AI - CRIMEVISION: ADVANCED CRIME CLASSIFICATION WITH DEEP LEARNING

PROJECT REPORT

Submitted By

THALIR SWATHI B

KISHORE G

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

KNOWLEDGE INSTITUTE OF TECHNOLOGY

SALEM-637504

ANNA UNIVERSITY::CHENNAI 600 025

ACKNOWLEDGEMENT

At the outset, we express our heartfelt gratitude to GOD, who has been our strength to bring this project to light.

At this pleasing moment of having successfully completed our project, we wish to convey our sincere thanks and gratitude to our beloved president **Mr. C. Balakrishnan**, who has provided all the facilities to us.

We would like to convey our sincere thanks to our beloved Principal **Dr. PSS. Srinivasan**, for forwarding us to do our projected offering adequate duration in completing our project.

We express our sincere thanks to our Head of the Department **Dr. V. Kumar**, Department of Computer Science and Engineering for fostering the excellent academic climate in the Department.

We express our pronounced sense of thanks with deepest respect and gratitude to our faculty mentor **Mr. Tamilselvan A** Assistant Professor Department of Computer Science and Engineering for his valuable and precious guidance and for having amicable relation.

With deep sense of gratitude, we extend our earnest and sincere thanks to our SPOC **Mr. T. Karthikeyan**, Assistant Professor, Department of Computer Science and Engineering for their guidance and encouragement during this project.

We would also like express our thanks to all the faculty members of our department, friends and students who helped us directly and indirectly in all aspects of the project work to get completed successfully.

CHAPTER NO	TABLE OF CONTENTS TITLE	PAGE NO
	ABSTRACT	
1	**LIST OF ABBREVATIONS**	
	INTRODUCTION	03
	1.1 PROJECT OVERVIEW	
	1.2 PURPOSE	
2	**LITERATURE SURVEY**	04
	2.1 REFERENCES (SURVEY CONTENT)	
	2.2 PROBLEM DEFINITION STATEMENT	
3	**IDEATION &PROPOSED SOLUTION**	07
	3.1 EMPATHY MAP CANVAS	
	3.2 IDEATION & BRAINSTORMING	
	3.3 PROPOSED SOLUTION	
	3.4 PROBLEM STATEMENT	
4	**REQUIREMENTS ANALAYSIS**	12
	4.1 FUNCTIONAL REQUIREMENT	
	4.2 NON - FUNCTIONAL REQUIREMENT	

ABSTRACT

Crime classification and analysis play crucial roles in understanding and combating criminal activities. In recent years, advancements in deep learning techniques have shown tremendous potential in various domains, including computer vision and natural language processing.

The proposed framework utilizes deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to extract meaningful patterns and features from diverse crime-related data sources.

The first stage of the framework involves pre processing and data integration, where various data sources are collected, cleaned, and combined. Next, deep learning models are employed to learn and represent the underlying patterns within the integrated data. CNNs are applied to process visual data, extracting features from surveillance videos or images, while RNNs are utilized to analyse sequential data, such as crime incident narratives or social media posts.

The classification phase employs the trained models to classify crime incidents into specific categories or subtypes. The outputs can range from identifying specific types of crimes (e.g., burglary, assault, drug trafficking) to identifying patterns, trends, or potential links between different incidents.

To evaluate the proposed framework, extensive experiments are conducted using real-world crime datasets.

The results indicate that the advanced crime classification framework outperforms traditional approaches in terms of accuracy and efficiency. The framework enables law enforcement agencies to leverage the power of deep learning algorithms to gain valuable insights into criminal activities, aiding in proactive crime prevention and efficient resource allocation.

LIST OF ABBREVATIONS

- CNN - Convolutional Neural Network
- RNN - Recurrent Neural Network
- GUI - Graphical User Interface
- API - Application Programming
- Interface UI - User Interface
- UX - User Experience
- LSTM - Long Short Term Memory
- DBMS - Database Management System
- CV - Computer Vision
- ML - Machine Learning
- DL - Deep Learning

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

CrimeVision: Advanced Crime Classification with Deep Learning is a cutting-edge project that harnesses the power of deep learning techniques to classify and analyze criminal activities. By integrating diverse data sources such as images, videos, and textual information, CrimeVision aims to provide law enforcement agencies with an advanced tool for accurate crime classification. Through the use of state-of-the-art deep learning models, the system can learn complex patterns and features, enabling faster and more efficient crime analysis. With its real-time classification capabilities, CrimeVision enhances public safety by enabling quicker responses and proactive crime prevention measures.

1.2 PURPOSE

Purpose: CrimeVision: Advanced Crime Classification with Deep Learning aims to revolutionize the field of crime analysis by leveraging state-of-the-art deep learning techniques to accurately classify and analyze criminal activities. By harnessing the power of multi-modal data integration and advanced neural networks, the project aims to provide law enforcement agencies with a robust and efficient tool for crime classification, aiding in investigations, resource allocation, and proactive crime prevention measures.

- To Enhance Law Enforcement
- To Improve Investigation Efficiency
- To Increase Accuracy and Reliability
- To Support Decision Making
- To Assist in Crime Pattern Analysis
- To Enable Predictive Policing

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Despite the efforts of law enforcement agencies, criminal activities in crowded areas still pose a significant threat to public safety. This results in delays in response time and increased risk to public safety. Therefore, there is a need for an advanced crime detection system that can accurately and quickly detect multiple criminal activities happening simultaneously in crowded areas. The system should leverage the latest technologies in computer vision, machine learning, and data analytics to enable law enforcement agencies to respond quickly and effectively to criminal activities, prevent crimes, and ensure public safety.

