DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

(Finding Missing Person Using Face Detection)

By

Md Ehsaan Shaikh – ENG18CS0163
Mohsin Omar Ahmed - ENG18CS0174
Namanpreet Singh – ENG18CS0182
Rishab Rk Shandilya – ENG18CS0228
Tushar Ranjan Jha – ENG18CS0302

Under the supervision of

Dr. Jayita Saha.
Assistant Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India.

CERTIFICATE

This is to certify that the Phase-II project work titled **“FINDING MISSING PERSON USING FACE DETECTION”** is carried out by **Md Ehssaan Shaikh(ENG18CS0163), Mohsin Omar Ahmed (ENG18CS0174),Namanpreet Singh (ENG18CS0182),Rishab RK Shandilya (ENG18CS0228), Tushar Ranjan Jha (ENG18CS0302)** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year 2021-2022.

Prof. Rashmi Mothkur
Assistant Professor
Dept. of CS&E,
School of Engineering
Dayananda Sagar University

Date:

Dr. Girisha G S
Chairman CSE
School of Engineering
Dayananda Sagar
University

Date:

Dr. A Srinivas
Dean
School of
Engineering
Dayananda Sagar
University

Date:

Name of the Examiner **Signature of the Examiner**

1.

2.

DECLARATION

We, **Md Ehsaan Shaikh (ENG18CS0163), Mohsin Omar Ahmed (ENG18CS0174), Namanpreet Singh (ENG18CS0182), Rishab Rk Shandilya (ENG18CS0228), Tushar Ranjan Jha (ENG18CS0302),** are students of the eighth semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **“Sign Language Translator Using Deep Learning”** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**

Student **Signature**

Name: MD Ehsaan Shaikh

USN: ENG18CS0163

Name:Mohsin Omar Ahmed

USN: ENG18CS0174

Name: Namanpreet Singh

USN: ENG18CS0182

Name: Rishab RK Shandilya

USN: ENG18CS0228

Name: Tushar Ranjan Jha

USN: ENG18CS0302

Place : Bangalore Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Dr Jayita Saha**, **Assistant Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinator **Dr. Meenakshi Malhotra** and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work

TABLE OF CONTENTS

Page

LIST OF ABBREVIATIONS...vi

LIST OF FIGURES.. vii

ABSTRACT... viii

CHAPTER 1 INTRODUCTION….. 1

1.1 INTRODUCTION...2

1.2 SCOPE...2

CHAPTER 2 PROBLEM DEFINITION..3

CHAPTER 3 LITERATURE SURVEY...5

CHAPTER 4 PROJECT DESCRIPTION...9

4.1 PROPOSED DESIGN..10

4.2 ASSUMPTIONS AND DEPENDENCIES......................................12

CHAPTER 5 REQUIREMENTS...13

5.1 FUNCTIONAL REQUIREMENTS...14

5.2 NON-FUNCTIONAL REQUIREMENTS.......................................14

5.3 SOFTWARE REQUIREMENTS...15

5.4 HARDWARE REQUIREMENTS...15

CHAPTER 6 METHODOLOGY..16

CHAPTER 7 EXPERIMENTATION..22

CHAPTER 8 TESTING AND RESULTS...24

CHAPTER 9 CONCLUSION.. 36

CHAPTER 10 FUTURE WORK...38

REFERENCES..40

APPENDIX ……………………………………………………………..

LIST OF ABBREVIATIONS

NCRB	The National Crime Records Bureau
MHA	Ministry of Home Affairs
GUI	Graphical Users Interface
CNN	Conventional Neural Network
SVM	Support Vector Machine
KNN	K-nearest neighbors

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
4.1(a)	High Level Design	9
4.1(b)	System Design	9
6.1	Dataset Split	15
6.2	Model Architecture	16
6.2.3(a)	Level 0 Design	18
6.2.3(d)	VGG16 Architecture	19
8.1	Home window	23
8.2	Login Window	23
8.3	Upload Window	24
8.4	VGG16 Model Fitting	25
8.5(a)	CNN Model Fitting	26
8.5(d)	Matching Found	26

ABSTRACT

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 untraceable. 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 2020, and 75,625 of them remained untraceable till the end of the year. This application contains functionality to add complaint as well as view all complaints. By using these complaints, Trust members will try to find lost person in various areas. This application will upload complaint on web server which can be accessed by any of the trust member having this application. This project Finding Missing Person using Face Detection on Android Application presents the solution for this problem.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

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. 392820-032018), 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. The 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

1.2. SCOPE

In the future, we are planning to extend this system further by connecting our system to public cameras and detect faces real-time. The frames will be continuously sent by the public cameras to our system where our system will be continually monitoring the frames. When a lost person is identified in any of the frames, it will be notified to the concerned authorities

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

In the world, a countless number of people are missing every day which includes kids, teens, mentally challenged, old-aged people with Alzheimer's, etc. Most of them remain untraced. Our team aims to help the public and the Police in finding these missing people as quickly and efficiently as possible.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

Nurul Azma Abdullah presents an automated facial recognition system using known Principal Component Analysis approach for criminal database was proposed. That system is able to detect and recognize face automatically. This system will help law enforcement to identify the suspect in case of thumbprint is not in the scene. [1]

Apoorva and Impana discussed an approach to recognize faces using haar classifier and it uses single classifier for recognizing various kind of images with different qualities and rotations. It uses several weak classifiers instead of a complex one. [2]

Kavushica Rasanayagam explains face detection and recognition with deep learning approach based on Convolution Neural Network (CNN) technique. It classifies data by using IMDb dataset with AWS cloud. [3]

Teddy Mantoro reviews eigenface methods to create a face vector or face print by using cascade classifiers and uses Principal Component Analysis (PCA) for training. [4]

