PROJECT REPORT FORMAT

DATE	02-11-2023
TEAM ID	TEAM-593175
PROJECT	CRIME VISION: ADVANCED CRIME CLASSIFICATION USING DEEP LEARNING

INTRODUCTION:

CRIME VISION: ADVANCED CRIME CLASSIFIACTION 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.

LITERATURE SURVEY:

There is a lot of crime happening in modern era. It is important to get awareness on crime which can lead to a better prevention of crime. Crime happens around every individual in one or the other way. Since, everyone is busy in their own work, no one cares about it. This project is about raising awareness among people so that one could identify the crime and fight against it. In this project a image is taken as an input and the classified crime is predicted and displayed as an output. 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.

IDEATION AND PROPOSED SOLUTION:

EMPATHY MAP:

BRAIN STORMING:

For this problem statement, we use deep learning algorithm called Convolutional Neural Network (CNN). By using deep learning, it is possible to analyse images and video footage and classify them into different crimes based on the activity depicted in the image or video. Deep Learning algorithms can be trained to recognize patterns and features in images and video that identifies types of crimes

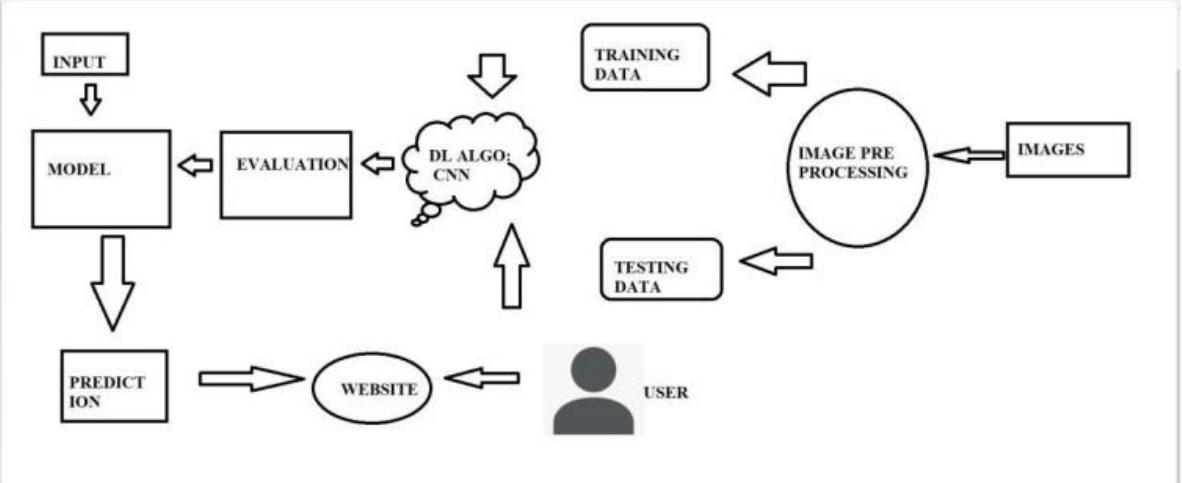
IDEA PRIORITIZATION: For this problem statement we are going to use deep learning algorithm called Convolutional Neural Network (CNN). After building the deep learning model, a website is created using flask, html, CSS, and java script. A user uploads an image or a video in this website and the predicted output is shown.

REQUIREMENT ANALYSIS:

The data is collected from Kaggle. The model is trained on this data and it is expected to classify the crime into one among the 14 classes given in the dataset accurately. This model should also predict if the user input is not from training or testing set. This model is built transfer learning and CNN with a batch size of 128. The model should be scalable.

PROJECT DESIGN:

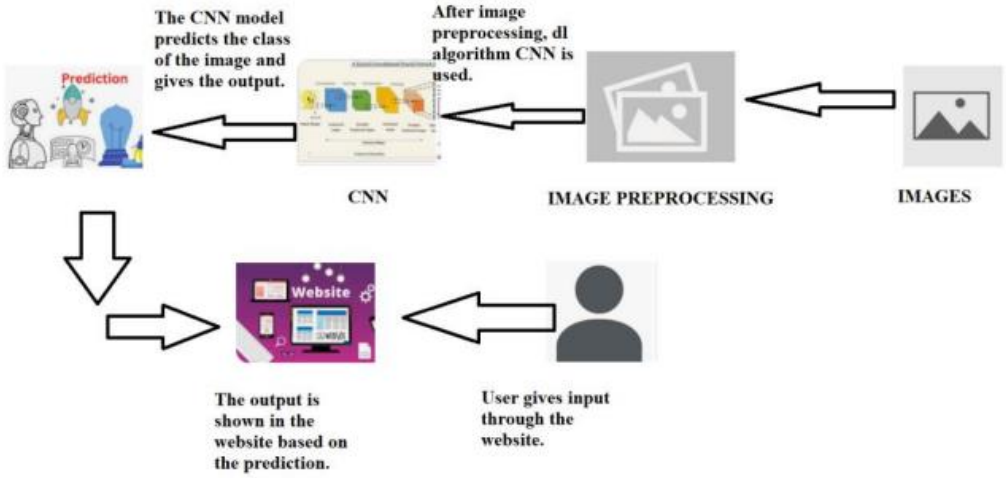
DATAFLOW DIAGARAM AND USER STORIES:



