MISSING CHILD IDENTIFICATION SYSTEM USING DEEPLEARNING AND MULTICLASS SVM

A Project Report Submitted in partial fulfillment of the requirement for the Award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

A. SARAN	**-**	**20T91A0507**
D. NIKHIL	**-**	**20T91A0521**
J.SAI KUMAR	**-**	**20T91A0532**

Under the Esteemed Guidance of

Mr. P. Sasi kumar, M. Tech., (Ph.D)

Assistant Professor

Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GIET ENGINEERING COLLEGE
[Affiliated to JNTUK, Kakinada |Approved by AICTE| Accredited by NAAC A+]

NH-16, CHAITANYA KNOWLEDGE CITY, RAJAMAHENDRAVARAM – 533 296,

ANDHRA PRADESH

2023 - 2024

GIET ENGINEERING COLLEGE

[Affiliated to JNTUK, Kakinada |Approved by AICTE| Accredited by NAAC A+]
NH-16, CHAITANYA KNOWLEDGE CITY, RAJAMAHENDRAVARAM – 533 296,
ANDHRA PRADESH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **A.SARAN (20T91A0507), D.NIKHIL (20T91A0521), J. SAI KUMAR (20T91A0532),** Studying in IV B .Tech II Semester of ComputerScience and Engineering have submitted their project "MISSING CHILD IDENTIFICATION USING DEEP LEARNING AND MULTICLASS SVM" during the academic year 2023-2024 inpartial fulfillment of the requirements for the award of degree in Bachelor of Technology, JNTUK,Kakinada

Project Guide	**Head of the Department**
Mr. P. Sasi Kumar, M. Tech.,(Ph.D)	**Dr. SK. Meera Sharif,M.Tech.,Ph.D.**
Assistant Professor	**Professor and HoD**
Dept. of CSE	**Dept. of CSE**

INTERNAL EXAMINAR **EXTERNAL EXAMINAR**

ACKNOWLEDGEMENT

It is a privilege for us to have undertaken the project **"MISSING CHILD IDENTIFICATION USING DEEP LEARNING AND MULTICLASS SVMs"** in **GIET ENGINEERING COLLEGE, RAJAMAHENDRAVARAM**.

We avail this opportunity to express our deep sense of gratitude and heart full thanks to **Sri K. SASI KIRAN VARMA**, Vice Chairman of **GIET ENGINEERING COLLEGE, RAJAMAHENDRAVARAM.**

We are thankful to our Principal **Dr. M. VIJAY SEKHAR BABU** and our HOD **Dr. SK. MEERA SHARIF** for encouraging us to do this project.

We are deeply indebted to our internal project guide **Mr. P. SASI KUMAR,** Assistant Professor in Computer Science and Engineering, **GIET ENGINEERING COLLEGE**, whose motivation in the field of software development made us overcome all the hardships during the course of study.

We are heartily thankful to our project coordinator **MR. J. BALA AMBEDKAR ,**Assistant Professor, **GIET ENGINEERING COLLEGE**, for his moral support which was alwaysthere to comfort and solace during tough times.

Finally, we would like to thank our **TEACHING AND NON-TEACHING STAFF** whose blessings and encouragement were always there as a source of strength and inspiration. Although the title "Acknowledgement" cannot represent our true feelings for all these Persons, we feel very much thankful to all of them and also to our **PARENTS** and **FRIENDS** for encouraging and giving us all the moral support required for making this endeavor a reality.

A. SARAN	**20T91A0507**
D. NIKHIL	**20T91A0521**
J.SAI KUMAR	**20T91A0532**

DECLARATION

We here by declare that this project entitled "**MISSING CHILD IDENTIFICATION USING DEEP LEARNING AND MULTICLASS SVM"** submitted to the Department of COMPUTERSCIENCE AND ENGINEERING, GIET ENGINEERING COLLEGE, affiliated to JNTUK, Kakinada, as partial fulfillment for the award of Bachelor of Technology degree is entirely the originalwork done byus and has not been submitted to anyother organization.

Project Members	Pin Numbers	Signature
A. SARAN	**20T91A0507**	
D. NIKHIL	**20T91A0521**	
J.SAI KUMAR	**20T91A0532**	

ABSTRACT

This study introduces a novel deep learning approach for identifying missing children using facial recognition. Leveraging public engagement, suspicious child photos can be uploaded to a central platform for comparison against a repository of missing children's images. Employing Convolutional Neural Network (CNN) technology, specifically VGG-Face deep architecture, facialdescriptors are extracted for recognition. Unlike typical deep learning applications, our algorithm uses CNN solely as a high-level feature extractor, employing a trained SVM classifier for child identification. Comparative analysis between established models like RESNET 50 and VGG 16 with a custom CNN algorithm demonstrates superior prediction accuracy. This innovative methodaims to address the challenges in accurately identifying missing children, presenting a promising application of deep learning in social welfare.

TABLE OF CONTENTS

LIST OF CONTENTS

LIST OF FIGURES

LIST OF KEY WORDS

1.	ML	Machine Learning
2.	DL	Deep Learning
3.	CNN	Convolutional Neural Network
4.	VGG 16	Visual Geomentry Group
5.	RESNET-50	Residual Network

CHAPTER – 1
INTRODUCTION

CHAPTER - 1
INTRODUCTION

1.1 Project 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 representa 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) inthe Parliament (LS Q no. 3928, 20-03- 2018), more than one lakh children (1,11,569 in actual numbers) were reported to have gone missing till 2016, and 55,625 of them remained untraced tillthe 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 an other region or another state, for various reasons. So even if a child is found, it is difficultto identify him/her from the reported missing cases. A framework and methodology for developingan assistive tool for tracing missing child is described in this paper. An idea for maintaining a virtualspace 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 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. Thisage gap reflects in the images since aging affects the shape of the face and texture of the skin. Thefeature 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. Theimage taken by public may not be of good quality, as some of them may be captured from a distancewithout the knowledge of the child. A deep learning [1] architecture considering all these constrainis designed here.

1.1.1 How Machine Learning Works

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses onthe use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is an important component of the growing field of data science. Through the useof statistical methods, algorithms are trained to make classifications or predictions, uncovering keyinsights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expandand grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them.

1.1.2 Machine Learning Algorithms

1. **A Decision Process:** In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabeled, your algorithm will produce an estimate about a pattern in the data.

2. **An Error Function:** An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.

3.A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate.The algorithm will repeat this evaluate and optimize process, updating weights autonomously until a threshold of accuracy has been met.

1.1.3 Types of Machine Learning Methods Supervised machine learning

Supervised learning also known as supervised machine learning, is defined by its use of labelled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fedinto the model, it adjusts its weights until the model has been fitted appropriately. This occurs as partof the cross-validation process to ensure that the model avoids over fitting or under fitting. Supervisedlearning helps organizations solve for a variety of real-world problems at scale, such as classifyingspam in a separate folder from your inbox. Some methods used in supervised learning include neuralnetworks, naïve bayes, linear regression, logistic regression, random forest, support vector machine.

1. **Unsupervised Machine Learning**

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyses and cluster unlabeled datasets. These algorithms discover hidden patterns ordata groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross selling strategies, customer segmentation, image and pattern recognition [2]. It's also used to reduce the number of features in a model through the process of dimensionality reduction; principal componentanalysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, means clustering, probabilistic clustering methods, and more [3].

2. **Semi-Supervised Learning**

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labelled data set to guide classification and feature extraction froma larger, unlabeled data set . Semi-supervised learning can solve the problem of having not enough labelled data (or not being able to afford to label enough data) to train a supervised learning algorithm.

1.1.4 Practical Use Of Machine Learning

1. **Speech Recognition**: It is also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, and it is a capability which uses natural language processing (NLP)to process human speech into a written format. Many mobile devices incorporate speech recognitioninto their systems to conduct voice search—e.g. Siri—or provide more accessibility around texting.

2. **Customer Service:** Online chatbots are replacing human agents along the customer journey. They answer frequently asked questions (FAQs) around topics, like shipping, or provide personalized advice, cross-selling products or suggesting sizes for users, changing the way we think about customer engagement across websites and social media platforms [4]. Examples include messagingbots on e-commerce sites with virtual agents, messaging apps, such as Slack and Facebook Messenger, and tasks usually done by virtual assistants and voice assistants.

3. **Computer Vision:** This AI technology enables computers and systems information from digital images, videos and other visual inputs, and based on those inputs, it can takerecognition tasks . Powered by convolutional neural networks, computer vision has applications within photo tagging.

4. **Recommendation Engines:** Using past consumption behaviour data, AI algorithms can help to discover data trends that can be used to develop more effective cross-selling strategies. This is usedto make relevant add-on recommendations to customers during the checkout process for online retailers [5].

5. **Automated Stock Trading:** Designed to optimize stock portfolios, AI-driven high-frequency trading platforms make thousands or even millions of trades per day without human intervention.

1.1.5 What Is Deep Learning

Deep learning is one of the foundations of artificial intelligence (AI), and the current interest in deep learning issue in part to the buzz surrounding AI. Deep learning techniques have improved theability to classify, recognize, detect and describe – in one word, understand [6]. For example, deep learning is used to classify images, recognize speech, detect objects and describe content.

Several developments are now advancing deep learning:

Algorithmic improvements have boosted the performance of deep learning methods.New machine

learning approaches have improved accuracy of models.

New classes of neural networks have been developed that fit well for applications like text translation and image classification.

We have a lot more data available to build neural networks with many deep layers, including streaming data from the Internet of Things, textual data from social media, physicians notes and investigative transcripts [7].

Computational advances of distributed cloud computing and graphics processing units have put incredible computing power at our disposal. This level of computing power is necessary to train deep algorithms.

At the same time, human-to-machine interfaces have evolved greatly as well. The mouse and the keyboard are being replaced with gesture, swipe, touch and natural language, ushering in a renewed interest in AI and deep learning [8].

1.1.6 How Deep Learning Works

Deep learning changes how you think about representing the problems that you're solving with analytics. It moves from telling the computer how to solve a problem to training the computer to solve the problem itself.

A traditional approach to analytics is to use the data at hand to engineer features to derive new variables, then select an analytic model and finally estimate the parameters (or the unknowns) of that model. These technique scan yield predictive systems that do not generalize well because completeness and correctness depend on the quality of the model and its features . For example, ifyou develop a fraud model with feature engineering, you start with a set of variables, and you mostlikely derive a model from those variables using data transformations. You may end up with 30,000variables that your model depends on, then you have to shape the model, figure out which variablesare meaningful, which ones are not, and so on. Adding more data requires you to do it all over again.

The new approach with deep learning is to replace the formulation and specification of the model with hierarchical characterizations (or layers) that learn to recognize latent features of the data fromthe regularities in the layers [9]. The paradigm shift with deep learning is a move from feature engineering to feature representation. The promise of deep learning is that it can lead to predictivesystems that generalize well, adapt well, continuously improve as new data arrives, and are more dynamic than predictive systems built on hard business rules. You no longer fit a model. Instead, you train the task.

Deep learning is making a big impact across industries. In life sciences, deep learning can be usedfor advanced image analysis, research, drug discovery, prediction of health problems and disease symptoms, and the acceleration of insights from genomic sequencing. In transportation, it can help autonomous vehicles adapt to changing conditions [10]. It is also used to protect critical infrastructure and speed response.

1.1.7 How Deep Learning Being Used

To the outside eye, deep learning may appear to be in a research phase as computer science researchers and data scientists continue to test its capabilities. However, deep learning has many practical applications that businesses are using today, and many more that will be used as researchcontinues . Popular uses today include:

1. **Speech Recognition**

Both the business and academic worlds have embraced deep learning for speech recognition. Xbox, Skype, Google Now and Apple's Siri, to name a few, are already employing deep learning technologies in their systems to recognize human speech and voice patterns.

2. **Natural Language Processing**

Neural networks, a central component of deep learning, have been used to process and analyse written text for many years. A specialization of text mining, this technique can be used to discoverpatterns in customer complaints, physician notes or news reports, to name a few.

3. **Image Recognition**

One practical application of image recognition is automatic image captioning and scene description. This could be crucial in law enforcement investigations for identifying criminal activityin thousands of photos submitted by bystanders in a crowded area where a crime has occurred. Self-driving cars will also benefit from image recognition through the use of 360- degree camera technology.

4. **Recommendation Systems**

Amazon and Netflix have popularized the notion of a recommendation system with a good chanceof knowing what you might be interested in next, based on past behaviour. Deep learning can be used to enhance recommendations in complex environments such as music interests or clothing preferences across multiple platforms.

Recent advances in deep learning have improved to the point where deep learning outperforms humans in sometasks like classifying objects in images [11]. While deep learning was first theorized in the 1980s, there are twomain reasons it has only recently become useful:

1. Deep learning requires large amounts of labelled data. For example, driverless car development requires millions of images and thousands of hours of video.

2. Deep learning requires substantial computing power.. When combined with clusters or cloud computing, network from weeks to hoursor less. When choosing between machinelearning and deep learning, consider whether you have ahigh-performance GPU and lots of labelled data. If youdon't have either ofthose things, it may makemore sense to use machine learning instead.

