CRIMINAL DETECTION SYSTEM

A Project Report
Submitted in partial fulfilment of the Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

SREYA VEGESINA (1602-18-737-092)

SANDALI NEMMANIWAR (1602-18-737-103)

HEMA KAMANI(1602-18-737-071)

RUCHITA ANANTHULA(1602-18-737-064)

Under the guidance of

DRL PRASANNA

Professor

Department of Information Technology
Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University) Ibrahimbagh,
Hyderabad-31

2020-2021

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500031

Department of Information Technology

DECLARATION BY THE CANDIDATES

We, **SREYA VEGESINA, SANDALI NEMMANIWAR, HEMA KAMANI, RUCHITA ANANTHULA** bearing hall ticket number , **1602-18-737-092, 1602-18-737-103, 1602-18-737-071, 1602-18-737-064** hereby declare that the project report entitled **"CRIMINAL DETECTION SYSTEM"** under the guidance of **DRL PRASANNA**, Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement of MINI PROJECT of VI semester of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

SREYA VEGESINA(1602-18-737-092)
SANDALI NEMMANIWAR(1602-18-737-103)
HEMA KAMANI(1602-18-737-071)
RUCHITA ANANTHULA(1602-18-737-064)

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**CRIMINAL DETECTION SYSTEM"** being submitted by **SREYA VEGESINA, SANDALI NEMMANIWAR, HEMA KAMANI, RUCHITA ANANTHULA** bearing **1602-18-737-092, 1602-18-737-103, 1602-18-737-071, 1602-18-737-064** in partial fulfilment of the requirements for the completion of MINI PROJECT of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

DRL Prasanna	**Dr. K. Ram Mohan Rao**
Professor	**HOD, IT**
Internal Guide	

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them. We would like to take the opportunity to express ourhumble gratitude to **DRL Prasanna(Professor)** under whom we executed this project. We would also use this opportunity to thank our Head Of Department **Dr.K.Ram Mohan Rao** . We would also like to thank all faculty members and staff of the Department ofInformation Technology for their generous help in various ways for the completion of this project.

Finally, We would like to express our heartfelt thanks to our senior **Rachana Yerram**(1602-17-737-098).We are grateful to his guidance, and constructive suggestions that helped us in the preparation of this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

ABSTRACT

The identification of interstate and intercountry criminals is one of those projects which can be used for the welfare of the society. Identification of criminals is important to ensure the safety of citizens and reduce the duration of investigation. Hence this project helps the police and research dept to identify those criminals who masquerade their identity.

Face is the primary means of recognizing a person, transmitting and communicating information with others. Our faces might disclose more than what we expect. A facial image can be informative of personal traits such as race, gender, age, health, emotion, psychology, and profession.

The base concept to implement the idea is neural networking. Neural networks have resurged and drawn much attention in the last decade with the new brand of deep learning, mainly due to the significant performance gain in visual recognition tasks. Neural networks are commonly used for pattern recognition and classification. They are also used to recognize aligned and normalized faces. In this project neural networks are used for extracting features from different images that are used to train the model.

In particular, one deep learning model, which is convolutional neural network (CNN) are applied to discriminate criminal and non-criminal facial images.

We feed the model with pre-planned dataset that has a collection of different images. A certain part of dataset is used for training the model and the rest for testing.

To put it in a nutshell, the project will accept an image from the user, surf through the dataset and pop out the confirmation of the presence of the image.

