


```
!git clone https://github.com/misbah4064/age_and_gender_detection.git
%cd age_and_gender_detection
```

```
Cloning into 'age_and_gender_detection'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
/content/age_and_gender_detection
```

```
# Downloading pretrained data and unzipping it
!gdown https://drive.google.com/uc?id=1_aDSc0vBeBLCn_iv0oxS08X1ySQpSbIS
# https://drive.google.com/uc?id=1_aDSc0vBeBLCn_iv0oxS08X1ySQpSbIS
!unzip modelNweight.zip
```

 Downloading...

From: [https://drive.google.com/uc?id=1\\_aDSc0vBeBLCn\\_iv0oxS08X1ySQpSbIS](https://drive.google.com/uc?id=1_aDSc0vBeBLCn_iv0oxS08X1ySQpSbIS)  
To: /content/age\_and\_gender\_detection/modelNweight.zip  
86.2MB [00:00, 151MB/s]  
Archive: modelNweight.zip  
creating: modelNweight/  
inflating: modelNweight/age\_deploy.prototxt  
inflating: modelNweight/age\_net.caffemodel  
inflating: modelNweight/gender\_deploy.prototxt  
inflating: modelNweight/gender\_net.caffemodel  
inflating: modelNweight/opencv\_face\_detector.pbtxt  
inflating: modelNweight/opencv\_face\_detector\_uint8.pb

```
# Import required modules
import cv2 as cv
import math
import time
from google.colab.patches import cv2_imshow

def getFaceBox(net, frame, conf_threshold=0.7):
    frameOpencvDnn = frame.copy()
    frameHeight = frameOpencvDnn.shape[0]
    frameWidth = frameOpencvDnn.shape[1]
    blob = cv.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

