FINDING MISSING PERSON USING DEEP LEARNING

UG PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

HYDERABAD

In partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

CHIDURALA SUPRITHA	**19UK1A05L1**
MADISHETTY AKSHAYA	**19UK1A05H6**
PENDOTA NAVANEETH	**19UK1A05H5**
AIREDDY RAKESH	**19UK1A05K8**

Under the Esteemed Guidance of

T. Sushma

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal-506005

2019-2023

FINDING MISSING PERSON USING DEEP LEARNING

UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

HYDERABAD

In partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

CHIDURALA SUPRITHA	**19UK1A05L1**
MADISHETTY AKSHAYA	**19UK1A05H6**
PENDOTA NAVANEETH	**19UK1A05H5**
AIREDDY RAKESH	**19UK1A05K8**

Under the Esteemed Guidance of

T.Sushma

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal-506005 **2019-2023**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal-506005

2019-2023

CERTIFICATE

This is to certify that the project report entitled **"FINDING MISSING PERSON USING DEEP LEARNING**" is being submitted by **CH.SUPRITHA(19UK1A05L1),M.AKSHAYA(19UK1A05H6),P.NAVANEETH(19UK1A05H5),A.RAKESH(19UK1A058)** under the guidance and supervision of **T.SUSHMA** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** to **Jawaharlal Nehru Technological University, Hyderabad** during the academic year **2022-23**,is a record work carried out by them under the guidance and supervision.

Project Guide **Head of the Department**

T. Sushma **Dr. R. Naveen Kumar**

(Assistant professor) (Professor)

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carryout this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1 Project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1and for their support in completing the UG Project Phase-1

We express heartfelt thanks to the guide, **T.SUSHMA,** Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

CH.SUPRITHA **(19UK1A05L1)**
M.AKSHAYA **(19UK1A05H6)**
P.NAVANEETH **(19UK1A05H5)**
A.RAKESH **(19UK1A05K8)**

ABSTRACT

Nowadays to find a missing person is very hard task, although we all are updated by social media, it requires numerous paper work to be done and it is time consuming process and also after doing this hard work, there are not much chances of proper result. This project gives out a system, which helps both police department and public by speeding up the process of searching using face recognition. Therefore, how this system works is that when the person goes missing the respective guardian of that person can upload the image, which then will get store in our database. Next is, the face recognition model in our system will find a match of that person in the database. If a match is found, it will be notified to the police and the guardian of that person.

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. This paper proposes a system that would help the police and the public by accelerating the process of searching using face recognition. When a person goes missing, the people related to that person or the police can upload the picture of the person which will get stored in the database. When the public encounter a suspicious person, they can capture and upload the picture of that person into our portal. The face recognition model in our system will try to find a match in the database with the help of face encodings. It is performed by comparing the face encodings of the uploaded image to the face encodings of the images in the database. If a match is found, it will be notified to the police and the people related to that person along with the location of where the person is found. The face recognition model that we have used maintains an accuracy of 99.38% on the Labelled Faces in the Wild Benchmark which comprises of 13,000 images

Keywords: Artificial Intelligence, KNN Algorithm, Support Vector Machine, Random Forest, Open CV, Dlib, Haar Cascade Classifier, Face Recognition, Face Modeling, Face Detection.

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1. MOTIVATION 1

1.2. PROBLEM DEFINATION 1

1.3. PROJECT OBJECTIVE 1

2. LITERATURE SURVEY 2-6

2.1. INTRODUCTION..... 2

2.2. EXISTING SYSTEM..... 2-3

2.3. PROPOSED SYSTEM..... 3-6

3. EXPERIMENTAL ANALYSIS 7-9

3.1. PROJECT ARCHITECTURE 8

3.2. SOFTWARE AND HARDWARE REQUIREMENTS 8

3.4. BLOCK DIAGRAM 9

3.5. PROJECT FLOW..... 9

4. DESIGN 10

5. APPLICATIONS 11

6. CONCLUSION 12

7. FUTURE SCOPE..... 13

LIST OF FIGURES PAGE NO

Figure 1: System Structure 4

Figure 2: CNN Algorithm 5

Figure 3: DNN Algorithm 5

Figure 4: Project Architecture 8

Figure 5: Block Diagram.....9

Figure 6: Usecase Diagram.....10

Figure 7: Activity Diagram 10

1. INTRODUCTION

1.1 MOTIVATION:

In India 174+ children's and 250+ persons are missing every day, and half of them are untraced so by this project we think the tracking rate will increase.

Children's or any person missing that is major issue and we as a society have to resolve it. For that purpose we decided to make system which will help to find missing persons by their face recognition using Machine Learning, Deep Learning and Artificial Intelligence.

1.2 PROBLEM DEFINITION:

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 2016, and 55,625 of them remaineduntraceable till the end of the year.

1.3 PROJECT OBJECTIVE:

In this scenario, missing case entries are updated with their photocopies in the police station. By using CCTV cameras we can compare each person with the available database and find these people. If the missing person is found in the CCTV Video stream then location which is tagged to the CCTV is sent as an SMS to the police station.

2. LITERATURE SURVEY

2.1 INTRODUCTION

We did lot of survey and summed up following regarding literature survey so firstly, S. AYYAPPAN and his fellow mates from IFET College of Engineering have a presented apaper which deals with a similar problem statement and objective.

2.2 EXISTING SYSTEM

The system proposed by users makes use of Deep Learning based Facial Feature Extraction and matching with stacked convolutional auto encoder (SCAE). The images of missing Persons are stored in a database. Faces are detected from those images, and a Convolutional Neural Network learns features. These learned features were utilized for training a multi-class SVM classifier. They used this method to identify and label the kid correctly. The main difference between their work and ours is that we are going to create a dataset of lost persons with the help of people who want to contribute to society (voluntary work). Also we are not going to disclose the details of lost person with the public. And their system involves complex algorithms which make the process of extraction and classification slower.

