Video Analysis Output & Result

```
import cv2

import math

import matplotlib.pyplot as plt

%matplotlib inline

import pandas as pd

from keras.preprocessing import image

import numpy as np

from keras.utils import np_utils

from skimage.transform import resize

count1 = 0

videoFile1 = "pickpocketing.mp4"

cap1 = cv21.VideoCapture(videoFile)   # capturing the video from the given path

frameRate1 = cap.get(5) #frame rate

x1=1

while(cap1.isOpened()):

    frameId1 = cap.get(1) #current frame number
```

```
    ret1, frame1 = cap.read()
    if (ret1 != True):
        break
    if (frameId1 % math.floor(frameRate) == 0):
        filename1 ="image%d.jpg" % count;count+=1
        cv2.imwrite(filename, frame)
cap.release()


count1 = 0
videoFile1 = "accident 1.mp4"
cap1 = cv2.VideoCapture(videoFile)   # capturing the
video from the given path
frameRate1 = cap.get(5) #frame rate
x1=1
while(cap.isOpened()):
    frameId1 = cap.get(1) #current frame number
    ret1, frame1 = cap.read()
    if (ret1 != True):
```

```
        break
    if (frameId1 % math.floor(frameRate) == 0):
        filename1 ="imagee%d.jpg" % count;count+=1
        cv2.imwrite(filename, frame)
cap.release()


count3 = 0
videoFile3 = "shoplifting.mp4"
cap3 = cv2.VideoCapture(videoFile)   # capturing the video from the given path
frameRate3 = cap.get(5) #frame rate
x3=1
while(cap.isOpened()):
    frameId3 = cap.get(1) #current frame number
    ret3, frame3 = cap.read()
    if (ret3 != True):
        break
    if (frameId3 % math.floor(frameRate) == 0):
```

```
        filename3 ="image%d.jpg" % count;count+=1
        cv2.imwrite(filename, frame)
cap.release()



X1 = [ ]     # creating an empty array
for img_name1 in data.Image_ID1:
    img1 = plt.imread(" + img_name)
    X1.append(img)  # storing each image in array X
X1 = np.array(X1)    # converting list to array


y1 = data.Class
dummy_y1 = np_utils.to_categorical(y1)    # one hot
encoding Classes


from keras.applications.vgg16 import preprocess_input
X1= preprocess_input(X, mode='tf')      # preprocessing
the input data
```

```
from sklearn.model_selection import train_test_split

X1_train, X1_valid, y1_train, y1_valid =
train_test_split(X, dummy_y1, test1_size=0.3,
random_state1=42)

from keras.models import Sequential

from keras.applications.vgg16 import VGG16

from keras.layers import Dense, InputLayer, Dropout



base_model1 = VGG16(weights='imagenet',
include_top=False, input_shape=(224, 224, 3))    #
include_top=False to remove the top layer

X_train1 = base_model.predict(X_train)

X_valid1 = base_model.predict(X_valid)

X_train1.shape, X_valid1.shape


X_train1 = X_train1.reshape(223, 7*7*512)      #
converting to 1-D
```

```
X_valid1 = X_valid1.reshape(95, 7*7*512)



# i. Building the model

model = Sequential()

model.add(InputLayer((9*9*512,)))    # input layer

model.add(Dense(units=1024, activation='sigmoid')) # hidden layer

model.add(Dense(5, activation='softmax'))    # output layer


model.summary()

# ii. Compiling the model

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])# preparing the validation set
```

Output

```
y1 = data1.Class

dummy_y1 = np_utils.to_categorical(y1)
```

#Traffic information

y1 = data1.Class

dummy_y1 = np_utils.to_categorical(y1)

#pickpocketing information

y1 = data1.Class

dummy_y1 = np_utils.to_categorical(y1)

#Accident Information


Social Media