USER STORIES:

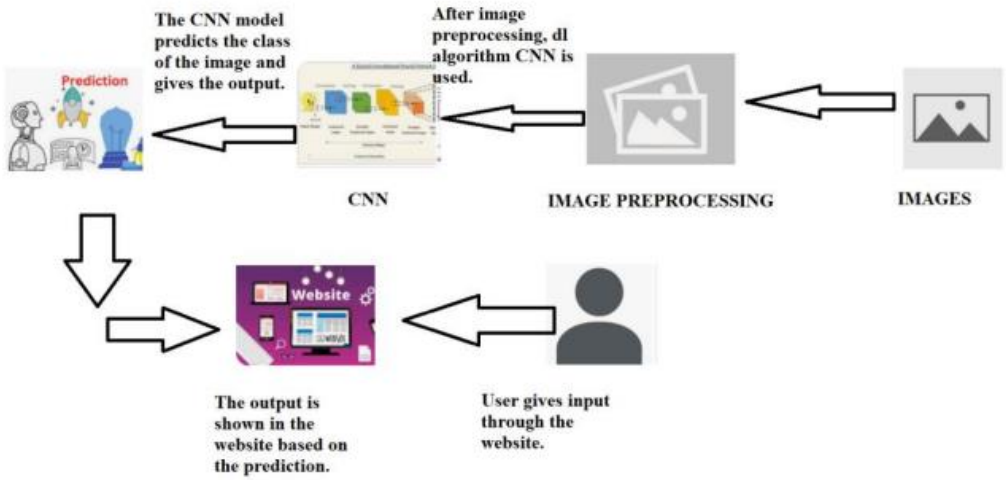
USER TYPE	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	USER STORY / TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
CUSTOMER (WEB)						

SOLUTION ARCHITECTURE:



PROJECT PLANNING AND SCHEDULING:

TECHNICAL ARCHITECTURE:



COMPONENTS AND TECHNOLOGIES:

SNO	COMPONENT	DESCRIPTION	TECHNOLOGY
1	USER INTERFACE	WEB APP	HTML, CSS, JAVASCRIPT
2	APPLICATION LOGIC – 1	Logic for the process in the application	Python
3	APPLICATION LOGIC - 2	Logic for the process in the application	Localhost
4	DATABASE	Data type, Configurations	MY SQL
5	CLOUD	Database services on cloud	GCP/ AWS
6	ML MODEL	ML and DL	CNN

APPLICATION CHARACTERISTICS:

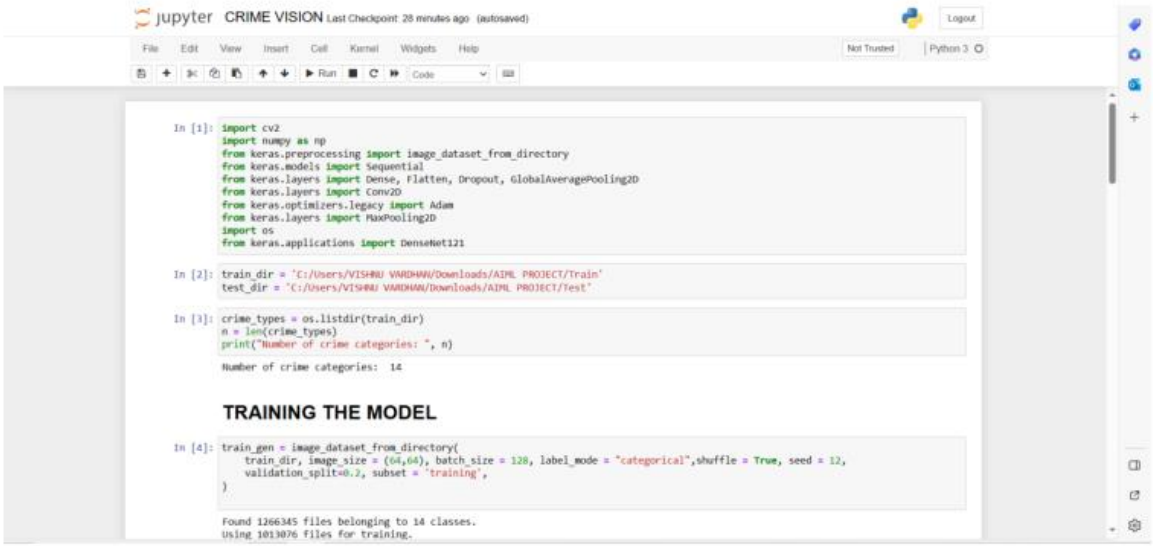
SNO	CHARACTERISTICS	DESCRIPTION	TECHNOLOGY
1	OPENSOURCE FRAMEWORK	List of opensource frameworks	Flask
2	SCALABILITY	API	Rest
3	AVAILABILITY	OS	It can run in Windows, macOS, Linux.