Liping Chang describes pattern recognition for classification in feature extraction based on a stacked convolutional auto encoder (SCAE) with deep learning. Further, it combines SRC with Local Binary Projection (LBP) features. [5]

MING Ju-wang exposes a 3D approach for choosing faces by simulating based on 2D images. It can also detect multiple faces dynamically with varying situations. [6]

Mohd Yusuf Firoz Siddiqui explains about training the faces efficiently based on symmetrical and mirror images by reflecting the original face using Principal Component Analysis and some fusion techniques. [7]

Hyung-Il Kim describes about face image assessment of objective visual quality with the improvement in quality of images for training a dataset in a convenient and reliable way. It extracts a way of automatic face recognition and the improvement in face image quality (FIQ) [8]

Piyush Kakkar explains detecting faces from video andnormalize the picture quality that is extracted, recognize and compare the faces. It uses so many datasets from different countries which leads to a bottleneck in performance. [9]

Lamiaa A. Elrefaei, Alaa Alharthi, Huda Alamoudi, and Shatha Almutairi presents a client-server approach for realtime tracking and it uses optical flow networks that is implemented in android lollipop version. It extracts features in an image such as corner features and regular features .It uses Adaptive Mean Algorithm for the camera based surveillance. [10]

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1. PROPOSED DESIGN

The High-Level Design of Our Project is as follows:

- Register/Login
- Image Upload
- Preprocessing of Image for Input
- Recognition of Image
- Display as Result

4.1.1. Register/Login

The User/Admin need to Login to Upload or search for missing person

4.1.2. Pre-processing of Uploaded Image

From the images, the region of the face will be recognized and then cropped. Then suitable processing of the image will be done to make it prepared for recognition.

4.1.3. Display as Result

The recognized Image will then be shown to the user via the screen/interface

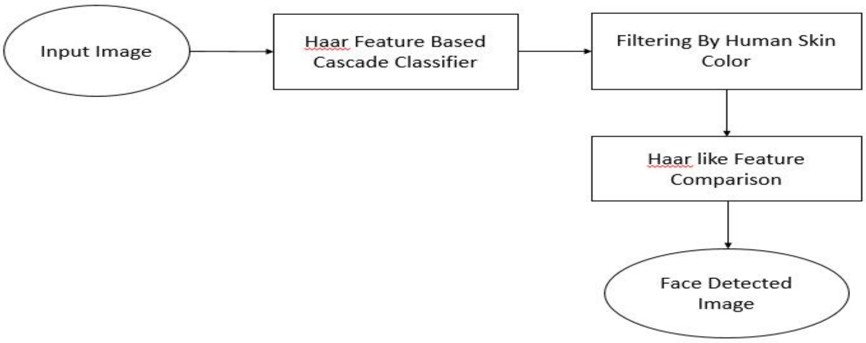

Input Image
Haar Feature Based Cascade Classifier
Filtering By Human Skin Color
Haar like Feature Comparison
Face Detected Image

Figure 4.1 (a) High Level Design

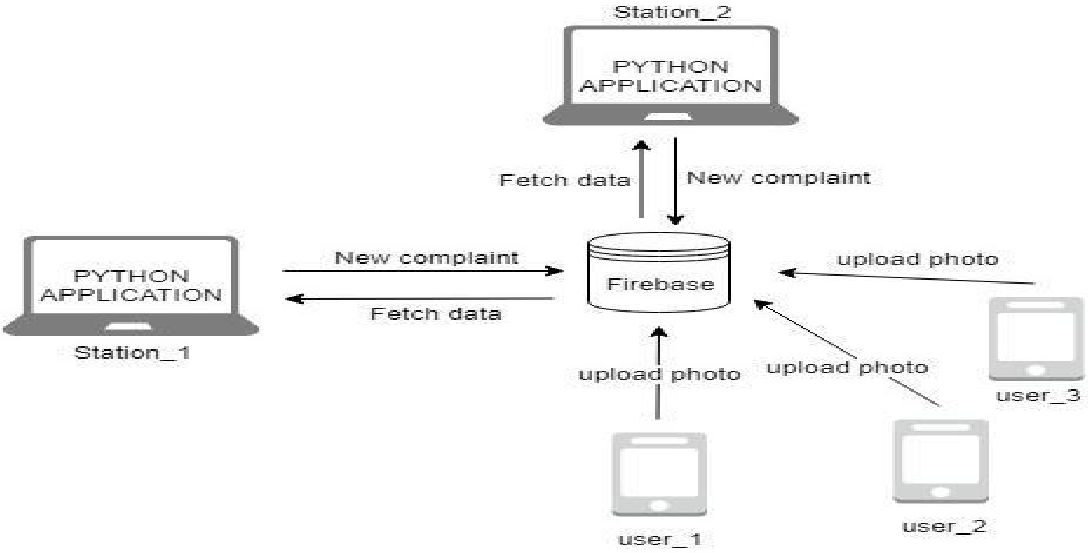


Figure 4.1 (b) System Layout

4.2. ASSUMPTIONS AND DEPENDENCIES

There are few assumptions and dependencies that are to be taken into consideration. The assumptions and dependencies are

1. Sketch:-Sometime a photograph of missing person is not available therefore our project should be working on sketches also.
2. Similar face:- Many person has similar face,so our project should be able to detect with accuracy
3. Detect in Aged-Can detect face from a age difference.

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1. FUNCTIONAL REQUIREMENTS

- Making use of a suitable Deep Learning Architecture, which is effective and efficient in face detection.
- It must be able to recognize a different faces with little or more similarity.

