

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 CLASSIFICATION USING DEEP LEARNING

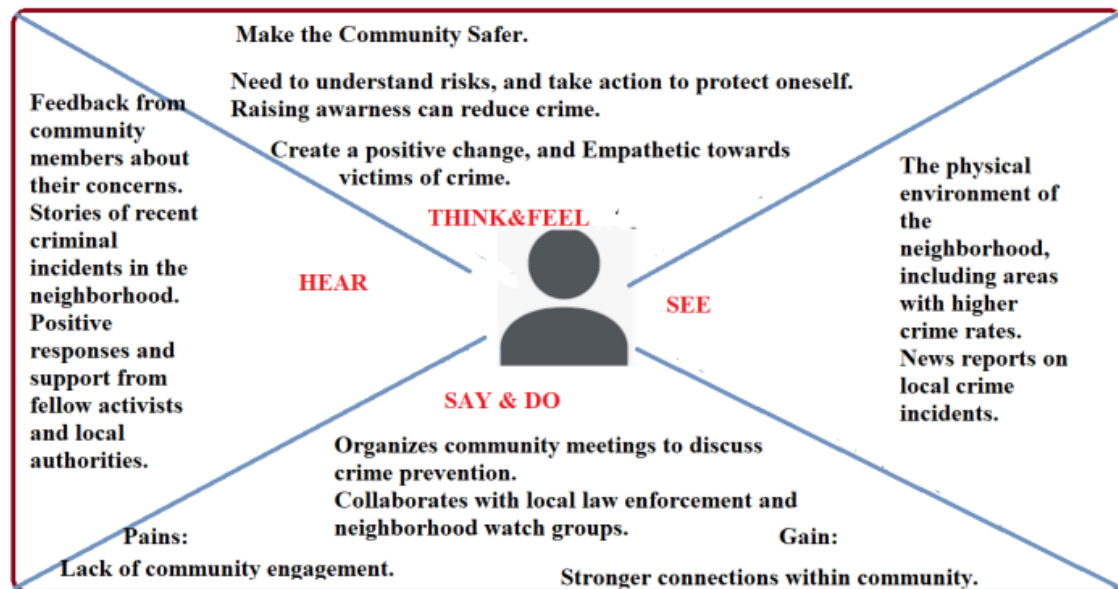
Crime Vision using Deep Learning is a technique that involves deep learning, to analyse images and video footage of crime scenes or incidents and identify and classify different types of crimes. This is very useful for criminal justice and law enforcement contexts, including crime scene investigation,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

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

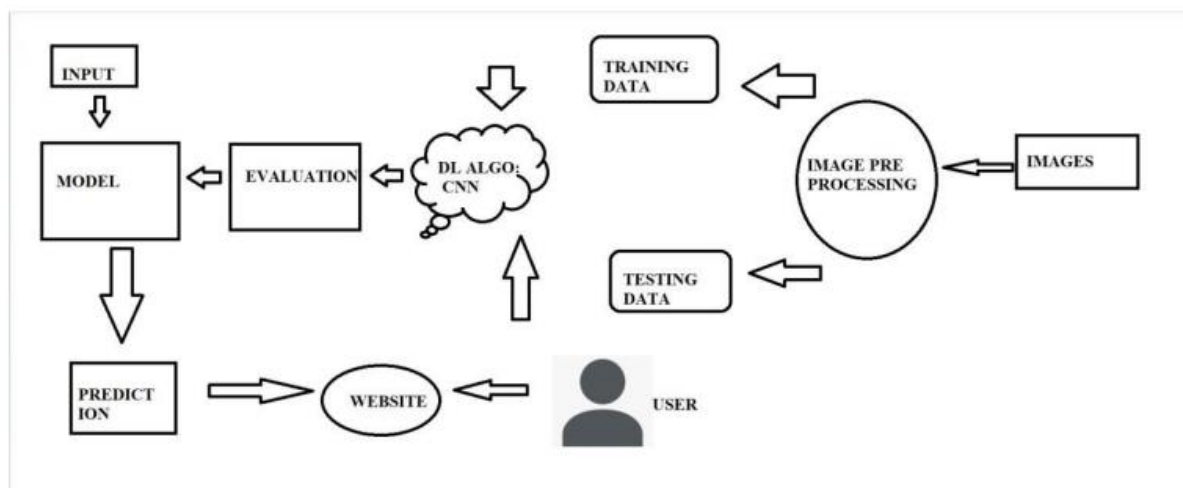
IDEA PRIORITIZATION: For this problem statement we are going to use deep learning algorithm called Convolutional Neural Network (CNN). After building the deep learning model, a website is created using flask, html, CSS, and java script. A user uploads an image or a video in this website and the predicted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