CHAPTER 1 8

Title and Description 8

1.1 Motivation 8

CHAPTER 2 9

Software Requirements Specification 9

2.1 Introduction 9

2.1.1 Purpose 9

2.1.2 Scope 9

2.1.3 Definitions, Acronyms, and Abbreviations 9

2.1.4 Overview 9

2.2 General Description 9

2.2.1 Product Perspective 9

2.2.2 Product Functions 10

2.2.3 User Characteristics 10

2.3 System Requirements 10

2.3.1 Hardware Requirements 10

2.3.2 Software Requirements 10

2.4 Design Constraints 11

CHAPTER 3 12

Related Work 12

CHAPTER 4 13

System Design 13

4.1 Architecture and Technology used 13

CHAPTER 5 14

Implementation 14

5.1 System Architecture (design) 14

5.2 Implementation and code 14

5.2.1. Html file 14

5.2.2 Python file 17

CHAPTER 6 21

Results 21

CHAPTER 7 24

Testing 24

CHAPTER 8 25

Conclusions and Future Scope 25

CHAPTER 9 26

References 26

CHAPTER 1
TITLE DESCRIPTION

Criminal Detection System is a web application which is developed to help the police and research department to detect if the given picture is that of a criminal or not. The application uses machine learning to predict the output.

1.1 Motivation

AI and machine learning is one of the booming technologies in the industry. Hence, utilizing our opportunity and to showcase our knowledge in this field we decided our thrust area to be machine learning. Since identification of criminals is important to ensure the safety of citizens and due to the shortage of such applications, “Criminal Detection System” was our go to choice.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 INTODUCTION

2.1.1 Purpose

Basic Description of Problem:
In the world where the crime rate is growing, we need technology which can be quicker than humans at solving criminal cases. The criminal detection system is that implementation where we use machine learning in order to figure out if the given input image is of a criminal or not.

2.1.2 Scope

This new application will help the police and research system to solve the cases a little quicker than usual. As it is known that criminals masquerade their identity in order to escape, this application detects the facial features and hence gives its output. This will result in accurate identification of criminals.

2.1.3 Definitions, Acronyms, and Abbreviations

Abbreviations:

CNN: Convolutional neural network
GUI: Graphical user interface
CDS: Criminal Detection System

Definitions:
Masquerade: Pretend to be someone one is not

2.1.4 Overview

The specifications include product perspective and the functionalities that the system will provide. The user characteristics, constraints, assumptions and dependencies are discussed below.
Requirements are categorized as performance, non-functional requirements and designed constraints. Non-functional requirements are scalability, maintainability and dependence.

2.2 GENERAL DESCRIPTION

2.2.1 Product perspective

CDS provides a new way for the investigation team to separate the criminals from the non-criminals. It also helps to identify those criminals who masquerade their identity.

2.2.2 Product functions

CDS allows users to input an image file (png, jpeg) and the result (criminal or non-criminal) is displayed on the user screen.

2.2.3 User characteristics

- User:
    - Can upload an image
    - Can view the output

2.3 SYSTEM REQUIREMENTS

2.3.1Hardware requirements

- 1 GB RAM
- 1 GB Core

2.3.2 Software requirement specifications

- **Python 3.8.9:**

  Python is the language used to build the Flask framework. It is a dynamic scripting language similar to Perl and Ruby. The principal author of Python is Guido Van Rossum. Python supports dynamic typing and has a garbage collector for automatic memory management. Another important feature of Python is dynamic name solutionwhich binds the names of functions and variables during execution.

- **Interpreter:**

  Visual Studio Code: It features a fast source code editor. It supports hundreds of languages and helps you with productive syntax highlighting, bracket matching, auto indentation, box selection, snippets, etc.

- **Flask:**

  Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

- **Bootstrap for web designing:**

  Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

2.4 DESIGN

2.5 CONSTRAINTS

- Software Constraints: Users can run this application on Linux and Windows.

- Hardware Constraints: The system will run on a 1 core processor.

- Acceptance Criteria: Before accepting the developer must check if the application is running properly or not and should also check whether the data is stored correctly.

CHAPTER 3

RELATED WORK

MARKET RESEARCH:

Facio: Facio is a facial recognition application which detects person from social network, media file or internet. You will have to give permissions to the application to access your social media, contacts, photos and internet. After which the user can take a picture or upload a picture from which you want to recognize a face.

