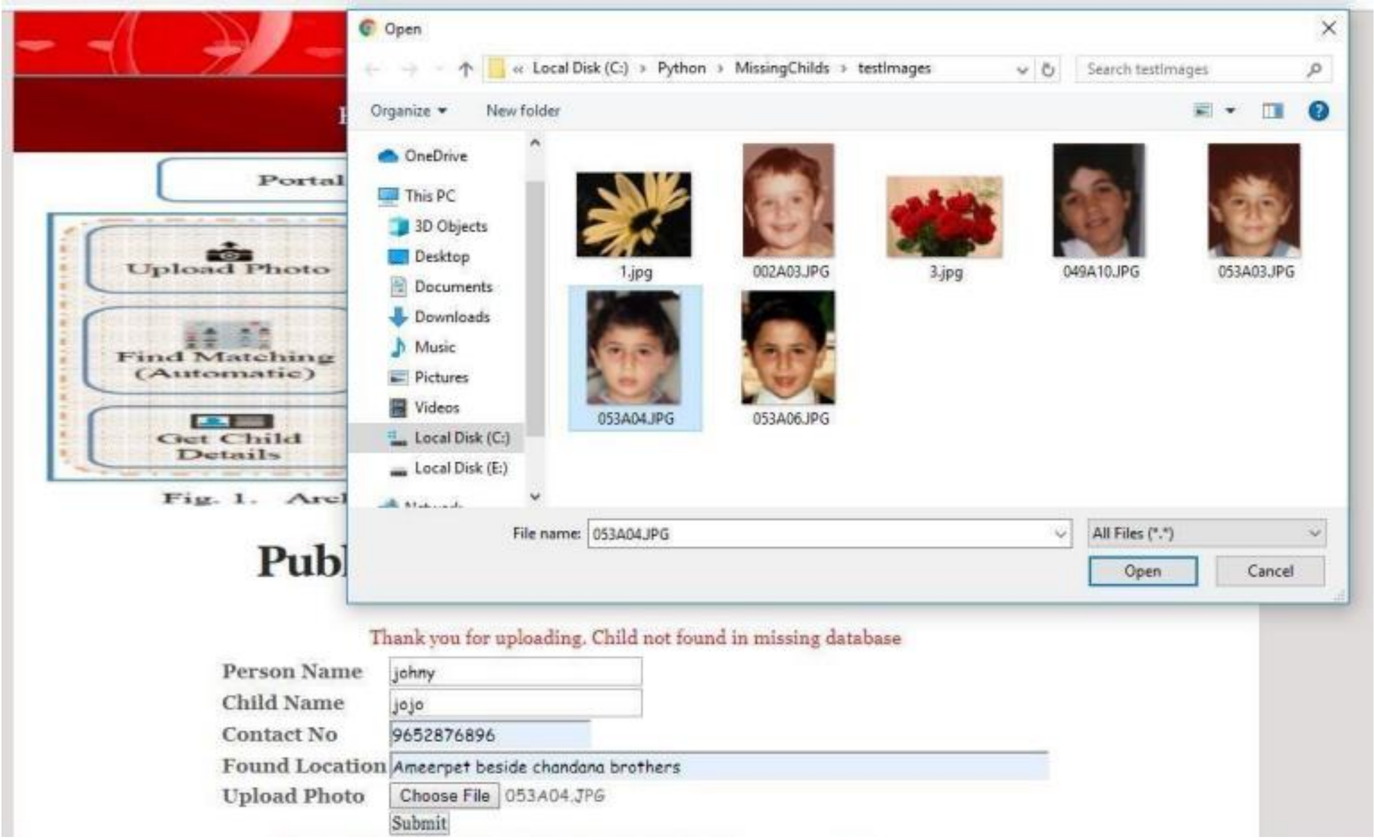
Home screen of application which shows detailed structure of the application how data can be processed in a manner.


Missing Kid App
127.0.0.1:8000/Upload.html
Missing Kid Distinguishing proof Framework Utilizing Profound Learning and SVM
Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Alert Message (Matching Found)
Mobile App/Portal (Public)
Find Matching (Automatic)
Statistical Reports
Upload photo of suspicious children with details
Place
Date and Time
Landmarks
Remarks
Get Child Details
REPORT
Architecture of proposed child identification system

Public Missing Child Upload Screen

Person Name
Child Name
Contact No
Found Location
Upload Photo Choose file No file chosen
Submit