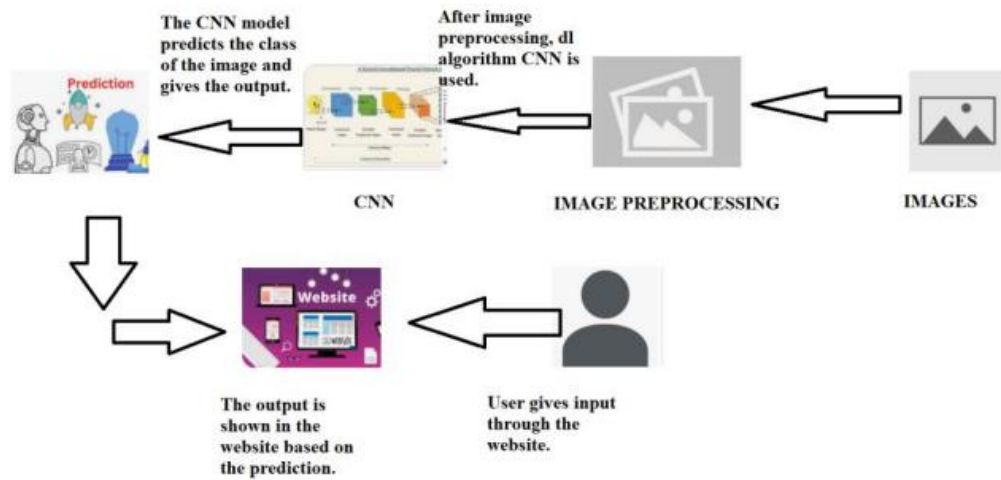
DATAFLOW DIAGARAM AND USER STORIES:



USER STORIES:

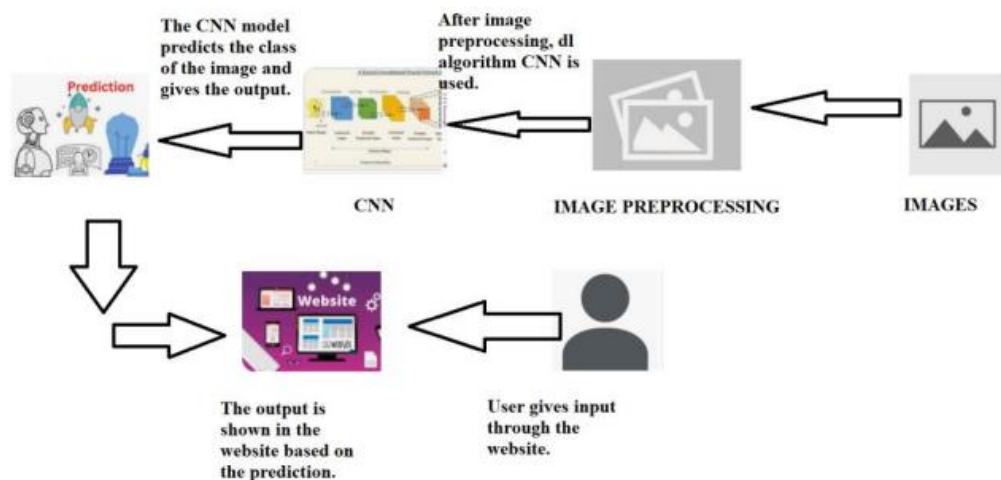
USER TYPE	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	USER STORY / TASK	ACCEPTANCE CRITERIA	PRIORITY	RELEASE
CUSTOMER (WEB)						

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:

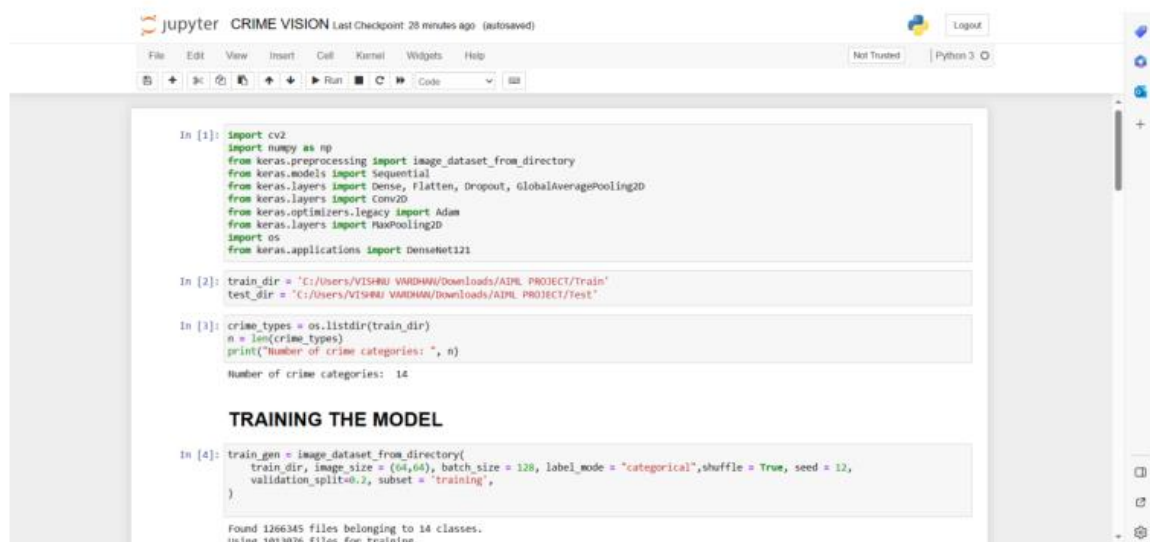
SNO	CHARACTERISTICS	DESCRIPTION	TECHNOLOGY
1	OPENSOURCE FRAMEWORK	List of opensource frameworks	Flask
2	SCALABILITY	API	Rest
3	AVAILABILITY	OS	It can run in Windows, macOS, Linux.

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:



```
jupyter CRIME VISION Last Checkpoint: 28 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
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/AIPL PROJECT/Train'
test_dir = 'C:/Users/VISHNU VARDHAN/Downloads/AIPL 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 = 32, 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:

```
jupyter CRIME VISION Last Checkpoint: 28 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
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 VARDHANU/downloads/AIPL PROJECT/train'
test_dir = 'C:/Users/VISHNU VARDHANU/downloads/AIPL 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.
```

```
jupyter CRIME VISION Last Checkpoint: 28 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
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
```

```
jupyter CRIME VISION Last Checkpoint: 28 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
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'])
```

```
jupyter CRIME VISION Last Checkpoint: 29 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
In [10]: model.summary()
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)

In [ ]: model.fit(x=train_gen, validation_data=val_set, epochs = 1)
7915/7915 [=====] - ETA: 0s - loss: 0.0846 - accuracy: 0.9779

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

TESTING THE MODEL:

```
jupyter CRIME VISION Last Checkpoint: 29 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
TESTING THE MODEL
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/VISHNU VARDHAN/crime.hs",compile=False)

In [ ]: img=image.load_img(r"C:/Users/VISHNU 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:

```
jupyter CRIME VISION Last Checkpoint: 29 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
In [ ]: x.ndim
In [ ]: x.shape
PREDICTION
In [ ]: pred=model.predict(x)
In [ ]: pred
In [ ]: {'Abuse':0,'Arrest':1,'Arson':2,'Assault':3,'Burglary':4,'Explosion':5,'Fighting':6,'NormalVideos':7,
'RoadAccidents':8,'Robbery':9,'Shooting':10,'Shoplifting':11,'Stealing':12,'Vandalism':13}
In [ ]: pred_class=np.argmax(pred,axis=1)
pred_class[0]
In [ ]: index = ['Abuse','Arrest','Arson','Assault','Burglary','Explosion','Fighting','NormalVideos',
'RoadAccidents','Robbery','Shooting','Shoplifting','Stealing','Vandalism']
result=str(index[pred_class[0]])
In [ ]:
```

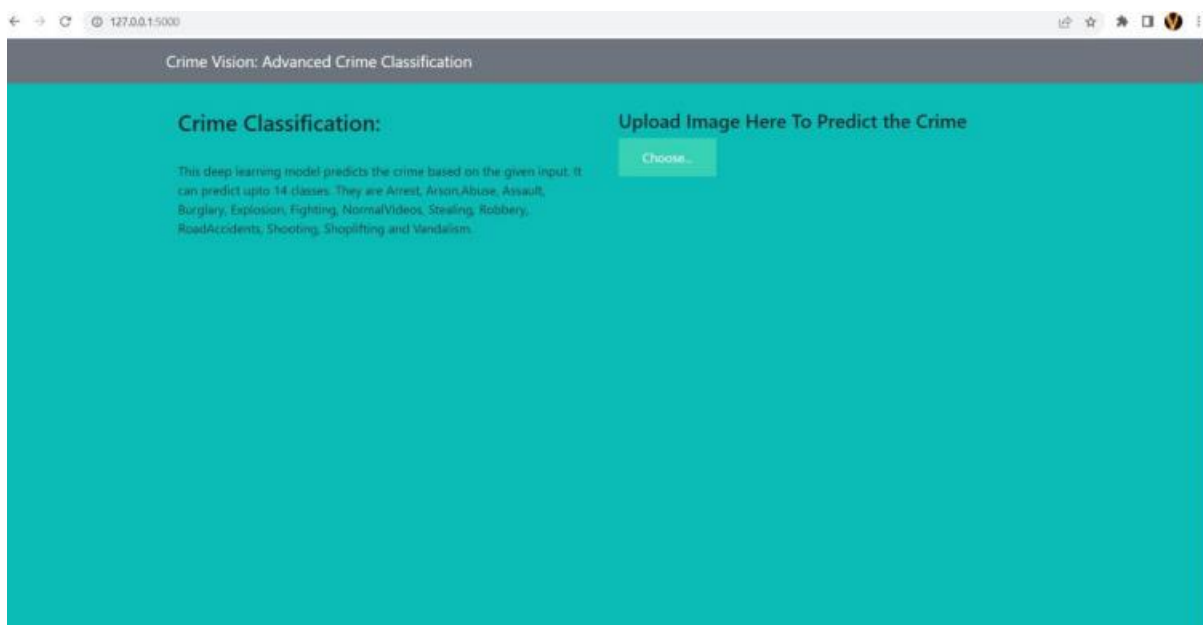

APPLICATION BUILDING:

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 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)
```

```
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
```