SPRINT PLANNING:

SPRINT	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	TASK	STORY POINTS
Sprint-1	Project setup	1	Set up the environment Using tools	1
Sprint-2	Data collection	2	Collect different images	2
Sprint-2	Data preprocessing	3	Preprocess the collected data	3
Sprint-3	Model Development	4	Develop the model using CNN	4
Sprint-3	Training	5	Train, it using dataset	6
Sprint-4	Model Deployment	6	Deploy the trained model	2
Sprint-5	Testing	7	Conduct testing of model	1

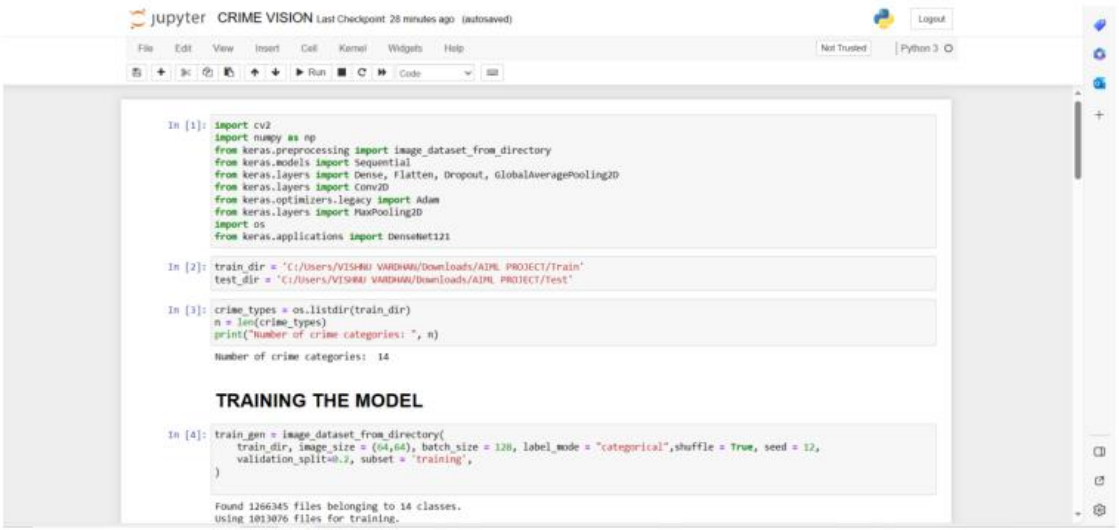
CODING AND SOLUTIONING:

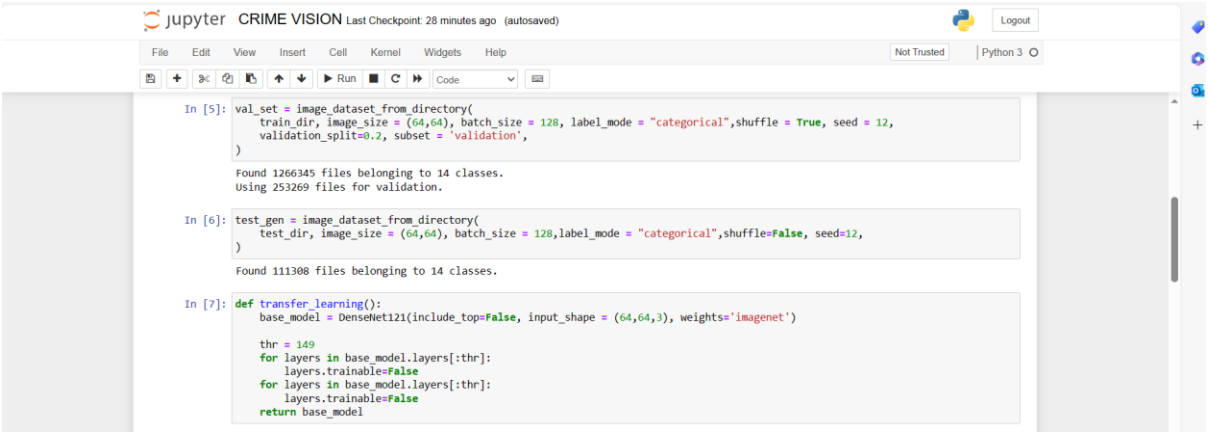
DATA PREPROCESSING AND IMPORTING LIBRARIES:

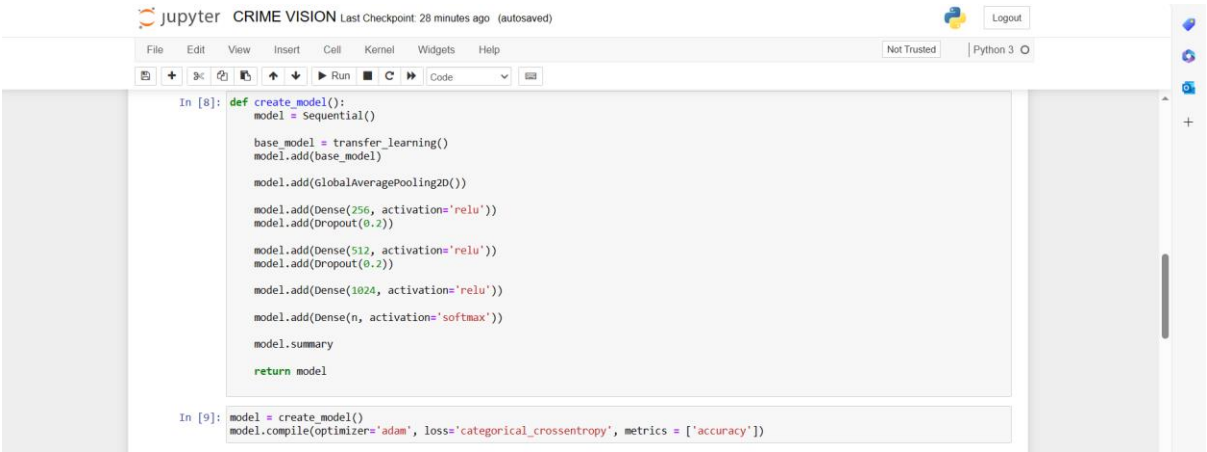


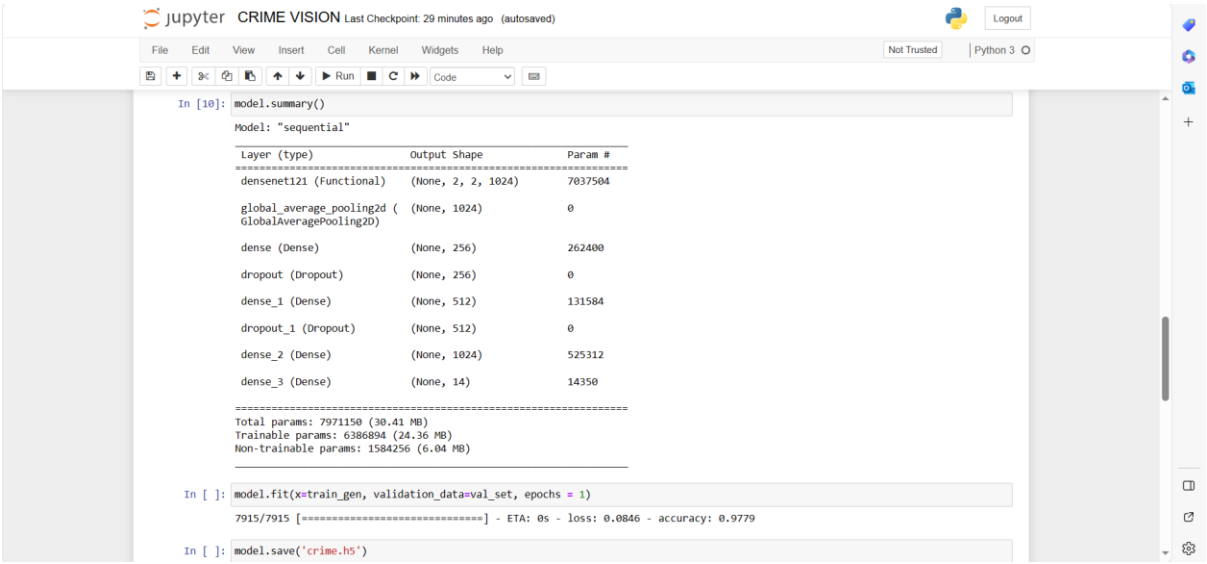
Global average pooling 2D (GAP 2D): It is a type of pooling operation commonly used in convolutional neural networks (CNNs) for image classification tasks.