5.2. NON-FUNCTIONAL REQUIREMENTS

- **User friendliness**

  The system should be easy to interact with and any user with no prior computer knowledge should be able to use it easily and effectively.

- **Performance and stability**

  The system should be stable and performance should be smooth with easy accessibility.

- **Reliability, availability, maintainability**

  Should be reliable so proper assessments are done with proper output, should be readily available and should be easy to maintain so future software handlers should also be able to continue to operate and update as and when required.

- **Robustness**

  The system should be able to recover if any problems occur.

- **User satisfaction**

  The system should meet the users' expectations.

5.3. SOFTWARE REQUIREMENTS

1. Operating System: Windows/Ubuntu/MacOS
2. Web Browser- Google Collab and Kaggle Notebooks
3. TensorFlow, Keras, Pandas and Scikit Learn
4. Framework: Vs code, Pycharm

5.4. HARDWARE REQUIREMENTS

1. Processor: Intel Core i3 and above
2. RAM: 4GB and above
3. Internet Connection

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

6.1. DATASET

- The data set is a collection of images of alphabets from the American Sign Language, separated in 29 folders which represent the various classes.
- The training data set contains 87,000 images which are 200x200 pixels. There are 29 classes, of which 26 are for the letters A-Z and 3 classes for SPACE, DELETE and NOTHING.
- The 3 classes are very helpful in real-time applications and classification. Each class has 3000 image samples.
- The test data set contains a mere 29 images, to encourage the use of real-world test images.
- The dataset includes image samples with variations in lighting conditions and skin tones as well to help make the model more robust.

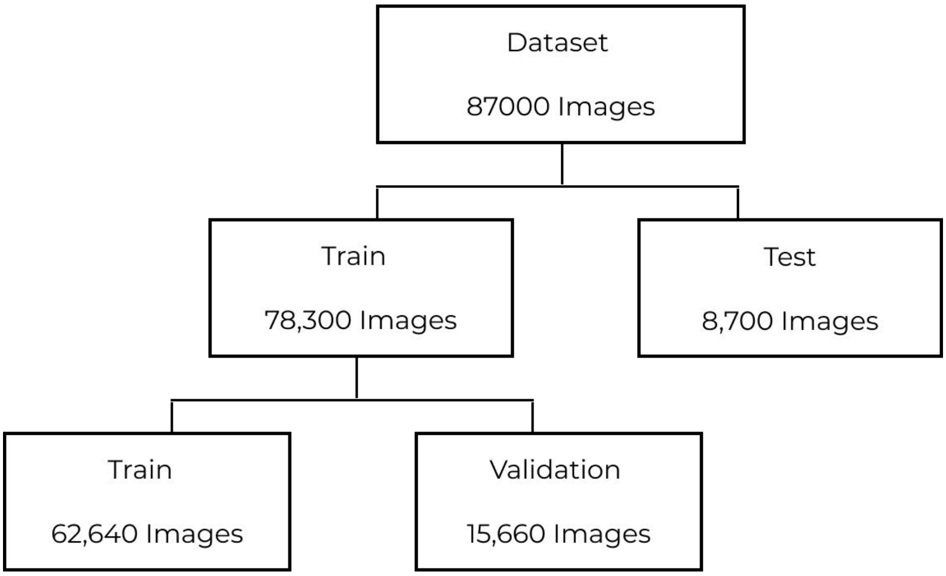


Figure 6.1 Dataset Split

6.2. DEEP LEARNING MODELS

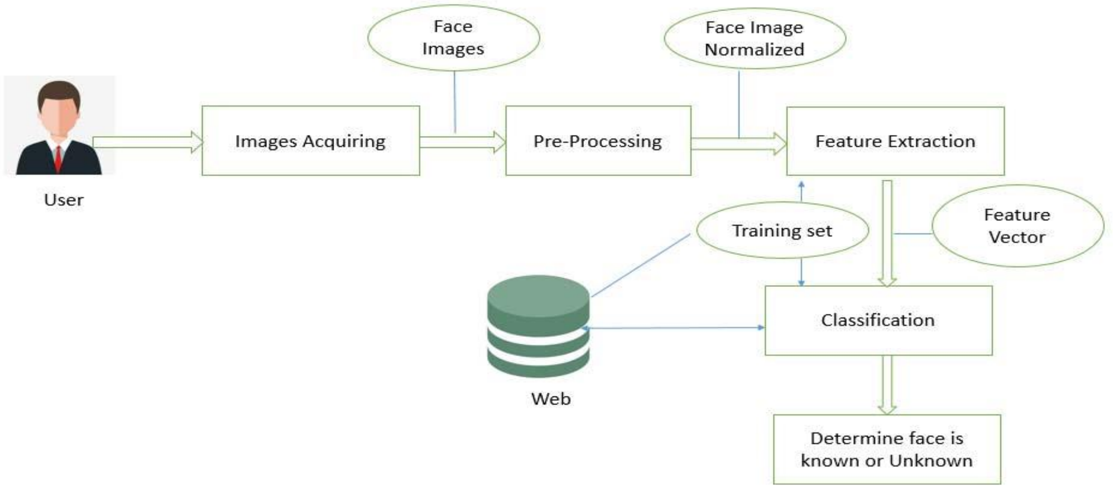


Figure 6.2 Model Architecture

6.2.1. Convolutional Neural Networks

- Convolutional Neural Networks popularly known as CNNs are one of the oldest deep neural networks that are widely used in computer vision tasks.
- They are composed of multiple building blocks such as, convolutional layers, pooling layers and fully connected layers.
- CNNs take in an input image, assign importance to different aspects and features within the image and are able to distinguish such images from others.
- CNNs form the basis of the various deep learning models that are implemented for this application.

