

PROJECT DEVELOPMENT PHASE

PROJECT MANUAL

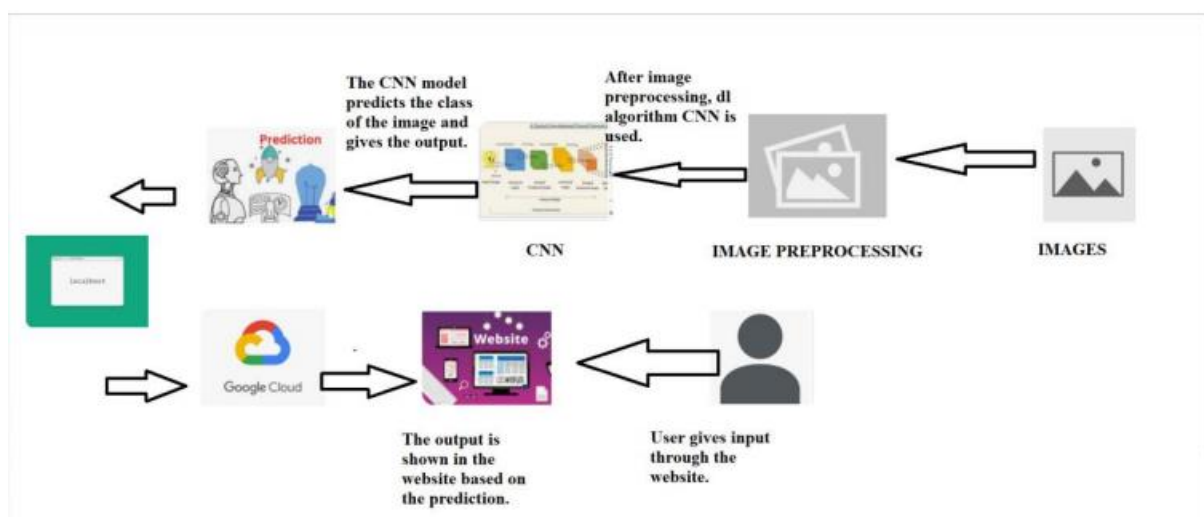
DATE	31-10-2023
TEAM ID	TEAM-593175
PROJECT	CRIME VISION: ADVANCED CRIME CLASSIFICATION USING DEEP LEARNING

PROJECT:

CRIME VISION: ADVANCED CRIME CLASSIFICATION USING DEEP LEARNING

Crime Vision using Deep Learning is a technique that involves deep learning, to analyse images and video footage of crime scenes or incidents and identify and classify different types of crimes. This is very useful for criminal justice and law enforcement contexts, including crime scene investigation, forensic analysis, and surveillance. This can allow law enforcement agencies to develop strategies and interventions to prevent crime

TECHNICAL ARCHITECTURE:



PROJECT FLOW:

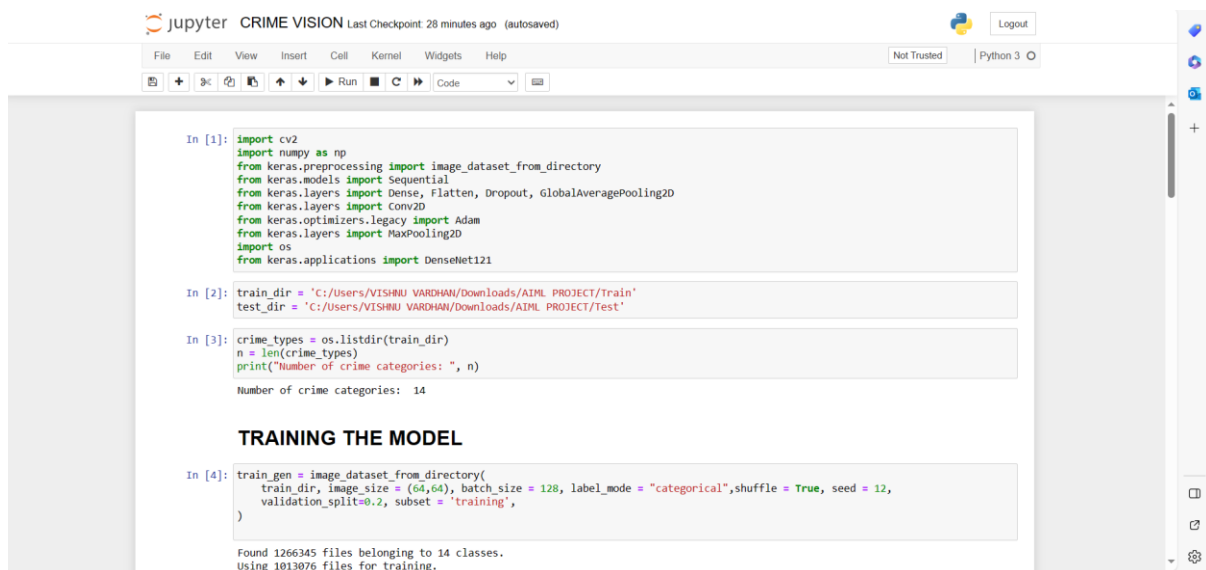
- The user interacts with the UI to choose an image.
- The chosen image is processed by a transfer learning deep learning model.
- The transfer learning model is integrated with a Flask application.
- The transfer learning model analyses the image and generates predictions.
- The predictions are displayed on the Flask UI for the user to see.
- This process enables users to input an image and receive accurate predictions quickly.

DATA COLLECTION:

DATASET LINK:

<https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset>

DATA PREPROCESSING AND IMPORTING LIBRARIES:



```
In [1]: import cv2
import numpy as np
from keras.preprocessing import image_dataset_from_directory
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout, GlobalAveragePooling2D
from keras.layers import conv2D
from keras.optimizers import Adam
from keras.layers import MaxPooling2D
import os
from keras.applications import DenseNet121

In [2]: train_dir = 'C:/Users/VISHNU VARDHAN/Downloads/AI ML PROJECT/Train'
test_dir = 'C:/Users/VISHNU VARDHAN/Downloads/AI ML PROJECT/Test'

In [3]: crime_types = os.listdir(train_dir)
n = len(crime_types)
print("Number of crime categories: ", n)

Number of crime categories: 14

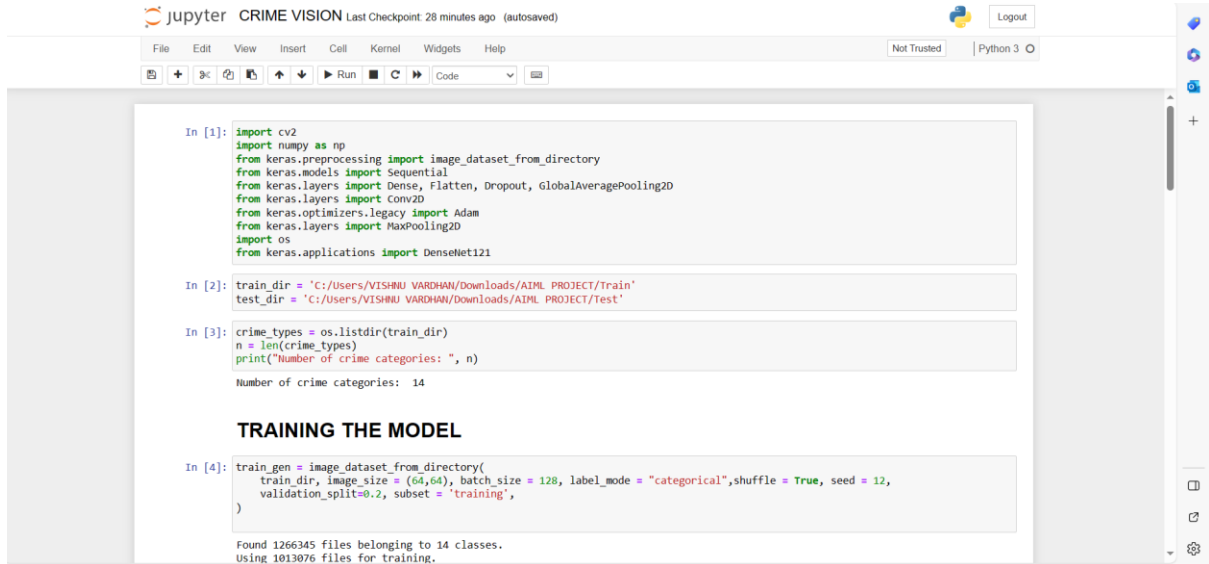
TRAINING THE MODEL

In [4]: train_gen = image_dataset_from_directory(
    train_dir, image_size = (64,64), batch_size = 128, label_mode = "categorical", shuffle = True, seed = 12,
    validation_split=0.2, subset = "training",
)
```