    net.setInput(blob)
    detections = net.forward()
    bboxes = []
    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > conf_threshold:
            x1 = int(detections[0, 0, i, 3] * frameWidth)
            y1 = int(detections[0, 0, i, 4] * frameHeight)
            x2 = int(detections[0, 0, i, 5] * frameWidth)
            y2 = int(detections[0, 0, i, 6] * frameHeight)
            bboxes.append([x1, y1, x2, y2])
```

```

        cv.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0), int(round(frameHeight/150)),
        return frameOpencvDnn, bboxes

faceProto = "modelNweight/opencv_face_detector.pbtxt"
faceModel = "modelNweight/opencv_face_detector_uint8.pb"

ageProto = "modelNweight/age_deploy.prototxt"
ageModel = "modelNweight/age_net.caffemodel"

genderProto = "modelNweight/gender_deploy.prototxt"
genderModel = "modelNweight/gender_net.caffemodel"

MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList = ['Male', 'Female']

# Load network
ageNet = cv.dnn.readNet(ageModel, ageProto)
genderNet = cv.dnn.readNet(genderModel, genderProto)
faceNet = cv.dnn.readNet(faceModel, faceProto)

padding = 20

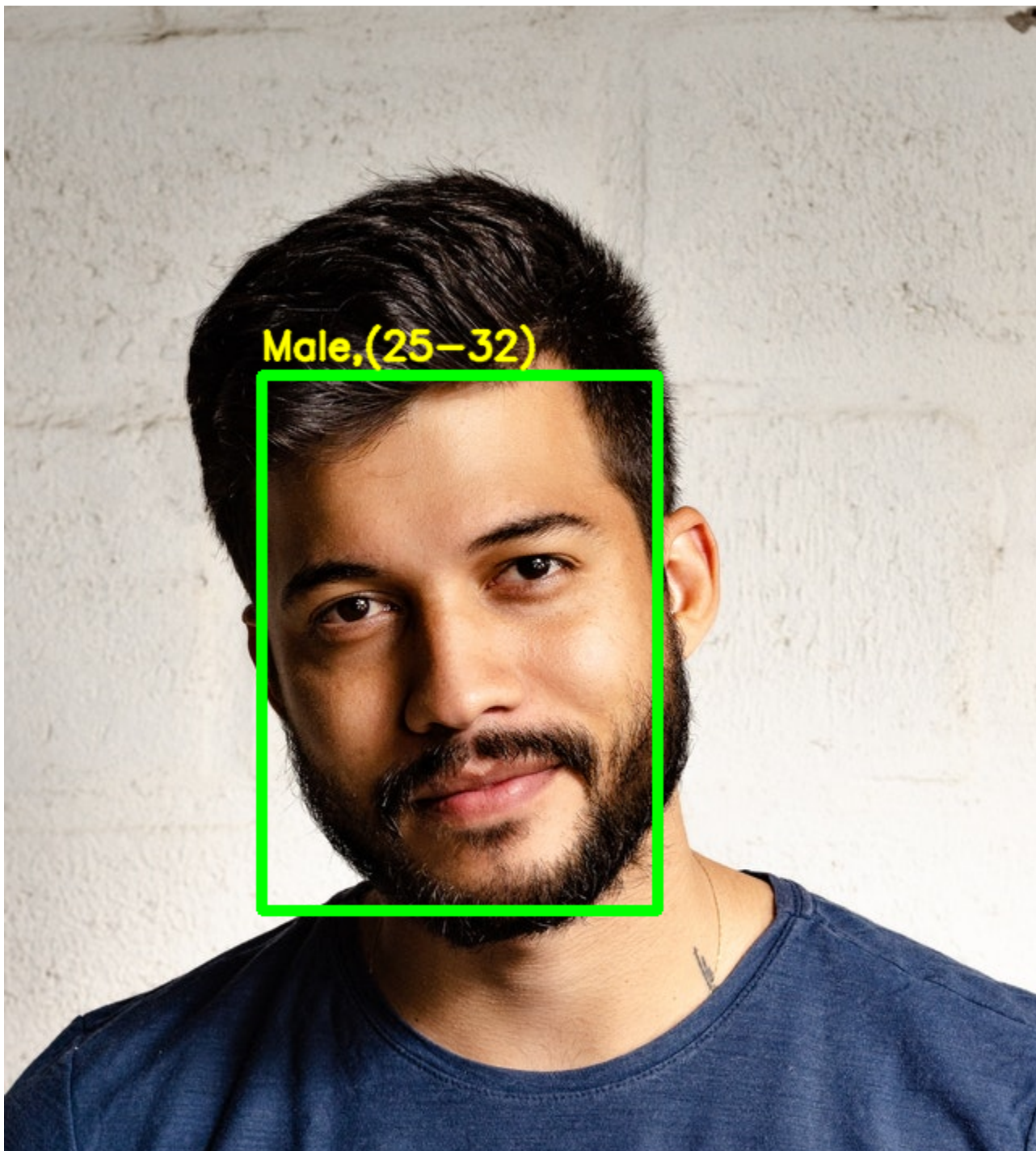
def age_gender_detector(frame):
    # Read frame
    t = time.time()
    frameFace, bboxes = getFaceBox(faceNet, frame)
    for bbox in bboxes:
        # print(bbox)
        face = frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[0]-1),max(0,bbox[0]-padding,frame.shape[1]-1)]