CHAPTER – 2
LITERATURE SURVEY

CHAPTER - 2
LITERATURE SURVEY

2.1 Face Recognition Using Histograms Of Oriented Gradients

Face recognition has been a long standing problem in computer vision. Recently, Histograms of Oriented Gradients (HOGs) have proven to be an effective descriptor for object recognition in general and face recognition in particular. In this paper, we investigate a simple but powerful approach to make robust use of HOG features for face recognition. The three main contributions of this work are: First, in order to compensate for errors in facial feature detection dueto occlusions,pose and illumination changes, we propose to extract HOG descriptors from a regular grid. Second,fusion of HOG descriptors at different scales allows to capture important structure for facerecognition. 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 ifHOG features are extracted from overlapping cells. Finally, experimental results on four databasesillustrate the benefits of our approach.

2.2 Face Recognition Using Sift Features

Scale Invariant Feature Transform (SIFT) has shown to be a powerful technique for general object recognition/detection. In this paper, we propose two new approaches: Volume-SIFT (VSIFT) and Partial- Descriptor-SIFT (PDSIFT) for face recognition based on the original SIFT algorithm. We compare holistic approaches: 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.

2.3 Missing Child Identification Using Face Recognition System

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. Face recognition, as one of the primary biometric technologies, became more and more important owingto rapid advances in technologies such as digital cameras, the Internet and mobile devices, and increased demands on security

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 system functions by projecting face image onto a feature space that spans the significant variations among known face images. 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 charactersize 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.

2.4 Very Deep Convolutional Networks for Large-Scale Image Recognition

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3x3) convolution filters, which shows thata significant improvement on the prior-art configurations can be achieved by pushing the depth to16-19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracksrespectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing Convnet models publiclyavailable to facilitate further research on the use of deep visual representations in computer vision.

2.5 Deep Face Recognition

Deep learning applies multiple processing layers to learn representations of data with multiple levels of feature extraction. This emerging technique has reshaped the research landscape of face recognition (FR) since 2014, launched by the breakthroughs of Deepface and Deepid.

CHAPTER – 3

SYSTEM ANALYSIS

CHAPTER - 3
SYSTEM ANALYSIS

3.1 Existing System

For addressing missing children in India relies heavily on police reports, lacking a centralizedand technologically-aided approach. Challenges arise due to regional displacements, making it difficult to link found children with reported missing cases. Aging effects on facial features pose aconsiderable hurdle in accurately identifying missing children using conventional face recognitionsystems. Public involvement in photo uploads is limited, hindering efficient and widespread child tracing efforts.

3.1.1 Disadvantages

1.Reliance on police reports' accuracy.

2.Lack of centralized approach.

3.Difficulty in linking found children.

4.Limitations in public involvement.

5.Problem for accurate identification.

6.Conventional face recognition limitations.

7.Limited efficient tracing efforts.

3.2 Proposed System

The proposed system establishes a deep learning-based framework for identifying missing children in India using facial recognition technology. A public portal allows uploading photos of suspicious children which are automatically compared with registered missing children's images. Utilizing Convolutional Neural Networks (CNN), specifically designed as a high-level feature extractor, the system classifies and matches facial images to the missing child database. Extensive testing of established models like RESNET 50 and VGG 16, alongs idea custom CNN, ensures enhanced prediction accuracy for efficient implementation in locating.

3.2.1 Advantages

1. Enhanced Missing Child Identification

2. Public Participation

3. Automated Image Comparison

4. Efficient Database Matching

5. Deep Learning Model Implementation

6. Use of Convolutional Neural Networks (CNN)

7. Improved Facial Extraction

8. Higher Prediction Accuracy

3.3 Functional Requirements

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior betweenoutputs and inputs.[1]

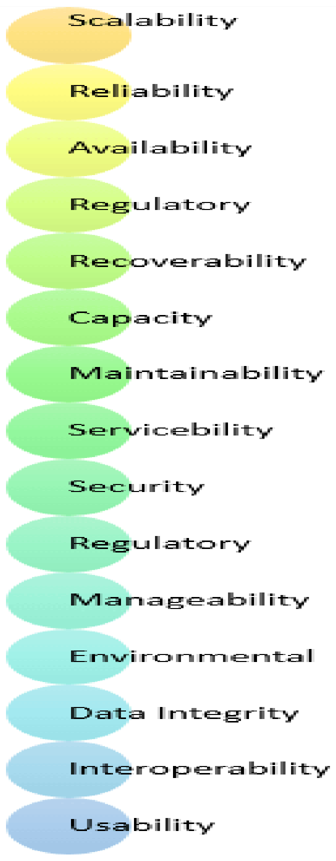
Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish.[2]Behavioral requirements describe all the cases where the system uses the functional requirements,these are captured in use cases. Functional requirements are supported by non-functional requirements (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).Generally, functional requirements are expressed in the form "system must do <requirement>," while non- functional requirements take the form "system shall be <requirement>." The plan for implementing functionalrequirements is detailed in the system design, whereas *non-functional* requirements are detailed inthe system architecture.[4

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements, which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture.

3.4 Non-Functional Requirements

(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 failto 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.



3.4.1 Advantages of Non-Functional Requirement

Benefits/pros of Non-functional testing are:

- The nonfunctional requirements ensure the software system follow legal and compliance.
- They ensure the reliability, availability, and performance of the software system
- They ensure good user experience and ease of operating the software.
- They help in formulating security policy of the software system.

3.4.2 Disadvantages of Non-functional requirement

Cons/drawbacks of Non-function requirement are:

- None functional requirement may affect the various high-level software subsystem
- They require special consideration during the software architecture/high-level design.
- Their implementation does not usually map to the specific software sub-system,It is tough to modify non-functional once you pass the architecture phase.

3.4.3 Key Learning

- A non-functional requirement defines the performance attribute of a software system.
- Types of Non-functional requirement are Scalability Capacity, Availability, Reliability.
- Example of Non Functional Requirement is Employees never allowed to update their salary.
- Functional Requirement is a verb while Non-Functional Requirement is an attribute.
- The advantage of Non-functional requirement is that it helps you to ensure good user.
- The biggest disadvantage of Non-functional requirement is that it may affect.

3.5 System Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is nota 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

3.5.1 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 ofthe system is limited. The expenditures must be justified. Thus the developed system as well withinthe budget and thiswas achieved because most of the technologies used are freely available. Onlythe customized products had to be purchased.

3.5.2 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.Thiswill lead to high demands on the available technical resources. This will lead to high demandsbeing placed on the client. The developed system must have a modest requirement, as only minimalor null changes are required for implementing this system.

3.5.3 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 onthemethods that are employed to educate the user about the system and to make him familiar withit. Hislevel of confidence must be raised so that he is also able to make some constructive criticism,which iswelcomed, as he is the final user of the system.

3.6 Requirement Specification

3.6.1 Hardware Requirements

- System : Pentium i3/i5.
- Hard Disk : 500 GB.
- Monitor : 15’’ LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB

3.6.2 Software Requirements

- Operating system : Windows 8/10.
- Coding Language : PYTHON

CHAPTER – 4
SYSTEM DESIGN

CHAPTER - 4
SYSTEM DESIGN

4.1 Module System

4.1.1 Data Collection and Database Setup

Collected dataset of missing children images with associated metadata (names, contact info, etc.).

Preprocess the dataset by resizing images and converting them to compatible formats. Setup a database (MySQL) to store information about missing children and uploaded images.

4.1.2 Web Interface Development

Develop web pages for user interaction using HTML, CSS, and Django templates.

Implement functionalities for user registration, login, image upload, and viewing uploadedimages.

4.1.3 Model Training and Evaluation

Pre-process the dataset and split it into training and validation sets.

Train the deep learning models (VGG16, ResNet50) using the prepared dataset.

Develop a custom CNN model for image classification specific to missing child identification.

Evaluate model performance using accuracy, loss, and other relevant metrics.

Save the trained models and their weights for future use.

4.1.4 VGG16

VGG16 deep learning model implemented in the project is utilized for image recognition tasks. It employs a pre-trained VGG16 model with additional layers for fine-tuning on a custom dataset. The model is loaded with pre-trained weights and customized by adding convolutional and poolinglayers, followed by dense layers for classification. It's trained using the Adam optimizer with categorical cross-entropy loss. The final accuracy is evaluated through training iterations.

1.Loading pre-trained VGG16 model

2.Defining additional layers for fine-tuning

3.Compiling the model

4.Saving training history for analysis

The process of configuring, training, and saving the VGG16 model for image classification withinthe Django project.

4.1.5 ResNet50

1. **Model Architecture**: ResNet50 comprises residual blocks that allow training of very deep neural networks. It has skip connections which help in avoiding the vanishing gradient problem.

2. **Training**: The model is trained using transfer learning where the pre-trained ResNet50 model is fine-tuned on a custom dataset for specific classification tasks.

3. **Performance**: ResNet50 achieves high accuracy in image classification tasks due to its deep architecture and skip connections.

4. **Usage**: The model is used for tasks like image classification, feature extraction, and object detection within the Django project.

5. **Integration**: The model is integrated into the project through loading the pre-trained weights and defining the necessary layers for the classification task.

The ResNet50 model implemented within the Django project is a convolutional neural network architecture that consists of 50 layers. It's pre-trained on the ImageNet dataset and has demonstrated significant performance in various computer vision tasks.

4.1.6 Model Training and Evaluation

Pre-process the dataset and split it into training and validation sets.

Train the deep learning models (VGG16, ResNet50) using the prepared dataset.

Develop a custom CNN model for image classification specific to missing child identification.Evaluate

model performance using accuracy, loss, and other relevant metrics.

Save the trained models and their weights for future use.

4.1.7 Image Classification

Implementing image classification functionality to identify missing children in uploaded

images.Integrate face detection algorithms to improve accuracy and identify faces within images.

Compare uploaded images against the database to determine potential matches.

4.1.8 User Authentication and Authorization

Implement user authentication and authorization mechanisms to secure access to the system.

roles (admin, regular user) and restrict access to sensitive functionalities.

4.1.9 Administrator Tools

Develop administrative features for managing uploaded images, viewing identificationresults, and updating database entries.

4.2 System Architecture:

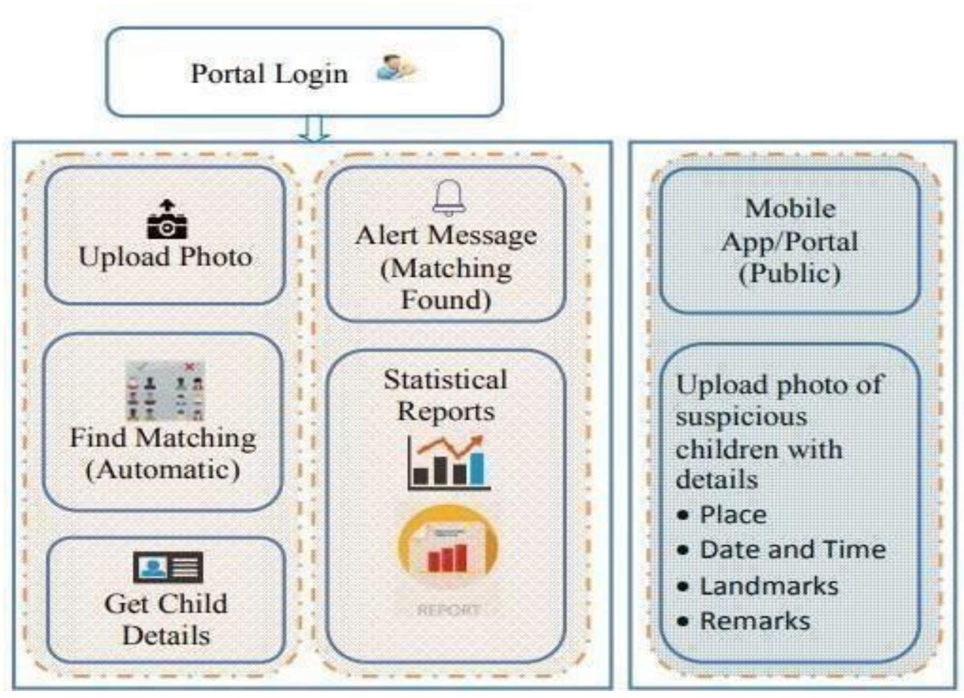

Portal Login
Upload Photo
Find Matching (Automatic)
Get Child Details
Alert Message (Matching Found)
Statistical Reports
REPORT
Mobile App/Portal (Public)
Upload photo of suspicious children with details
• Place
• Date and Time
• Landmarks
• Remarks

Fig1.1:Architecture

4.3 UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; orassociated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelingand other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling oflarge and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals:

The Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they candevelop and exchange meaningful models.

4.1.1 Provide extendibility and specialization mechanisms to extend the core concepts.

4.1.2 Be independent of particular programming languages and development process.

4.1.3 Provide a formal basis for understanding the modeling language.

4.1.4 Encourage the growth of OO tools market.

4.1.5 Support higher level development concepts such as collaborations, frameworks.

4.1.6 Integrate best practice

4.3.1 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), andany dependencies between those use cases.

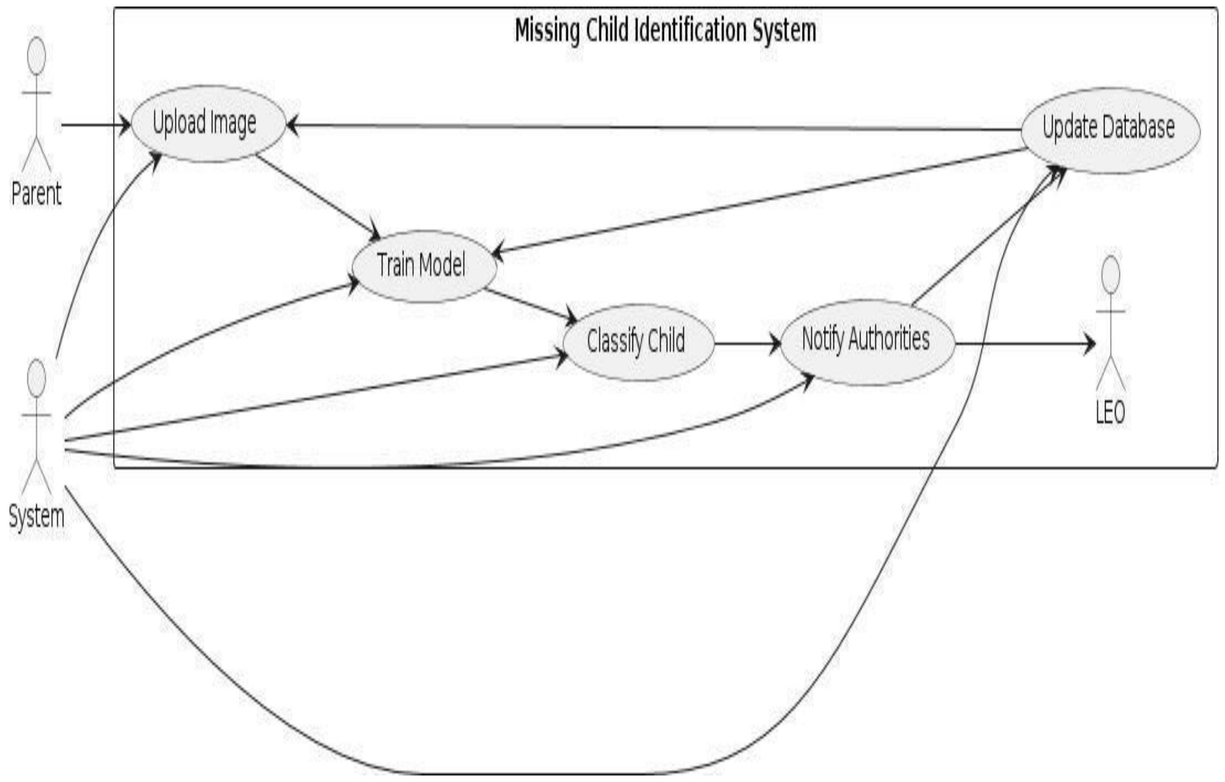


Fig 4.3.1 Use case diagram

4.3.2 Class Diagram

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

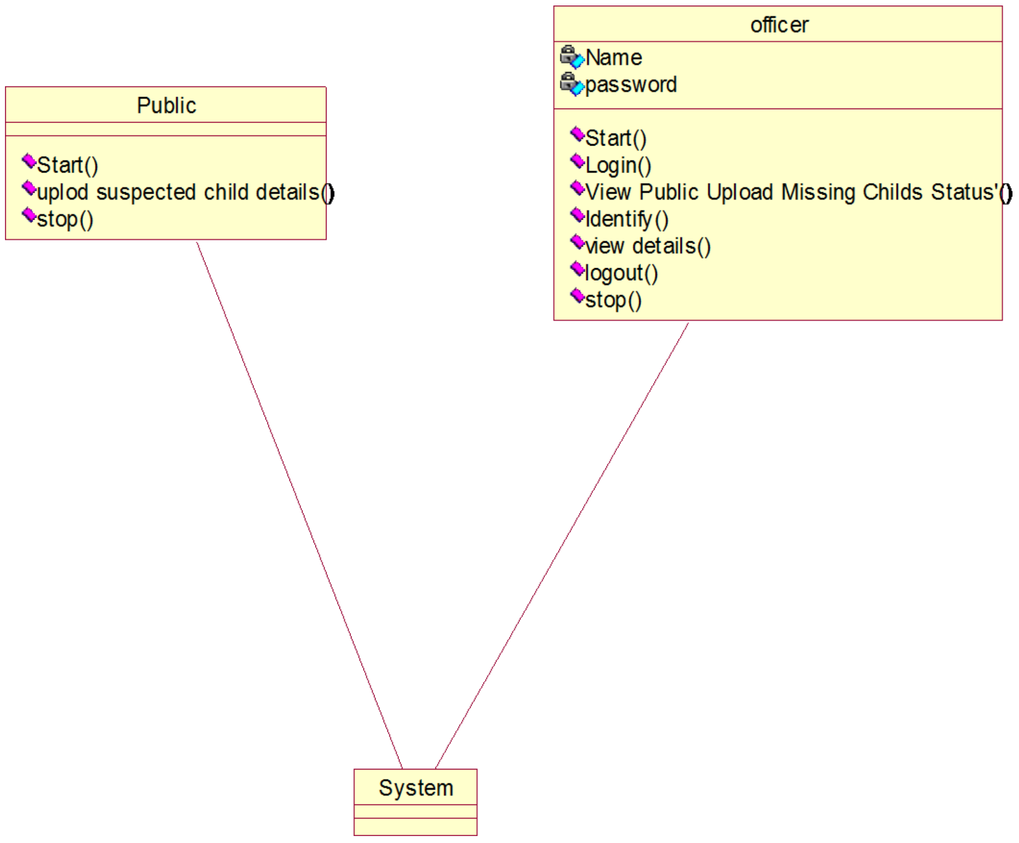


Fig 4.3.2 Class diagram

4.3.3 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

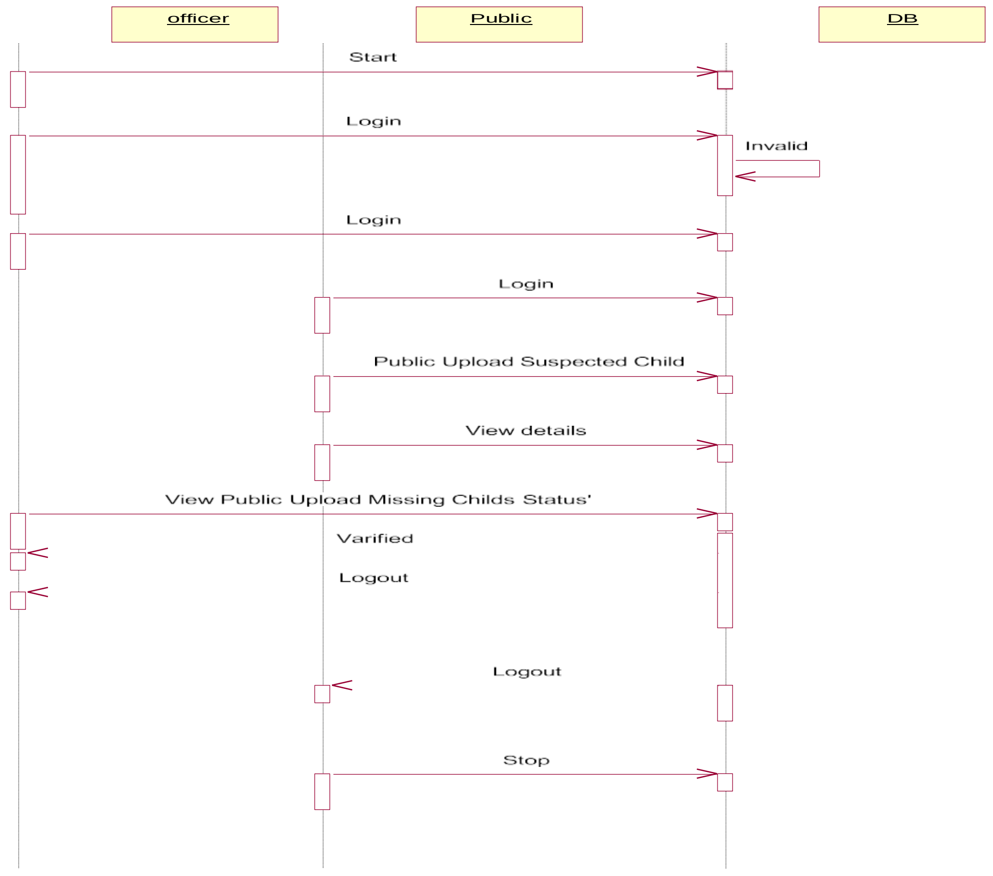


Fig 4.3.3 Sequence diagram

4.3.4 Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of severalfeatures. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

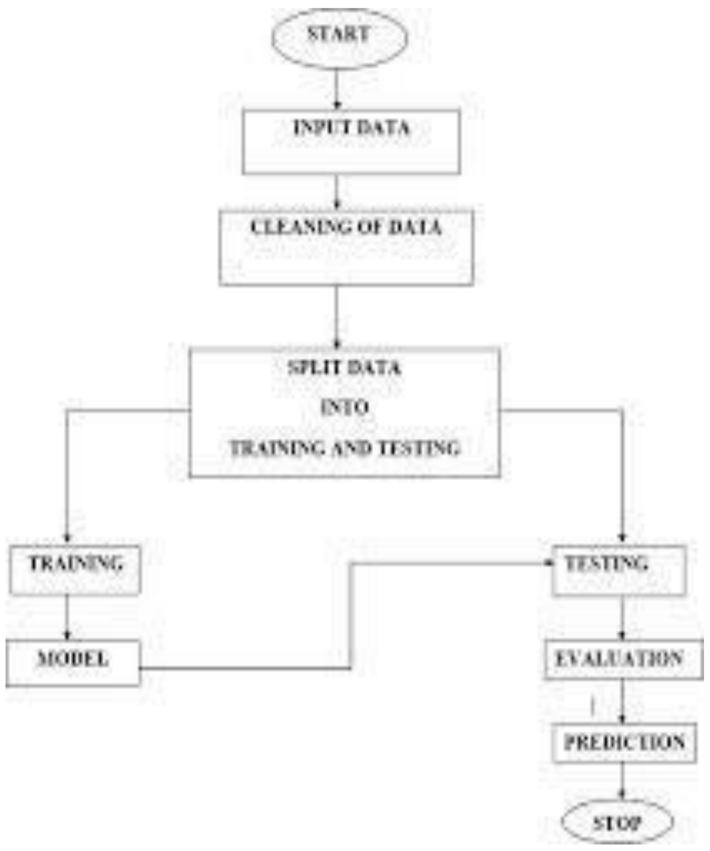


Fig 4.3.4 Collaboration diagram

CHAPTER – 5

TECHNOLOGIES USED

CHAPTER – 5
TECHNOLOGIES USED

5.1 Python Introduction

Python is a general purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature, makes it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object oriented, imperative and functional or procedural programming styles.

Python is not intended to work on special area such as web programming. That is why it is knownas multipurpose because it can be used with web, enterprise, 3D CAD etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

5.1.1 Python History

- Python laid its foundation in the late 1980s.
- The implementation of Python was started in the December 1989 by Guido VanRossum at CWI in Netherland.
- In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.
- In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.

- Python 2.0 added new features like: list comprehensions, garbage collection system.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed torectify fundamental flaw of the language.
- ABC programming language is said to be the predecessor of Python language which wascapable of Exception Handling and interfacing with Amoeba Operating System.
- Python is influenced by following programming languages:
- ABC language.
- Modula-3

5.1.2 Python Features

Python provides lots of features that are listed below.

1.Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

2.Expressive Language

Python language is more expressive means that it is more understandable and readable.

3.Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4.Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc.So, we can say that Python is a portable language.

5.Free and Open Source

Python language is freely available at offical web address.The source-code is also available.

6. Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

7. Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

8. Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

9. GUI Programming Support

Graphical user interfaces can be developed using Python.

10. Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

11. Free and Open Source

Python language is freely available at offical web address.The source-code is also available. Therefore it is open source.

12. Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

13. Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

14. Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid.

15. **Programming Support**

Graphical user interfaces can be developed using Python.

16. **Integrated**

It can be easily integrated with languages like C, C++, JAVA etc.

5.1.3 Python Applications

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.
Here, we are specifing applications areas where python can be applied.

1. **Web Applications**

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautiful Soup, Feed parser etc. It alsoprovides Frameworks such as Django, Pyramid, Flask etc to design and develop web based applications. Some important developments are: Python Wiki Engines, Pocoo, Python Blog Software etc.

2. **Desktop GUI Applications**

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

3. **Software Development**

Python is helpful for software development process. It works as a support language and can be usedfor build control and management, testing etc.

4. **Scientific and Numeric**

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science.