6.2.2 Transfer Learning

- Transfer Learning is the reuse of a model on a problem that is different from the original one that it was previously developed and trained for.
- Transfer Learning allows us to use previously gained knowledge to newer problems without having to start from scratch.
- The benefits include shorter training times to develop models that work well in most cases.

6.2.3.1 Face detection

Python opencv is used to detect faces and it usually uses principal component analysis [4] or convolution neural network (CNN) [3] technique. These techniques are used to classify images as face and non-face regions based on features obtained from an image. For such a task the classifier must be trained with positive and negative face images to identify the faces in the given image.

6.2.3.2 Preprocessing

The detected image is cropped and stored in the form of a numpy array. The detected image is preprocessed in order to remove the unwanted features to reduce the amount of computation performed. It provides a 3D view of the face that is stimulated from the face image. It fills up the regions on face automatically by applying deep convolutional network

6.2.3.3 Feature Extraction

It extracts the features of the face such as the color, width and height of the face, color of hair, spectacles, and facial hair and so on by employing deep learning. These features can be represented as an array of numbers which will be unique for all the faces and they are stored in Dataset.

6.2.3.4 VGG16

VGG16 is a simple and widely used Convolutional Neural Network (CNN) Architecture used for ImageNet, a large visual database project used in visual object recognition software research. It won ILSVR(Imagenet) in 2014.

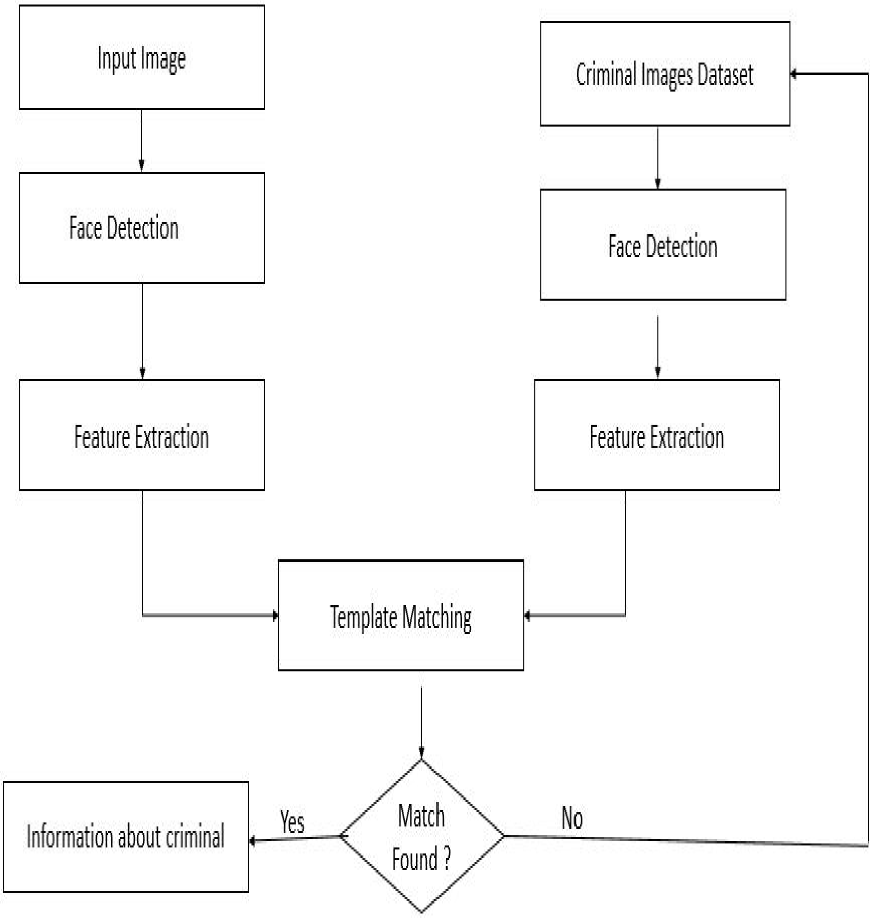


Figure 6.2 Level 0 Design

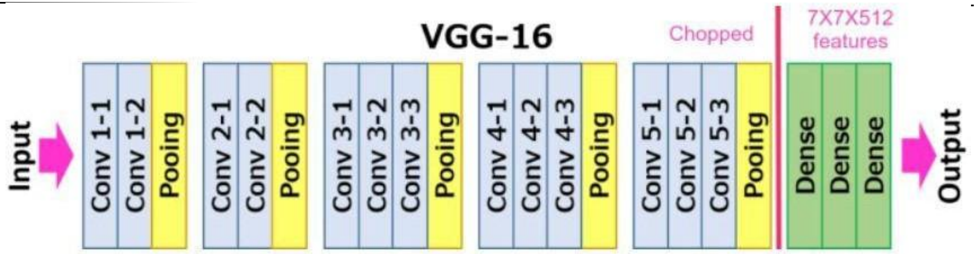


Figure 6.2.3 (d) VGG-16 architecture

The transfer learning models are developed using the above pre-trained models and the weights obtained on training them on the ImageNet dataset. The proposed approach involves removing the existing fully connected layers from the above pre-trained models and replacing them with another set that would cater to this specific application. The base models would be frozen and hence would not be able to update weights. The models are trained on images from the specified dataset.

Fine-tuning these pre-existing models in this fashion allows us to classify images for our specific application.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

- **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 at any time using the proposed systemDataset Split:- Initially since the dataset split was not random. The last few image samples were used for validation (which were taken under darker lighting conditions compared to the rest of the set), which resulted in a very low validation accuracy. Compiling the dataset into a Pandas Dataframe and splitting it with the help of the Scikit Learn ‘train_test_split’ method helped overcome this issue.