Previously, Shefali Patil and his fellow mates from SNDT Women's University, Juhu, Mumbai have a presented a paper which deals with a similar problem statement and objective. The system proposed by them uses KNN Algorithm which makes use of 136 * 3 data points to recognize Face. The main disadvantage of using the KNN method is its accuracy 71.28% and also it does not address cross-age face recognition. The main difference between their work and ours is that here we are going to create a dataset using a mobile application with voluntary work of people. we are going to use AWS facial reorganization which has cross-age face recognition. Also, our dataset is going to be stored in the cloud database.

In 2020, Sarthak Babbar, Navroz Dewan, Kartik Shangle and his fellow mates from Jaypee Institute of Information Technology, Noida, India team gave out a paper in which gave us very clear idea about how Amazon Web Services (AWS) Recognition works and it compares AWS recognition with other algorithms and to

use in our project e.g Amazon Web Services (AWS) Recognition Our faces will change with time as our age increases, while the pictures in our dataset remain the same. We intend to study the accuracy of Residual Network (ResNet) for the purpose of cross-age face recognition. The performance is compared to cross-age reference coding (CARC), Amazon Web Services (AWS) Recognition and other techniques on the various data set viz.

2.3 PROPOSED SYSTEM

The proposed system makes use of various methods for finding missing people. Thesystem structure is presented in **Fig.1**.

Overall Structure of Proposed System To prevail over the drawbacks of previous systems. We are building a system

Structure of System that existing systems were not having. We plan to add concepts regarding how the interface ought for adding new complaints and how to register the new case.

The face recognition model in our system will try to find a match in the database with the help of AWS recognition. It is performed by comparing the face encoding of the uploaded image to the face encoding of the images in the database. If a match is found, it will be notified to the police and the people related to that person along with the location of where the person is found.

The proposed systemcontains the following Modules:

Volunteer Module:

• Volunteer Registration/ Login (Using E-mail ID,

• Filling in the details such as the location, age. Architecture suspected missing personand then upload the image of a suspected missing person.

Police / Authority Module:

• Police/ Authority Login (Using Email ID, Mobile Number, Password)

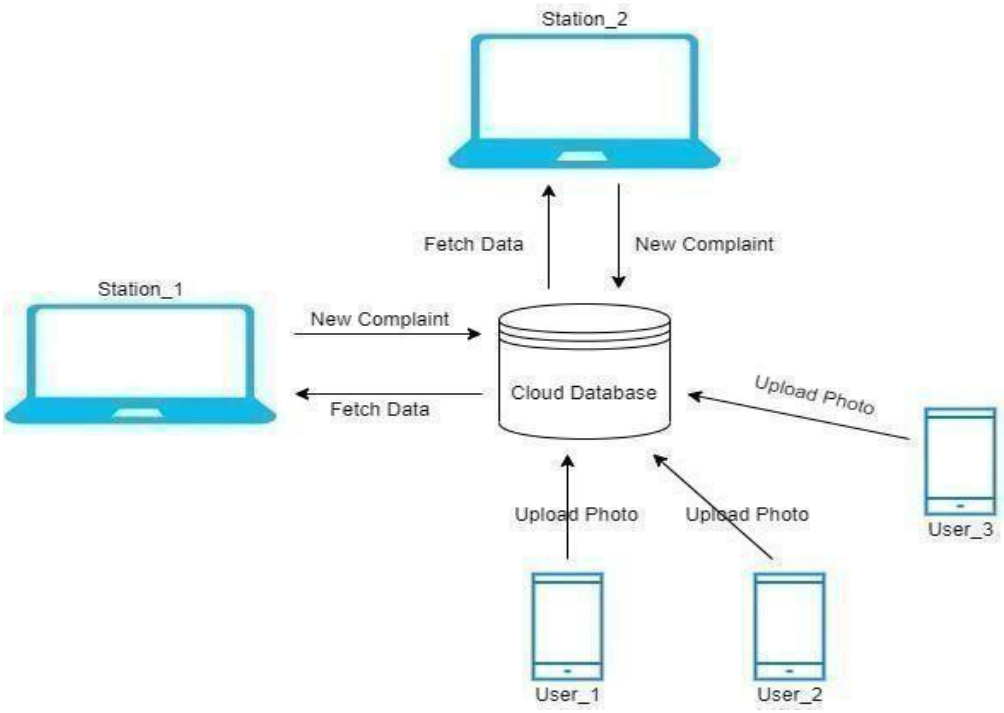

Figure 1: System Structure

Here we are using various machine learning algorithms, some of them are:

1. CNN(Convolutional Neural Network):

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre- processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as theReceptive Field. A collection ofsuch fields overlap to cover the entire visual area.

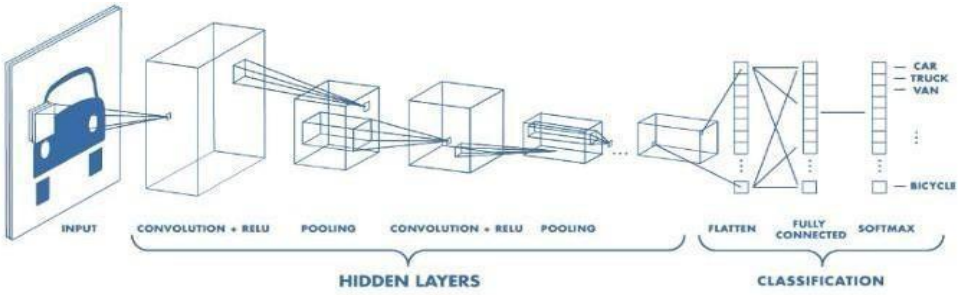

Figure 2: CNN Algorithm

2. DNN(Deep Neural Network):

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships.

The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks.

We have an input, an output, and a flow of sequential data in a deep network.