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

Global average pooling 2D (GAP 2D): It is a type of pooling operation commonly used in convolutional neural networks (CNNs) for image classification tasks.

TRAINING THE MODEL:



```
In [1]: import cv2
import numpy as np
from keras.preprocessing import image_dataset_from_directory
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout, GlobalAveragePooling2D
from keras.layers import Conv2D
from keras.optimizers import Adam
from keras.layers import MaxPooling2D
import os
from keras.applications import DenseNet121

In [2]: train_dir = 'C:/Users/VISHNU VARDHAN/Downloads/AI/ML PROJECT/Train'
test_dir = 'C:/Users/VISHNU VARDHAN/Downloads/AI/ML PROJECT/Test'

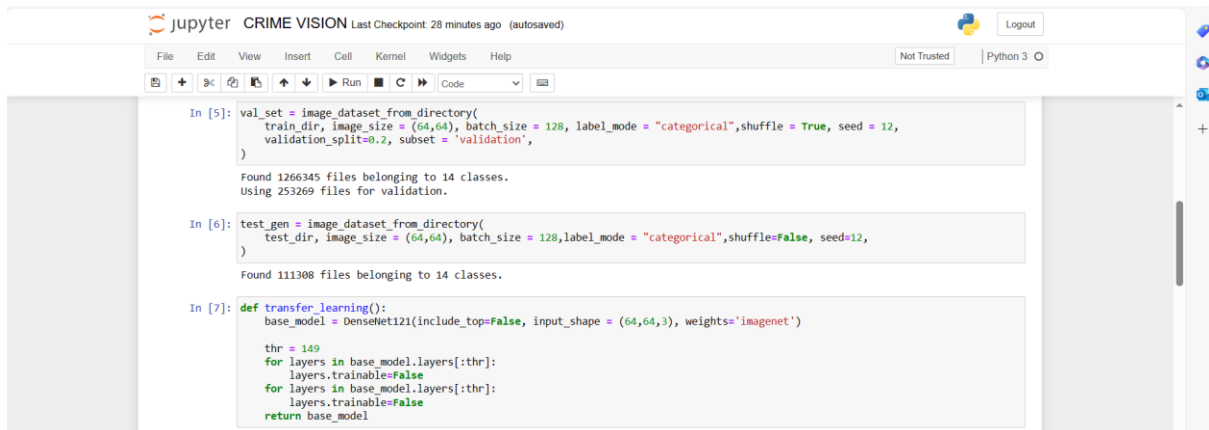
In [3]: crime_types = os.listdir(train_dir)
n = len(crime_types)
print("Number of crime categories: ", n)

Number of crime categories: 14

TRAINING THE MODEL

In [4]: train_gen = image_dataset_from_directory(
    train_dir, image_size = (64,64), batch_size = 128, label_mode = "categorical", shuffle = True, seed = 12,
    validation_split=0.2, subset = 'training',
)

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



```
In [5]: val_set = image_dataset_from_directory(
    train_dir, image_size = (64,64), batch_size = 128, label_mode = "categorical", shuffle = True, seed = 12,
    validation_split=0.2, subset = 'validation',
)

Found 1266345 files belonging to 14 classes.
Using 253269 files for validation.

In [6]: test_gen = image_dataset_from_directory(
    test_dir, image_size = (64,64), batch_size = 128, label_mode = "categorical", shuffle=False, seed=12,
)

Found 111308 files belonging to 14 classes.