WEB APPLICATION:



PERFORMANCE TETSING:

Model: "sequential"

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
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_accuracy: 0.9914

RESULTS:

```
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)
```

Crime Vision: Advanced Crime Classification

Crime Classification:

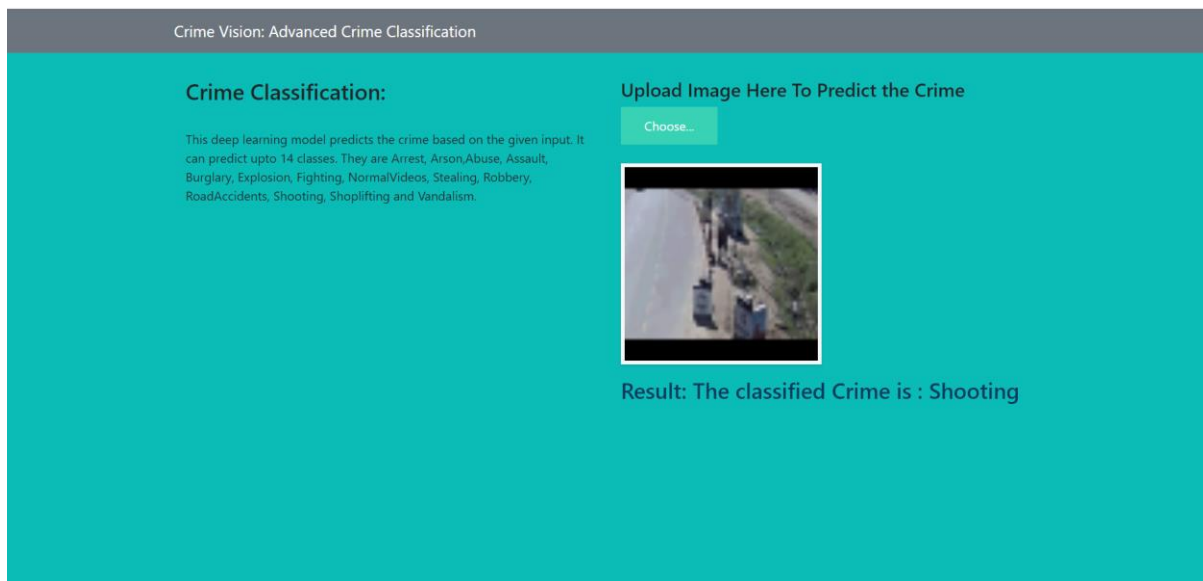
This deep learning model predicts the crime based on the given input. It can predict upto 14 classes. They are Arrest, Arson, Abuse, Assault, Burglary, Explosion, Fighting, NormalVideos, Stealing, Robbery, RoadAccidents, Shooting, Shoplifting and Vandalism.

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/PERFORMANCE%20AND%20FINAL%20SUBMISSION%20PHASE>

PROJECT DEMO LINK:

https://drive.google.com/file/d/1wzA65c10SL2UUFznIe-xtdH3f-6JY7Bf/view?usp=drive_link