TRAINING THE MODEL:

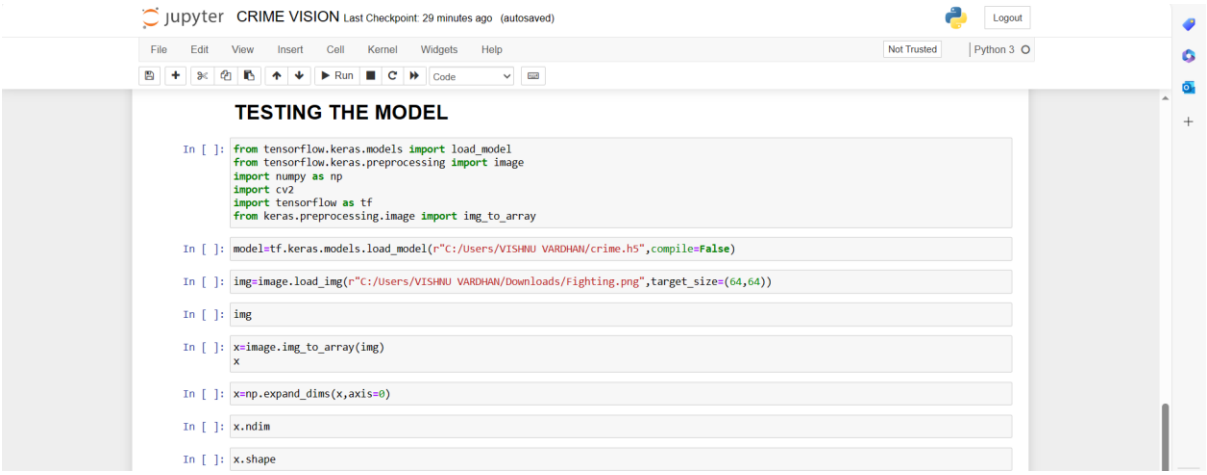




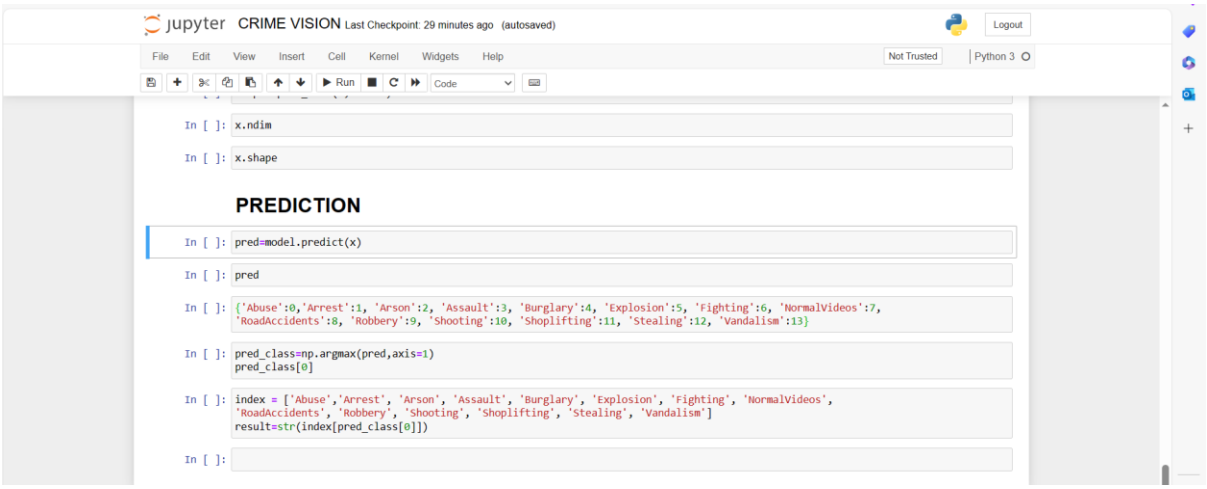




TESTING THE MODEL:



PREDICTION:



APPLICATION BUILDING:

FLASK:

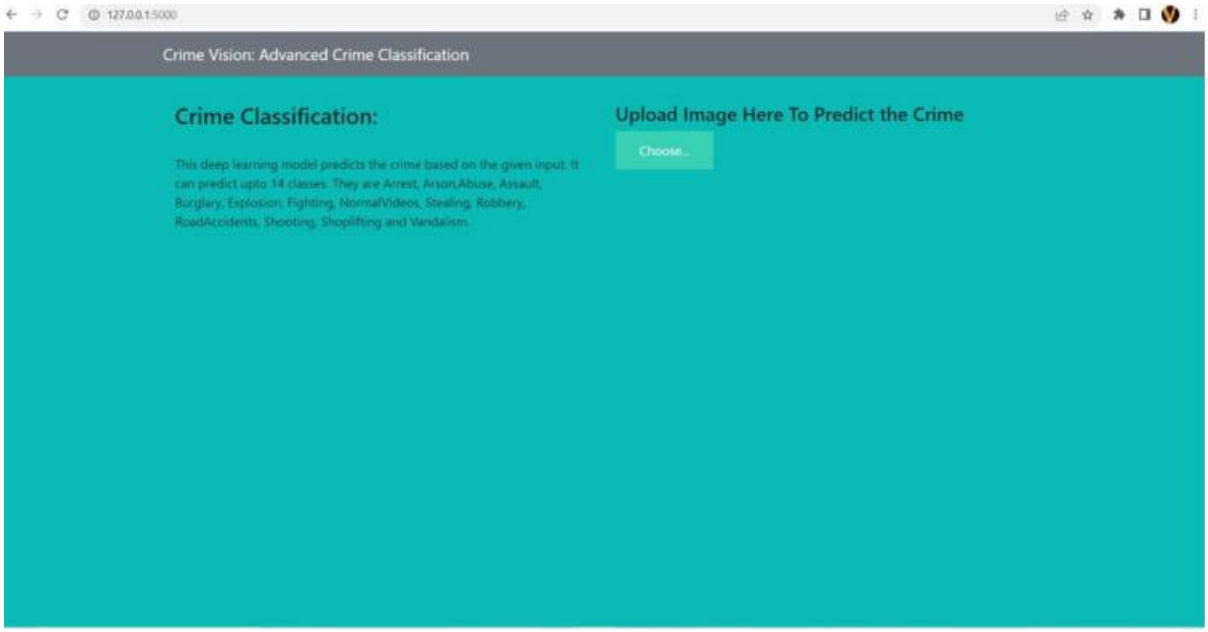
```python
# -*- coding: utf-8 -*-
"""
Created on Thu Nov  2 13:24:08 2023

@author: VISHNU VARDHAN
"""


import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask , request, render_template
#from werkzeug.utils import secure_filename
#from gevent.pywsgi import WSGIServer

app = Flask(__name__)

model = load_model("crime.h5",compile=False)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']
        print("current path")
        basepath = os.path.dirname(__file__)
        print("current path", basepath)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)

        img = image.load_img(filepath,target_size = (64,64))
        x = image.img_to_array(img)
        print(x)
        x = np.expand_dims(x,axis =0)
        print(x)
        y=model.predict(x)
```

C:\Users\VISHNU VARDHAN\Downloads\AIML FLASK\app1.py

main.css | main.js | index.html | app1.py

```python
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask , request, render_template
#from werkzeug.utils import secure_filename
#from gevent.pywsgi import WSGIServer

app = Flask(__name__)

model = load_model("crime.h5",compile=False)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']
        print("current path")
        basepath = os.path.dirname(__file__)
        print("current path", basepath)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)

        img = image.load_img(filepath,target_size = (64,64))
        x = image.img_to_array(img)
        print(x)
        x = np.expand_dims(x,axis =0)
        print(x)
        y=model.predict(x)
        preds=np.argmax(y, axis=1)
        #preds = model.predict_classes(x)
        print("prediction",preds)
        index = ['Abuse','Arrest', 'Arson', 'Assault', 'Burglary', 'Explosion', 'Fighting', 'NormalVideos',
'RoadAccidents', 'Robbery', 'Shooting', 'Shoplifting', 'Stealing', 'Vandalism']
        text = "The classified Crime is : " + str(index[preds[0]])
    return text
if __name__ == '__main__':
    app.run(debug = False, threaded = False)

```

WEB APPLICATION:



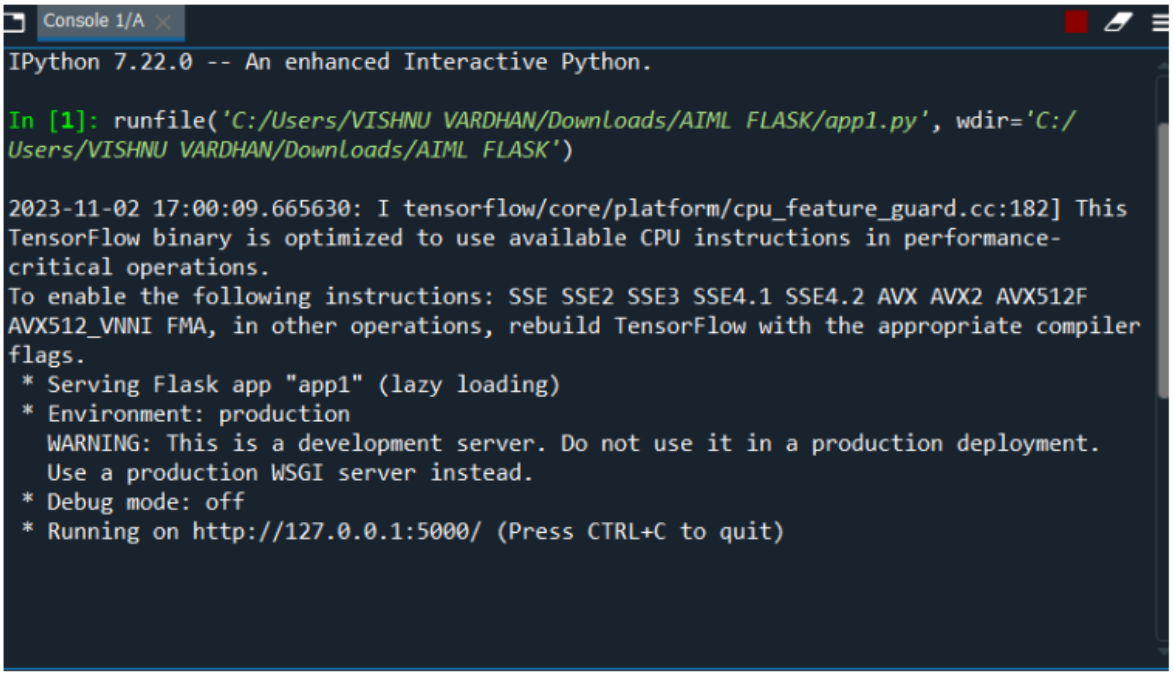
PERFORMANCE TETSING:

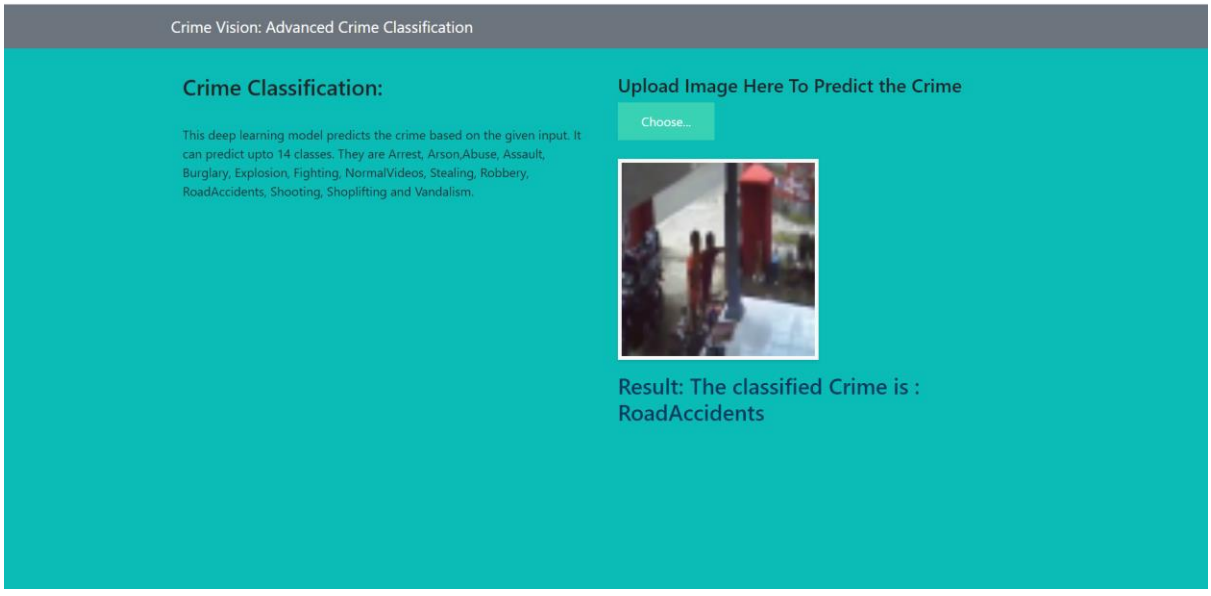
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 densenet121 (Functional)    (None, 2, 2, 1024)        7037504

 global_average_pooling2d (  (None, 1024)              0
 GlobalAveragePooling2D)

