MAJOR PROJECT 3

Object Detection (in both real-time and from a video file) using OpenCV

```
#OBJECT DETECTION USING OPENCV

import cv2

#importing the tensorflow trained model:
conf_file = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
weights_file = 'frozen_inference_graph.pb'

mod = cv2.dnn_DetectionModel(conf_file, weights_file)

#defining the object classes using the coco dataset:
classObjs = []
obj_file = 'objs.txt'
with open(obj_file,'rt') as fr:
   classObjs = fr.read().rstrip('\n').split('\n')

#defining input frame parameters according to the model configuration:
mod.setInputSize(320,320)
mod.setInputScale(1.0/127.5)
mod.setInputMean((127.5,127.5,127.5))
mod.setInputSwapRB(True)



#cap= cv2.VideoCapture(0)    #for video capture through webcam
cap= cv2.VideoCapture('Busy_city.mp4')     #for video capture through video file
```

```
while True:
    ret, frame = cap.read()

    #passing the frames to the model with a detection threshold of 0.5
    clsIn, confidence, bbox = mod.detect(frame,confThreshold=0.5)
    print(clsIn)

    if (len(clsIn) != 0):
        #zip is used as we have to consider 3 variables
        #flattening is used to convert the variables to an 1D-array
        for ClassIndex, conf, boxes in zip(clsIn.flatten(), confidence.flatten(), bbox):
            if (ClassIndex<=80):
                cv2.rectangle(frame,boxes,(255,0,0),2)    #blue rectangle frame
                cv2.putText(frame,classObjs[ClassIndex-1].upper(),
                        (boxes[0]+10,boxes[1]+35), cv2.FONT_HERSHEY_TRIPLEX, fontScale=0.7,
                        color= (0,255,0), thickness=2)
                #this shows the name of the objects in the rectangular frame in uppercase in green
color

    cv2.imshow('Object Detection',frame)
    if cv2.waitKey(1) == 27:  #ESC key to close the frame
        break

cap.release()
cv2.destroyAllWindows()
```

Output of the above code: