

```
1 import numpy as np
2 import imutils
3 import cv2
4
5 # List of class labels
6 CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
7            "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
8            "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
9            "sofa", "train", "tvmonitor"]
10
11 # Different colors for persons and other objects
12 PERSON_COLOR = (0, 255, 0) # Green for persons
13 OBJECT_COLOR = (0, 0, 255) # Red for other objects
14
15 # Load the pre-trained MobileNetSSD model
16 model = cv2.dnn.readNetFromCaffe(
17     'MobileNetSSD_deploy.prototxt',
18     'MobileNetSSD_deploy.caffemodel'
19 )
20
21 # Open video file (or use 0 for webcam)
22 # cap = cv2.VideoCapture(0) # For webcam
23 cap = cv2.VideoCapture('videoplayback.mp4') # For video file
24
25 while True:
26     ret, frame = cap.read()
27     if not ret:
28         break
29
30     # Resize the frame for better visualization
31     frame = imutils.resize(frame, width=1000)
32     (h, w) = frame.shape[:2]
```

```
14
15 # Load the pre-trained MobileNetSSD model
16 model = cv2.dnn.readNetFromCaffe(
17     'MobileNetSSD_deploy.prototxt',
18     'MobileNetSSD_deploy.caffemodel'
19 )
20
21 # Open video file (or use 0 for webcam)
22 # cap = cv2.VideoCapture(0) # For webcam
23 cap = cv2.VideoCapture('videoplayback.mp4') # For video file
24
25 while True:
26     ret, frame = cap.read()
27     if not ret:
28         break
29
30     # Resize the frame for better visualization
31     frame = imutils.resize(frame, width=1000)
32     (h, w) = frame.shape[:2]
33
34     # Prepare the frame for the model
35     blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
36                                 0.007843, (300, 300), 127.5)
37     model.setInput(blob)
38     detections = model.forward()
39
40     # Loop over the detections
41     for i in range(detections.shape[2]):
42         confidence = detections[0, 0, i, 2]
43
44         if confidence > 0.5:
45             idx = int(detections[0, 0, i, 1])
```

```
EXPLORER
v UNTITLED (WORKSPACE)
  ffgfg
  MobileNetSSD_deploy.caffemodel
  MobileNetSSD_deploy.prototxt
  Object detection.py
  videoplayback.mp4

Object detection.py
ffgfg > Object detection.py > ...
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

model.setInput(blob)
detections = model.forward()

# Loop over the detections
for i in range(detections.shape[2]):
    confidence = detections[0, 0, i, 2]

    if confidence > 0.5:
        idx = int(detections[0, 0, i, 1])
        label = CLASSES[idx]
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # Choose color based on the class
        color = PERSON_COLOR if label == "person" else OBJECT_COLOR

        # Draw the bounding box and label
        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
        text = "{}: {:.2f}%".format(label, confidence * 100)
        y = startY - 15 if startY - 15 > 15 else startY + 15
        cv2.putText(frame, text, (startX, y),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, color, 2)

# Display the result
cv2.imshow("Object & Person Detection", frame)
key = cv2.waitKey(1) & 0xFF
if key == ord('q'):
    break

# Release resources
cap.release()
cv2.destroyAllWindows()
```