CHAPTER 4

SYSTEM DESIGN

4.1 ARCHITECTURE AND TECHNOLOGY USED

Front End:

1.HTML

2.CSS

3.BOOTSTRAP

Back end:

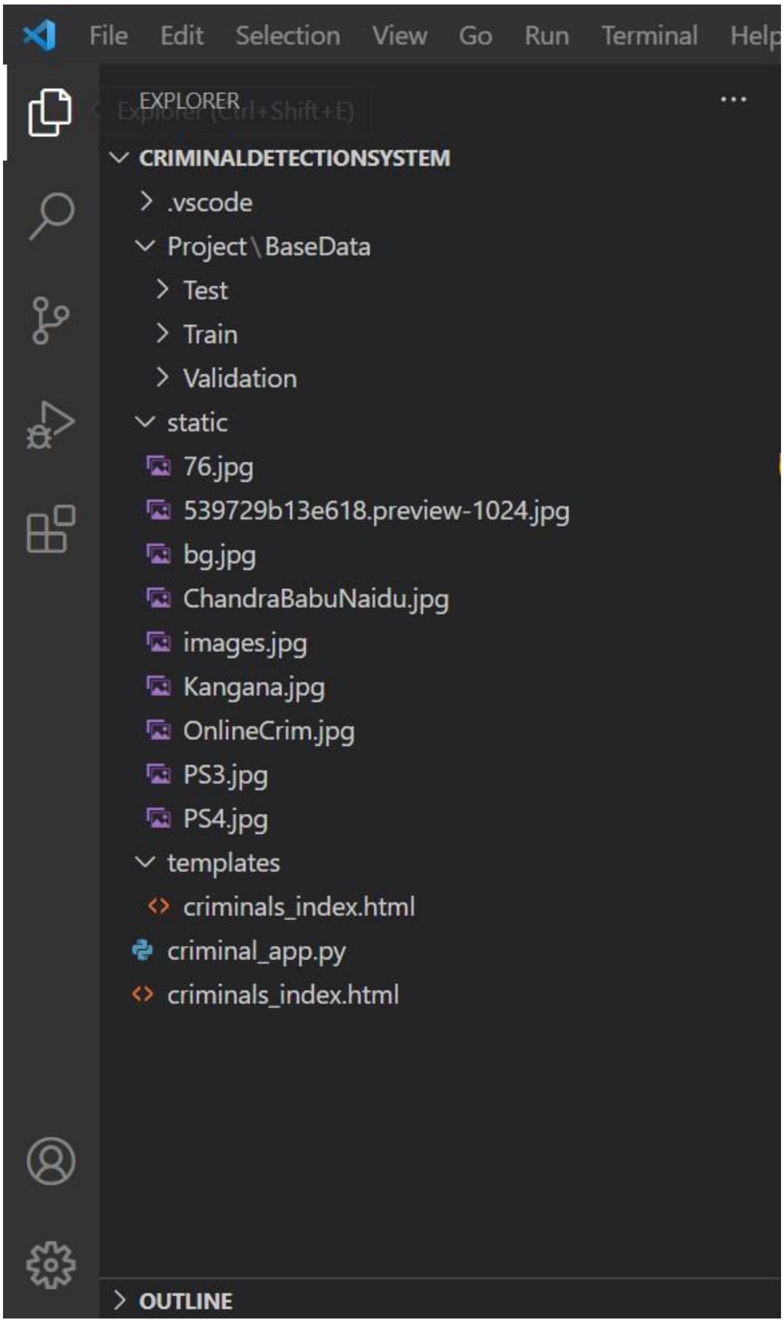
1. FLASK

2. PYTHON

CHAPTER 5

CODE IMPLEMENTATION

5.1 SYSTEM ARCHITECTURE(DESIGN)



```
CRIMINALDETECTIONSYSTEM
  .vscode
  Project \ BaseData
    Test
    Train
    Validation
  static
    76.jpg
    539729b13e618.preview-1024.jpg
    bg.jpg
    ChandraBabuNaidu.jpg
    images.jpg
    Kangana.jpg
    OnlineCrim.jpg
    PS3.jpg
    PS4.jpg
  templates
    criminals_index.html
  criminal_app.py
  criminals_index.html
```

5.2 IMPLEMENTATION AND CODE

5.2.1 HTML FILE

i) criminals_index.html

```
<!doctype html>

<html>

<head>
```

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x"
crossorigin="anonymous">
    <title>Criminal Detection System</title>
    <style>
        body{
            background: linear-gradient(rgba(0,0,50,0.5),rgba(0,0,50,0.5)),url(static/bg.jpg);
            margin: 0;
            padding: 0;
            background-size: cover;
            background-position: center;
            text-align: center;
            color: white;
            font-family: sans-serif;

        }
        .crim
        {
            color: red;
            background-color: white;
        }
        .non-crim{
            color: green;
            background-color: white;
        }
        .form-title
        {
            margin-top: 50px;
            color: white;
            text-transform: uppercase;
            transition: all 4s ease-in-out;
        }
        .form-title h1{
            font-size: 50px;
            line-height: 10px;
        }
        .sub{
            color:black;

        }
    </style>
</head>

<body class="text-center">
    <div class="container">
        <div class="form-title">
        <h1>Criminal Detection System</h1>
        </div>
        <br>
        <div class="sub">
            <marquee><h5>Please upload an image.</h5></marquee>
        </div>
        <form method=post enctype=multipart/form-data>
            <div class="form-floating">
                <input type=file name=image>
            </div>
            <br>
            <input type=submit class="btn btn-primary" value=Predict>
            <br>
        </form>
        <br>
        <h4>Image chosen: {{image_loc}}</h4>
        <br>
        {% if image_loc %}
            <img src="static/{{image_loc}}" width="200" height="200">
        {% endif %}
        <br>
        <br>
        {% if pred=='Criminal' %}
        <div class="crim">
        <h3><b>Prediction: This person is a CRIMINAL.</b></h3>
        </div>
        {% endif %}
        {% if pred=='notCriminal' %}
        <div class="non-crim">
        <h3><b>Prediction: This person is NOT A CRIMINAL.</b></h3>
        </div>
        {% endif %}


    </div>
</body>
</html>
```

5.2.2 PYTHON FILE

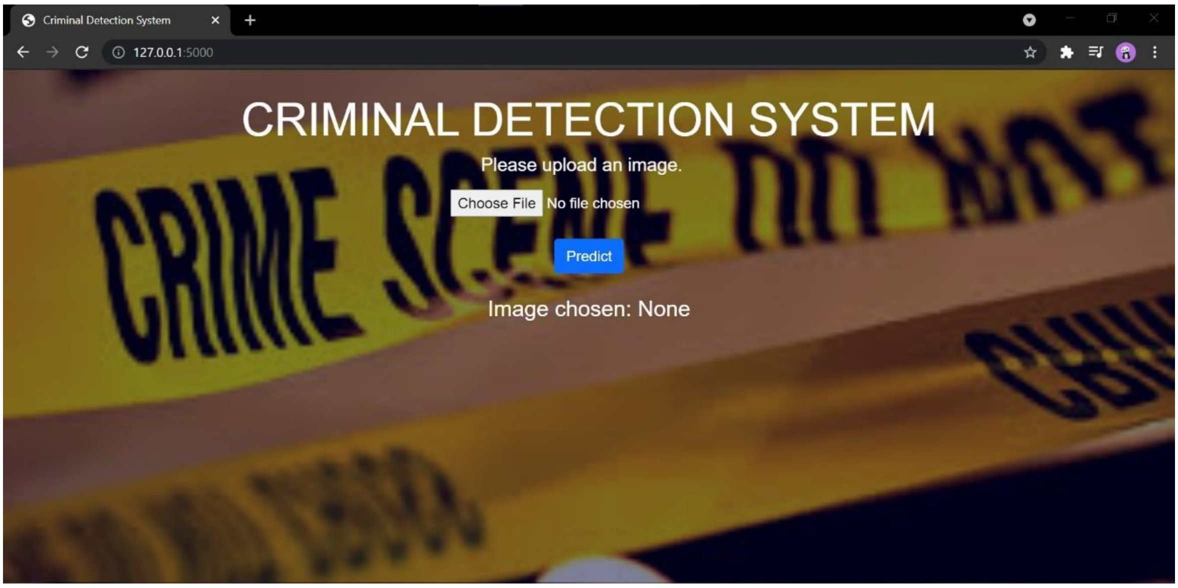
i)criminal_app.py

```
#way to upload image
#way to save the image
#function to make prediction on the image


import os
import PIL
import numpy as np
import tensorflow as tf

import matplotlib.pyplot as plt

from flask import Flask,render_template,request

from tensorflow.keras.optimizers import RMSprop

from tensorflow.keras.preprocessing import image

from tensorflow.keras.preprocessing.image import load_img

from tensorflow.keras.preprocessing.image import ImageDataGenerator



app=Flask(_name_)

upload_path="C://Users//Sreya//Desktop//CriminalDetectionSystem//static"

model=None

def training():

    global model

    train = ImageDataGenerator(rescale=1/201)

    validation=ImageDataGenerator(rescale=1/201)

    train_dataset = train.flow_from_directory('Project\\BaseData\\Train',

                                        target_size = (200,200),

                                        batch_size = 15,

                                        class_mode = 'binary')

    validation_dataset = train.flow_from_directory('Project\\BaseData\\Validation',

                                        target_size = (200,200),

                                        batch_size = 15,

                                        class_mode = 'binary')

    model = tf.keras.models.Sequential([ tf.keras.layers.Conv2D(16,(3,3),activation = 
'relu',input_shape=(200,200,3)),

                                    tf.keras.layers.MaxPool2D(2,2),

                                    #

                                    tf.keras.layers.Conv2D(32,(3,3),activation='relu'),

                                    tf.keras.layers.MaxPool2D(2,2),

                                    #
                                    tf.keras.layers.Conv2D(64,(3,3),activation='relu'),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    #
                                    tf.keras.layers.Flatten(),
                                    #
                                    tf.keras.layers.Dense(512,activation='relu'),
                                    #
                                    tf.keras.layers.Dense(1,activation='sigmoid')
                                   ])
    model.compile(loss='binary_crossentropy',
              optimizer=RMSprop(learning_rate=0.001),
              metrics=['accuracy'])
    model_fit = model.fit(train_dataset,
                     steps_per_epoch=5,
                     epochs=50,
                     validation_data= validation_dataset)

@app.route("/",methods=["GET","POST"])
def upload_predict():
    image_loc="
    pred="
    if(request.method=="POST"):
        image_file=request.files["image"]

        if image_file:
            image_loc=os.path.join(
                upload_path,
                image_file.filename
            )
            image_file.save(image_loc)
            img=load_img(upload_path+'/'+image_file.filename,target_size=(200,200))

            x=image.img_to_array(img)
            x=np.expand_dims(x,axis=0)
            images=np.vstack([x])
            if(model):
                val=model.predict(images)
                if val==0:
                    print("Criminal")
                    pred="Criminal"
                else:
                    print("not a Criminal")
                    pred="notCriminal"