 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

=================================================================
Total params: 7971150 (30.41 MB)
Trainable params: 6386894 (24.36 MB)
Non-trainable params: 1584256 (6.04 MB)
_________________________________________________________________
```

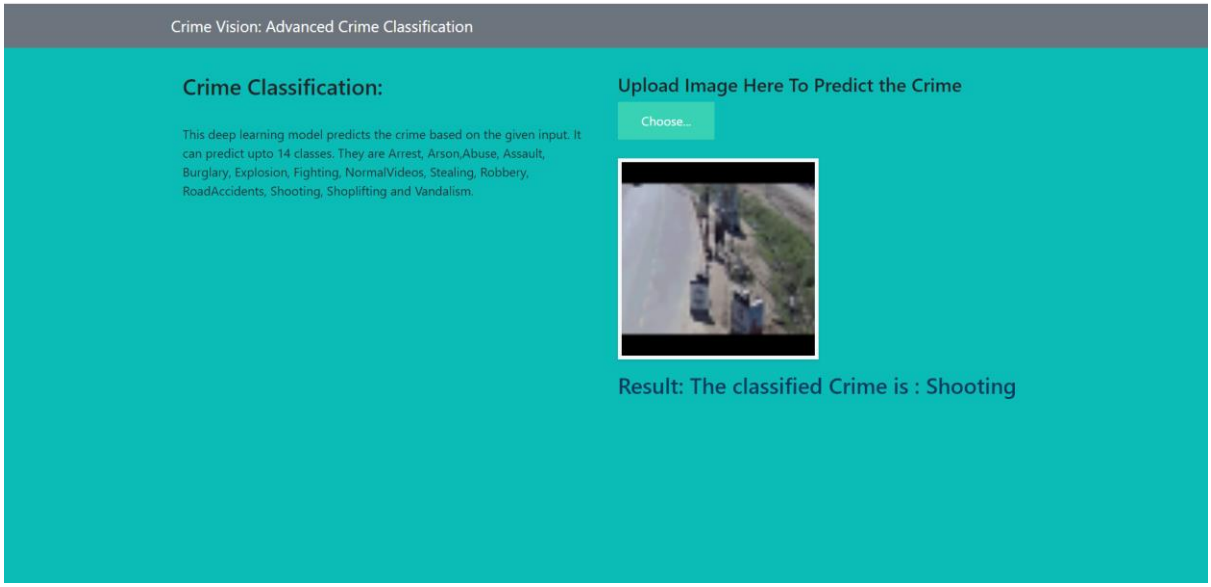
ACCURACY:

```
7915/7915 [==============================] - 21468s 3s/step - loss: 0.0846 - accuracy: 0.9779 - val_loss: 0.0397 - val_accurac
y: 0.9914
```

RESULTS:


Console 1/A
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)


Crime Vision: Advanced Crime Classification
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



ADVANTAGES:

The training accuracy of the model is 97.79% and the validation accuracy of the model is 99.14%.

This model can predict the crime shown in the image accurately.

Since, the model is trained using transfer learning it is cheaper to afford.

The training time taken is less.

DISADVANTAGES:

If not executed properly, transfer learning algorithms can lead to negative transfer, where the learning performance is impaired instead of improved.

CONCLUSION:

This deep learning model is used to predict the crime around us. It needs a user input and then it processes the deep learning algorithm to predict the most accurate crime and displays back to user. It has a training accuracy of 97.79% and validation accuracy of 99.14%. This model is scalable and easy to use as well. It takes less time to train on the given dataset as it uses Global Average Pooling 2D which is a pre trained model to classify images.

FUTURE SCOPE:

This model is to bring awareness among people on the crime happening in today's world. This could also help agencies to bring awareness among people and plan effective strategies in the prevention of crime.

APPENDIX:

GITHUB LINK FOR SOURCE CODE:

https://github.com/smartinternz02/SI-GuidedProject-610027-1698382468/tree/main/PERFORMACE%20AND%20FINAL%20SUBMISSION%20PHASE

PROJECT DEMO LINK:

https://drive.google.com/file/d/1wzA65c10SL2UUfZnIe-xtdH3f-6JY7Bf/view?usp=drive_link