```
! pip install -q kaggle
```

```
!mkdir ~/.kaggle
```

```
!cp kaggle.json ~/.kaggle
```

```
!kaggle datasets download -d odins0n/ucf-crime-dataset
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.jso
Downloading ucf-crime-dataset.zip to /content
100% 11.0G/11.0G [08:27<00:00, 20.5MB/s]
100% 11.0G/11.0G [08:27<00:00, 23.3MB/s]
```

```
!unzip /content/ucf-crime-dataset.zip
```

```
  inflating: Train/Vandalism/Vandalism050_x264_380.png
  inflating: Train/Vandalism/Vandalism050_x264_390.png
  inflating: Train/Vandalism/Vandalism050_x264_40.png
  inflating: Train/Vandalism/Vandalism050_x264_400.png
  inflating: Train/Vandalism/Vandalism050_x264_410.png
  inflating: Train/Vandalism/Vandalism050_x264_420.png
  inflating: Train/Vandalism/Vandalism050_x264_430.png
  inflating: Train/Vandalism/Vandalism050_x264_440.png
  inflating: Train/Vandalism/Vandalism050_x264_450.png
  inflating: Train/Vandalism/Vandalism050_x264_460.png
  inflating: Train/Vandalism/Vandalism050_x264_470.png
  inflating: Train/Vandalism/Vandalism050_x264_480.png
  inflating: Train/Vandalism/Vandalism050_x264_490.png
  inflating: Train/Vandalism/Vandalism050_x264_50.png
  inflating: Train/Vandalism/Vandalism050_x264_500.png
  inflating: Train/Vandalism/Vandalism050_x264_510.png
  inflating: Train/Vandalism/Vandalism050_x264_520.png
  inflating: Train/Vandalism/Vandalism050_x264_530.png
  inflating: Train/Vandalism/Vandalism050_x264_540.png
  inflating: Train/Vandalism/Vandalism050_x264_550.png
  inflating: Train/Vandalism/Vandalism050_x264_560.png
  inflating: Train/Vandalism/Vandalism050_x264_570.png
  inflating: Train/Vandalism/Vandalism050_x264_580.png
  inflating: Train/Vandalism/Vandalism050_x264_590.png
  inflating: Train/Vandalism/Vandalism050_x264_60.png
  inflating: Train/Vandalism/Vandalism050_x264_600.png
  inflating: Train/Vandalism/Vandalism050_x264_610.png
  inflating: Train/Vandalism/Vandalism050_x264_620.png
  inflating: Train/Vandalism/Vandalism050_x264_630.png
  inflating: Train/Vandalism/Vandalism050_x264_640.png
  inflating: Train/Vandalism/Vandalism050_x264_650.png
  inflating: Train/Vandalism/Vandalism050_x264_660.png
  inflating: Train/Vandalism/Vandalism050_x264_670.png
  inflating: Train/Vandalism/Vandalism050_x264_680.png
  inflating: Train/Vandalism/Vandalism050_x264_690.png
  inflating: Train/Vandalism/Vandalism050_x264_70.png
  inflating: Train/Vandalism/Vandalism050_x264_700.png
  inflating: Train/Vandalism/Vandalism050_x264_710.png
  inflating: Train/Vandalism/Vandalism050_x264_720.png
  inflating: Train/Vandalism/Vandalism050_x264_730.png
  inflating: Train/Vandalism/Vandalism050_x264_740.png
  inflating: Train/Vandalism/Vandalism050_x264_750.png
  inflating: Train/Vandalism/Vandalism050_x264_760.png
  inflating: Train/Vandalism/Vandalism050_x264_770.png
  inflating: Train/Vandalism/Vandalism050_x264_780.png
  inflating: Train/Vandalism/Vandalism050_x264_790.png
  inflating: Train/Vandalism/Vandalism050_x264_80.png
  inflating: Train/Vandalism/Vandalism050_x264_800.png
  inflating: Train/Vandalism/Vandalism050_x264_810.png
  inflating: Train/Vandalism/Vandalism050_x264_820.png
  inflating: Train/Vandalism/Vandalism050_x264_830.png
  inflating: Train/Vandalism/Vandalism050_x264_840.png
  inflating: Train/Vandalism/Vandalism050_x264_850.png
  inflating: Train/Vandalism/Vandalism050_x264_860.png
  inflating: Train/Vandalism/Vandalism050_x264_870.png
  inflating: Train/Vandalism/Vandalism050_x264_880.png
  inflating: Train/Vandalism/Vandalism050_x264_890.png
  inflating: Train/Vandalism/Vandalism050_x264_90.png
```

```
train_path = '/content/Train'
test_path = '/content/Test'
```

```
from tensorflow.keras.preprocessing import image_dataset_from_directory
train_datagen = image_dataset_from_directory(
    train_path,
    validation_split = 0.2,
    subset = 'training',
    shuffle = True,
    seed = 69,
    label_mode = 'categorical',
    image_size = (64,64),
    batch_size = 64)
```