        return render_template("criminals_index.html",pred=pred,image_loc = image_file.filename)
    return render_template("criminals_index.html",pred=pred, image_loc = None)
if _name=="main_":
    training()
    app.run(debug=True)
```

CHAPTER 6

RESULTS

Home page:

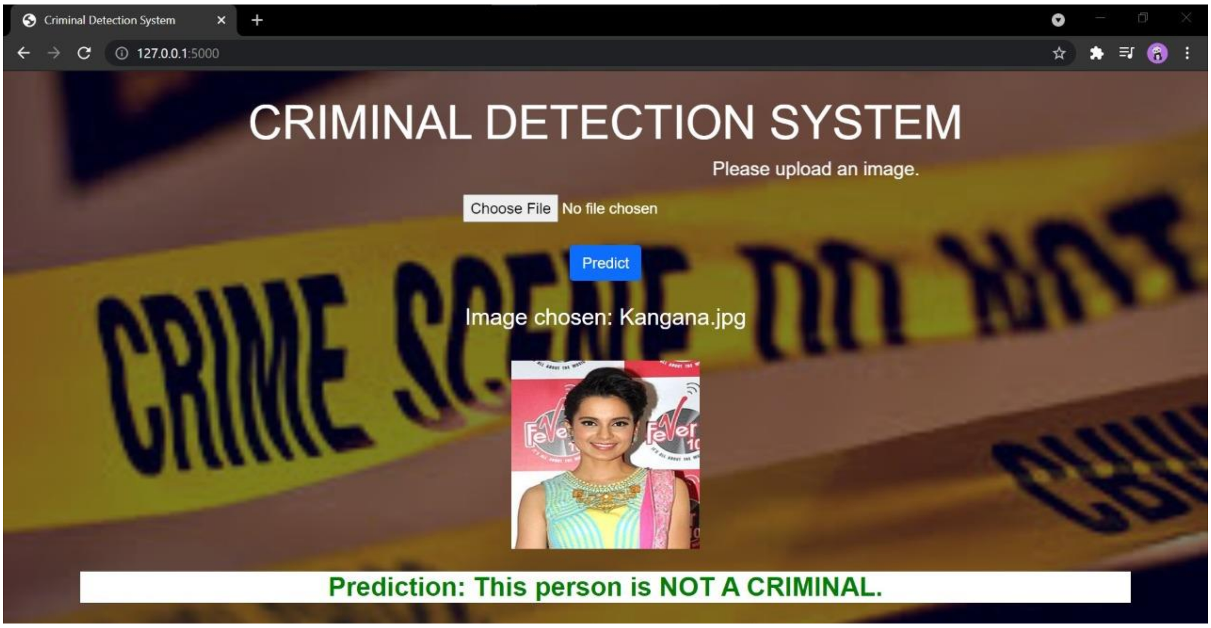


CRIMINAL DETECTION SYSTEM

Please upload an image.

Image chosen: None

Non-criminal detection:



CRIMINAL DETECTION SYSTEM

Please upload an image.

Image chosen: Kangana.jpg

Prediction: This person is NOT A CRIMINAL.

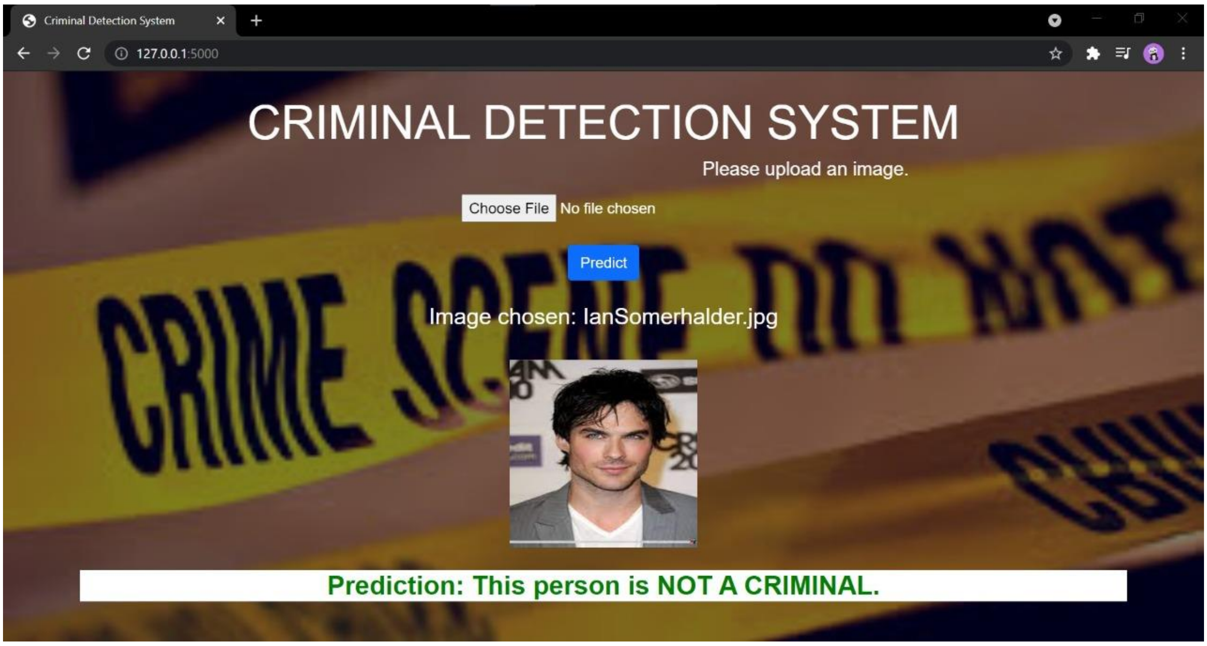

Criminal Detection System
127.0.0.1:5000
CRIMINAL DETECTION SYSTEM
Please upload an image.
Choose File No file chosen
Predict