5. Business Applications

Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform.

6. Console Based Application

We can use Python to develop console based applications. For example: IPython.Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

7.3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

8. Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

9. Applications for Images

Using Python several application can be developed for image. Applications developed are:VPython, Gogh, imgSeek etc.

There are several such applications which can be developed using PythonHow to Install Python

(Environment Set-up)

In this section of the tutorial, we will discuss the installation of python on various operating systems.

5.1.4 Why Python

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than someother
- programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it iswritten. This means that prototyping can be very quick.
  Python can be treated in a procedural way, an object-orientated way or a functional way. Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

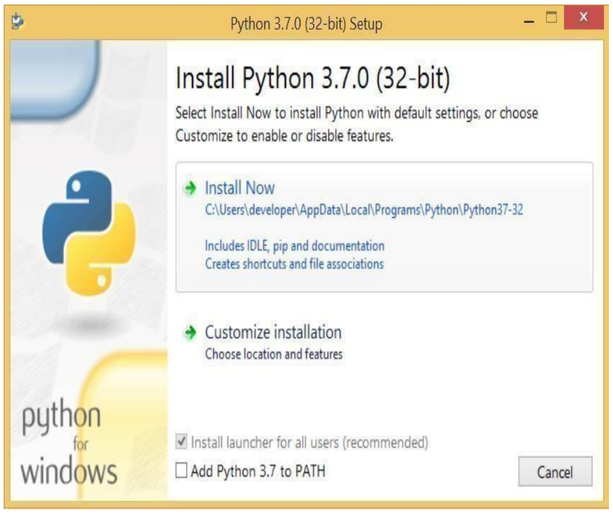
5.1.4 Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languageswhich often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Installation on Windows

Visit the link *https://www.python.org/downloads/* to download the latest release of Python. In this process, we will install Python 3.6.7 on our Windows operating system.

Double-click the executable file which is downloaded; the following window will open. Select Customize installation and proceed



Python 3.7.0 (32-bit) Setup

Install Python 3.7.0 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\developer\AppData\Local\Programs\Python\Python37-32

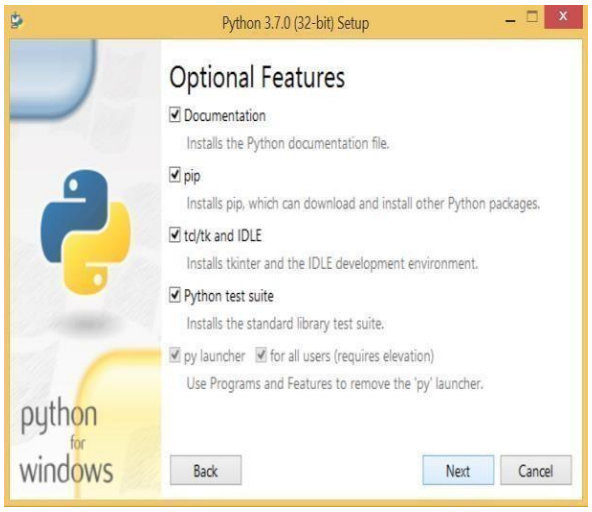
Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☐ Add Python 3.7 to PATH

Fig 5.1.1 Python installation

The following window shows all the optional features. All the features need to be installed and are checked by default; we need to click next to continue.

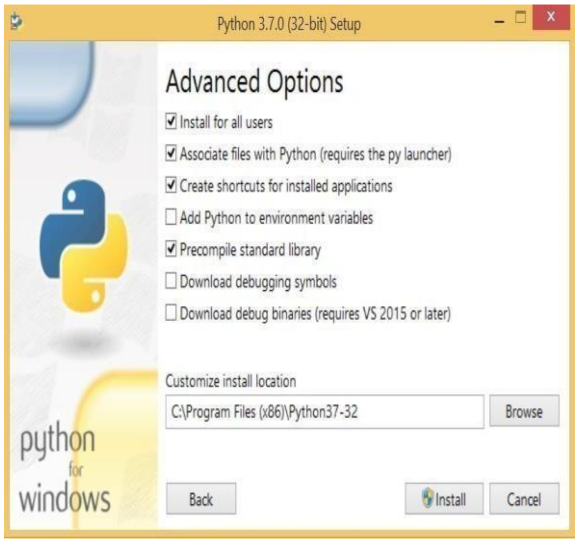


Optional Features

☑ Documentation
Installs the Python documentation file.
☑ pip
Installs pip, which can download and install other Python packages.
☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.
☑ Python test suite
Installs the standard library test suite.
☑ py launcher ☑ for all users (requires elevation)
Use Programs and Features to remove the 'py' launcher.

Fig 5.1.2 Python setup

The following window shows a list of advanced options. Check all the options which you want to install and click next. Here, we must notice that the first check-box (install for all users) must be checked.


Python 3.7.0 (32-bit) Setup

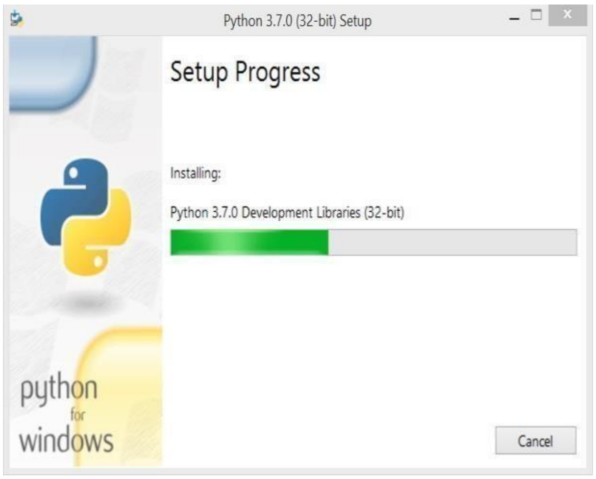
Advanced Options

- [✓] Install for all users
- [✓] Associate files with Python (requires the py launcher)
- [✓] Create shortcuts for installed applications
- [ ] Add Python to environment variables
- [✓] Precompile standard library
- [ ] Download debugging symbols
- [ ] Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Program Files (x86)\Python37-32

Fig 5.1.3 Install Python


Python 3.7.0 (32-bit) Setup

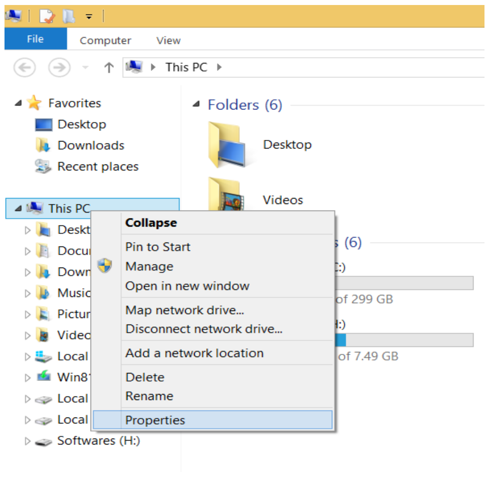
Setup Progress

Installing:

Python 3.7.0 Development Libraries (32-bit)

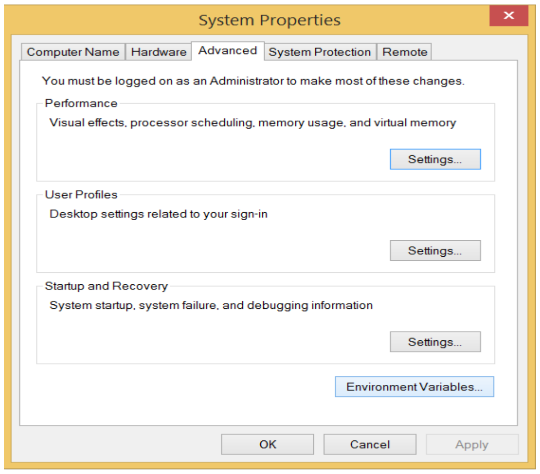
Fig 5.1.4 Run on command prompt

Now, try to run python on the command prompt. Type the command python in case of python2 or python3 in case of python3. It will show an error as given in the below image. It is because we haven't set the path.



To set the path of python, we need to the right click on "my zcomputer" and go to Properties →Advanced → Environment Variables.

Add the new path variable in the user variable section



You must be logged on as an Administrator to make most of these changes.

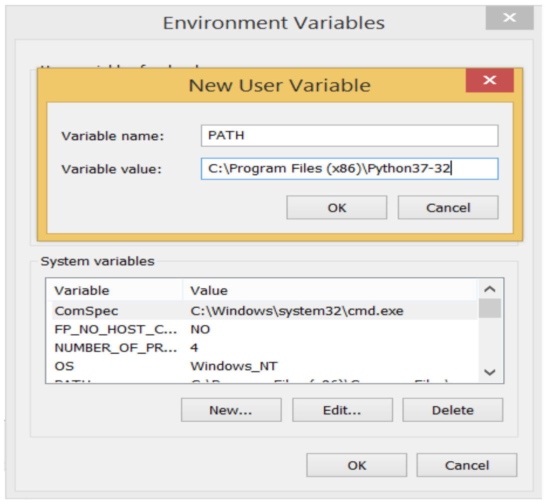
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...

User Profiles
Desktop settings related to your sign-in
Settings...

Startup and Recovery
System startup, system failure, and debugging information
Settings...

Environment Variables...

OK Cancel Apply



Variable name: PATH
Variable value: C:\Program Files (x86)\Python37-32

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_PR...	4
OS	Windows_NT

Type PATH as the variable name and set the path to the installation directory of the python shownin the below image.

Now, the path is set, we are ready to run python on our local system. Restart CMD, and type python again. It will open the python interpreter shell where we can execute the python statements.