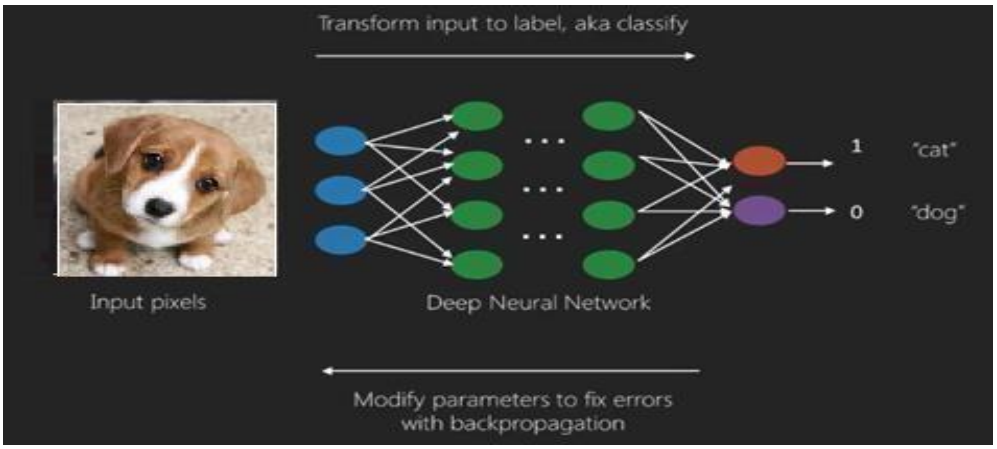

Figure 3: DNN Algorithm

3. OpenCV For Video Processing

Open CV is an open-source library which provides us with the tools to perform almost any kind of image and video processing.

Task 1: Capture Video from Camera

Often, we have to capture a live stream with a camera. OpenCV provides a very simple interface to this. Let's capture a video from the camera (I am using the in-built webcam of my laptop), convert it into grayscale video and display it.

To capture a video, you need to create a VideoCapture object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Normally one camera will be connected (as in my case). So I simply pass 0(or -1). You can select the second camera bypassing 1 and so on. After that, you can capture frame-by-frame. But in the end, don't forget to release the capture.

Task 2: Importing the required libraries.

Before importing the libraries, Install Twilio library, using below command in anaconda prompt, "pip install twilio".

Task 3: Loading our saved model file using load_model from Keras library

3.EXPERIMENTAL ANALYSIS

Milestone 1:Data Collection:

Artificial Intelligence is a data hunger technology, it depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Convolutional Neural Networks, as it deals with images, we need training and testing data set. It is the actual data set used to train the model for performing various actions.

- Collect the data set or create the data set.
- You can collect datasets from different open sources like kaggle.com..
- The data set used for this project was obtained from kaggle.

Milestone 2:Image Preprocessing:

Image Pre-processing includes the following main tasks

- Import Image Data Generator Library.
- Configure Image Data Generator Class.
- Applying Image Data Generator functionality to trainset and testset.

The Image Data Generator accepts the original data, randomly transformsit,and returns only the new, transformed data.

Milestone 3:Model Building

The neural network model is to be built by adding different network layers likeconvolution,pooling, flattening, dropout, and neural layers.
In this milestone, we start building our model by:
1.Initializing the mode
2.Adding Convolution layers
3.Adding Pooling layers
4.Flatten layer
5.Full connection layers which include hidden layers

At last, we compile the model with layers we added to complete the neural network Structure.

Milestone 4:Video Streaming and alerting:

1.OpenCV for video processing

2.Creating an account in twilio service

3.Use API to send Alert Message.

3.1 PROJECT ARCHITECTURE:

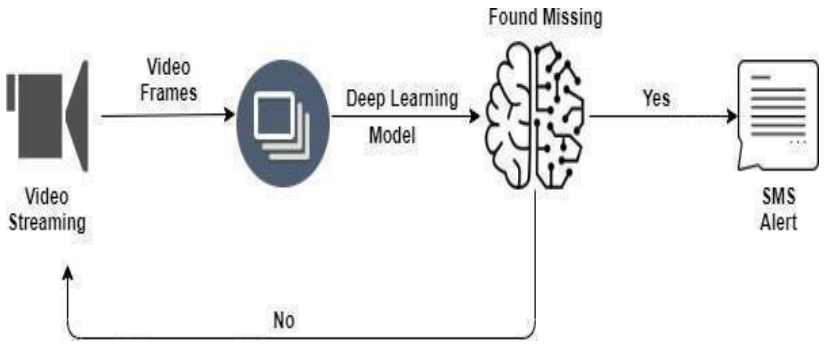

Figure 4: Project Architecture

3.2 SOFTWARE AND HARDWARE REQUIREMENTS:

- Processor-i5
- RAM-8GB
- Anaconda3
- Jupyter Notebook
- Google calloboratory
- Spyder

3.3 BLOCK DIAGRAM:

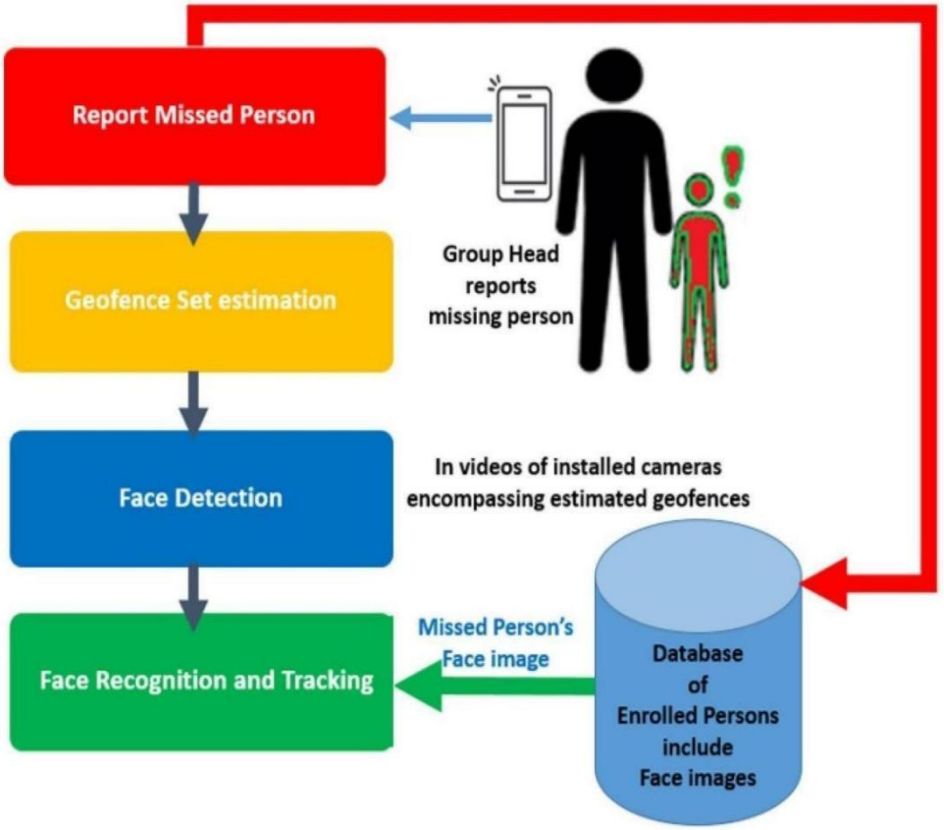

Figure 5: Block Diagram

3.4 PROJECT FLOW

A)User interacts with a web camera to read the video.

B)Once model analyses the input image from the video frame, when the missing person is detected, triggers an alert message to the Police/Family Members.

4.DESIGN

4.1 USECASE DIAGRAM

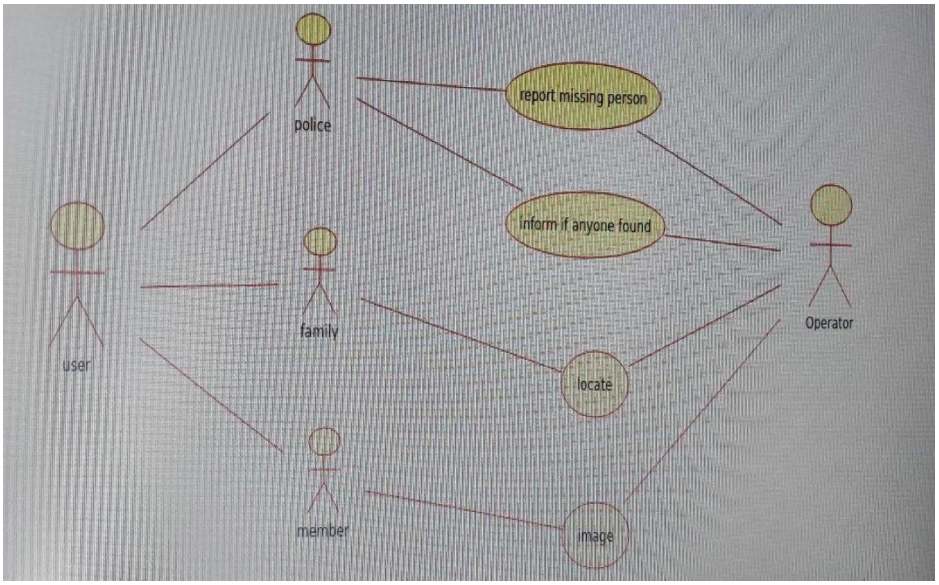

Figure 7 : Usecase diagram

4.2 ACTIVITY DIAGRAM

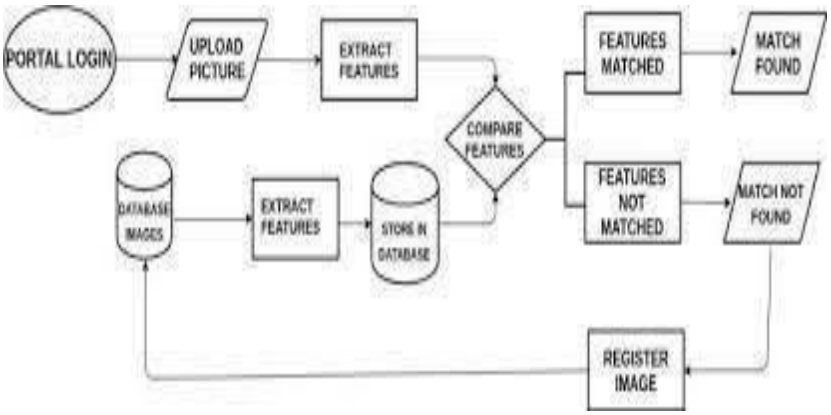

Figure 8: Activity diagram

5.APPLICATIONS

The system proposed by us makes use of Deep Learning based Facial Feature Extractionand matching with stacked convolutional auto encoder (SCAE). The images of missing Persons are stored in a database. Faces are detected from those images, and a ConvolutionalNeural Network learns features.Hence, the goal of this research is to solve this issue by developing a deep learning-based system for identifying missing individuals

6.CONCLUSION

Today's advancement is just the beginning in the onset of face detection technology there are many other applications where face detection can be introduced. So we found we use dataset of "HaarCascade_frontalface_default.xml" in our project. Which gives better results and easily available also. We found that using OpenCV with KNN is giving better results. Thus our project will improve the existing system of findinga person. Image recognition with the use of one-shot learning has become very powerful. This technology when put into good use can be extremely helpful. It can be even used in Hotels, Hospitals, etc. to find criminals in an instant.

7.FUTURE SCOPE

In this project we have used the convolution Neural Network which yielded a good accuracy around 92% but there are far more advanced techniques including Resnet networks, VGG networks and Dense Net networks which can work much better when compared to traditional CNN.

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.

FINDING MISSING PERSON USING DEEP LEARNING

UG PHASE-2 PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

CHIDURALA SUPRITHA	**19UK1A05L1**
MADISHETTY AKSHAYA	**19UK1A05H6**
PENDOTA NAVANEETH	**19UK1A05H5**
AIREDDY RAKESH	**19UK1A05K8**

Under the Esteemed Guidance of

T. Sushma

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal-506005

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal-506005

2019-2023

CERTIFICATE

This is to certify that the UG Project Phase-2 entitled **"FINDING MISSING PERSON USING DEEP LEARNING**" is being submitted by **CH.SUPRITHA(19UK1A05L1),M.AKSHAYA(19UK1A05H6),P.NAVANEETH(19UK1A05H5),A.RAKESH(19UK1A058)** under the guidance and supervision of **T.SUSHMA** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** to **Jawaharlal Nehru Technological University, Hyderabad** during the academic year **2022-23**,is a record work carried out by them under the guidance and supervision.

Project Guide	**Head of the Department**
T. Sushma	**Dr. R. Naveen Kumar**
Assistant professor	professor

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carryout this Industry UG Project Phase-2 in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-2 Project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1and for their support in completing the UG Project Phase-2

We express heartfelt thanks to the guide, **T.SUSHMA,** Assistant professor, Departmentof CSE for her constant support and giving necessary guidance for completionof this UG Project Phase-2.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

CH.SUPRITHA (19UK1A05L1)
M.AKSHAYA (19UK1A05H6)
P.NAVANEETH (19UK1A05H5)
A.RAKESH (19UK1A05K8)

TABLE OF CONTENTS

1.INTRODUCTION

2.IMPLEMENTATION AND RESULTS

2.1 Importing The Image Data Generator Library

2.2 Define The Parameters /Arguments For Image Data Generator Class

2.3 Applying Image Data Generator Functionality To Trainset And Testset

2.4 Importing The Model Building Libraries

2.5 Initializing The Model

2.6 Adding Dense Layers

2.7 Configuring The Learning Process

2.8 Training The Model

2.9 Save The Model

2.10 Predictions

2.11 Video Analysis

2.12 OpenCV For Video Processing

2.13 Creating An Account In Twilio Service

2.14 Sending Alert Message By Combining All Codes

2.15 Alert Message

3.APPLICATIONS

4.CONCLUSION

LIST OF FIGURES PAGE NO

Figure 1: System Structure 4

1.INTRODUCTION

In India 174+ children's and 250+ persons are missing every day, and half of them are untraced so by this project we think the tracking rate will increase.

Children's or any person missing that is major issue and we as a society have to resolve it. For that purpose we decided to make system which will help to find missing persons by their face recognition using Machine Learning, Deep Learning and Artificial Intelligence.

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 2016, and 55,625 of them remaineduntraceable till the end of the year.

In this scenario, missing case entries are updated with their photocopies in the police station. By using CCTV cameras we can compare each person with the available database and find these people. If the missing person is found in the CCTV Video stream then location which is tagged to the CCTV is sent as an SMS to the police station.

2.IMPLEMENTATION AND RESULTS

2.1 Importing The Image Data Generator Library

The first step is usually importing the libraries that will be needed in the program.

Import Keras library from that library import the ImageDataGenerator Library to your Python script:

#Importing Keras Library

import keras

#Importing ImageDataGenerator class from keras

from keras.preprocessing.image import ImageDataGenerator

Using TensorFlow backend.

2.2 Define The Parameters /Arguments For Image Data Generator Class

# Define the parameters / arguments for ImageDataGenerator class
train_datagen= ImageDataGenerator (rescale=1./255,
shear_range=0.2,
rotation_range=180,
horizontal_flip=True,
zoom range=0.2)
test_datagen= ImageDataGenerator (rescale=1./255)

2.3 Applying ImageDataGenerator Functionality To Trainset And Testset:

The ImageDataGenerator class has three methods flow ( ), flow_from_directory ( ) and flow_from_dataframe ( ) to read the images from a big numpy array and folders containing images.

flow_from_directory ( ) expects at least one directory under the given directory path.

Task1: Apply flow_from_directory ( ) method for Train folder.

#Applying ImageDataGenerator functionality to training_set
x_train train_datagen.flow_from_directory(r'directory_path\trainset',
target_size=(64,64), batch_size=32,
class_mode='binary')
# Note if more than 2 categories class_mode='categorical # Note Change your Directory path before executing this Cell
Note: Change directory path before executing this cell.

Task 2: Now will apply flow_from_directory ( ) method for test folder.

#Applying ImageDataGenerator, functionality to testing set

x_test-train_datagen.flow_from_directory(r'directory_path\testset',

target_size=(64,64), batch_size=32,

class_mode='binary')

#Note if more than 2 categories class_mode='categorical" #Note Change your Directory path before executing this Cell

The directory must be set to the path where your training folders are present.
The target_size is the size of your input images, every image will be resized to this size.
batch_size: No. of images to be yielded from the generator per batch.
"batch_size" in both train and test generators is to some number that divides your total number of images in your train set and train set respectively.
class_mode: Set "binary" if you have only two classes to predict, if not set to "categorical".

2.4 Importing The Model Building Libraries

#Importing Model Building Libraries

#To define linear intialisation import sequential

from tensorflow.keras.models import Sequential

#To add Hidden layers import Dense

from tensorflow.keras.layers import Dense

To Create Convolution Layer import convolution2D

from ensorflow.keras.layers import Conv2D

Import Max pooling Layer to extra maximum features

from tensorflow.keras.layers import MaxPoo12D

#Importing Flatten Layer

from tensorflow.keras.layers import Flatten

2.5 Initializing The Model

Keras has 2 ways to define a neural network:

Sequential
Function API

The Sequential class is used to define linear initializations of network layers which then, collectively, constitute a model. In our example below, we will use the Sequential constructor to create a model, which will then have layers added to it using the add () method.

Now will initialise our model.

#Intialising the model

model-Sequential()

Adding CNN Layers

We will be adding three layers for CNN

Convolution layer
Max Pooling layer
Flattening layer

Task 1: Adding Convolutional Layer

The convolutional layer is the first and core layer of CNN. It is one of the building blocks of a CNN and is used for extracting important features from the image.

In Convolution operation, the input image will be convolved with a feature detector/filters to get a feature map. The important role of the feature detector is to extract the features from the image. The group of feature maps is called a feature layer.

```
# Adding Convolutional Layer

model.add(Conv2D( 32,3,3,input_shape=(64,64,3), activation='relu'))
```

In the convolution2D function, we have given arguments like 32,(3,3), that means we are applying 32 filters of 3x3 matrix filter, and input_shape is the input image shape with RGB, here 64x64 is the size and 3 represent the channel, RGB colour images.

And Activation function defines the output of input or set of inputs or in other terms defines node of the output of node that is given in inputs. They basically decide to deactivate neurons or activate them to get the desired output.

Task 2: Adding Max Pooling Layer

Pooling reduces the dimensionality of images by reducing the number of pixels in the output from the previous convolutional layer. It keeps only the necessary details. Pooling is a technique in CNN which helps us to avoid overfitting of data, spatial invariance and distortion. After applying max-pooling we will get another feature map called Pooled Feature Map.

Let's apply the pooling technique now.

```
# Adding Max Pooling Layer

model.add(MaxPool2D(pool_size=(2,2))) # 2,2 size of matrix
```

In the above code, pool_size is the pooling filter or kernel size.

Task 3: Adding Flatten Layer

Now the pooled feature map from the pooling layer will be converted into a one single dimension matrix or map, where each pixel in one single column, nothing but flattening. Flattening layer converts multi-dimension matrix to one single dimension layer.

```
#Adding Flatten Layer
```

model.add(Flatten())

2.6 Adding Dense Layers

The name suggests that layers are fully connected (dense) by the neurons in a network layer. Each neuron in a layer receives input from all the neurons present in the previous layer. Dense is used to add the layers.

Task 1: Adding Hidden layers

This step is to add a dense layer (hidden layer). We flatten the feature map and convert it into a vector or single dimensional array in the Flatten layer. This vector array is fed it as an input to the neural network and applies an activation function, such as sigmoid or relu or other, and returns the output.

Adding Hidden Layer

model.add(Dense(units=128, activation='relu', kernel_initializer= random_uniform'))

Key terms:

init is the weight initialization; initialization function is the network initialization function which sets all the weights and biases of a network to values suitable as a starting point for training. Units, which denotes is the number of neurons in the hidden layer. Activation function defines the output of input or set of inputs or in other terms defines node of the output of node that is given in inputs. They basically decide to deactivate neurons or activate them to get the desired output. It also performs a nonlinear transformation on the input to get better results on a complex neural network. Its nodes here just pass on the information (features) to the hidden layer.

Note: Add multiple hidden layers for more accuracy.

Task 2: Adding output layer

This step is to add a dense layer (output layer) where you will be specifying the number of classes your dependent variable has, activation function and weight initializer as the arguments. We use add () method to add dense layers. In this layer, no need of mentioning input dimensions as we have mentions them in the above layer itself.

Adding Output Layer

model.add(Dense(units=1,activation='sigmoid', kernel_initializer=random_uniform'))

Note: if you have only one or two class in output put “output_dim = 1” and “activation = sigmoid”. If you have more number of classes for supposing there are 3 classes then put “output_dim = 3” and “activation = softmax”.

2.7 Configuring The Learning Process

With both the training data defined and model defined, it's time to configure the learning process. This is accomplished with a call to the compile () method of the Sequential model class. Compilation requires 3 arguments: an optimizer, a loss function, and a list of metrics.

Configure the Learning Process

model.compile(optimizer=''adam', loss='binary_crossentropy', metrics=['accuracy'])

Note: In our project, we have 2 classes in the output, so the loss is binary_crossentropy.

If you more than two classes in output put “loss = categorical_crossentropy”.

2.8 Training The Model

At this point, we have training data and a fully configured neural network to train with said data. All that is left is to pass the data to the model for the training process to commence, a process which is completed by iterating on the training data. Training begins by calling the fit ( ) method.

The arguments are the batch size as you are using “adam”.

Training the model

model.fit_generator(x_train, steps_per_epoch=8,

validation_data=x_test, epochs=128,

validation steps=2)

steps_per_epoch: It specifies the total number of steps taken from the generator as soon as one epoch is finished and next epoch has started. We can calculate the value

of steps_per_epoch as the total number of samples in your training folder divided by the batch size. Epochs: an integer and number of epochs we want to train our model for.Validation_data can be either input and targets list generator inputs, targets, and sample_weights list which can be used to evaluate.

The loss and metrics for any model after any epoch has ended.

Validation_steps

Only if the validation_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

Note: This step takes a few minutes based on the epochs (no: of time you would like to train the machine with the given data set) you give.

2.9 Save The Model

Your model is to be saved for future purpose. This saved model also is integrated with the android application or web application in order to predict something.

#Saving the trained model with .h5 extension

model.save('Missing_1.h5')

2.10 Predictions

The last and final step is to make use of our saved model to do predictions. For that we have a class in keras called load_model. Load_model is used to load our saved model h5 file (alert.h5).

Importing the Model Libraries

import cv2

import numpy as np

import smtplib

from keras.preprocessing import image

import tensorflow as tf

```
import os

name = ["Found Missing", "Normal"]

#Loading the Saved model

model = tf.keras.models.load_model('Missing_1.h5')

#GivingRandomImage Path img = image.load_img(r"image_path\t1.jpg",target_size = (64,64)) x=image.img_to_array(img) x=np.expand_dims (x,axis=0)

# Classes of Prediction pred=model.predict_classes (x)

pred[0][0]
```

2.11 Video Analysis

In this, we are going to perform video analysis using opencv

2.12 OpenCV For Video Processing

OpenCV is an open-source library which provides us with the tools to perform almost any kind of image and video processing.

Task 1: Capture Video from Camera

Often, we have to capture a live stream with a camera. OpenCV provides a very simple interface to this. Let's capture a video from the camera (I am using the in-built webcam of my laptop), convert it into grayscale video and display it.

To capture a video, you need to create a VideoCapture object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Normally one camera will be connected (as in my case). So I simply pass 0 (or -1). You can select the second camera bypassing 1 and so on. After that, you can capture frame-by-frame. But in the end, don't forget to release the capture.

Task 2: Importing the required libraries.

Before importing the libraries, Install Twilio library, using below command in anaconda prompt,

"pip install twilio".

```
#Importing of Libraries #Import opencv Library

import cv2

# Import numpy Library

import numpy as np

# Import Keras image processing Library from keras.preprocessing import image

#Import Tensorflow Library

import tensorflow as tf

#Import Client Library from twilio

from twilio.rest import Client
```

Task 3: Loading our saved model file using load_model from Keras library

```
#Loading the Saved Model

model=tf.keras.models.load_model( 'Missing 1.h5')

#Initialising the video

video=cv2.VideoCapture(0)

# Destred outputs

name= ["Found Missing", "Normal"]
```

2.13 Creating An Account In Twilio Service

Twillio account for finding missing persons using deep learning methods. Twilio is a cloud communications platform that offers various services like SMS, voice calls, and video calls, which can be utilized in different applications including missing person cases.

1. To create a project using Twilio for finding missing persons with deep learning methods, you would typically need to integrate Twilio's communication capabilities with your own deep learning algorithms or models. Here's a high-level overview of the steps you might consider:Data collection: Gather relevant data such as images, videos, text descriptions, or any available information related to missing persons.

This data can come from various sources, including law enforcement agencies, social media platforms, or public databases.

2. Deep learning model development: Train or use pre-existing deep learning models for tasks like face recognition, object detection, or anomaly detection, depending on the specific requirements of your project, There are several popular deep learning frameworks like TensorFlow or PyTorch that can be used for this purpose

3.Integration with Twilio: Utilize Twilio's APIs and services to enable communication functionalities such as sending SMS messages or making voice calls. For example, you could configure Twilio to send notifications or alerts to a specified list of contacts when relevant information or potential sightings of missing persons are detected by your deep learning models.

4. Automation and alerting: Set up automated processes that analyze incoming data in real-time or on a scheduled basis. When the deep learning models detect significant matches or anomalies related to missing persons, trigger Twilio to send notifications or alerts to designated individuals or groups.

5. User interaction: Develop a user interface or application where authorized individuals,such as law enforcement personnel or search and rescue teams, can interact with the system, view alerts, and provide updates or additional information

Please note that implementing a full project involving deep learning and" Twilio for finding missing persons can be a complex task that requires expertise in deep learning, software development, and integration with external APIs. It's recommended to consult with professionals or developers experienced in these areas to ensure a successful implementation.

Go to this link and create a twilio account. It is free and no credit card required.

https://www.twilio.com/try-twilio

After creating account login to twilio. Using this link,

https://www.twilio.com/login/

After login, you will be redirected to home page or dashboard. Its looks like this,

Buy a Number

To get a free number to refer the steps in the below link,

https://www.twilio.com/docs/usage/tutorials/how-to-use-your-free-trial-account

Once after you are done with buy a number, you can able to see your free trail number and total balance in the dashboard.

Along with that, you can able to see the Account SID and Authentication token.

By using that free trial number and the Account SID and Authentication token we alerting the people.

We are going to integrating this twilio credentials with our OpenCV. When the missing person is detected it will give an alert through a message.

Task 1: Use API to send message.

We got the Account SID and Authentication token from twilio. Now we have to integrate all the opencv and twilio to send an alert message.

To integrate it with python, the code is looks like.

```
# Twilio Account Service ID

account_sid='ACec5c29dade0d5d23f9b93ea3f0c105e0'

#Twilio Account Auth Token

auth_token='72ed0ca874bc4114cd0f47ec02d4ff60'

#Initialise the client

client=Client (account_sid, auth_token)

# Creation of Message API

message=client.messages.create(

to="+91XXXXXXXXXX", # Fill the contact to your desired one
from_="XXXXXXXXX", # Fill with your created Twilio number

body=" Found the Missing at 17.3984° N, 78.5583° E" # Alert SMS Text

)
```

2.14 Sending Alert Message By Combining All Codes

The Entire Code of Video Streaming

```
# Importing of Libraries

#Import opencv Library

import cv2

# Inport numpy Library

import numpy as np

# Inport Keras image processing Library

from keras.preprocessing import image

# Inport Tensorflow Library

import tensorflow as tf

# Import Client Library from twilio

from twilio.rest import Client

# Loading the Saved Model

model = tf.keras.models.load_model( "Missing 1.h5")

# Initialising the video

video = cv2.VideoCapture(@)

# Desired outputs

name = [“Found Missing”, Normal”]

while(True):

success, frame = video.read()

cv2.imwrite("image.jpg", frame)

img = image.load_img("image.jpg", target_size = (64,64))

X = image.img_to_array(img)

x = np.expand_dims (x, axis = 0)
```

```
pred = model.predict_classes(x)

P = pred[0][0]

print (p)

cv2.putText(frame, "Predicted Class = "+str(name[p]), (100,100), cv2.FONT HERSHEY SIMPLEX, 1, (0,0,0), 1)

if pred[0][0]==0:

#Tilto Account Service ID

 account_sid='ACec5c29dade0d5d23f9b93ea3f0c105e0'

# Twilio Account Auth Token

auth_token='72ed0ca874bc4114cd0f47ec02d4ff60'

#Initialise the client

client=Client(account_sid, auth_token)

# Creation of Message API

message=client.messages.create(

 to="+91XXXXXXXXXX", # Fill the contact to your desired one from_="XXXXXXXXXXXXXX", # Fill with your created Twilio number body=" Found the Missing at 17.3984° N, 78.5583° E" # Alert SMS Text

 )

print (message.sid)

print("Found Missing")

print('SMS Sent')

else:

print("Normal")

cv2.imshow("frame", frame)

if cv2.waitKey(1) & 0xFF ord('a'):
```

```
break

video.release()

cv2.destroyAllWindows()
```

Once you run the above code after loading your model, and the label names (classes) that you have considered in the project, and also the authentication token and Account SID from twilio, it will open one video frame pop on your desktop/laptop screen.