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

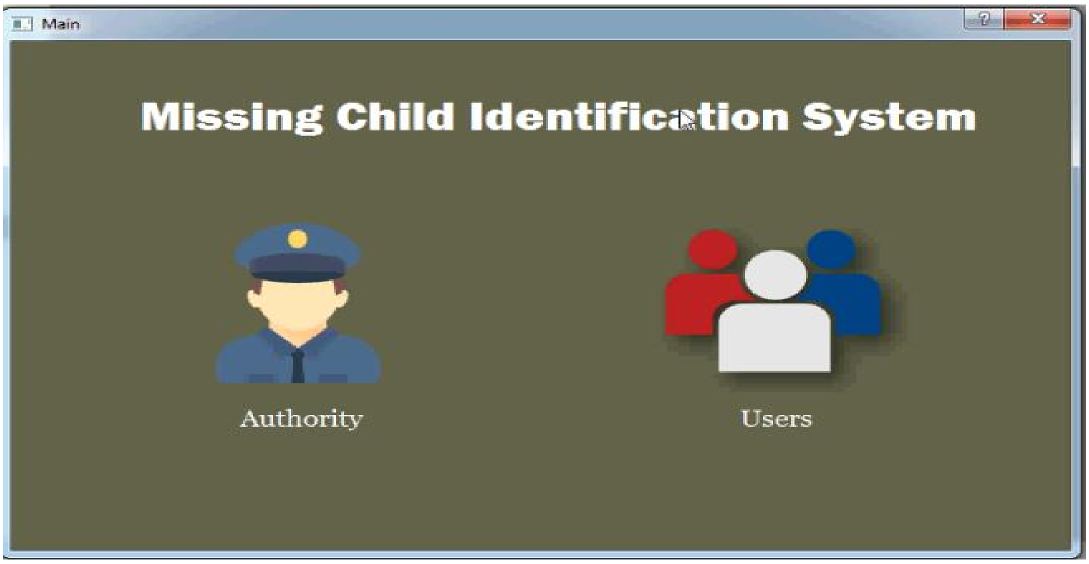


Figure 8.1 Home Window

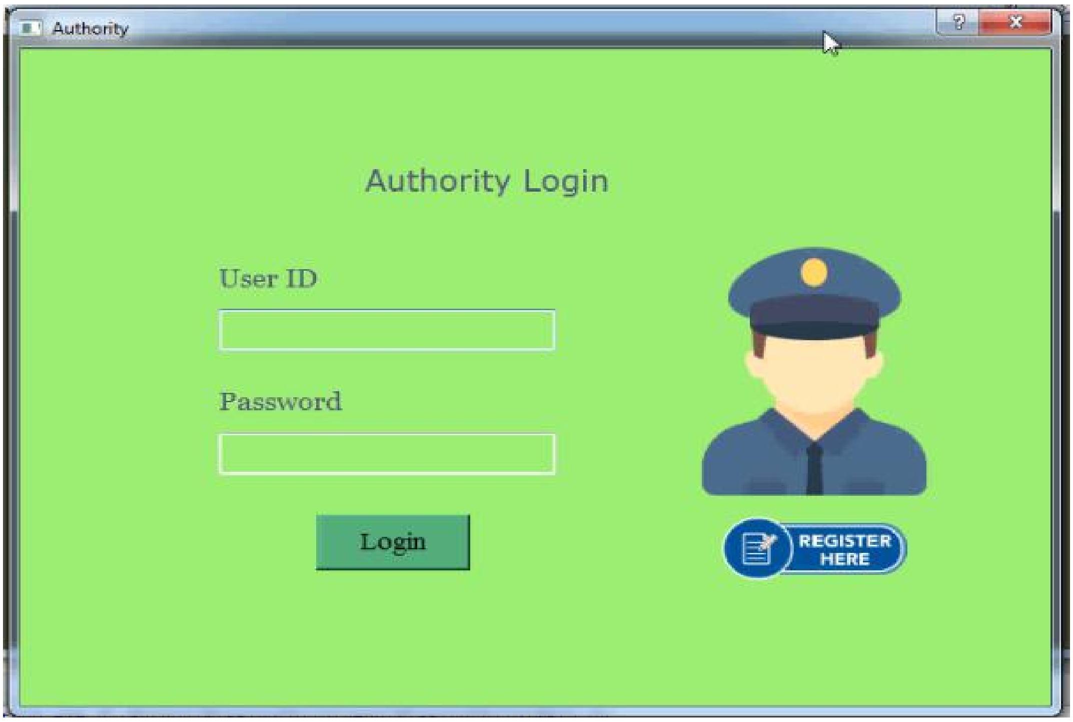
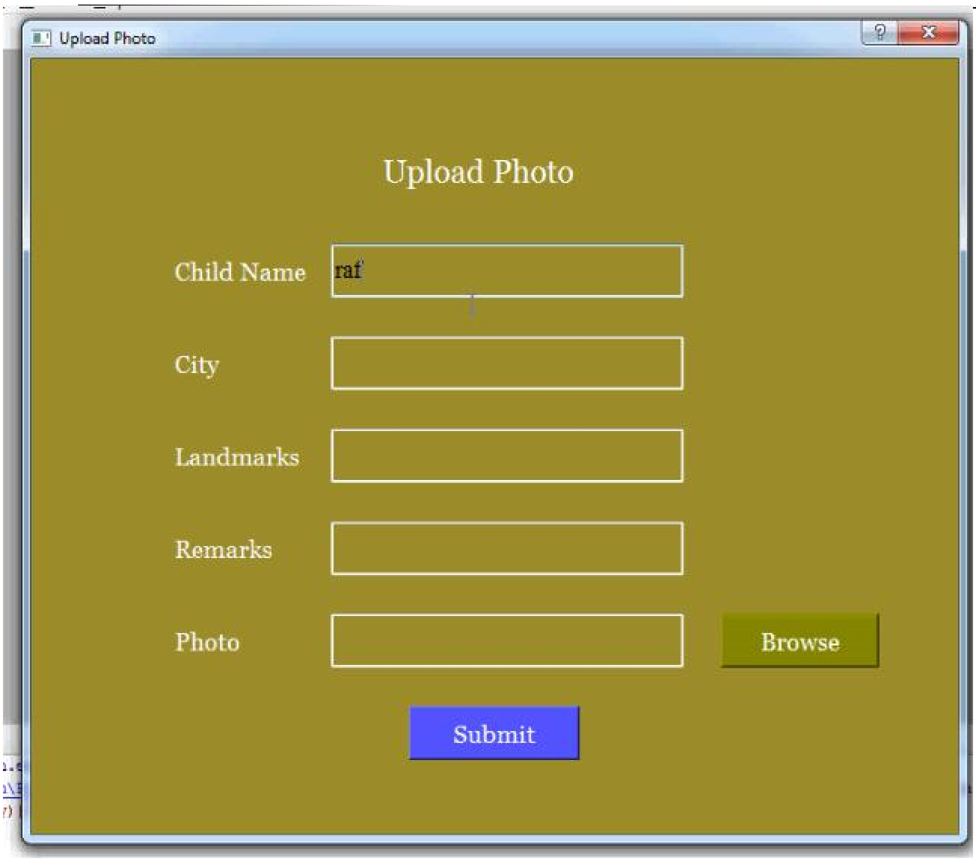


