

```
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.legacy import Adam
from keras.layers import MaxPooling2D
import os
from keras.applications import DenseNet121
```

```
In [2]: train_dir = 'C:/Users/VISHNU VARDHAN/Downloads/AIML PROJECT/Train'
test_dir = 'C:/Users/VISHNU VARDHAN/Downloads/AIML 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
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [4]: train_gen = image_dataset_from_directory(
    train_dir, image_size = (64,64), batch_size = 128, label_mode = "categorical", shuffle=True,
    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,
    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=True,
)

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
```

```
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'])
```

```
In [10]: model.summary()

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)
_____
```

```
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.h5')
```

```
In [ ]:
```

```
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.h5",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
```

```
In [ ]: pred=model.predict(x)
```

```
In [ ]: pred
```

```
In [ ]: {'Abuse':0, 'Arrest':1, 'Arson':2, 'Assault':3, 'Burglary':4, 'Explosion':5, 'Fighting':6, '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', 'RoadAccidents', 'Robbery', 'Shooting', 'Shoplifting', 'Stealing', 'Vandalism']
result=str(index[pred_class[0]])
```

```
In [ ]:
```