Image chosen: IanSomerhalder.jpg

Prediction: This person is NOT A CRIMINAL.

Criminal detection:

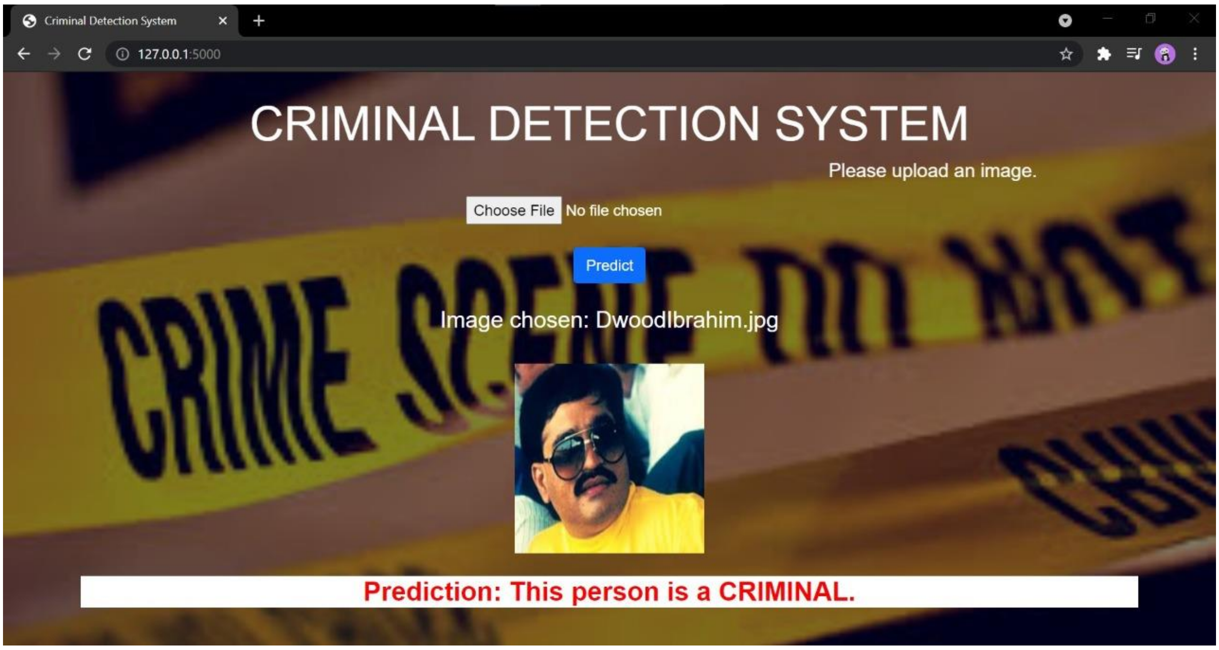
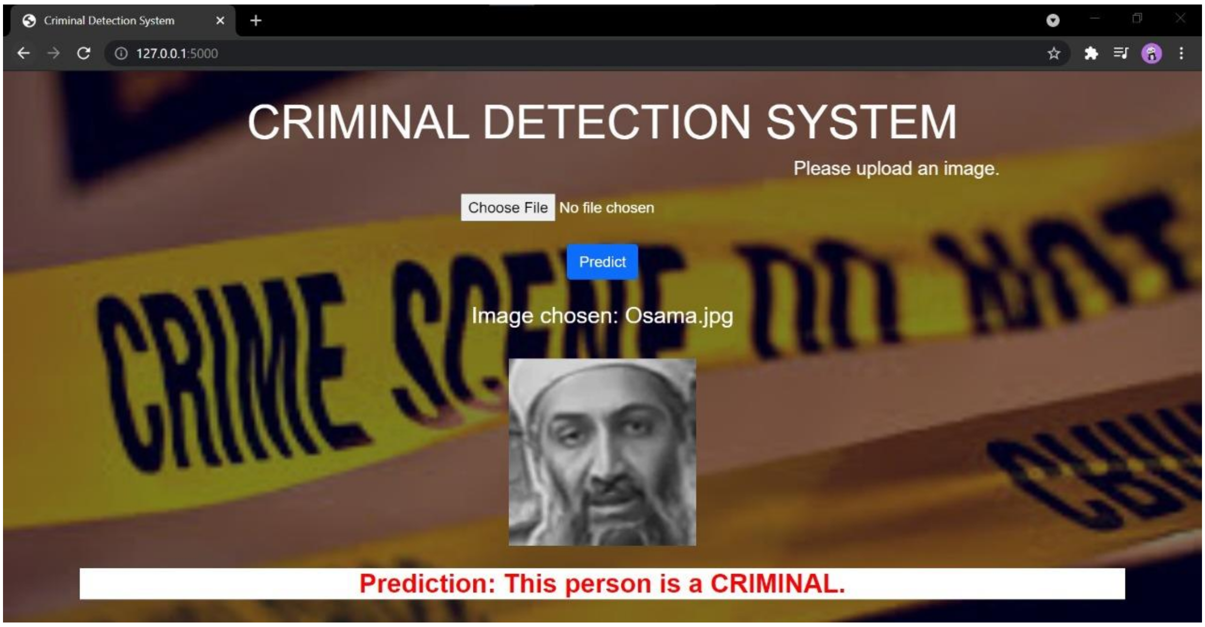

Criminal Detection System
127.0.0.1:5000
CRIMINAL DETECTION SYSTEM
Please upload an image.
Choose File No file chosen
Predict

Image chosen: DwoodIbrahim.jpg

Prediction: This person is a CRIMINAL.



CRIMINAL DETECTION SYSTEM

Please upload an image.

Image chosen: Osama.jpg

Prediction: This person is a CRIMINAL.

CHAPTER 7

TESTING

Validation was performed using various criminals and non-criminal images. Acceptance of input was tested using png and jpeg pictures.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

We think this application will be of great help to the society. It will reduce the time taken by the research departments to solve a case. It gives an accuracy of approximately 87% which makes it dependable.

FUTURE SCOPE:

- To generate a database that stores past record of the criminals.
- To make the application available on IOS
- Video format acceptance of input

CHAPTER 9

REFERENCES

1. https://www.ijert.org/research/criminal-face-recognition-system-IJERTV4IS030165.pdf
2. https://ieeexplore.ieee.org/document/9377205
3. https://ieeexplore.ieee.org/document/8819850
4. https://ijarcce.com/upload/2018/march-18/IJARCCE%2046.pdf
5. https://www.interpol.int/en/How-we-work/Forensics/Facial-Recognition