2.2 SURVEY WORK

2.2.1 PERFORMANCE ANALYSIS OF MOST PROMINENT MACHINE LEARNING AND DEEP LEARNING ALGORITHMS IN CRIME NEWS ARTICLES

[Ariful Islam et al., 2020]

This work is dedicated to Crime Type Classification. As very few works had been done for crime classifier. To carry out this research, first we have developed a crime dataset which contains around 24,295 news articles and made most of them publicly available at Github. Then we have built our crime classifier model and trained the classifier with our own dataset. We have analysed word vectors like bag of words, TF-IDF in state-of-art machine learning algorithms as well as most promising semantic and syntactic word embeddings like Word2Vec, GloVe, fast-Text in both shallow and deep CNN and RNN to select best word embeddings. Finally we have summarized the experimental result in tabular form. We can see that significant improved accuracy can be achieved using deep learning algorithms over state-of-art machine learning algorithms in classifying crime data. The final experimental result shows that the proposed model is able to achieve 93.70% accuracy.

2.2.2 INTRUSION DETECTION AND CLASSIFICATION BASED ON DEEP LEARNING

[Habibe GUler et al., 2020]

Cyberattacks aiming to disrupt the confidentiality, integrity and availability of systems by penetrating the network infrastructure of organizations are becoming increasingly widespread. These attacks carried out by attackers cause anomalies in normally functioning networks. Detection of these intrusions have of great importance in the protection of networks. Basically, Network Intrusion Detection Systems are tools that prevent and detect malicious activities or policy violations against networks by monitoring network traffic. In the scope of this study, supervised learning classification-based RNN, LSTM and GRU algorithms for intrusion detection on networks are applied comparatively on the UNSW-NB15 dataset.

2.2.3 EXAMINING DEEP LEARNING ARCHITECTURES FOR CRIME CLASSIFICATION AND PREDICTION

[Petros Daras et al., 2021]

"Examining Deep Learning Architectures for Crime Classification and Prediction" focuses on exploring and analysing the effectiveness of various deep learning architectures in the domain of crime classification and prediction. Through this project, we aim to investigate the potential of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and other advanced deep learning models in accurately classifying and predicting criminal activities. By utilizing diverse crime datasets and evaluating the performance of different architectures, we strive to provide valuable insights into the development of robust and efficient deep learning models for crime analysis, enabling law enforcement agencies to enhance their crime prevention and investigation capabilities.

2.2.4 SUPERVISED LEARNING BASED ON LOCAL RECURRENT SPIKING NEURAL NETWORKS

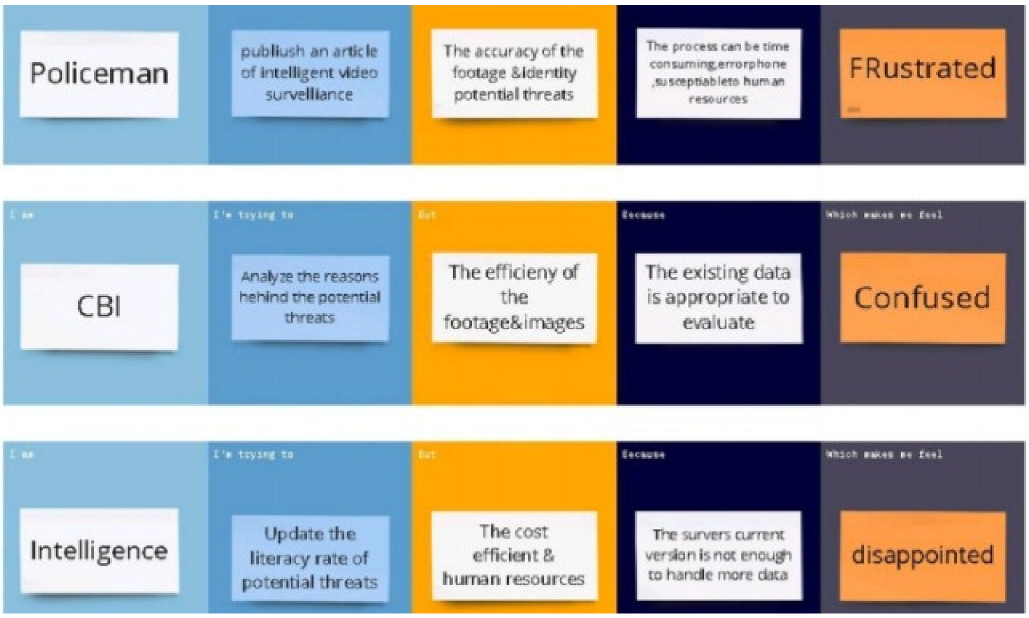
[Yuping Zhang et al., 2021]

Supervised Learning based on Local Recurrent Spiking Neural Networks" is a cutting-edge topic in the field of neural networks and machine learning. This approach focuses on using spiking neural networks (SNNs), a biologically-inspired model of computation, for supervised learning tasks. Unlike traditional artificial neural networks, SNNs mimic the behaviour of neurons in the brain, where information is encoded and processed through the timing of spikes.

In this context, the emphasis is on local recurrent connections within the SNN architecture. Local recurrence allows neurons to communicate and exchange information within their local neighbourhoods, enabling more complex and dynamic computations. By leveraging this property, supervised learning algorithms can be developed to train SNNs on labelled datasets, enabling the network to learn and classify patterns with improved accuracy.

2.3 PROBLEM STATEMENT DEFINITION

Devloping an accurate deep learning model for action recognition to classify violent crimes in real-time from surveillance footage. The goal is to detect and classify violent actions such as punching, stabbing, or shooting, to provide law enforcement with real-time alerts for violent crimes in progress.