Figure 8.2 Login Window



Upload Photo

Upload Photo

Child Name raf

City

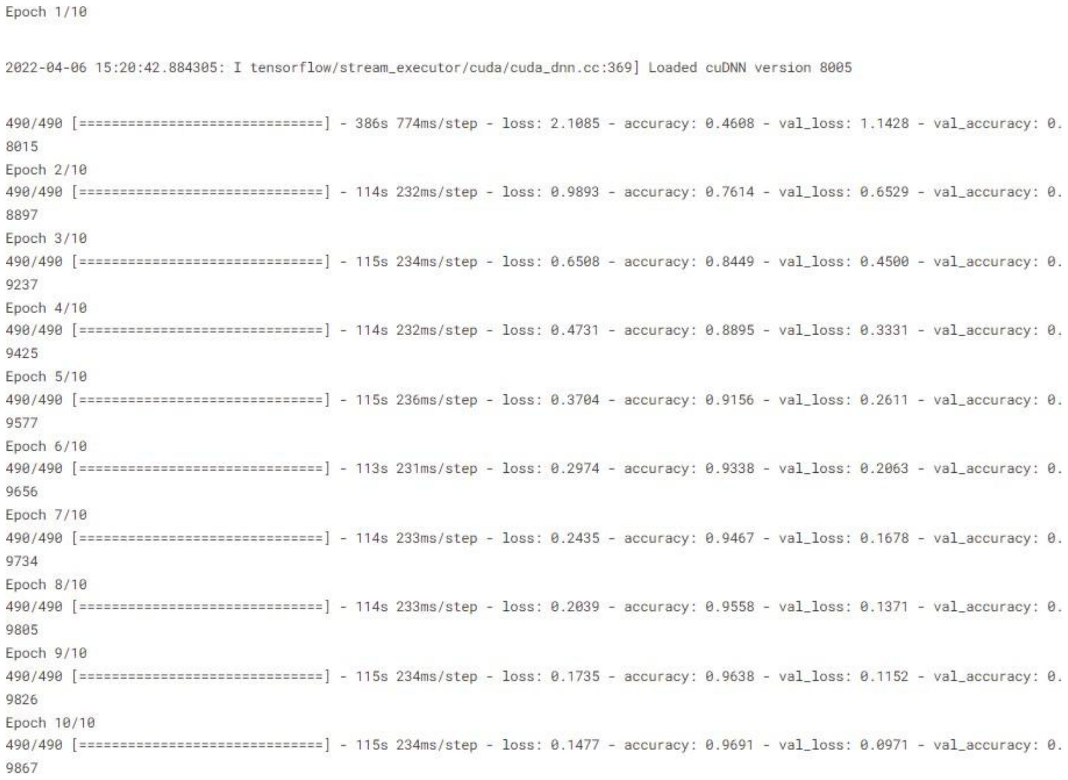
Landmarks

Remarks

Photo

Figure 8.3 Upload Window

8.1. VGG16



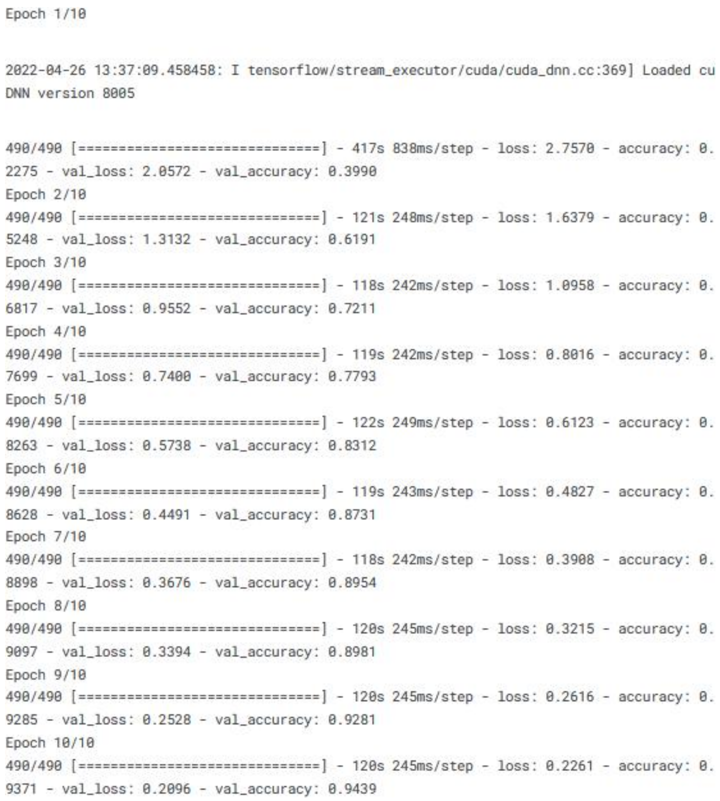
```
Epoch 1/10

2022-04-06 15:20:42.884305: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8005

490/490 [==============================] - 386s 774ms/step - loss: 2.1085 - accuracy: 0.4608 - val_loss: 1.1428 - val_accuracy: 0.
8015
Epoch 2/10
490/490 [==============================] - 114s 232ms/step - loss: 0.9893 - accuracy: 0.7614 - val_loss: 0.6529 - val_accuracy: 0.
8897
Epoch 3/10
490/490 [==============================] - 115s 234ms/step - loss: 0.6508 - accuracy: 0.8449 - val_loss: 0.4500 - val_accuracy: 0.
9237
Epoch 4/10
490/490 [==============================] - 114s 232ms/step - loss: 0.4731 - accuracy: 0.8895 - val_loss: 0.3331 - val_accuracy: 0.
9425
Epoch 5/10
490/490 [==============================] - 115s 236ms/step - loss: 0.3704 - accuracy: 0.9156 - val_loss: 0.2611 - val_accuracy: 0.
9577
Epoch 6/10
490/490 [==============================] - 113s 231ms/step - loss: 0.2974 - accuracy: 0.9338 - val_loss: 0.2063 - val_accuracy: 0.
9656
Epoch 7/10
490/490 [==============================] - 114s 233ms/step - loss: 0.2435 - accuracy: 0.9467 - val_loss: 0.1678 - val_accuracy: 0.
9734
Epoch 8/10
490/490 [==============================] - 114s 233ms/step - loss: 0.2039 - accuracy: 0.9558 - val_loss: 0.1371 - val_accuracy: 0.
9805
Epoch 9/10
490/490 [==============================] - 115s 234ms/step - loss: 0.1735 - accuracy: 0.9638 - val_loss: 0.1152 - val_accuracy: 0.
9826
Epoch 10/10
490/490 [==============================] - 115s 234ms/step - loss: 0.1477 - accuracy: 0.9691 - val_loss: 0.0971 - val_accuracy: 0.
