

```

import cv2
import numpy as np
#https://www.youtube.com/results?search_query=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3D1LCb1PVqzeY

net = cv2.dnn.readNet('yolov3.weights', 'yolov3.cfg')

classes = []
with open(r"C:\Users\nurau\Downloads\Sairam opencv\coco.names", "r") as f:
    classes = f.read().splitlines()

# Read the input image
img = cv2.imread(r"C:\Users\nurau\Downloads\Sairam opencv\Swami3.jpg")

# Define the desired width and height of the display window
window_width = 800
window_height = 600

# Get the original width and height of the image
height, width, _ = img.shape

# Calculate the aspect ratio
aspect_ratio = width / height

# Calculate the new width and height based on the aspect ratio
if aspect_ratio > 1:
    new_width = window_width
    new_height = int(window_width / aspect_ratio)
else:
    new_height = window_height
    new_width = int(window_height * aspect_ratio)

# Resize the image
resized_img = cv2.resize(img, (new_width, new_height))

# Create a blob from the resized image
blob = cv2.dnn.blobFromImage(resized_img, 1/255, (416, 416), (0, 0, 0), swapRB=True, crop=False)

net.setInput(blob)
output_layers_names = net.getUnconnectedOutLayersNames()
layerOutputs = net.forward(output_layers_names)

boxes = []
confidences = []
class_ids = []

for output in layerOutputs:
    for detection in output:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.2:
            center_x = int(detection[0] * new_width)
            center_y = int(detection[1] * new_height)
            w = int(detection[2] * new_width)
            h = int(detection[3] * new_height)

            x = int(center_x - w/2)
            y = int(center_y - h/2)

            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

# Define font
font = cv2.FONT_HERSHEY_SIMPLEX

```

```
num_classes = len(classes)
colors = np.random.randint(0, 255, size=(num_classes, 3), dtype=np.uint8)

for i in indexes.flatten():
    x, y, w, h = boxes[i]
    label = str(classes[class_ids[i]])
    confidence = str(round(confidences[i], 2))
    color = tuple(map(int, colors[i]))

    # Reduce font size and change font color
    font_size = 0.5
    #white if all the three have 255
    font_color = (255,255,255) # Colour changing variable

    cv2.rectangle(resized_img, (x, y), (x+w, y+h), color, 2)
    cv2.putText(resized_img, label + " " + confidence, (x, y+20), font, font_size, font_color, 1)

# Display the resized image
cv2.imshow('Image', resized_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```