CHAPTER -3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP

Fig 3.1.1 - Empathy Map

3.2 IDEATION AND BRANISTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Fig 3.2.1.a - Brainstorming and Idea Prioritization

Step-2: Brainstorm, Idea Listing and Grouping

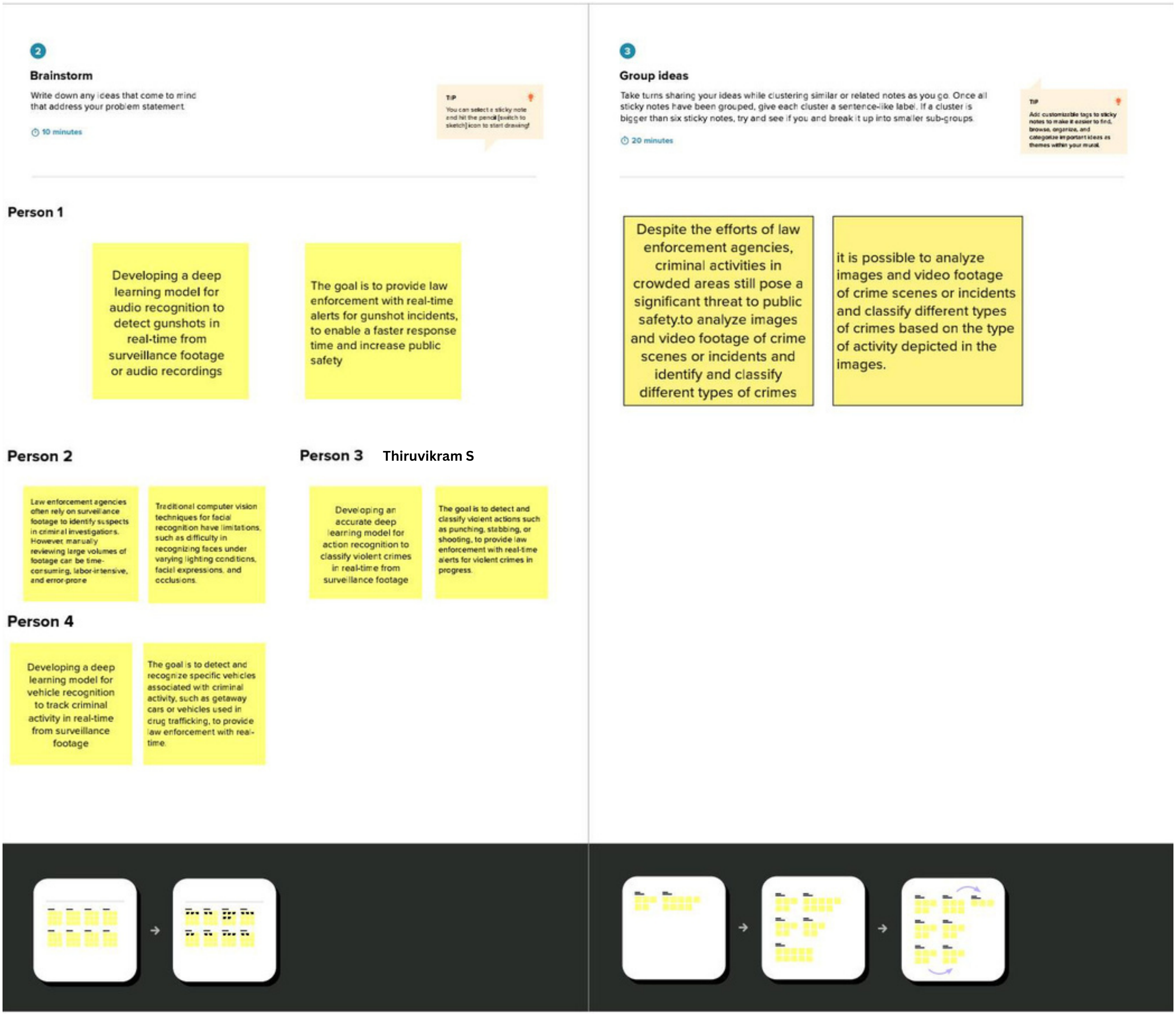


Fig 3.2.1.b - Brainstorming and Idea Prioritization

Step-3: Idea Prioritization

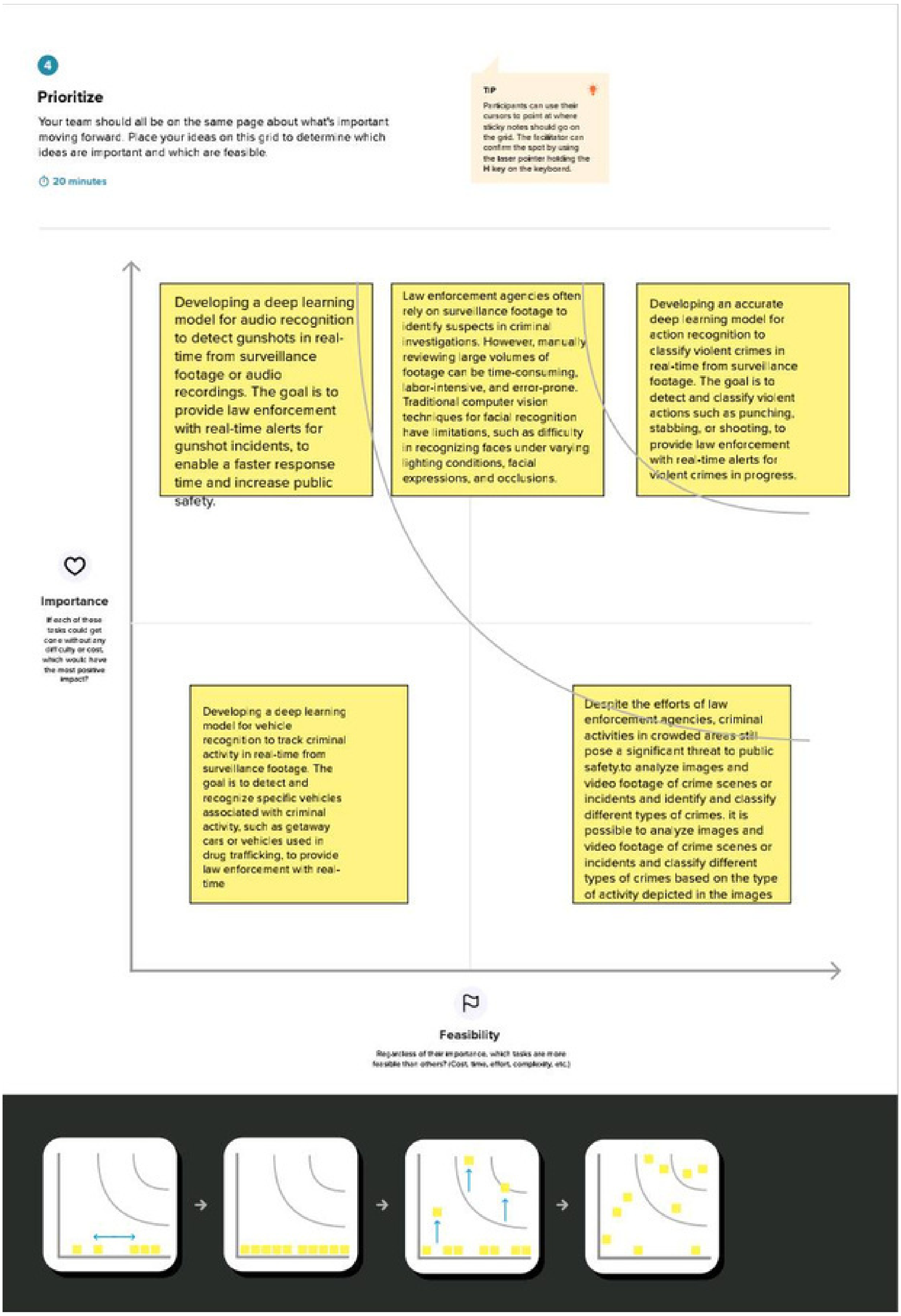


Fig 3.2.1.c - Brainstorming and Idea Prioritization

3.3 PROPOSED SOLUTION

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Despite the efforts of law enforcement agencies, criminal activities in crowded areas still pose a significant threat to public safety. The current crime detection systems are limited in their ability to detect multiple criminal activities happening simultaneously in crowded areas. This results in delays in response time and increased risk to public safety.
2.	Idea / Solution description	There is a need for an advanced crime detection system that can quickly and accurately detect multiple criminal activities in crowded areas. The system uses various technologies, such as computer vision, machine learning and data analytics.
3.	Novelty / Uniqueness	Focus on addressing a specific problem area where current crime detection systems fall short.
4.	Social Impact / Customer Satisfaction	Social Impact: Before: Frustrated & Hopeless After : Hopeful & Confident With a more efficient and effective crime detection system in place, people will feel safer while visiting public places and will be more likely to spend time and money there .
5.	Business Model (Revenue Model)	The system can be sold as a subscription-based model or pay-per-use model, depending on the requirements of the customers. The revenue can be generated through advertising, sponsorships or partnerships with related businesses or organizations.
6.	Scalability of the Solution	Ability to implement it in different locations and regions. The system can be deployed in high-density areas such as malls, transportation hubs, stadiums and other public places. And to cover a larger area and provide a wider range of services.