9867
```

Figure 8.4 (a) VGG16 Model Fitting

8.2. SIMPLE CNN

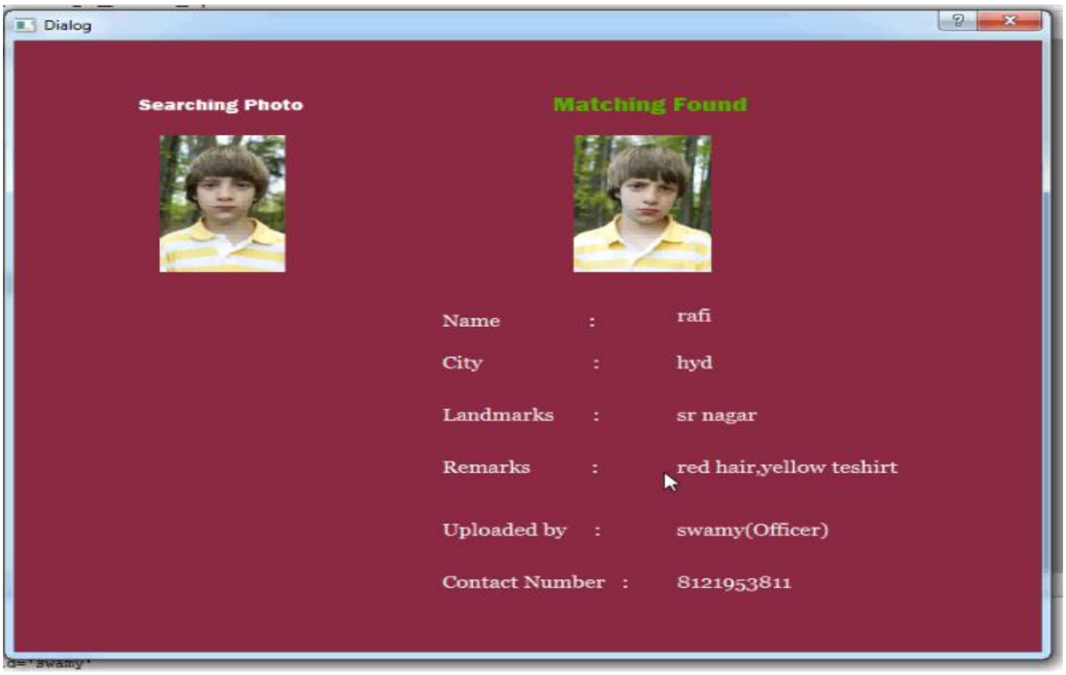


```
Epoch 1/10

2022-04-26 13:37:09.458458: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cu
DNN version 8005

490/490 [==============================] - 417s 838ms/step - loss: 2.7570 - accuracy: 0.
2275 - val_loss: 2.0572 - val_accuracy: 0.3990
Epoch 2/10
490/490 [==============================] - 121s 248ms/step - loss: 1.6379 - accuracy: 0.
5248 - val_loss: 1.3132 - val_accuracy: 0.6191
Epoch 3/10
490/490 [==============================] - 118s 242ms/step - loss: 1.0958 - accuracy: 0.
6817 - val_loss: 0.9552 - val_accuracy: 0.7211
Epoch 4/10
490/490 [==============================] - 119s 242ms/step - loss: 0.8016 - accuracy: 0.
7699 - val_loss: 0.7400 - val_accuracy: 0.7793
Epoch 5/10
490/490 [==============================] - 122s 249ms/step - loss: 0.6123 - accuracy: 0.
8263 - val_loss: 0.5738 - val_accuracy: 0.8312
Epoch 6/10
490/490 [==============================] - 119s 243ms/step - loss: 0.4827 - accuracy: 0.
8628 - val_loss: 0.4491 - val_accuracy: 0.8731
Epoch 7/10
490/490 [==============================] - 118s 242ms/step - loss: 0.3908 - accuracy: 0.
8898 - val_loss: 0.3676 - val_accuracy: 0.8954
Epoch 8/10
490/490 [==============================] - 120s 245ms/step - loss: 0.3215 - accuracy: 0.
9097 - val_loss: 0.3394 - val_accuracy: 0.8981
Epoch 9/10
490/490 [==============================] - 120s 245ms/step - loss: 0.2616 - accuracy: 0.
9285 - val_loss: 0.2528 - val_accuracy: 0.9281
Epoch 10/10
490/490 [==============================] - 120s 245ms/step - loss: 0.2261 - accuracy: 0.
9371 - val_loss: 0.2096 - val_accuracy: 0.9439
```