Show any testing image to the camera, the output will be triggered as accordingly, either the Normal or Found missing person with the text on the console.

2.15 Alert Message

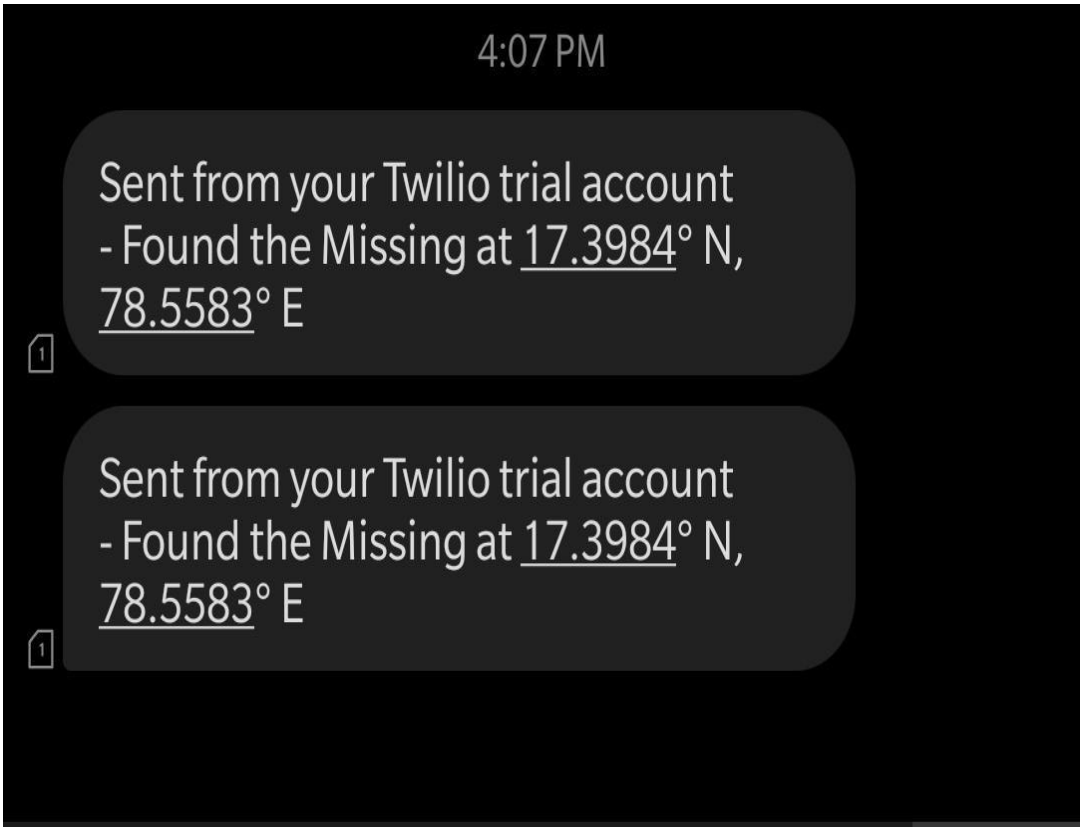

Fig :1 Alert Message

3.APPLICATIONS

Deep learning models can be trained to recognize patterns and detect anomalies in large amounts of data. This can be useful in identifying suspicious activities, such as unusual online behavior, communication patterns, or financial transactions associated with the missing person. It's important to note that while deep learning can be a powerful tool in finding missing persons, it should be used in conjunction with other investigative techniques and human expertise. Collaborating with law enforcement agencies, search and rescue teams, and other professionals is crucial for a comprehensive and effective search effort.

Deep learning models can continuously monitor social media platforms, news feeds, and online forums in real-time. By analyzing posts, comments, and Geo-location data, these models can identify relevant information such as potential sightings, updates from concerned individuals, or any other leads that can aid in the search for missing persons.

4.CONCLUSION

In conclusion, deep learning techniques offer promising applications in the search for missing persons. By analyzing message formats such as text messages or social media posts, deep learning models can aid in the identification, tracking, and location of missing individuals. Through text analysis, sentiment classification, language translation transcription, image and video analysis, social network analysis, NLP-powered chat bots, and pattern recognition, deep learning can provide valuable insights and assist investigators in their search efforts. However, it's important to emphasize that deep learning should be used as a tool in conjunction with other investigative techniques and human expertise.

Collaboration with law enforcement agencies, search and rescue teams, and other professionals is crucial for a comprehensive and effective search operation. Deep learning models are not a standalone solution but can significantly enhance the capabilities of investigators by processing large volumes of data, identifying patterns, and extracting relevant information.Ethical considerations, privacy concerns, and legal guidelines should also be taken into account when implementing deep learning in the search for missing persons.

Safeguarding individuals' privacy and ensuring the responsible use of personal data are essential aspects of any application of deep learning technologies. Overall, finding missing persons using deep learning holds great potential to assist in search and rescue operations, improve the efficiency of investigations, and ultimately contribute to reuniting families and bringing closure to missing person cases.

4. REFERENCES

1. Gumus E, Kilic N, Sertbas A, Osman NU (2010) Face recognition using PCA (principal component analysis). Expert Syst Appl 37:6404–6408

2. Bansal A, et al (2012) Face recognition using PCA and LDA algorithm. In: 2012 second international conference on advanced computing and communication technologies, vol 5. pp 251–254

3.Manoharan and J. Samuel, "Study of Variants of Extreme Learning Machine (ELM) Brands and its Performance Measure on Classification Algorithm", Journal of Soft Computing Paradigm (JSCP), vol. 3, no. 02, pp. 83-95, 2021.