```
!git clone https://github.com/moses1626/Project-No-34-Age-and-Gender-
Prediction
%cd Age-and-Gender-Prediction

# Downloading pretrained data and unzipping it
!gdown https://drive.google.com/uc?id=1_aDScOvBeBLCn_iv0oxSO8X1ySQpSbIS
# https://drive.google.com/uc?id=1_aDScOvBeBLCn_iv0oxSO8X1ySQpSbIS
!unzip modelNweight.zip

# 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)), 8)
    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):min(bbox[2]+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, 0.4, (0, 0, 255), 1, cv.LINE_AA)
    return frameFace
```

Explained:

The provided code is for an age and gender prediction model using deep learning and computer vision techniques. Let's go through the code step by step:

1. `!git clone https://github.com/moses1626/Project-No-34-Age-and-Gender-Prediction`

   This command clones a GitHub repository containing the code and necessary files for the age and gender prediction project.

2. `%cd Age-and-Gender-Prediction`

This command changes the current working directory to the "Age-and-Gender-Prediction" directory.

3. `!gdown https://drive.google.com/uc?id=1_aDScOvBeBLCn_iv0oxSO8X1ySQpSbIS`

This command downloads a file named "modelNweight.zip" from Google Drive. The file likely contains pre-trained weights and models required for the age and gender prediction.

4. `!unzip modelNweight.zip`

This command unzips the downloaded "modelNweight.zip" file, extracting the contents.

5. `import cv2 as cv`

This imports the OpenCV library, which provides functions for computer vision tasks.

6. `import math`

This imports the math module, which provides mathematical functions and constants.

7. `import time`

This imports the time module, which is used for time-related operations.

8. `from google.colab.patches import cv2_imshow`

This imports the `cv2_imshow` function from the `google.colab.patches` module. It is used for displaying images in the Colab notebook.

9. The `getFaceBox` function takes the neural network, frame, and a confidence threshold as input. It performs face detection on the given frame using the neural network and returns the modified frame with bounding boxes around detected faces and the coordinates of the bounding boxes.

10. The code defines several file paths for different models and weights required for the age and gender prediction. These paths specify the locations of pre-trained models and configuration files.

11. `MODEL_MEAN_VALUES` is a tuple containing mean values used for preprocessing the input image.

12. `ageList` is a list of age ranges used for age prediction.

13. `genderList` is a list of gender labels used for gender prediction.

14. The code loads the pre-trained age, gender, and face detection networks using the `cv.dnn.readNet` function.

15. `padding` is a variable that determines the padding around the detected face region.

16. The `age_gender_detector` function takes a frame as input. It detects faces in the frame using the `getFaceBox` function and performs age and gender prediction on each detected face. It returns the modified frame with labels indicating the predicted age and gender for each face.

17. Lastly, the code defines the main logic for running the age and gender prediction on a video or series of images. It utilizes the `age_gender_detector` function to process each frame and display the output.

Note: This code appears to be designed to run in a specific environment, such as Google Colab, which provides access to GPU resources for accelerated deep learning computations.