In [7]: def transfer_learning():
    base_model = DenseNet121(include_top=False, input_shape = (64,64,3), weights='imagenet')

    thr = 149
    for layers in base_model.layers[:thr]:
        layers.trainable=False
    for layers in base_model.layers[thr]:
        layers.trainable=False
    return base_model
```

The image shows a Jupyter Notebook titled "CRIME VISION" with a last checkpoint 28 minutes ago. The notebook contains two code cells. The first cell defines a function `create_model()` that builds a Keras model. The model starts with a `Sequential` container, followed by a `transfer_learning()` layer. It then adds a `GlobalAveragePooling2D` layer, a `Dense(256, activation='relu')` layer with `Dropout(0.2)`, another `Dense(512, activation='relu')` layer with `Dropout(0.2)`, a `Dense(1024, activation='relu')` layer, and finally a `Dense(n, activation='softmax')` layer. The model is summarized and returned. The second cell creates the model and compiles it with the `adam` optimizer, `categorical_crossentropy` loss, and `accuracy` metric.

```
In [8]: def create_model():
        model = Sequential()

        base_model = transfer_learning()
        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(n, activation='softmax'))

        model.summary

        return model

In [9]: model = create_model()
        model.compile(optimizer='adam', loss='categorical_crossentropy', metrics = ['accuracy'])
```

The second screenshot shows the notebook with a last checkpoint 29 minutes ago. The first code cell calls `model.summary()`, which displays the following table:

Layer (type)	Output shape	Param #
densenet121 (Functional)	(None, 2, 2, 1024)	7037504
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0
dense (Dense)	(None, 256)	262400
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1024)	525312
dense_3 (Dense)	(None, 14)	14350

Below the table, the summary shows: Total params: 7971150 (30.41 MB), Trainable params: 6386894 (24.36 MB), and Non-trainable params: 1584256 (6.04 MB). The second code cell runs `model.fit(x=train_gen, validation_data=val_set, epochs = 1)`, which outputs: 7915/7915 [=====] - ETA: 0s - loss: 0.0846 - accuracy: 0.9779. The third code cell runs `model.save('crime.hs')`.

TESTING THE MODEL:

The image shows a Jupyter Notebook titled "CRIME VISION" with a last checkpoint 29 minutes ago. The notebook contains a section titled "TESTING THE MODEL" with several code cells. The first cell imports the necessary libraries: `load_model` from `tensorflow.keras.models`, `image` from `tensorflow.keras.preprocessing`, `numpy` as `np`, `cv2`, `tensorflow` as `tf`, and `img_to_array` from `keras.preprocessing.image`. The subsequent cells load the model from `"C:/Users/VISHWU VARDHAN/crime.hs"` with `compile=False`, load the image `"C:/Users/VISHWU VARDHAN/Downloads/Fighting.png"` with `target_size=(64,64)`, convert the image to an array, expand its dimensions, and finally print the shape of the array `x`.

```
In [ ]: from tensorflow.keras.models import load_model
        from tensorflow.keras.preprocessing import image
        import numpy as np
        import cv2
        import tensorflow as tf
        from keras.preprocessing.image import img_to_array

In [ ]: model=tf.keras.models.load_model(r"C:/Users/VISHWU VARDHAN/crime.hs", compile=False)

In [ ]: img=image.load_img(r"C:/Users/VISHWU VARDHAN/Downloads/Fighting.png", target_size=(64,64))