Public can click on" Public can upload the suspected child" in this screen

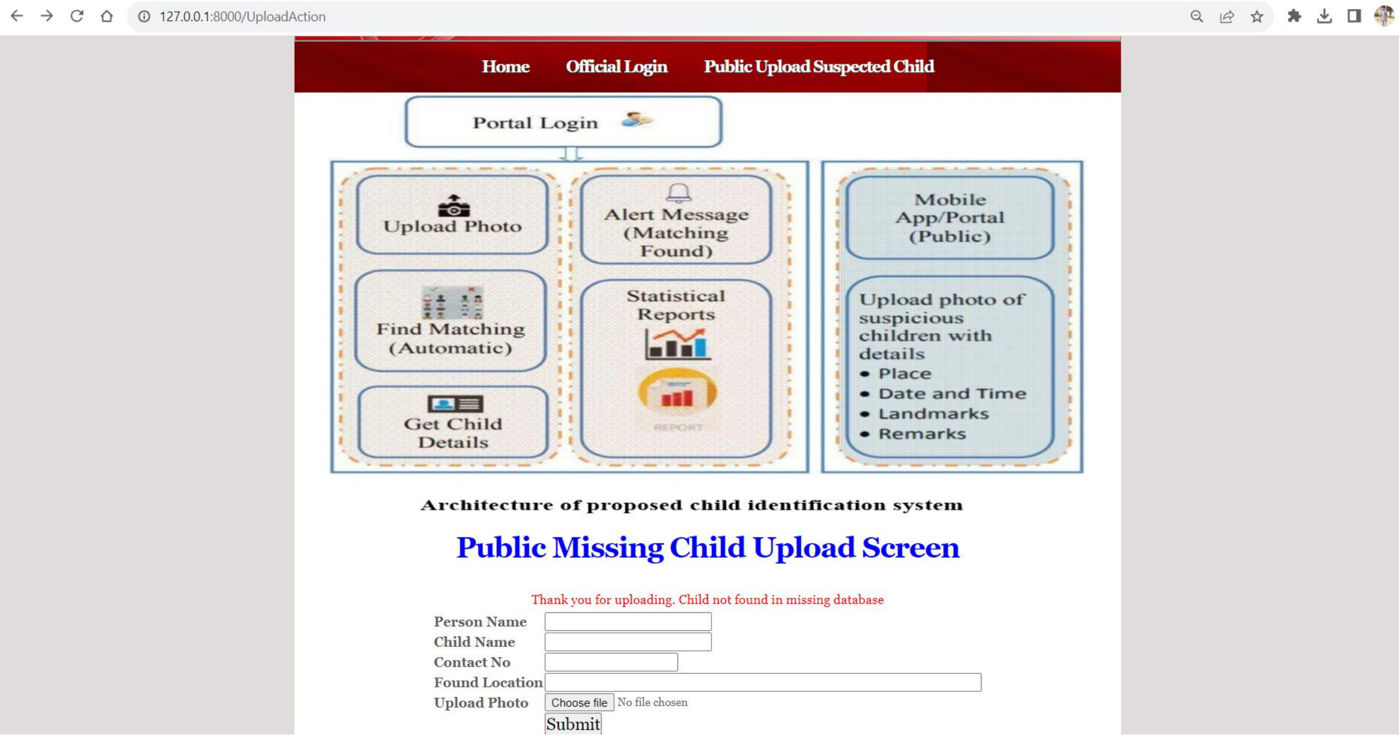


File name: 053A04.JPG

Thank you for uploading. Child not found in missing database

Person Name: johny
Child Name: jojo
Contact No: 9652876896
Found Location: Ameerpet beside chandana brothers
Upload Photo: 053A04.JPG

public can upload the details in this screen when a suspicious child will occur

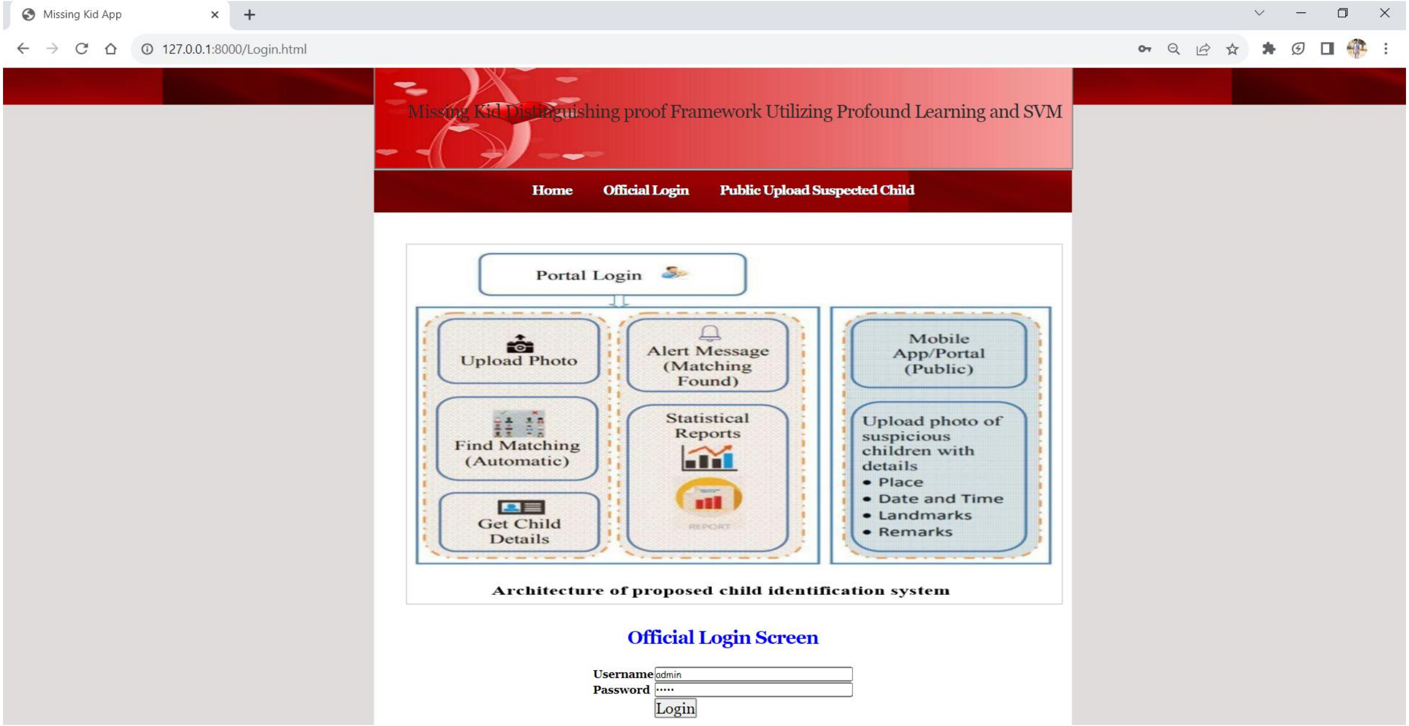

127.0.0.1:8000/UploadAction
Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Alert Message (Matching Found)
Mobile App/Portal (Public)
Find Matching (Automatic)
Statistical Reports
REPORT
Upload photo of suspicious children with details
Place
Date and Time
Landmarks
Remarks
Get Child Details
Architecture of proposed child identification system
Choose file
No file chosen
Submit

Public Missing Child Upload Screen

Thank you for uploading. Child not found in missing database

Person Name
Child Name
Contact No
Found Location
Upload Photo

view public upload missing child status in this screen

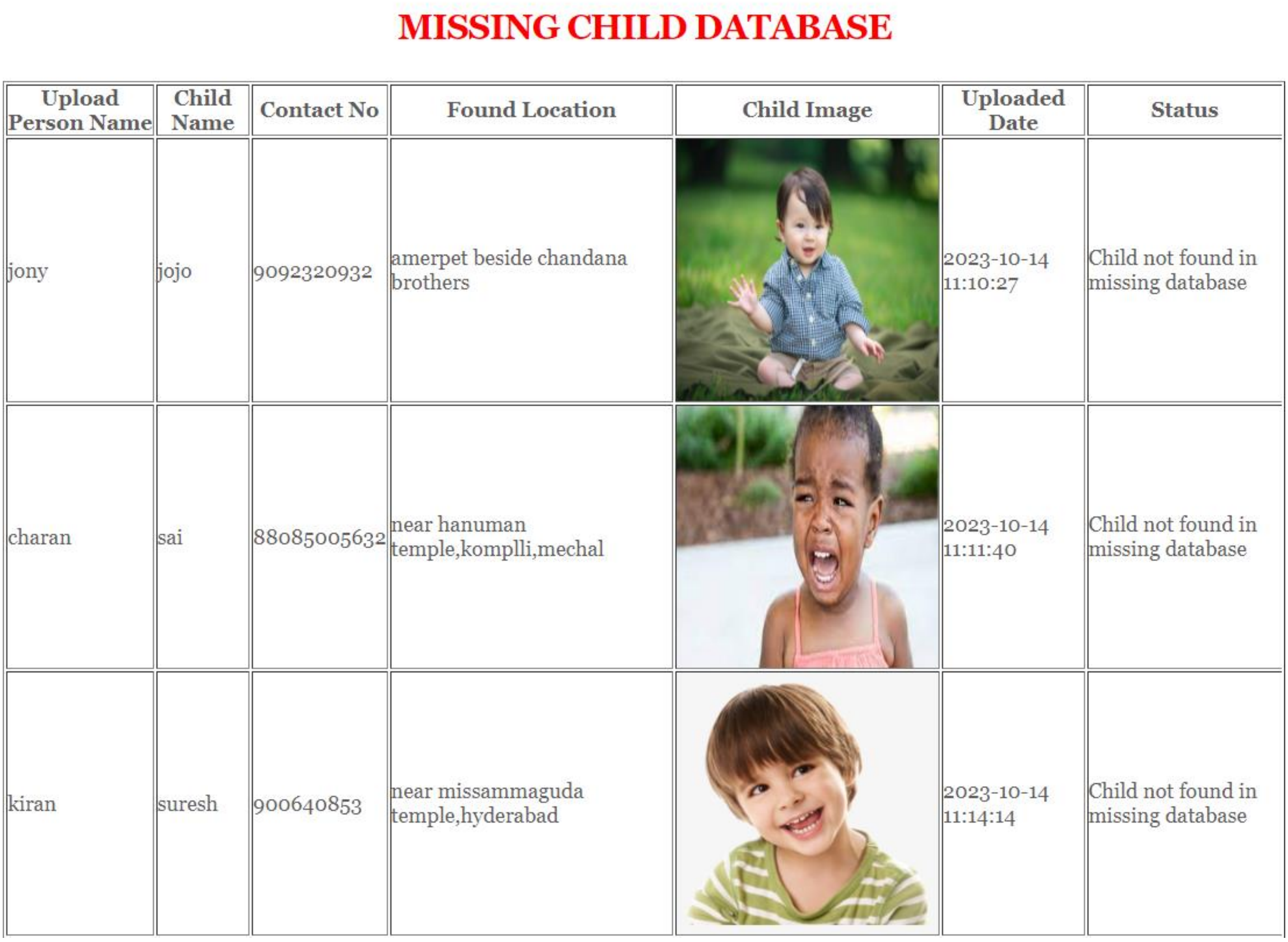

Missing Kid App
127.0.0.1:8000/Login.html
Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Alert Message (Matching Found)
Mobile App/Portal (Public)
Find Matching (Automatic)
Statistical Reports
REPORT
Upload photo of suspicious children with details
• Place
• Date and Time
• Landmarks
• Remarks
Get Child Details
Architecture of proposed child identification system
Login

Missing Kid Distinguishing proof Framework Utilizing Profound Learning and SVM

Official Login Screen

Username admin
Password ·····

Official login screen,access to only the admin



MISSING CHILD DATABASE

Upload Person Name	Child Name	Contact No	Found Location	Child Image	Uploaded Date	Status
jony	jojo	9092320932	amerpet beside chandana brothers		2023-10-14 11:10:27	Child not found in missing database
charan	sai	88085005632	near hanuman temple,kompli,mechal		2023-10-14 11:11:40	Child not found in missing database
kiran	suresh	900640853	near missammaguda temple,hyderabad		2023-10-14 11:14:14	Child not found in missing database

view all the details of the public uploaded photos it can be seen by admin when he is logged into the database.

9.CONCLUSION & FUTURE SCOPE

9.1 CONCLUSION:

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 offace recognition could be used for reliable missing children identification.

9.2 FUTURE SCOPE:

In future we will use some more efficient algorithms to improve accuracy and efficiency for my application. The future of such a system lies in the integration of advanced technologies, improved collaboration, and a commitment to addressing the challenges associated with privacy and ethics.

10.Bibliography

10.1 WEBSITES:

1. https://en.wikipedia.org/wiki/FindFace
2. https://www.reuters.com/article/us-china-trafficking-apps/mobileapp-helps-chinarecover-hundreds-of-missing-childrenidUSKBN15J0GU

10.2 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. Rohit Satle, Vishnuprasad Poojary, John Abraham, Shilpa Wakode, "Missing child identification using face recognition system", International Journal of Advanced Engineering and Innovative Technology (IJAEIT), Volume 3 Issue 1 July - August 2016.
5. Simonyan, Karen and Andrew Zisserman, "Very deep convolutional networks for largescale image recognition", International Conference on Learning Representations (ICLR), April 2015.
6. O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in British Machine Vision Conference, vol. 1, no. 3, pp. 1-12, 2015.
7. Vedaldi, and K. Lenc, "MatConvNet: Convolutional Neural Networks for MATLAB", ACM International Conference on Multimedia, Brisbane, October 2016.