Figure 8.5 (a) CNN Model Fitt



Name : rafi

City : hyd

Landmarks : sr nagar

Remarks : red hair,yellow teshirt

Uploaded by : swamy(Officer)

Contact Number : 8121953811

Figure 8.6 (a) Match Found

CHAPTER 9

CONCLUSION

CHAPTER 9 CONCLUSION

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

CHAPTER 10

FUTURE WORK

CHAPTER 10 FUTURE WORK

Future research and work should be devoted to:

- Integrating CCTV of Public Places.
- Development of a smartphone application.
- Add Map Location.

REFERENCES

[1] Nurul Azma Abdullah, Md. Jamri Saidi and Nurul Hidayah Ab Rahman “Face recognition for criminal identification: An implementation of principal component analysis for face recognition”The 2nd International Conference on Applied Science and Technology 2017 (ICAST’17)

[2] Apoorva.P, Ramesh.B and Varshitha.M.R “Automated criminal identification by face recognition using open computer vision classifiers” Third International Conference on Computing Methodologies and Communication (ICCMC 2019).

[3] Rasanayagam, K.Kumarasiri, S.D.D, Tharuka, W. A. D. Samaranayake, N. Samarasinghe and P. Siriwardana “CIS: An Automated Criminal Identification System”. 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] Mantoro, T., Ayu, M. A., & Suhendi. (2018).” Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods” 2018 6th International Conference on Multimedia Computing and Systems (ICMCS).

[5] Chang L , Yang J, Li S, Xu H, Liu K & Huan, C. (2018). ”Face Recognition Based on Stacked Convolutional Autoencoder and Sparse Representation”. 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)

[6] MING Ju-wang (2018), “Face Feature Dynamic Recognition Method Based on Intelligent Image”. International Conference on Virtual Reality.

[7] Mohd Yusuf Firoz Siddiqui and Sukesha (2015), “Face Recognition using Original an Deep Learning

[8] Hyung-Il Kim, Seung Ho Lee, and Yong Man (2015), “Face Image Assessment Learned

[9] Piyush Kakkar, Mr. Vibhor Sharma (2018) “Criminal Identification System Using Face Detection and Recognition”. International Journal of Advanced Research in Computer and Communication Engineering.

[10] Lamiaa A. Elrefaei, Alaa Alharthi, Huda Alamoudi, Shatha Almutairi (2017) "Real-time Face Detection and Tracking on Mobile Phones for Criminal Detection"

APPENDIX

Convolutional Neural Networks:

```
input_image=”person.jpg”
image=face_recognition.load_image_file(inputimage)
face_locations = face_recognition.face_locations(image)
for face_location in face_locations:
    top, right, bottom, left = face_location
    face_image = image[top:bottom, left:right]
    cv2.rectangle(inputimage, (left, top),(right,bottom)
(0,0, 255), 2)
```

Main

```
from PyQt5 import QtCore, QtGui, QtWidgets
from Authority import Ui_Authority
from User import Ui_User
class Ui_Dialog(object):

  def authrtylogin(self, event):

    event.accept()

  def userlogin(self, event):
    try:
    except Exception as e:
        print(e.args[0])
        tb = sys.exc_info()[2]
        print(tb.tb_lineno)
    event.accept()

  def setupUi(self, Dialog):

  def retranslateUi(self, Dialog):
    _translate = QtCore.QCoreApplication.translate
    Dialog.setWindowTitle(_translate("Dialog", "Main"))
    self.label.setText(_translate("Dialog", "Missing Person Identification System"))
    self.label_2.setText(_translate("Dialog", "Authority"))
    self.label_3.setText(_translate("Dialog", "Users"))




if __name__ == "__main__":
  import sys
  app = QtWidgets.QApplication(sys.argv)
  Dialog = QtWidgets.QDialog()
  ui = Ui_Dialog()
```

```
ui.setupUi(Dialog)
Dialog.show()
sys.exit(app.exec_())
```