        blob = cv.dnn.blobFromImage(face, 1.0, (227, 227), MODEL_MEAN_VALUES, swapRB=False)
        genderNet.setInput(blob)
        genderPreds = genderNet.forward()
        gender = genderList[genderPreds[0].argmax()]
        ageNet.setInput(blob)
        agePreds = ageNet.forward()
        age = ageList[agePreds[0].argmax()]

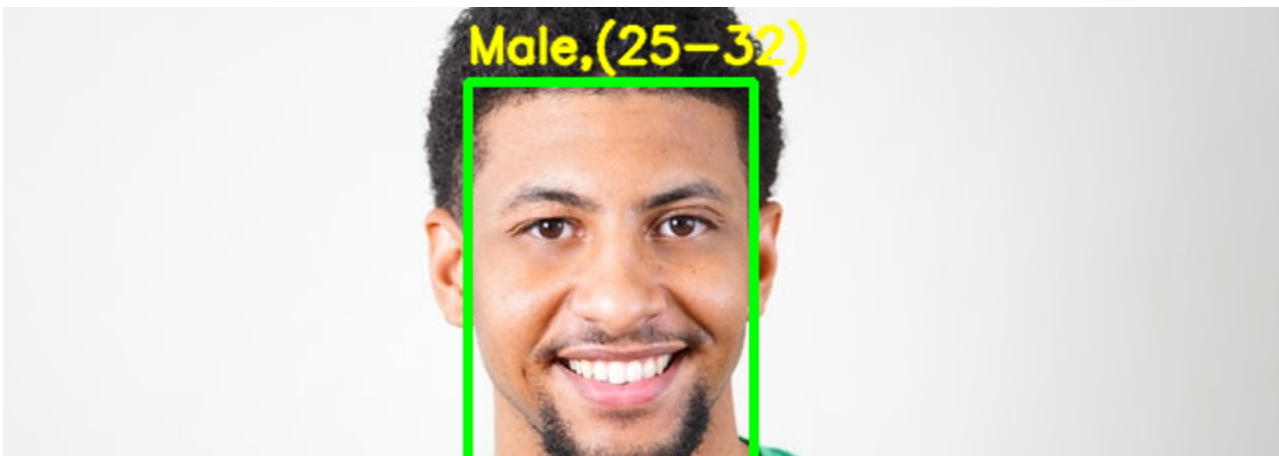
        label = "{},{}".format(gender, age)
        cv.putText(frameFace, label, (bbox[0], bbox[1]-10), cv.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 255))
    return frameFace

input = cv.imread("image.jpg")
output = age_gender_detector(input)
cv2.imshow(output)

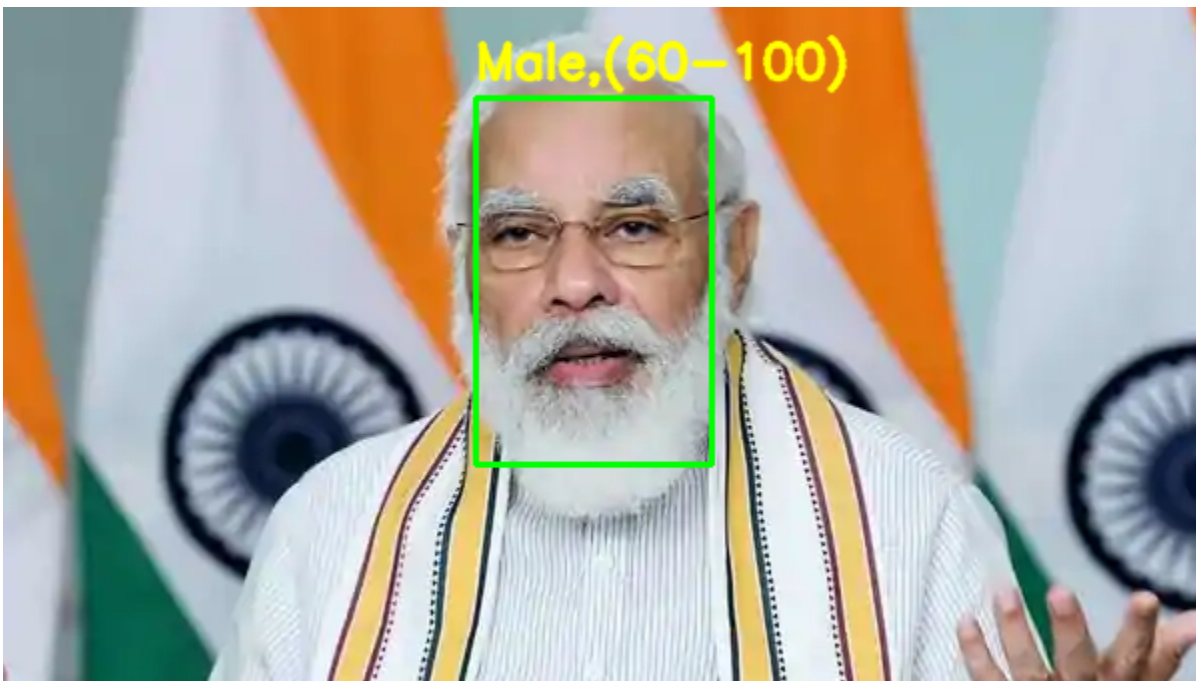
```



```
input = cv.imread("image1.jpg")  
output = age_gender_detector(input)  
cv2.imshow(output)
```



```
input = cv.imread("image2.jpg")
output = age_gender_detector(input)
cv2.imshow(output)
```



```
import cv2
cap = cv2.VideoCapture('video.mp4')
ret, frame = cap.read()
frame_height, frame_width, _ = frame.shape
out = cv2.VideoWriter('output.avi',cv2.VideoWriter_fourcc('M','J','P','G'), 10, (frame_width,frame_height))
print("Processing Video...")
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        out.release()
        break
    output = age_gender_detector(frame)
    out.write(output)
out.release()
print("Done processing video")
```