In [ ]: img

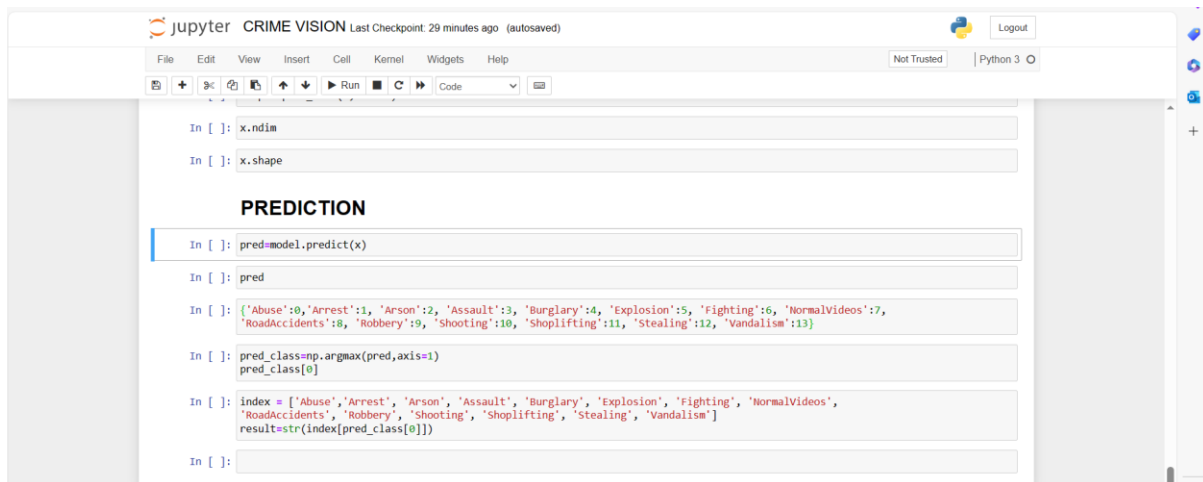
In [ ]: x=image.img_to_array(img)
        x

In [ ]: x=np.expand_dims(x,axis=0)

In [ ]: x.ndim

In [ ]: x.shape
```

PREDICTION:



APPLICATION BUILDING:

FLASK:

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Nov  2 13:24:08 2023
4
5  @author: VISHNU VARDHAN
6  """
7
8
9  import numpy as np
10 import os
11 from tensorflow.keras.models import load_model
12 from tensorflow.keras.preprocessing import image
13 from flask import Flask, request, render_template
14 #from werkzeug.utils import secure_filename
15 #from gevent.pywsgi import WSGIServer
16
17 app = Flask(__name__)
18
19 model = load_model("crime.h5",compile=False)
20
21 @app.route('/')
22 def index():
23     return render_template('index.html')
24
25 @app.route('/predict',methods = ['GET','POST'])
26 def upload():
27     if request.method == 'POST':
28         f = request.files['image']
29         print("current path")
30         basepath = os.path.dirname(__file__)
31         print("current path", basepath)
32         filepath = os.path.join(basepath,'uploads',f.filename)
33         print("upload folder is ", filepath)
34         f.save(filepath)
35
36         img = image.load_img(filepath,target_size = (64,64))
37         x = image.img_to_array(img)
38         print(x)
39         x = np.expand_dims(x,axis =0)
40         print(x)
41         y=model.predict(x)