CHAPTER – 6
CODING AND IMPLEMENTATION

CHAPTER - 6
CODING AND IMPLEMENTATION

6.1 Source Code

Database.py

```
from django.shortcuts import render
from django.template import RequestContextimport pymysql
from django.http import HttpResponsefrom django.conf import settings from
django.core.files.storage import FileSystemStorageimport datetimeimport numpy as np
from keras.utils.np_utils import to_categoricalfrom keras.layers import MaxPooling2Dfrom
keras.layers import Dense, Dropout, Activation, Flattenfrom keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_jsonimport matplotlib.pyplot as pltimport numpy as np
from keras.models import Sequential
from keras.layers import Dense, Flatten, Activationfrom keras.layers import Dropout
from keras.layers.convolutional import Conv2D, MaxPooling2Dfrom keras.models import
model_from_json
import pickle
from keras import Model from keras import optimizers
from keras.applications.vgg16 import VGG16from keras.optimizers import Adamfrom keras.layers
import Inputfrom keras import applications.
from keras.applications import ResNet50from keras import Model, layersglobal cascPath
global faceCascadedef vgg16():
if os.path.exists('model/vgg_model.json'):
with open('model/vgg_model.json', "r") as json_file:loaded_model_json = json_file.read()model =
model_from_json(loaded_model_json)
model.load_weights("model/vgg_model_weights.h5")model._make_predict_function()
print(model.summary())
f = open('model/vgg_history.pckl', 'rb')data = pickle.load(f)
f.close()
accuracy = data['val_accuracy']acc = accuracy[9] * 100
print("VGG16 model generated with final accuracy as : "+str(acc)+"\n\n")else: X_train =
np.load('model/X.txt.npy')Y_train = np.load('model/Y.txt.npy') print(str(X_train.shape)+"
    "+str(Y_train.shape))input_tensor = Input(shape=(64, 64,3)
vgg_model = applications.VGG16(weights='imagenet', include_top=False,input_shape=(64,64, 3))
#VGG16 transfer learning code here
print(vgg_model.summary())
layer_dict = dict([(layer.name, layer) for layer in vgg_model.layers])x =
layer_dict['block2_pool'].output
x = Conv2D(filters=64, kernel_size=(3, 3), activation='relu')(x)x = MaxPooling2D(pool_size=(2,
2))(x)
x = Flatten()(x)
x = Dense(256, activation='relu')(x)x = Dropout(0.5)(x)x = Dense(2, activation='softmax')(x)
model_final = Model(input=vgg_model.input, output=x)opt = Adam(lr=0.0001)model_final.compile(loss =
'categorical_crossentropy', optimizer = opt,
metrics=["accuracy"])
print(model_final.summary())
cnn_acc = model_final.fit(X_train, Y_train, batch_size=32, epochs=10,validation_split=0.2,
shuffle=True, verbose=2)
model_final.save_weights('model/vgg_model_weights.h5')model_json = model_final.to_json()with
open("model/vgg_model.json", "w") as json_file:json_file.write(model_json)

data = cnn_acc.history #save each epoch accuracy and lossvalues = data['accuracy']acc = values[9] *
100
f = open('model/vgg_history.pckl', 'wb')pickle.dump(data, f)
f.close()
print("VGG16 model generated with final accuracy as : "+str(acc)+"\n\n");def resnet50():
if os.path.exists('model/resnet_model_weights.h5'):
 with open('model/resnet_model.json', "r") as json_file:loaded_model_json = json_file.read()model =
model_from_json(loaded_model_json)
model.load_weights("model/resnet_model_weights.h5")model._make_predict_function()
print(model.summary())
text.insert(END,"model output can bee seen in black console\n\n");f
=open('model/resnet_history.pckl', 'rb')
data = pickle.load(f)f.close()
```

```
accuracy = data['accuracy']acc = accuracy[9] * 100
print("Resnet50 Model generated with final accuracy as : "+str(acc)+"\n")else:
X_train = np.load('model/X.txt.npy')Y_train =
np.load('model/Y.txt.npy')print(X_train.shape)
print(Y_train.shape)
#y_train = np.argmax(y_train, axis=1)
conv_base = ResNet50(include_top=False, weights=None, input_shape=(64, 64, 3))for layer
in
conv_base.layers:
layer.trainable = Falsex = conv_base.output
x = layers.GlobalAveragePooling2D()(x) x = layers.Dense(128, activation='relu')(x)#x
=Flatten()(x)
predictions = layers.Dense(Y_train.shape[1], activation='softmax')(x)rrcnn
=Model(conv_base.input, predictions)
optimizer = keras.optimizers.Adam()
rrcnn.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])hist
=rrcnn.fit(X_train, Y_train, batch_size=16, epochs=10, shuffle=True, verbose=2)
rrcnn.save_weights('model/resnet_model_weights.h5')
model_json = rrcnn.to_json()
 with open("model/resnet_model.json", "w") as
json_file:json_file.write(model_json)f = open('model/resnet_history.pckl',
'wb')pickle.dump(hist.history, f)
f.close()
f = open('model/resnet_history.pckl', 'rb')data =
pickle.load(f)f.close()
acc = data['accuracy'] accuracy = acc[9] * 100
print("Resnet50 model generated with final accuracy as :
"+str(accuracy)+"\n\n")def graph(request):
if request.method == 'GET':
f = open('model/resnet_history.pckl', 'rb')data = pickle.load(f)f.close()
resnet_acc = data['accuracy']resnet_loss = data['loss']
```

```
f = open('model/vgg_history.pckl', 'rb')data =

pickle.load(f)f.close()

vgg_acc = data['accuracy']vgg_loss = data['loss']
f = open('model/history.pckl', 'rb')data = pickle.load(f)f.close()
cnn_acc = data['accuracy']cnn_loss = data['loss']
strdata = '<table border=1 align=center width=100%><tr><th>Algorithm
Name</th><th>Accuracy</th><th>Loss</th></tr><tr>'
 strdata+='<tr><td><font size="" color="black">Resnet 50</td><td><font size=""
color="black">'+str(resnet_acc[9])+'</td><td><font size=""
color="black">'+str(resnet_loss[9])+'</td></tr>'
strdata+='<tr><td><font size="" color="black">VGG 16</td><td><font size=""
color="black">'+str(vgg_acc[9])+'</td><td><font size=""
color="black">'+str(vgg_loss[9])+'</td></tr>'
strdata+='<tr><td><font size="" color="black">CNN</td><td><font size=""
color="black">'+str(cnn_acc[9])+'</td><td><font size=""
color="black">'+str(cnn_loss[9])+'</td></tr></table>'
plt.figure(figsize=(10,6)) plt.grid(True) plt.xlabel('Iterations') plt.ylabel('Accuracy/Loss')
plt.plot(resnet_acc, 'ro-', color = 'green')plt.plot(vgg_acc, 'ro-', color = 'blue') plt.plot(cnn_acc, 'ro-',
color = 'orange')
plt.legend(['Resnet Accuracy', 'VGG 16 Accuracy','CNN Accuracy'], loc='upper left')
#plt.xticks(wordloss.index)
plt.title('Resnet, VGG & CNN Accuracy Graph')plt.show()context= {'data':strdata}

return render(request, 'graph.html', context)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>ChildImage
<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/photo/'+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_classifierglobal 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/photo/'+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 = 1if 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)
```

Explanation:

This Django web application implements functionalities for aiding in the search for missing children:

1. It utilizes Django for web development, integrating machine learning models trained with Keras (VGG16 and ResNet50) for image classification tasks.
2. The models are trained and stored with their weights and architectures.
3. Views are created for user interactions such as logging in, uploading images, and viewing uploaded data.
4. MySQL database is used to store information about missing children and their images.

5. Image processing techniques, including face detection, are applied to uploaded images to aid in classification.
6. Graphs are generated to visualize accuracy and loss trends of trained models.
7. HTML templates are rendered dynamically using context data passed from Django views.

CHAPTER – 7
TESTING AND VALIDATIONS

CHAPTER – 7
TESTING AND VALIDATION

7.1.1 Introduction

In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computerprogram. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

7.1.2 System Testing and Implementation

The purpose is to exercise the different parts of the module code to detect coding errors. After this the modules are gradually integrated into subsystems, which are then integrated themselves too eventually forming the entire system. During integration of module integration testing is performed. The goal of this isto detect designing errors, while focusing the interconnection between modules. After the system was put together, system testing is performed. Here the system is tested against the system requirements to see if allrequirements were met and the system performs as specified by the requirements. Finally accepting testingis performed to demonstrate to the client for the operation of the system.

For the testing to be successful, proper selection of the test case is essential. There are two different approaches for selecting test case. The software or the module to be tested is treated as a black box, and thetest cases are decided based on the specifications of the system or module. For this reason, this form of testing is also called “black box testing”.

The focus here is on testing the external behavior of the system. In structural testing the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. The two forms of testing are complementary: one tests the external behavior, the other tests the internal structure.

Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. Frequently the testing process starts with the test plan. This plan identifies all testing relatedactivities that must be performed and specifies the schedule, allocates the resources, and specifies guidelines for testing. The test plan specifies conditions that should be tested; different units to be tested, and themanner in which the module will be integrated together. Then for different test unit, a test case specificationdocument is produced, which lists all the different test cases, together with the expected outputs, that will be used for testing. During the testing of the unit the specified test cases are executed and the actual resultsare compared with the expected outputs. The final output of the testing phase is the testing report and the error report, or a set of such reports. Each test report contains a set of test cases and the result of executing the code with the test cases. The error report describes the errors encountered and the action taken to removethe error.

7.1.3.1 Testing Techniques

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test casesand the output is evaluated to determine whether the program is performing as expected. In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differingphases of software development are:

7.1.3 Black Box Testing

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

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

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

7.1.4 White Box Testing

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

4.1.6.1.1.1 Guarantee that all independent paths have been executed.

4.1.6.1.1.2 Execute all logical decisions on their true and false sides

4.1.6.1.1.3 Execute all loops at their boundaries and within their operational

4.1.6.1.1.4 Execute internal data structures to ensure their validity.

7.1.5 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done afterthe completion of an individual unit before integration. This is a structural testing, that relies on knowledgeof its construction and is invasive. Unit tests perform basic tests at component level and test a specific businessprocess, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

This System consists of 3 modules. Those are Reputation module, route discovery module, audit module. Each module is taken as unit and tested. Identified errors are corrected and executable unit are obtained.

7.1.6 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.6.1 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configurationto ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven processlinks and integration points.

7.1.7 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.

7.1.7.1 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 beexercised

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacingsystems 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, predefinedprocesses, and successive processes.

7.2 Test Cases:

The performance, reliability, and ethical considerations of the missing child identification system using deep le should cover various aspects of the identification process to ensure the accuracy, reliability, and ethical considerations are met. Here are some example test cases:

1. **Image Recognition Accuracy**:

Test Case 1: Provide the system with a database of images of missing children and evaluate the accuracy of deep learning algorithms in correctly identifying them.

Test Case 2: Introduce variations in lighting conditions, angles, and facial expressions to assessthe robustness of the system in recognizing missing children under different scenarios.

2. **False Positive Rate**:

Test Case 3: Evaluate the system's false positive rate by presenting it with images of individualswho are not missing children and assessing whether it incorrectly identifies them as missing.

Test Case 4: Introduce images of individuals who closely resemble missing children to test the system's ability to differentiate between similar-looking individuals.

3. **Speed and Efficiency**:

Test Case 5: Measure the processing speed of the deep learning algorithms when analyzingimages and data to identify missing children.

Test Case 6: Evaluate the system's efficiency in handling a large volume of data and imageswhile maintaining accurate identification results.

4. **Data Privacy and Security**:

Test Case 7: Ensure that the system complies with data privacy regulations by testing its ability to handle sensitive information securely.

Test Case 8: Evaluate the system's vulnerability to cyber attacks or unauthorized access and assess itssecurity measures to protect sensitive data.

5. **Ethical Considerations**:

Test Case 9: Assess the system's adherence to ethical guidelines regarding consent and data usage, ensuring that it only utilizes data obtained through legal and ethical means.

Test Case 10: Evaluate the system's handling of false positives and negatives, ensuring that it minimizespotential harm to individuals and respects their rights and dignity.

6. **Cross-Dataset Generalization**:

Test Case 11: Test the system's ability to generalize across different datasets of missing children fromvarious demographics, ages, and geographic regions.

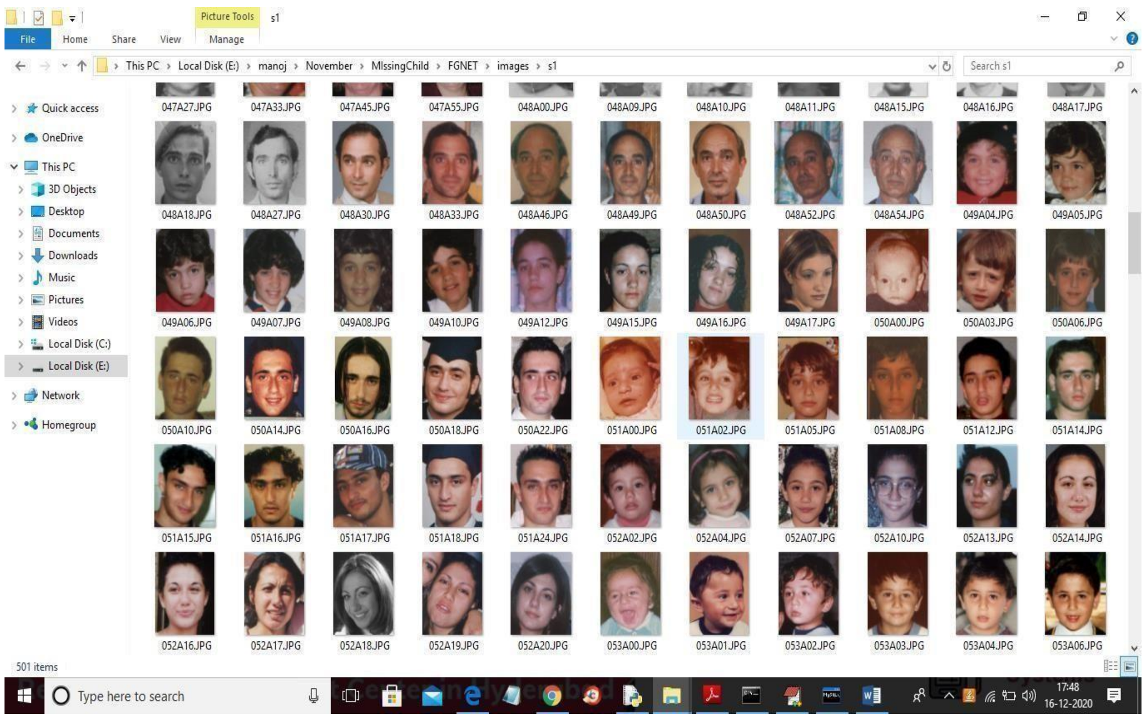
Test Case 12: Evaluate the system's performance when identifying missing children from diverse backgrounds to ensure unbiased and inclusive identification results.

These test cases aim to comprehensively assess learning techniques. Adjustments and additional test cases may be necessary based on specific requirements and context.

CHAPTER – 8
SCREENSHOTS

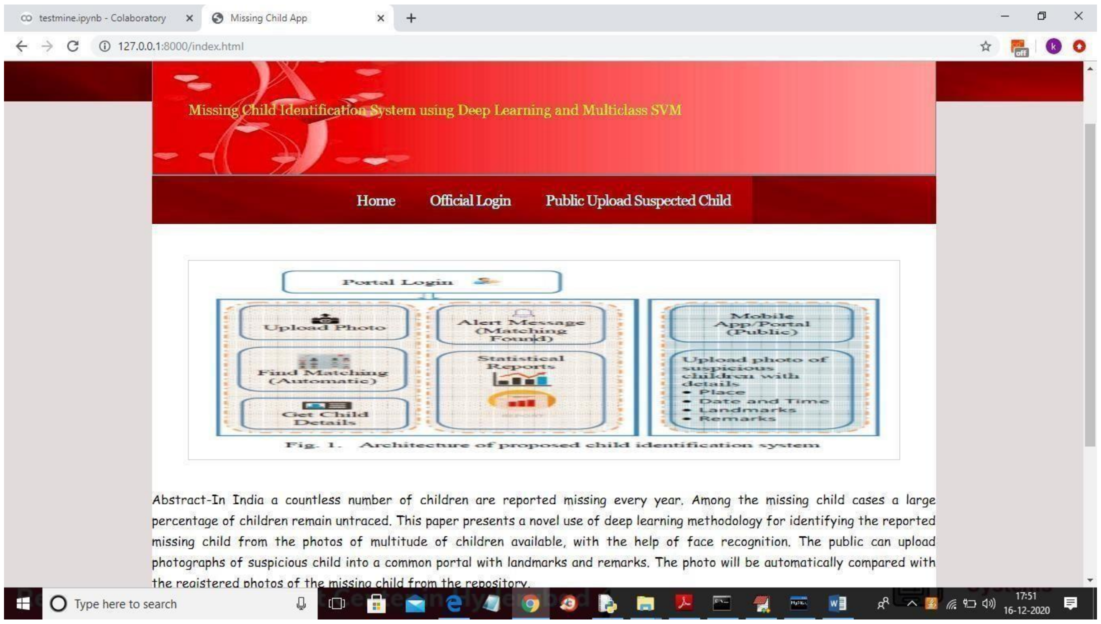
CHAPTER - 8
SCREENSHOTS

8.1 Screenshots


Picture Tools
s1
File
Home
Share
View
Manage
This PC › Local Disk (E:) › manoj › November › MissingChild › FGNET › images › s1
Search s1
Quick access
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Network
Homegroup
047A27.JPG 047A33.JPG 047A45.JPG 047A55.JPG 048A00.JPG 048A09.JPG 048A10.JPG 048A11.JPG 048A15.JPG 048A16.JPG 048A17.JPG
048A18.JPG 048A27.JPG 048A30.JPG 048A33.JPG 048A46.JPG 048A49.JPG 048A50.JPG 048A52.JPG 048A54.JPG 049A04.JPG 049A05.JPG
049A06.JPG 049A07.JPG 049A08.JPG 049A10.JPG 049A12.JPG 049A15.JPG 049A16.JPG 049A17.JPG 050A00.JPG 050A03.JPG 050A06.JPG
050A10.JPG 050A14.JPG 050A16.JPG 050A18.JPG 050A22.JPG 051A00.JPG 051A02.JPG 051A05.JPG 051A08.JPG 051A12.JPG 051A14.JPG
051A15.JPG 051A16.JPG 051A17.JPG 051A18.JPG 051A24.JPG 052A02.JPG 052A04.JPG 052A07.JPG 052A10.JPG 052A13.JPG 052A14.JPG
052A16.JPG 052A17.JPG 052A18.JPG 052A19.JPG 052A20.JPG 053A00.JPG 053A01.JPG 053A02.JPG 053A03.JPG 053A04.JPG 053A06.JPG
501 items
Type here to search
17:48
16-12-2020

First we used below dataset to train deep learning CNN model

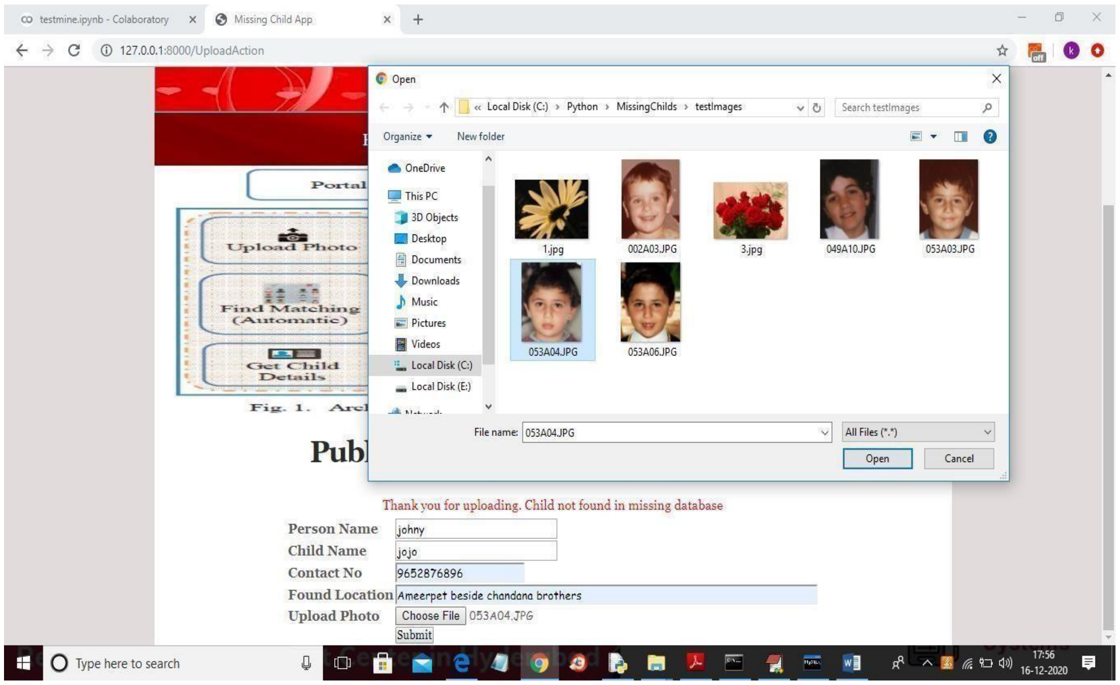
Fig 8.1.1 Dataset Traning


testmine.ipynb - Colaboratory
Missing Child App
127.0.0.1:8000/index.html
Missing Child Identification System using Deep Learning and Multiclass 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
Fig. 1. Architecture of proposed child identification system
Type here to search
17:51
16-12-2020

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.

above screen public can click on 'Public Upload Suspected Child' link to get belowpage and to add missing child details.

Fig 8.1.2 Home Page



File name: 053A04.JPG

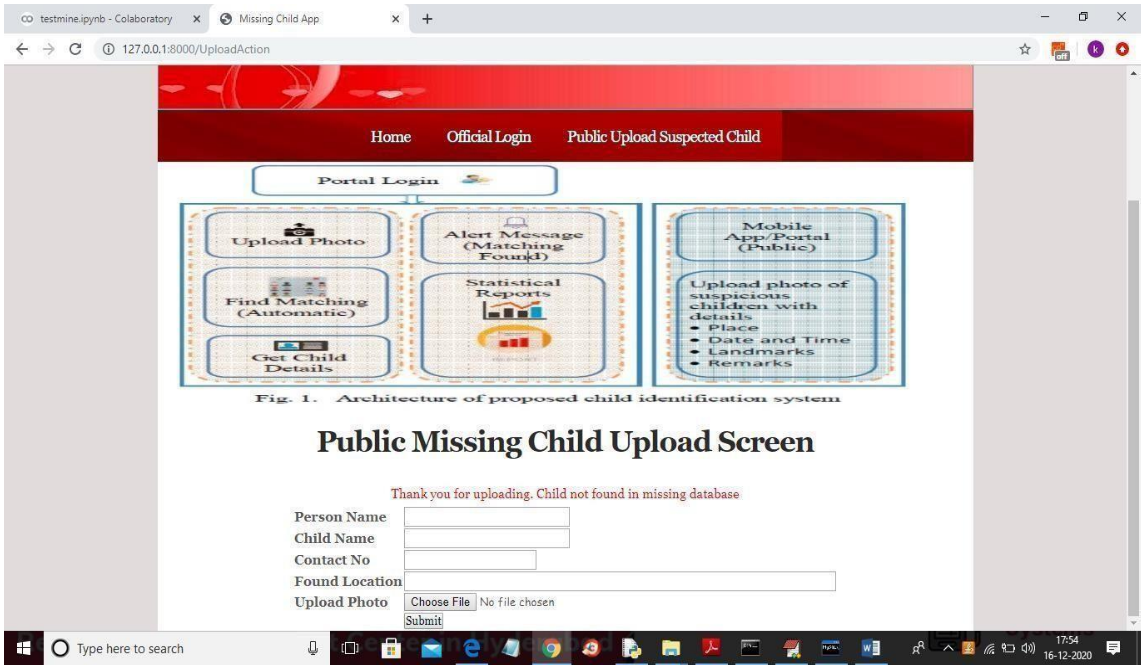
Thank you for uploading. Child not found in missing database

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

Submit

Fig 8.1.3 Public Uploads Photos

In above screen public will enter suspected child details and then upload photo and the click on 'Submit' button and to get below result


Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Find Matching (Automatic)
Get Child Details
Alert Message (Matching Found)
Statistical Reports
Mobile App/Portal (Public)
Upload photo of suspicious children with details
Place
Date and Time
Landmarks
Remarks
Fig. 1. Architecture of proposed child identification system

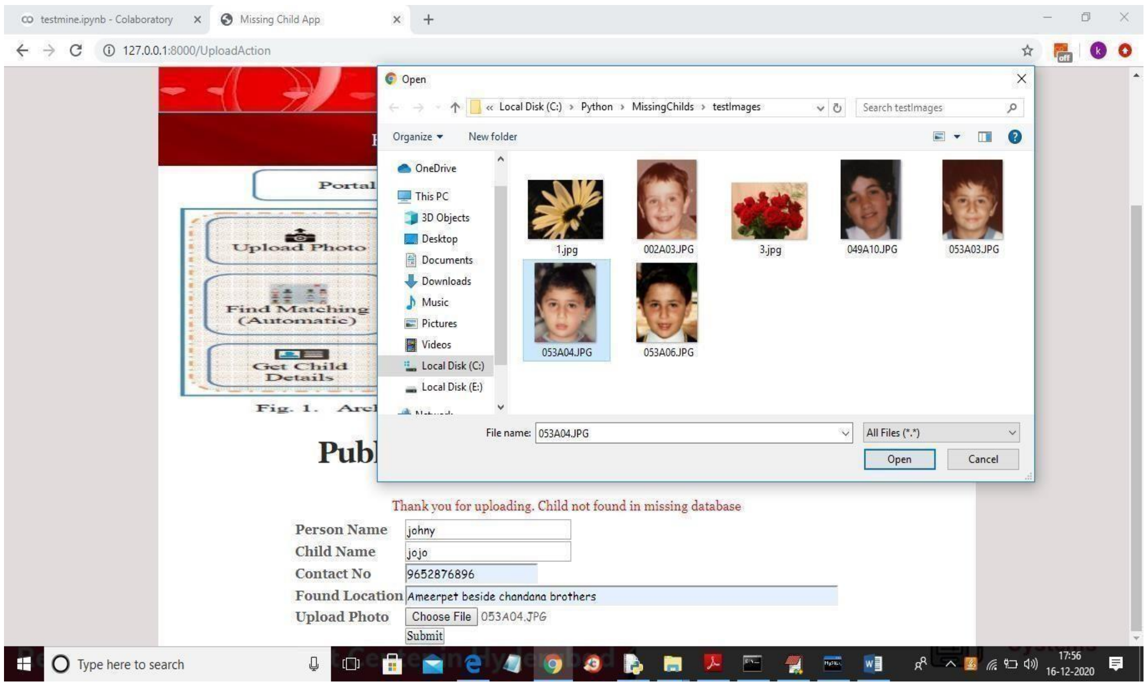
Public Missing Child Upload Screen

Thank you for uploading. Child not found in missing database

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

Submit

Fig 8.1.4 Child Not Found In Missing Database

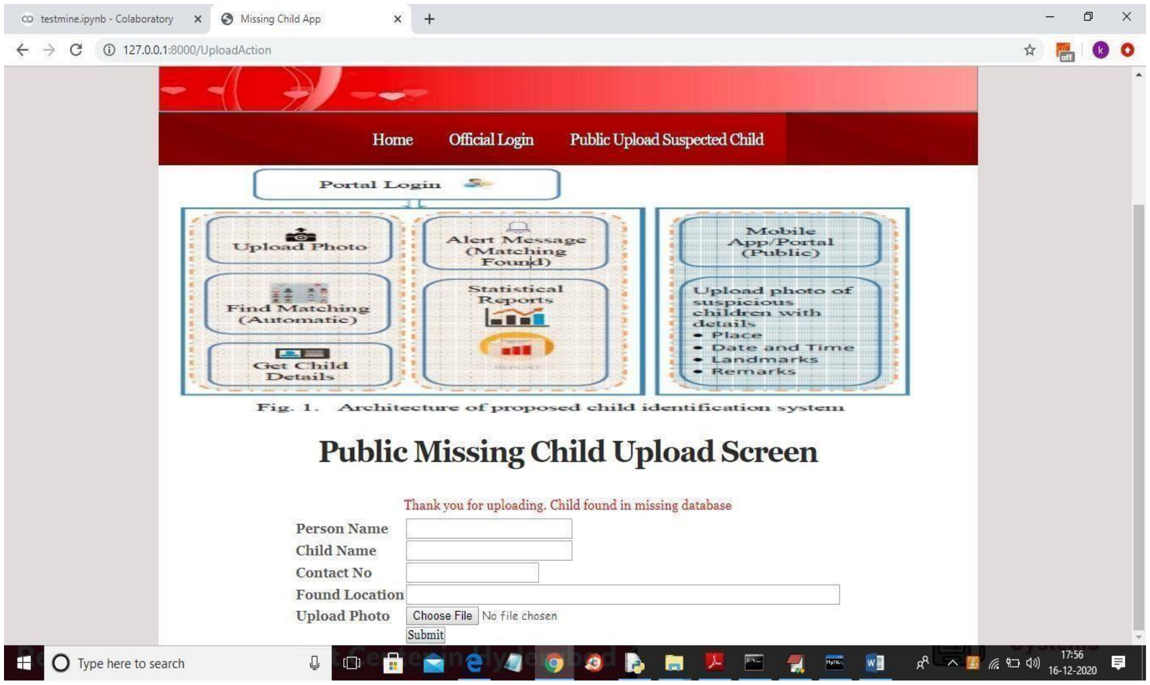


Thank you for uploading. Child not found in missing database

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

Fig 8.1.5 Child Not Found In Missing Database

And below is the result for new above child details


Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Find Matching (Automatic)
Get Child Details
Alert Message (Matching Found)
Statistical Reports
Mobile App/Portal (Public)
Upload photo of suspicious children with details
Place
Date and Time
Landmarks
Remarks
Fig. 1. Architecture of proposed child identification system

Public Missing Child Upload Screen

Thank you for uploading. Child found in missing database

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

In above screen uploaded child found in database and now click on ‘Official Login’ link to getbelow login screen

Fig 8.1.6 Child Found In Missing Database

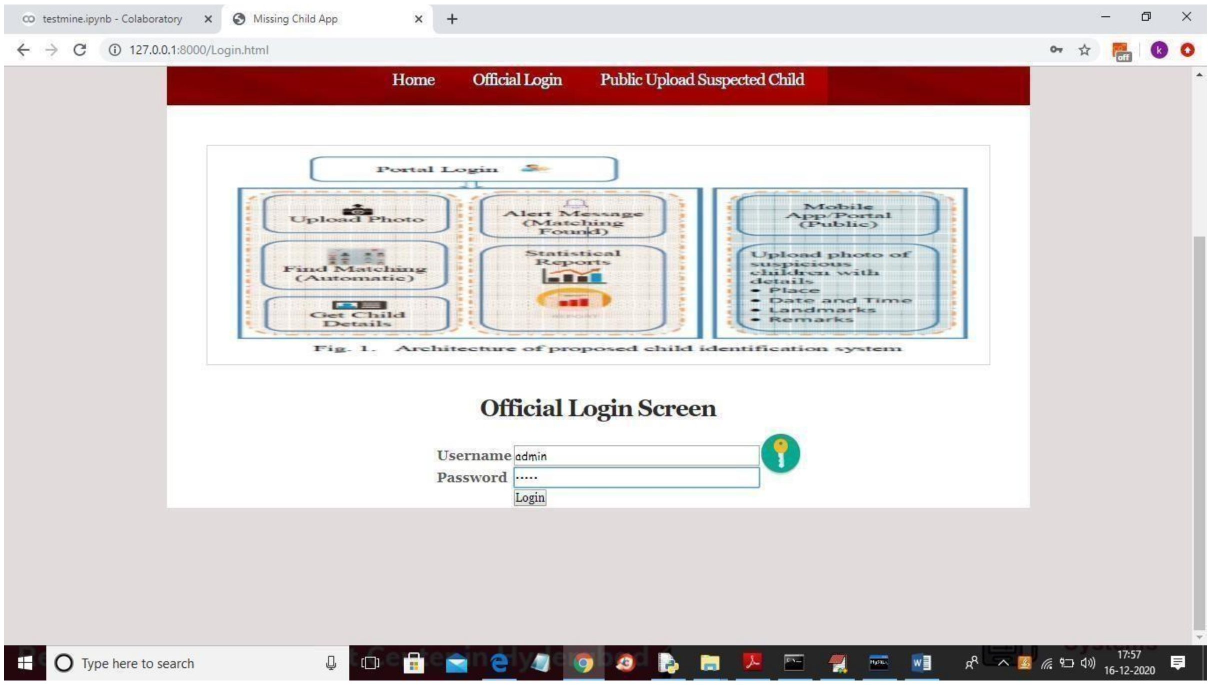
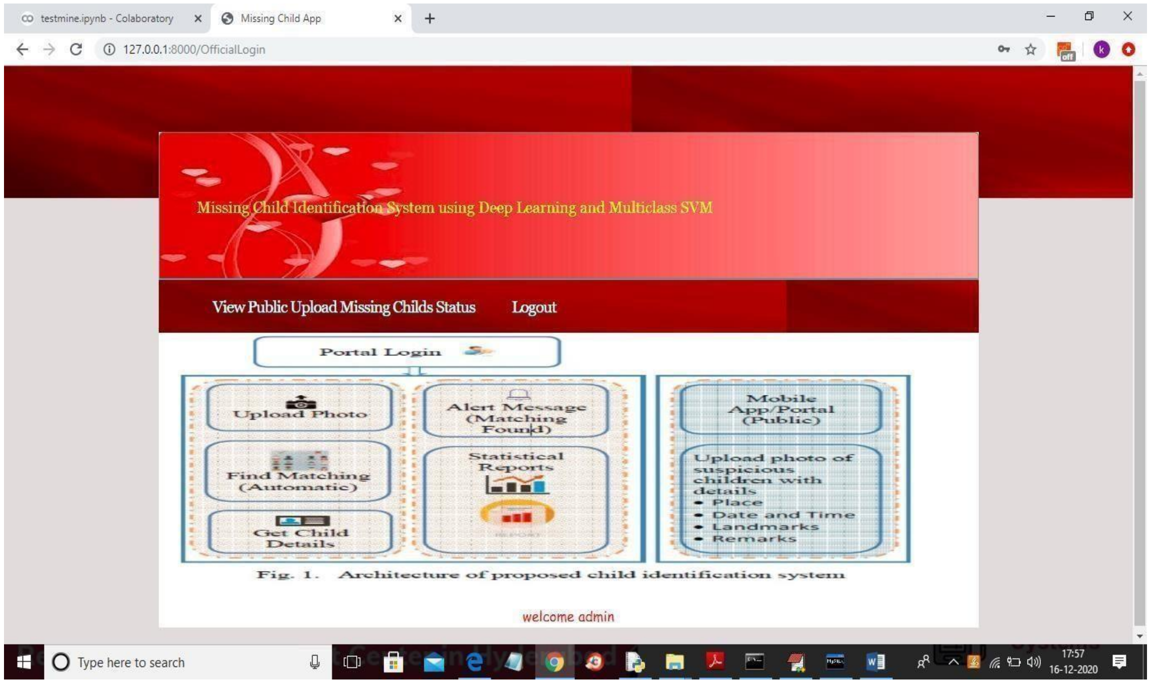
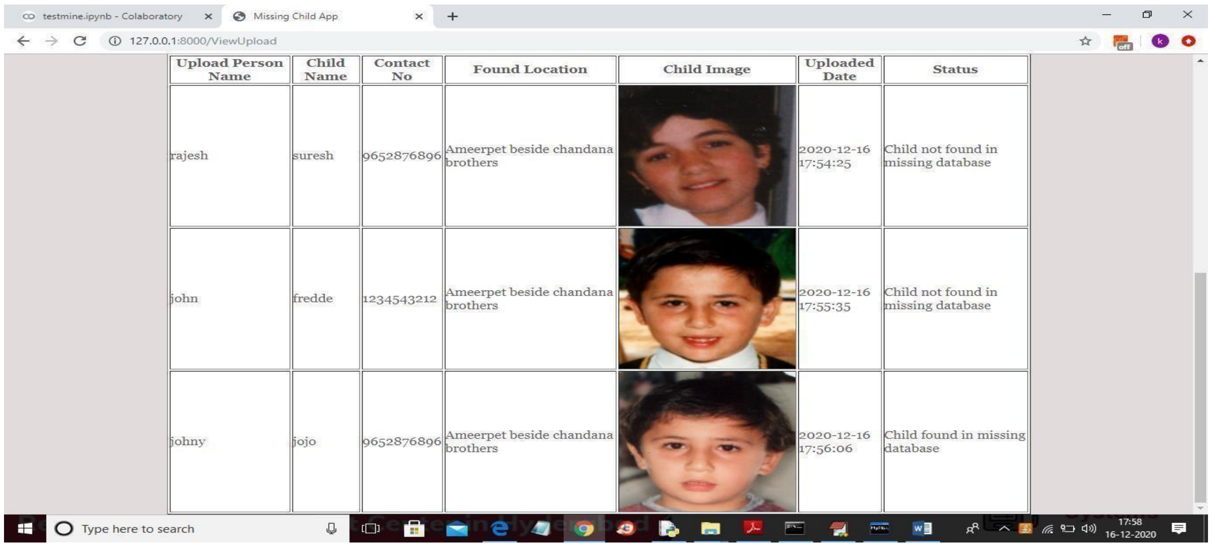

testmine.ipynb - Colaboratory
Missing Child App
127.0.0.1:8000/Login.html
Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Alert Message (Matching Found)
Mobile App/Portal (Public)
Find Matching (Automatic)
Statistical Reports
Upload photo of suspicious children with details
Place
Date and Time
Landmarks
Remarks
Get Child Details
Fig. 1. Architecture of proposed child identification system
Official Login Screen
Username
admin
Password
Login
Type here to search
17:57
16-12-2020

Fig 8.1.7 Offical Login Screen

In above screen admin can login by entering username and password as ‘admin’ and ‘admin’ and after clicking on ‘Login’ button will get below screen


testmine.ipynb - Colaboratory
Missing Child App
127.0.0.1:8000/OfficialLogin
Missing Child Identification System using Deep Learning and Multiclass SVM
View Public Upload Missing Childs Status
Logout
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
Fig. 1. Architecture of proposed child identification system
welcome admin
Type here to search
17:57
16-12-2020

In above screen official can click on ‘View Public Upload Missing Childs Status’ link to view alluploads and its result done by public



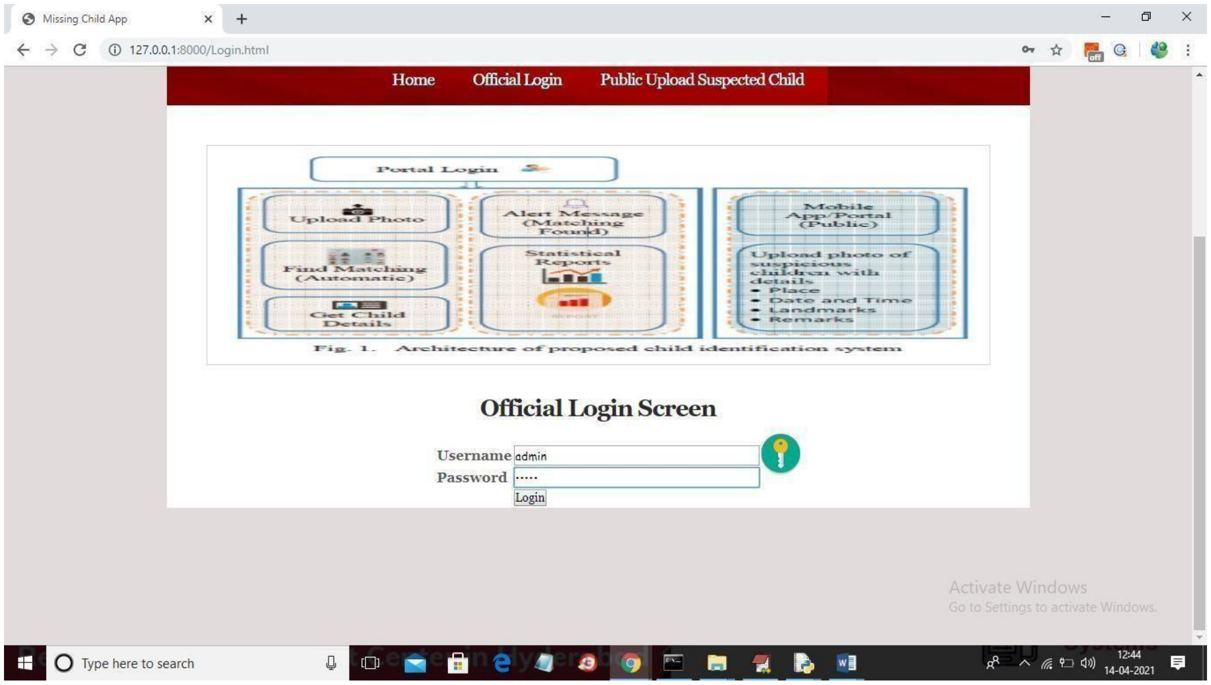
Upload Person Name	Child Name	Contact No	Found Location	Child Image	Uploaded Date	Status
rajesh	suresh	9652876896	Ameerpet beside chandana brothers		2020-12-16 17:54:25	Child not found in missing database
john	fredde	1234543212	Ameerpet beside chandana brothers		2020-12-16 17:55:35	Child not found in missing database
johny	jojo	9652876896	Ameerpet beside chandana brothers		2020-12-16 17:56:06	Child found in missing database

Fig 8.1.8 Screen Can See All Details

In above screen officials can see all details and then take action to find that child

Extension

We have implemented existing RESNET 50 and VGG 16 and compare their accuracy with extension custom CNN algorithm. In all 3 algorithms custom CNN is giving better prediction accuracy and to implement this algorithms.

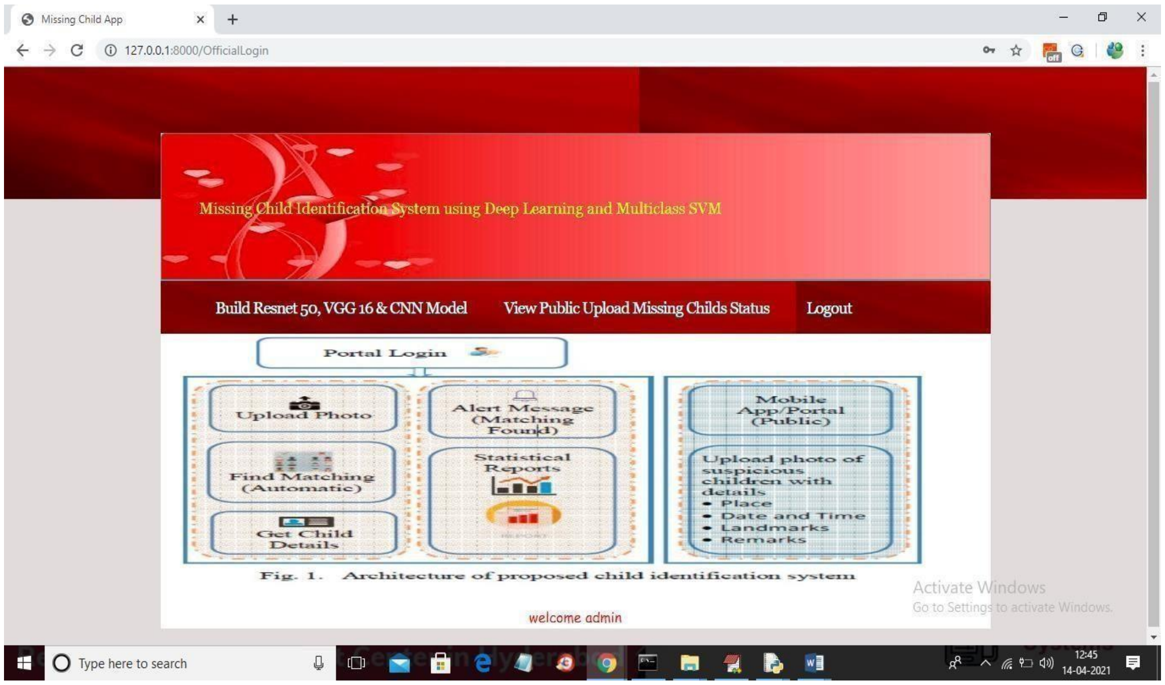

Missing Child App
127.0.0.1:8000/Login.html
Home
Official Login
Public Upload Suspected Child
Portal Login
Upload Photo
Find Matching (Automatic)
Get Child Details
Alert Message (Matching Found)
Statistical Reports
Mobile App/Portal (Public)
Upload photo of suspicious children with details
• Place
• Date and Time
• Landmarks
• Remarks
Fig. 1. Architecture of proposed child identification system
Activate Windows
Go to Settings to activate Windows.
Type here to search
12:44
14-04-2021

Official Login Screen

Username admin
Password •••••
Login

To see comparison graph login as OFFICIAL and then click on ‘Build Resnet 50, VGG 16 & CNN

In above screen login as official and then click on ‘Login’ button to get below screen


Missing Child App
127.0.0.1:8000/OfficialLogin
Missing Child Identification System using Deep Learning and Multiclass SVM
Build Resnet 50, VGG 16 & CNN Model
View Public Upload Missing Childs Status
Logout
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
Fig. 1. Architecture of proposed child identification system
welcome admin

above screen now click on ‘Build Resnet 50, VGG 16 & CNN Model’ link to get below graph
Fig 8.1.9 Click On Resnet,Vgg16,Cnn

Model’link to get below graph

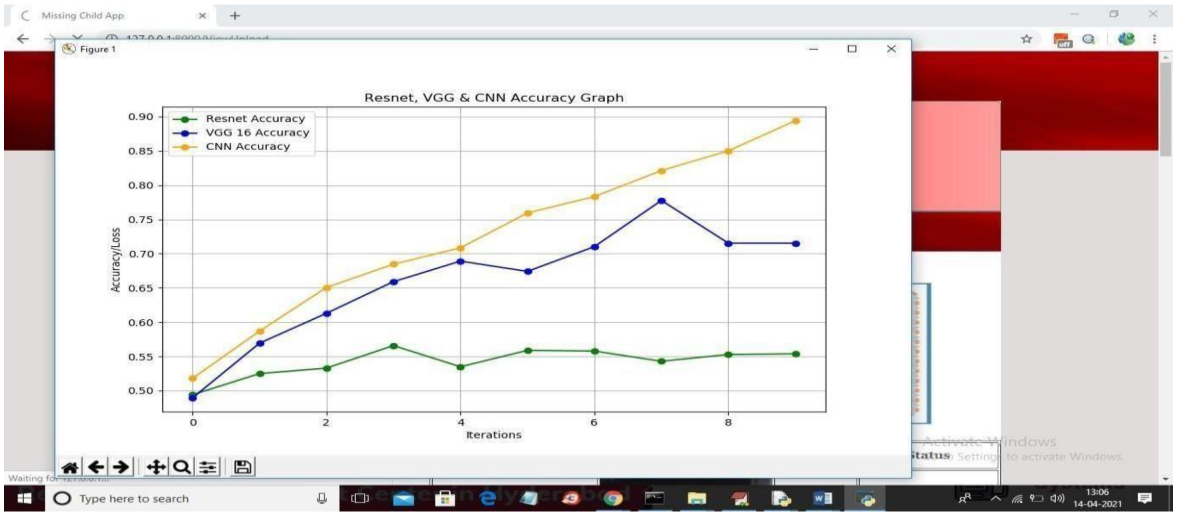
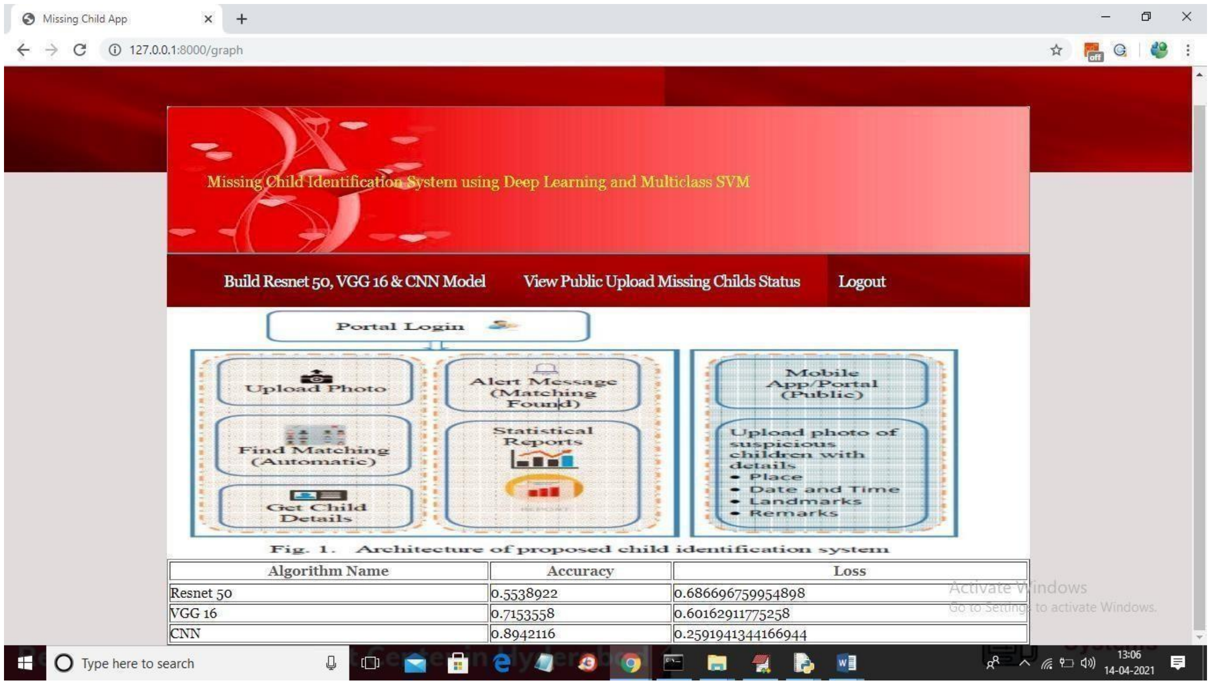


Fig 8.2.0 Graph

In above graph x-axis represents epoch/iteration and y-axis represents accuracy and in above graphwith increasing epoch all algorithms accuracy is getting better and better and in above graph greenline represents RESNET and blue line represents VGG 16 and orange line represents CNN


Missing Child App
127.0.0.1:8000/graph
Missing Child Identification System using Deep Learning and Multiclass SVM
Build Resnet 50, VGG 16 & CNN Model
View Public Upload Missing Childs Status
Logout
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
Fig. 1. Architecture of proposed child identification system
Type here to search
13:06
14-04-2021

Algorithm Name	Accuracy	Loss
Resnet 50	0.5538922	0.686696759954898
VGG 16	0.7153558	0.60162911775258
CNN	0.8942116	0.2591941344166944

accuracy and now close above graph to get below above screen in table we can see final accuracy and loss value of each algorithm and to get better model algorithm accuracy must be

Fig 8.2.1 Algorithms Accuracy

CHAPTER – 9
CONCLUSION AND FUTURE SCOPE

CHAPTER - 9
CONCLUSION AND FUTURE SCOPE

9.1 Conclusion

Missing Child Identification System, employing Deep Learning and Multiclass SVM, offers a robust solution for locating missing children. Leveraging FGNET dataset, our Deep Learning CNNmodel efficiently detects suspected children uploaded by the public, aiding officials in identifyingmissing individuals. Additionally, the Multiclass SVM classifier enhances facial feature extraction,optimizing the accuracy of child identification. Through systematic steps outlined for project implementation andcomprehensive screenshots provided, users can seamlessly navigate and utilize the system. Furthermore, extension efforts integrating existing models like RESNET 50 and VGG16 showcase the superiority of the custom CNN algorithm, ensuring enhanced prediction accuracy.This system stands as a vital tool in reuniting missing children with their families, underscoring the significance of technological innovation in humanitarian efforts.

9.2 Future Scope

Through systematic steps outlined for project implementation and comprehensive screenshots provided, users can seamlessly navigate and utilize the system. Furthermore, extension efforts integrating existing models like RESNET 50 and VGG16 showcase the superiority of the custom CNN algorithm, ensuring enhanced prediction accuracy.This system stands as a vital tool in reuniting missing children with their families, underscoring the significance of technological innovation in humanitarian efforts.

CHAPTER – 10
REFERENCES/WEB REFERENCES

CHAPTER - 10
REFERENCES/WEB 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, 2015. [3] C. Geng and X. Jiang, "Face recognition using sift features", IEEE International Conference on Image Processing(ICIP).

[3] 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 2015.

[4] Kavitha, K., & Samundeswari, S. (2016, June). Missing Children Face Identification Using Deep Learning Algorithm. In *2016 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)* (pp. 27-34). IEEE.

[5] Kamble, V. H., & Dale, M. P. (2016). Machine learning approach for longitudinal face recognition of children. In *Machine learning for biometrics* (pp. 1-27). Academic Press.

[6] Simonyan, Karen and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition", International Conference on Learning Representations (ICLR), April 2017.

[7] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in British Machine Vision Conference, vol. 1, no. 3, pp. 1-12, 2018.

[8] A. Vedaldi, and K.Lenc, "Mat ConvNet: Convolutional Neural Networks for MATLAB", ACM International Conference on Multimedia, Brisbane, October 2019.

[9] Chandran, P. S., Byju, N. B., Deepak, R. U., Nisha kumari, K. N., Devanand, P., & Sasi, P. M. (2018, December). Missing child identification system using deep learning and multiclass SVM. In *2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS)* (pp. 113- 116). IEEE.

[10] Kumar, K. A., Anupama, P., Naveen, P., & Poojitha, S. (2020). Missing Child Identification System using Deep Learning and Multiclass SVM. *Volume XII, Issue II, ISSN*, 0975-4520.

[11] Bhanumathi, P., & Lavanya, B. Missing Child Identification System using Deep Learning braham, N. S., Rajan, R. A., George, R. E., Gopinath, S., & Jeya krishnan, V. (2021). Finding missing child in shopping mall using deep learning. In *Advances in Smart System Technologies: Select*

Proceedings of ICFSST 2019 (pp. 477-482). Springer Singapore.

[12] Ahsan, M., Based, M. A., Haider, J., & Kowalski, M. (2021). An intelligent system for automatic fingerprint identification using feature fusion by Gabor filter and deep learning. *Computers and Electrical Engineering*, *95*, 107387.