CHAPTER -4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Input	Allow user to analyse the stored videos in DBMS through GUI
FR-2	Model and Pre-processing	A file that has been trained to recognize certain type of videos. The app would use computer vision algorithms to detect objects, such as people or vehicles, in the video frames.
FR-3	Detection	The app would use CV algorithms to detect objects, such as people or vehicles, in the video frames. Object detection algorithms such as YOLO or CNN could be used.
FR-4	Evaluation	This step done by using machine learning algorithms to identify suspicious behaviour or anomalies in the video footage. Techniques such as anomaly detection, clustering, or classification could be used for behaviour analysis.
FR-5	Report Generation	The report include details about the objects detected, their movements, and any suspicious behaviour observed. The app would use tools such as Python, R, or MATLAB to generate and present it in various formats such as PDF, HTML.

4.2 NON - FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Highly usable and beneficial for various stakeholders in the field of law enforcement and public safety. Useful in Improved Crime Detection, Enhanced Investigation , Increased Public Safety.
NFR-2	**Security**	It deals with sensitive data and information. Also Includes Secure Video Storage, Secure Data Transmission(SSL,TLS), User Authentication, Data Privacy(GDPR,CCPA), Regular Security Audits.
NFR-3	**Reliability**	The reliability depends on the quality of the hardware and software, the accuracy of the algorithms relevant data, Availability of Relevant Data, Technical Infrastructure, User Training and Education.
NFR-4	**Performance**	The performance can be evaluated based on the speed and accuracy of video analysis, the ability to detect and prevent crime, and the app's scalability and flexibility.
NFR-5	**Availability**	The crime vision apps can available to law enforcement agencies or other authorized users, while others may be available to the general public.
NFR-6	**Scalability**	A scalable crime vision system can handle large volumes of data and users, without sacrificing performance or reliability.