```

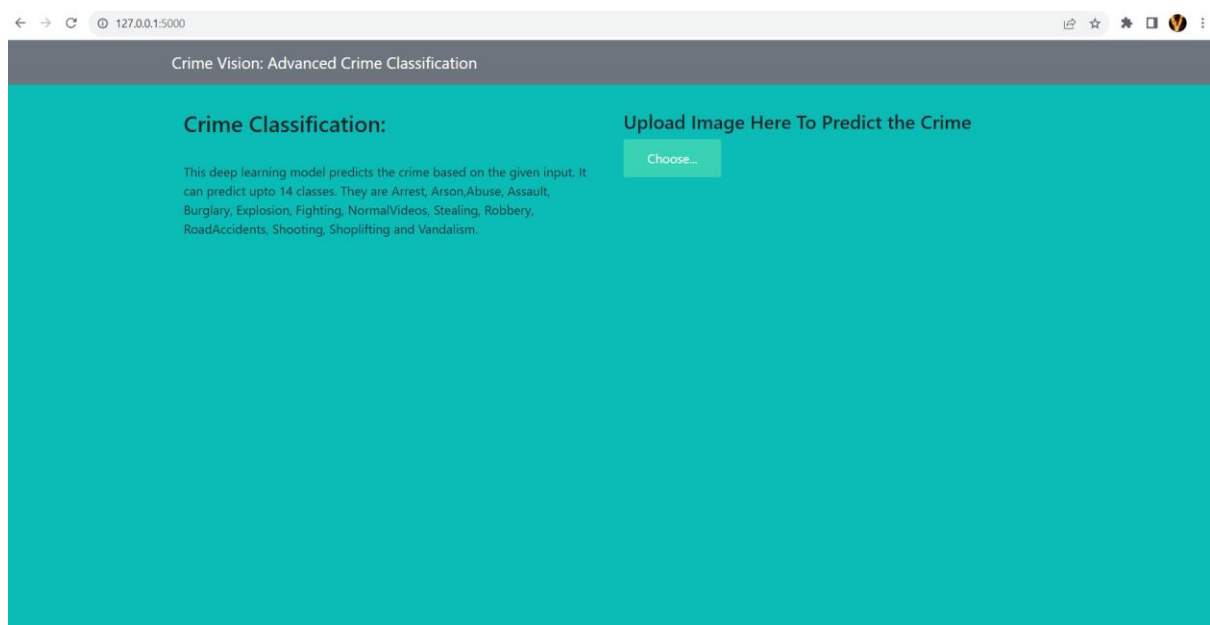
```
C:\Users\VISHNU VARDHAN\Downloads\AIML FLASK\app1.py
main.css × main.js × index.html × app1.py ×
11 from tensorflow.keras.models import load_model
12 from tensorflow.keras.preprocessing import image
13 from flask import Flask, request, render_template
14 #from werkzeug.utils import secure_filename
15 #from event.pywsgi import WSGIServer
16
17 app = Flask(__name__)
18
19 model = load_model("crime.h5", compile=False)
20
21 @app.route('/')
22 def index():
23     return render_template('index.html')
24
25 @app.route('/predict', methods = ['GET', 'POST'])
26 def upload():
27     if request.method == 'POST':
28         f = request.files['image']
29         print("current path")
30         basepath = os.path.dirname(__file__)
31         print("current path", basepath)
32         filepath = os.path.join(basepath, 'uploads', f.filename)
33         print("upload folder is ", filepath)
34         f.save(filepath)
35
36         img = image.load_img(filepath, target_size = (64,64))
37         x = image.img_to_array(img)
38         print(x)
39         x = np.expand_dims(x, axis = 0)
40         print(x)
41         y = model.predict(x)
42         preds = np.argmax(y, axis = 1)
43         #preds = model.predict_classes(x)
44         print("prediction", preds)
45         index = ['Abuse', 'Arrest', 'Arson', 'Assault', 'Burglary', 'Explosion', 'Fighting', 'NormalVideos',
46 'RoadAccidents', 'Robbery', 'Shooting', 'Shoplifting', 'Stealing', 'Vandalism']
47         text = "The classified Crime is : " + str(index[preds[0]])
48         return text
49 if __name__ == '__main__':
50     app.run(debug = False, threaded = False)
51
```

```
Console 1/A X
IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/VISHNU VARDHAN/Downloads/AIML FLASK/app1.py', wdir='C:/Users/VISHNU VARDHAN/Downloads/AIML FLASK')

2023-11-02 17:00:09.665630: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Serving Flask app "app1" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

WEB APPLICATION



PREDICTION OF CRIME IN WEB APPLICATION:

Crime Classification:

This deep learning model predicts the crime based on the given input. It can predict upto 14 classes. They are Arrest, Arson, Abuse, Assault, Burglary, Explosion, Fighting, NormalVideos, Stealing, Robbery, RoadAccidents, Shooting, Shoplifting and Vandalism.

Upload Image Here To Predict the Crime

Choose...



Result: The classified Crime is :
RoadAccidents

Crime Classification:

This deep learning model predicts the crime based on the given input. It can predict upto 14 classes. They are Arrest, Arson, Abuse, Assault, Burglary, Explosion, Fighting, NormalVideos, Stealing, Robbery, RoadAccidents, Shooting, Shoplifting and Vandalism.

Upload Image Here To Predict the Crime

Choose...



Result: The classified Crime is : Shooting