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

```
test_datagen = image_dataset_from_directory(
    test_path,
    seed = 69,
    shuffle = False,
    label_mode = 'categorical',
    class_names = None, #
    image_size = (64,64),
    batch_size = 64)
```

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

```
val_datagen = image_dataset_from_directory(
    train_path,
    validation_split = 0.2,
    subset = 'validation',
    shuffle = True,
    seed = 69,
    label_mode = 'categorical',
    image_size = (64,64),
    batch_size = 64)
```

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

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.regularizers import l2
resnet_model = Sequential()

from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.layers import Dense, Flatten, Dropout

pre_trained_model = ResNet50(include_top = False, input_shape = (64,64,3), pooling = 'max', classes = 14, weights = 'imagenet')

for layer in pre_trained_model.layers:
  layer.trainable = False

resnet_model.add(pre_trained_model)
resnet_model.add(Flatten())
resnet_model.add(Dense(512, activation = 'relu'))#, kernel_regularizer = l2(0.1)))
resnet_model.add(Dense(14, activation = 'softmax'))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kerne
94765736/94765736 [==============================] - 5s 0us/step
```

```
resnet_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2048)              23587712

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dense_1 (Dense)             (None, 14)                7182

=================================================================
Total params: 24643982 (94.01 MB)
Trainable params: 1056270 (4.03 MB)
Non-trainable params: 23587712 (89.98 MB)
_________________________________________________________________
```

```
from tensorflow.keras.optimizers import Adam
resnet_model.compile(optimizer = Adam(learning_rate = 0.00003), loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

```
resnet_model.fit(train_datagen, validation_data = val_datagen, epochs = 7)
```

```
Epoch 1/7
15830/15830 [==============================] - 1447s 91ms/step - loss: 0.1420 - accur
Epoch 2/7
15830/15830 [==============================] - 1308s 83ms/step - loss: 0.0178 - accur
Epoch 3/7
15830/15830 [==============================] - 1290s 81ms/step - loss: 0.0087 - accur
Epoch 4/7
  108/15830 [..............................] - ETA: 16:13 - loss: 0.0074 - accuracy:
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-12-6eebb463ac5a> in <cell line: 1>()
----> 1 resnet_model.fit(train_datagen, validation_data = val_datagen, epochs = 7)

                    10 frames
/usr/local/lib/python3.10/dist-packages/tensorflow/python/eager/execute.py in
quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     58         for t in inputs
     59     ]
---> 60     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
     61                                         inputs, attrs, num_outputs)
     62   except core._NotOkStatusException as e:

KeyboardInterrupt:
```

```
resnet_model.save('UCF.h5')
```

```
from tensorflow.keras.preprocessing import image
import numpy as np
```

```
img = image.load_img('/content/Test/NormalVideos/Normal_Videos_003_x264_1820.png', target_size = (64,64))
img
```

```
x = image.img_to_array(img)
x = np.expand_dims(x, axis = 0) #expanding the dimension of the array
pred = np.argmax(resnet_model.predict(x)) #predict the higher probability index
op =['abuse', 'arrest', 'arson', 'assult', 'burglary', 'explosion', 'fighting', 'normal', 'road accident', 'robbery', 'shooting', 'shopli
op[pred]
```

```
1/1 [==============================] - 0s 35ms/step
'normal'
```