CHAPTER -5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

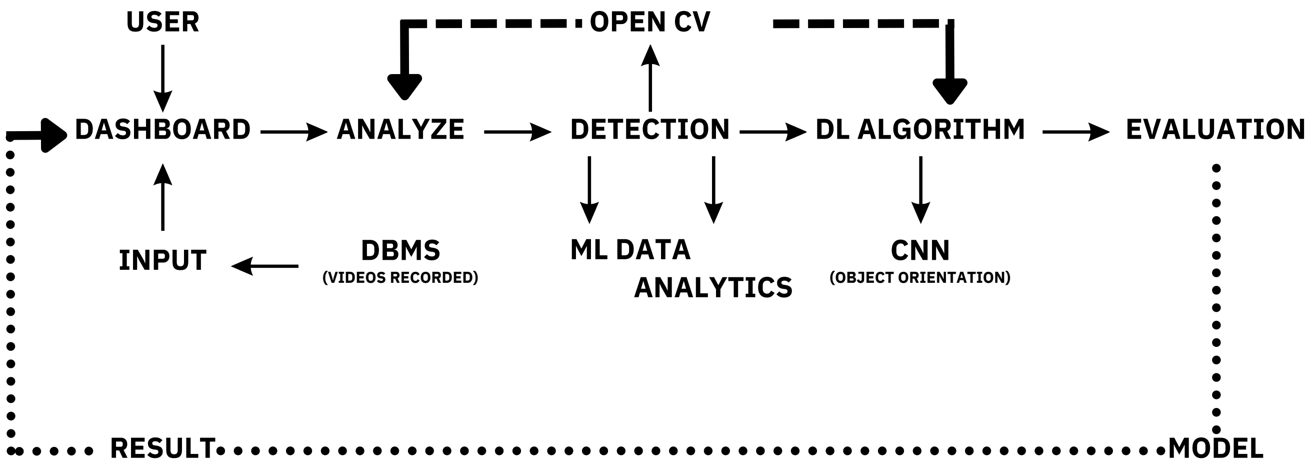


Fig 5.1.1 - Data Flow Diagram

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

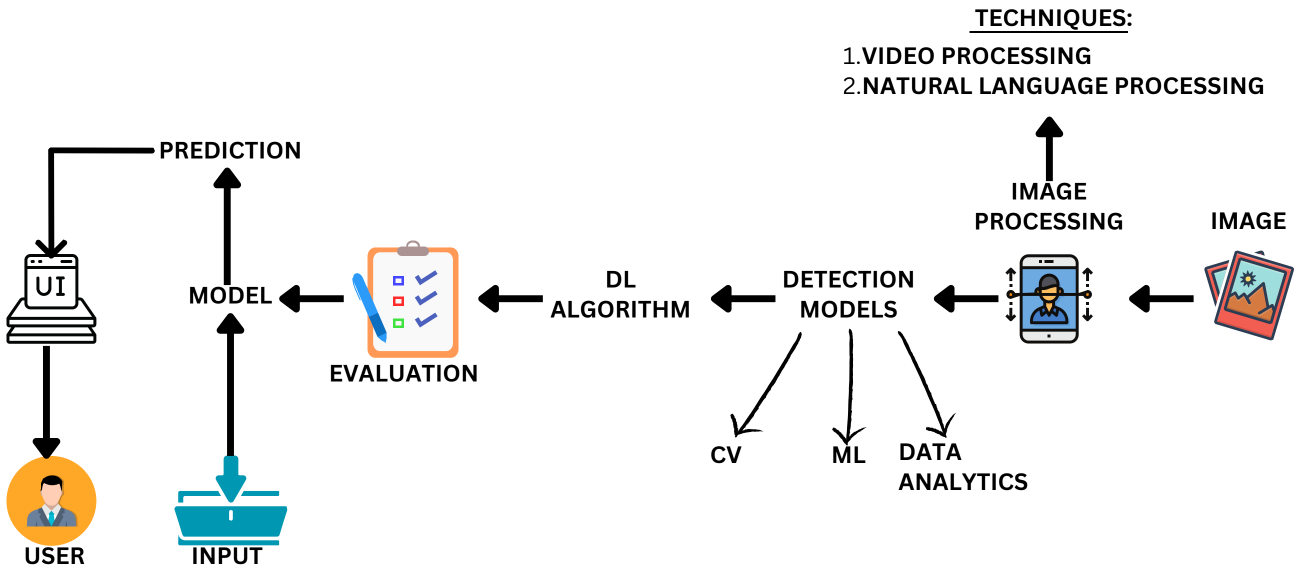


Fig 5.2.1 - Technical Architecture

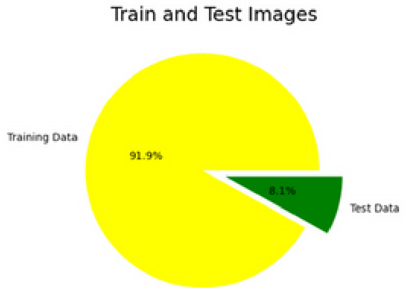
5.3 USER STORIES

User Type	Functional Requirement (EPIC)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Team Member
Policeman /Cop	Sign Up/Sign In	USN 1	In this page user can login/signup to use the app.	I can view the Login page.	Low	Hari
	Home/Dashboard	USN 2	In the Homepage, User can view the guidelines of how to use the app. In the Dashboard, User can view icons like Area, CCTV's & Crimes.	Guidelines & Surveys about area.	Low	Hari
	Complaints /Queries	USN 3	In the Complaints/Queries, User can view Complaints and Queries from public.	I can view Complaints & Queries	Medium	Hari
	Analyse	USN 4	Based on the requirements, User can select Video Footage.	I can analyse the uploaded videos on dashboard	High	Hari
	Detection	USN 5	By analysing the footages DL Algorithm detects the crime.	I can Detect the crime using DL Algorithm	High	Hari
	Evaluation	USN 6	After Detecting the crime, DL Algorithm can provide the accuracy of the crime.	Based on the detection, I can view the evaluation of crime.	High	Hari
	Result	USN 7	As a User, I can get the predicted output by GUI.	At Last, I can get the overview of the crime with the accurate rate.	High	Hari

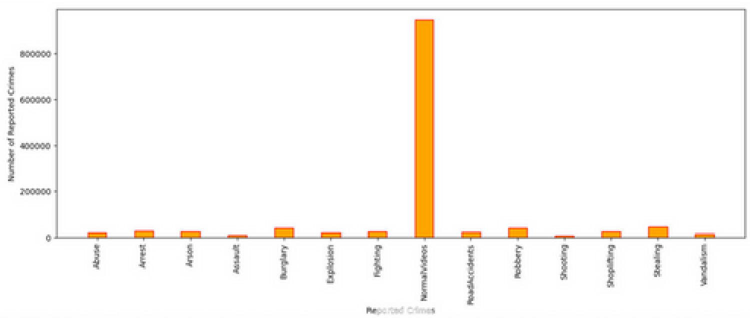
CHAPTER - 6

CODING & SOLUTIONING

```
In [6]: plt.figure(figsize=(8,5))
        plt.pie(x=np.array([train, test]), autopct="%.1f%%", explode=[0.1, 0.1],
                labels=["Training Data", "Test Data"], pctdistance=0.5,colors=['yellow', 'green'])
        plt.title("Train and Test Images", fontsize=18);
```




```
In [7]: import matplotlib.pyplot as plt

        plt.figure(figsize=(15,5))

        plt.bar(list(crimes.keys()), list(crimes.values()), width=0.4,align="center", edgecolor=['red'],color=['orange'])
        plt.xticks(rotation=90)

        plt.xlabel("Reported Crimes")
        plt.ylabel("Number of Reported Crimes")
        plt.show()
```




```
In [8]: train_set= image_dataset_from_directory(
            train_dir,
            label_mode="categorical",
            batch_size=BATCH_SIZE,
            image_size=IMG_SHAPE,
            shuffle=True,
            seed=seed,
            validation_split=0.2,
            subset="training",
        )

        Found 1266345 files belonging to 14 classes.
        Using 1013076 files for training.
```

```
In [46]: def transfer_learning(input_shape):
             base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=input_shape)
             return base_model

         def create_model(input_shape, num_classes):
             model = Sequential()
             base_model = transfer_learning(input_shape)
             model.add(base_model)

             model.add(GlobalAveragePooling2D())

             model.add(Dense(256, activation="relu"))
             model.add(Dropout(0.2))

             model.add(Dense(512, activation="relu"))
             model.add(Dropout(0.2))

             model.add(Dense(1024, activation="relu"))

             model.add(Dense(num_classes, activation="softmax"))

             model.summary()

             return model

         input_shape = (224, 224, 3)
         num_classes = 10
         model = create_model(input_shape, num_classes)

         model.compile(optimizer="adam",
                       loss='categorical_crossentropy',
                       metrics=['accuracy'])
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_224 (Funct  (None, 7, 7, 1280)       2257984
 ional)

 global_average_pooling2d_3   (None, 1280)             0
 (GlobalAveragePooling2D)

 dense_3 (Dense)             (None, 256)               327936

 dropout (Dropout)           (None, 256)               0

 dense_4 (Dense)             (None, 512)               131584

 dropout_1 (Dropout)         (None, 512)               0

 dense_5 (Dense)             (None, 1024)              525312

 dense_6 (Dense)             (None, 10)                10250

=================================================================
Total params: 3,253,066
Trainable params: 3,218,954
Non-trainable params: 34,112
_________________________________________________________________
```

```
In [17]: import tensorflow as tf
         from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input
         from tensorflow.keras.preprocessing.image import ImageDataGenerator
         from tensorflow.keras.layers import Dense

         EPOCHS = 1
         BATCH_SIZE = 24

         train_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Train"
         val_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Train"

         train_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
         train_set = train_datagen.flow_from_directory(
             train_dir,
             target_size=(64, 64),
             batch_size=BATCH_SIZE,
             class_mode='categorical'
         )

         val_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
         val_set = val_datagen.flow_from_directory(
             val_dir,
             target_size=(64, 64),
             batch_size=BATCH_SIZE,
             class_mode='categorical'
         )

         n = train_set.num_classes

         input_shape = (224, 224, 3)
         base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=input_shape)
         model = tf.keras.Sequential([
             base_model,
             tf.keras.layers.GlobalAveragePooling2D(),
             Dense(n, activation='softmax')
         ])
         model.summary()

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

         history = model.fit(x=train_set, validation_data=val_set, epochs=EPOCHS)
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_224 (Funct  (None, 7, 7, 1280)       2257984
 ional)

 global_average_pooling2d (G  (None, 1280)             0
 lobalAveragePooling2D)

 dense (Dense)               (None, 14)                17934

=================================================================
Total params: 2,275,918
Trainable params: 2,241,806
Non-trainable params: 34,112
_________________________________________________________________
52765/52765 [==============================] - ETA: 0s - loss: 0.1091 - accuracy: 0.9699
```

```
In [18]: model.save('crime.h5')
```

```
In [19]: from tensorflow.keras.models import load_model
         model.load_weights('crime.h5')
```

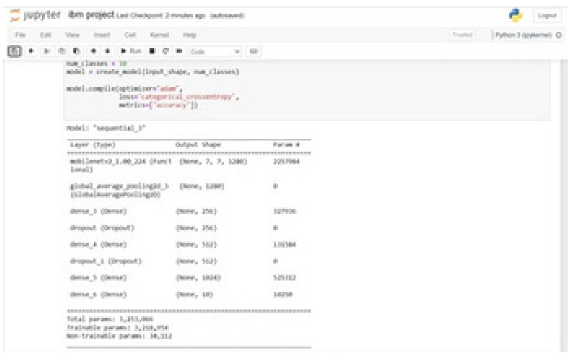
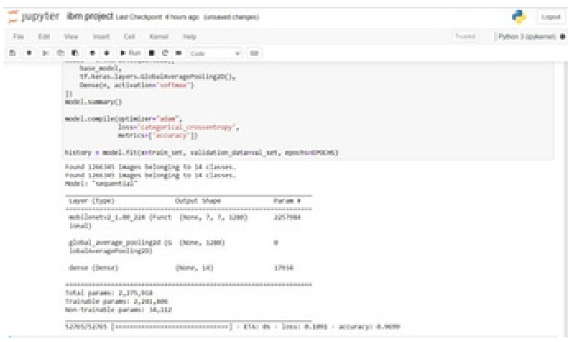
```
In [24]: test_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Test"

         test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
         test_set = test_datagen.flow_from_directory(
             test_dir,
             target_size=(64, 64),
             batch_size=BATCH_SIZE,
             class_mode='categorical',
             shuffle=False
         )

         import numpy as np

         y_true = np.array([])

         for x, y in test_set:
             y_true = np.concatenate([y_true, np.argmax(y, axis=-1)])
```

```
Found 111308 images belonging to 14 classes.
```

CHAPTER - 7

RESULT

Date	19 May 2023
Team ID	NM2023TMID16648
Project Name	CrimeVision: Advanced Crime Classification with Deep Learning

S. No	Parameter	Value	Screenshot
01	Model Summary		
02	Accuracy	Training Accuracy & Validation Accuracy	

CHAPTER - 8

ADVANTAGES

Accurate crime detection: Deep learning models, such as convolutional neural networks (CNNs), excel at image recognition tasks. They can effectively analyze visual data and accurately classify crime-related images or videos, enabling the detection of criminal activities more reliably.

Real-time monitoring: Deep learning models can process visual data in real-time, making it possible to monitor live video feeds and identify potential criminal activities as they occur.

Improved crime prevention: By analyzing patterns and trends in crime-related images or videos, deep learning algorithms can provide valuable insights for crime prevention strategies.

Enhanced investigative capabilities: Deep learning-powered crime vision classification can assist investigators in identifying suspects or related individuals by matching visual data against existing databases.

Scalability and adaptability: Deep learning models can be trained on large datasets and can learn to recognize various types of crimes.

DISADVANTAGES

Bias and fairness concerns: Deep learning models are trained on large datasets that may contain biases and imbalances. If the training data is not representative of the entire population or is biased towards certain demographics, the model's predictions can be biased as well, leading to unfair or discriminatory outcomes.

Lack of interpretability: Deep learning models are often considered black boxes, meaning it can be difficult to understand how they arrive at their predictions. This lack of interpretability can be problematic in the context of crime classification, as it becomes challenging to explain or justify the reasoning behind the model's decisions.

High computational requirements: Deep learning models typically require significant computational power and resources, including GPUs, to train and deploy. Running and maintaining these models can be costly, especially for smaller organizations or individuals with limited resources.

CHAPTER - 9

CONCLUSION

Deep learning techniques have shown great potential in advancing crime classification and improving the accuracy of crime vision systems. By leveraging deep neural networks and training them on large datasets, researchers have made significant progress in automating the process of identifying and categorizing criminal activities.

The use of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has allowed for the extraction of meaningful features from crime-related data, including images, videos, and textual information. These models can effectively capture complex patterns and relationships, enabling accurate classification of various types of crimes.

In conclusion, advanced crime classification using deep learning has shown promising results in automating crime vision systems and improving the accuracy of crime detection. While there are challenges to overcome, further research and development in this field hold the potential to enhance law enforcement efforts, assist in crime prevention, and contribute to building safer communities.

CHAPTER - 10

FUTURE SCOPE

Enhanced accuracy: As deep learning models continue to evolve, their accuracy in crime classification is likely to improve. With more refined architectures, better training algorithms, and increased availability of labeled datasets, these models can achieve higher precision and recall rates, leading to more reliable crime classification outcomes.

Real-time crime analysis: Deep learning models can be integrated into surveillance systems to provide real-time crime analysis.

Multi-modal crime classification: Deep learning models can be trained to analyze multiple modalities of data, such as images, videos, audio recordings, and text descriptions.

Privacy-preserving techniques: Addressing privacy concerns, future developments may focus on the development of privacy-preserving techniques for deep learning models.

CHAPTER - 11

APPENDIX

A.1 SOURCE CODE

```
train = r"C:\Users\vikra\OneDrive\Documents\Train"
test = r"C:\Users\vikra\OneDrive\Documents\Test"
```

#IMPORTING LIBRARIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import os

import tensorflow as tf
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.applications import DenseNet121
from sklearn.preprocessing import LabelBinarizer
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout, MaxPooling2D, Conv2D, Flatten
from tensorflow.keras.models import Sequential
from IPython.display import clear_output
import warnings
warnings.filterwarnings('ignore')

seed = 12
IMG_HEIGHT = 64
IMG_WIDTH = 64
IMG_SHAPE = (64,64)
BATCH_SIZE = 24
EPOCHS = 1
LR = 0.00003

import os

train_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Train"
crime_types = os.listdir(train_dir)
n = len(crime_types)
print("Number of crime categories:", n)
plt.figure(figsize=(8,5))
plt.pie(x=np.array([train, test]), autopct="%.1f%%", explode=[0.1, 0.1],
 labels=["Training Data", "Test Data"], pctdistance=0.5,colors=['yellow', 'green'])
plt.title("Train and Test Images", fontsize=18);
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(15,5))

plt.bar(list(crimes.keys()), list(crimes.values()), width=0.4,align="center", edgecolor=['red'],color=['orange'])
plt.xticks(rotation=90)

plt.xlabel("Reported Crimes")
plt.ylabel("Number of Reported Crimes")
plt.show()


train_set= image_dataset_from_directory(
train_dir,
label_mode="categorical",
batch_size=BATCH_SIZE,
image_size=IMG_SHAPE,
shuffle=True,
seed=seed,
validation_split=0.2,
subset="training",
)


import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Train"
BATCH_SIZE = 24
IMG_SHAPE = (224, 224)
seed = 42

datagen = ImageDataGenerator(
 rescale=1./255,
 validation_split=0.2
)

val_set = datagen.flow_from_directory(
 train_dir,
 target_size=IMG_SHAPE,
 batch_size=BATCH_SIZE,
 shuffle=True,
 seed=seed,
 subset="validation"
)
```

#MODEL SUMMARY

```
def transfer_learning(input_shape):
 base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=input_shape)
 return base_model

def create_model(input_shape, num_classes):
 model = Sequential()
 base_model = transfer_learning(input_shape)
 model.add(base_model)

 model.add(GlobalAveragePooling2D())

 model.add(Dense(256, activation="relu"))
 model.add(Dropout(0.2))

 model.add(Dense(512, activation="relu"))
 model.add(Dropout(0.2))

 model.add(Dense(1024, activation="relu"))

 model.add(Dense(num_classes, activation="softmax"))

 model.summary()

 return model

input_shape = (224, 224, 3)
num_classes = 10
model = create_model(input_shape, num_classes)

model.compile(optimizer="adam",
 loss='categorical_crossentropy',
 metrics=['accuracy'])
```

#TRAINING A DATASET

```
import tensorflow as tf
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense

EPOCHS = 1
BATCH_SIZE = 24

train_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Train"
val_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Train"

train_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
train_set = train_datagen.flow_from_directory(
 train_dir,
 target_size=(64, 64),
 batch_size=BATCH_SIZE,
 class_mode='categorical'
)

val_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
val_set = val_datagen.flow_from_directory(
 val_dir,
 target_size=(64, 64),
 batch_size=BATCH_SIZE,
 class_mode='categorical'
)

n = train_set.num_classes

input_shape = (224, 224, 3)
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=input_shape)
model = tf.keras.Sequential([
 base_model,
 tf.keras.layers.GlobalAveragePooling2D(),
 Dense(n, activation='softmax')
])
model.summary()

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

history = model.fit(x=train_set, validation_data=val_set, epochs=EPOCHS)
```

#TESTING A DATASET

```
test_dir = "C:\\Users\\vikra\\OneDrive\\Documents\\Test"

test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
test_set = test_datagen.flow_from_directory(
 test_dir,
 target_size=(64, 64),
 batch_size=BATCH_SIZE,
 class_mode='categorical',
 shuffle=False
)

import numpy as np

y_true = np.array([])

for x, y in test_set:
 y_true = np.concatenate([y_true, np.argmax(y, axis=-1)])
```

A.2 WEB GUI

```
<!DOCTYPE html>
<html>
 <head class="colorinform">
 <link rel="stylesheet" href="style.css">
 <title>iWitness-Login</title>
 </head>
 <body class="colorinform">
<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>

<div class="form-main" style="font-size: large;"> <p ><b>WELCOME!</b></p> <form >
 <label>Mobile Number:</label>
 <br/>
 <input type="tel" placeholder="Enter Mobile Number" minlength="10" maxlength="10"><br/>
 <label>Password:</label>
 <br/>
 <input type="password" maxlength="12" minlength="6" placeholder="Enter Password"><br/><br/>




</form>
<a href="mainpage.html"><button type="button" > Sign in</button></a>
<p><a href="mainsignup.html">Create account</a></p>
</div>
 </body>
</html>
```

```html
<!DOCTYPE html>
<html>
 <head class="signcolor">
 <link rel="stylesheet" href="style.css">
 <title>Sign up</title>
 </head >

 <body class="signcolor"><br/><br/><br/><br/><br/>
 <form class="mainsignup">
 <label>First Name:<span style="color: red;">*</span></label><br/>
 <input type="text" placeholder="First Name as per Aadhar" maxlength="25" required><br/><br/>
 <label>Last Name:</label><br/>
 <input type="text" placeholder="Last Name" maxlength="3"><br/><br/>
 <label>Gender:<span style="color: red;">*</span></label><br/>
 <input type="radio" name="Gender">
 <label>Male</label>
 <input type="radio" name="Gender">
 <label>Female</label>
 <input type="radio" Name="Gender">
 <label>Others</label><br/><br/>
 <label>Aadhar Number:<span style="color: red;">*</span></label><br/>
 <input type="tel" maxlength="12" placeholder="Enter Your Aadhar" required><br/><br/>
 <label>Mobile Number:<span style="color: red;">*</span></label><br/>
 <input type="tel" maxlength="10" placeholder="Enter Mobile Number"><br/><br/>
 <label>Password:<span style="color: red;">*</span></label><br/>
  <input  placeholder="Enter  Password"  minlength="8"  maxlength="15"  type="password"  required><br/><br/>
 <label>Confirm Password:<span style="color: red;">*</span></label><br/>
 <input placeholder="Confirm Password" minlength="8" maxlength="15" type="password"><br/><br/><br/>

 <a href="signup.html"><button type="button" style="color: blue;"> Create</button></a>

 </form>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head class="color">
<title>iWitness</title>
<link rel="stylesheet" href="style.css">
</head>
<body class="color">
 <header><nav>
 <ul>
 <li><a href=mainpage.html>Home</a></li>
 <li><a href="complaints.html">Complaints</a></li>
 <li><a href=profile.html>My Profile</a></li>
 <li><a href="about.html">About</a></li>
 <li><a href="help.html">Help</a></li>
 </ul>
 <img src="D:\IBM\images\png_20230516_211610_0000.png" height="150" width="400" alt="logo"
class="centre">
</nav>
 </header>
<main>
<div>
 <section>
 <p class="para">This portal is a platform for Citizens to file crime related complaints online and seek antecedent
verification of prospective employees (including for domestic help, drivers etc.), tenants or for any other purpose.
Citizens can also seek certification of their own antecedents.
 </p>
 </section>
 <img src="D:\IBM\images\Kallakurichi_Arrests_Kuravar_171121_1200x800.jpeg" width="600" height="400"
class="image"><br/>
 <p class="q" style="font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;">"With no warrants, TN
police barge into homes of Kuravar men, take them into custody."</p>

<img src="D:\IBM\images\gurgaon.jpg" width="600" height="400" class="image"><br/>
 <p class="q" style="font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;">"The accused have been
identified as Asif Hussain, K Jirvani Babu, Moh
 ammad Azad, Mohammad Karim and Sonu."</p>


</div>
</main>

<footer>
 <br/><br/><br/>
 <p>&#169; Copyrights iWitness,2023 &#124; All Rights Reserved &#124; Mail us at<a
href="mailto:iwitness@yahoo.com">  Send Mail</a></p>
</footer>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
 <head class="color">
 <link rel="stylesheet" href="style.css">
 <title>Prediction</title>
 </head >

 <body class="crimebody" ><p>
 <img src="D:\IBM\images\istockphoto-921664276-612x612.jpg" alt="crime" height="600" width="600"
class="crime">
 <label>
 <br><br> Drop in the image to get the prediction<br/><br/><br/>
 </label>
 <input type="file" name="Choose file" style="color: red;">
</p>

</body>
</html>
```