

importing librairies

```
In [1]: import cv2 as cv
import matplotlib.pyplot as plt
```

loading our model

```
In [2]: configFile = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
frozenModel = 'frozen_inference_graph.pb'
model = cv.dnn_DetectionModel(frozenModel, configFile)
```

Creating the class file

```
In [3]: classFile = 'labels.txt'
classNames = []
with open(classFile, 'rt') as fpt:
    classNames = fpt.read().rstrip('\n').split('\n')
```

```
In [4]: print(classNames)
```

```
['person', 'bicycle', 'car', 'motorbike', 'aeroplane', 'bus', 'train', 'truck', 'boat',
'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat',
'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'sofa', 'pottedplant', 'bed', 'diningtable', 'toilet', 'tvmonitor', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']
```

```
In [5]: print(classNames, len(classNames)) #classNames ==> Contains all the 80 classes from the
```

```
['person', 'bicycle', 'car', 'motorbike', 'aeroplane', 'bus', 'train', 'truck', 'boat',
'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat',
'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'sofa', 'pottedplant', 'bed', 'diningtable', 'toilet', 'tvmonitor', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'] 80
```

Setting up the configuration of the model

```
In [6]: model.setInputSize(320,320)
model.setInputScale(1.0/127.5)
model.setInputMean((127.5,127.5,127.5))
model.setInputSwapRB(True)
```

```
Out[6]: <dnn_Model 000001F59D74CCD0>
```

Capturing the Video

```
In [7]: cap = cv.VideoCapture('traffic.mp4')
#cap = cv.VideoCapture(0) ==> webcam
if not cap.isOpened():
    raise IOError("Cannot Open Video")
```

Setting the font scale and font style

```
In [8]: font_scale = 1.1
font = cv.FONT_HERSHEY_COMPLEX
```

Reading each frame and detecting the objects in it

```
In [9]: while True:
    ret, frame = cap.read()
    ClassIndex, confidence, bbox = model.detect(frame, confThreshold=0.62)
    if len(ClassIndex) != 0:
        for ClassInd, conf, boxes in zip(ClassIndex.flatten(), confidence.flatten(), bbox):
            if ClassInd <= 80:
                cv.rectangle(frame, boxes, (255, 0, 0), 2)
                cv.putText(frame, classNames[ClassInd-1], (boxes[0]+10, boxes[1]+40), font,
                    cv.imshow("output", frame)
                if cv.waitKey(2) & 0xFF == ord('q'):
                    break #breaks out of the loop as soon as we press 'q' from our keyboard
cap.release()
cv.destroyAllWindows()
```

```
-----
error                                Traceback (most recent call last)
<ipython-input-9-303f9b4a4e35> in <module>
      1 while True:
      2     ret, frame = cap.read()
----> 3     ClassIndex, confidence, bbox = model.detect(frame, confThreshold=0.62)
      4     if len(ClassIndex) != 0:
      5         for ClassInd, conf, boxes in zip(ClassIndex.flatten(), confidence.flatten(
), bbox):

error: OpenCV(4.5.2) C:\Users\runneradmin\AppData\Local\Temp\pip-req-build-_8k9tw8n\open
cv\modules\imgproc\src\resize.cpp:4051: error: (-215:Assertion failed) !ssize.empty() in
function 'cv::resize'
```

```
In [ ]:
```