

Project Based Learning Report

on

Design Gender and Age Detection with Data Science with OpenCV

Submitted in the partial fulfillment of the requirements
For the Project based learning in (**Essentials of Data Science**)
in
Electronics & Communication Engineering

By

2014111043	Kushagra Goel
2014111076	Mansi Lal
2014111050	Rohit Kumar Gupta

Under the guidance of Course In-charge

Prof. Dnyanesh S.Lavhkare

Department of Electronics & Communication Engineering

Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 4110043

Academic Year: 2021-22

**Bharati Vidyapeeth
(Deemed to be University)
College of Engineering,
Pune – 411043**

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

**Certified that the Project Based Learning report entitled,
“Design Gender and Age Detection with Data Science with
OpenCV” is work done by**

**2014111043 Kushagra Goel
2014111076 Mansi Lal
2014111050 Rohit Kumar Gupta**

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL)
in **Essentials of Data Science Course** of Bachelor of Technology Semester IV, in Branch name.

Date: 24 May 2022

**Prof. Dnyanesh S. Lavhkare
Course In-charge**

**Dr. Tanuja S. Dhope
PBL Co-Ordinator**

Dr. Arundhati A Shinde

Professor & Head

ELECTRONICS & COMMUNICATION ENGINEERING

INDEX: -

Page No.	Contents
1-3	Problem Statement with Solution
4-8	Description about project
9-9	Software used
10-15	Result with Analysis

Problem Statement: -

What is Data Science? Why learn Data Science?

Solution:-

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models. The data used for analysis can come from many different sources and presented in various formats.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate insights which analysts and business users can translate into tangible business value.

Reasons to learn Data Science are: -

- 1) Learning about data science provides an opportunity for you to recreate yourself.
- 2) We live in a digital world, everything is data-driven. There is data science in business, accounting, education, science, engineering, healthcare, technology, energy sector, government, and so on.
- 3) Data science is also a very promising field with lots of high paying job opportunities.
- 4) Basic data science skills are important for personal use.
- 5) Great potential to branch out with different options.
- 6) Become a decision-maker, not every job opportunity will give you the power to make informed business decisions. For a data scientist, that is the core responsibility.
- 7) Less competitive because it is a highly analytical role, competition is less, but demand is not. With a limited talent pool, there is always a challenge for businesses to hire in these roles.

What is Computer Vision?

Computer Vision is the field of study that enables computers to see and identify digital images and videos as a human would. The challenges it faces largely follow from the limited understanding of biological vision. Computer Vision involves acquiring, processing, analysing, and understanding digital images to extract high-dimensional data from the real world in order to generate symbolic or numerical information which can then be used to make decisions. The process often includes practices like object recognition, video tracking, motion estimation, and image restoration.

What is OpenCV?

OpenCV is short for Open-Source Computer Vision. Intuitively by the name, it is an open-source Computer Vision and Machine Learning library. This library is capable of processing real-time image and video while also boasting analytical capabilities. It supports the Deep Learning frameworks **TensorFlow**, **Caffe**, and **PyTorch**.

What is a CNN?

A **Convolutional Neural Network** is a deep neural network (DNN) widely used for the purposes of image recognition and processing and **NLP**. Also known as a ConvNet, a CNN has input and output layers, and multiple hidden layers, many of which are convolutional. In a way, CNNs are regularized multilayer perceptrons.

Gender and Age Detection Python Project- Objective

To build a gender and age detector that can approximately guess the gender and age of the person (face) in a picture using **Deep Learning** on the Adience dataset.

The CNN Architecture

The convolutional neural network for this python project has 3 convolutional layers:

- Convolutional layer; 96 nodes, kernel size 7
- Convolutional layer; 256 nodes, kernel size 5
- Convolutional layer; 384 nodes, kernel size 3

It has 2 fully connected layers, each with 512 nodes, and a final output layer of softmax type.

To go about the python project, we'll:

- Detect faces
- Classify into Male/Female
- Classify into one of the 8 age ranges
- Put the results on the image and display it

The Dataset

For this python project, we'll use the **Adience dataset**; the dataset is available in the public domain and is available on Kaggle. This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It has a total of 26,580 photos of 2,284 subjects in eight age ranges (as mentioned above) and is about 1GB in size. The models we will use have been trained on this dataset.

Prerequisites

You'll need to install OpenCV (cv2) to be able to run this project. You can do this with pip-

```
pip install opencv-python
```

Other packages you'll be needing are math and argparse, but those come as part of the standard Python library.

Building Gender and Age Detection Project Using OpenCV

1.Import the libraries and load the dataset

First, we are going to import all the modules that we are going to need for training our model.

Let's get started. If you haven't OpenCV already installed, make sure to do so:

```
$ pip install opencv-python numpy
```

Open up a new file. Importing the libraries:

```
import cv2
import math
import argparse
```

Next, defining the variables of weights and architectures for face, age, and gender detection models:

```
# face model
faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
# Age model
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
# Gender model
genderProto="gender_deploy.prototxt"
genderModel="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']
```

Below are the necessary files to be included in the project directory:

- `gender_net.caffemodel`: It is the pre-trained model weights for gender detection.
- `gender_deploy.prototxt`: is the model architecture for the gender detection model (a plain text file with a JSON-like structure containing all the neural network layer's definitions).
- `age_net.caffemodel`: It is the pre-trained model weights for age detection.
- `age_deploy.prototxt`: is the model architecture for the age detection model (a plain text file with a JSON-like structure containing all the neural network layer's definitions).
- `opencv_face_detector_uint8.pb`: The pre-trained model weights for face detection.
- `opencv_face_detector.pbtxt`: This is the model architecture for the face detection model.

Next, loading the models:

```
#To load face model
faceNet=cv2.dnn.readNet(faceModel,faceProto)
#To load Age model
ageNet=cv2.dnn.readNet(ageModel,ageProto)
#To load Gender model
genderNet=cv2.dnn.readNet(genderModel,genderProto)
```

2. We use the argparse library to create an argument parser so we can get the image argument from the command prompt. We make it parse the argument holding the path to the image to classify gender and age for.

3. For face, age, and gender, initialize protocol buffer and model.

4. Initialize the mean values for the model and the lists of age ranges and genders to classify from.

5. Now, use the readNet() method to load the networks. The first parameter holds trained weights and the second carries network configuration.

6. Let's capture video stream in case you'd like to classify on a webcam's stream. Set padding to 20.

7. Now until any key is pressed, we read the stream and store the content into the names hasFrame and frame. If it isn't a video, it must wait, and so we call up waitKey() from cv2, then break.

8. Let's make a call to the `highlightFace()` function with the `faceNet` and `frame` parameters, and what this returns, we will store in the names `resultImg` and `faceBoxes`. And if we got 0 `faceBoxes`, it means there was no face to detect.

Here, `net` is `faceNet`- this model is the DNN Face Detector and holds only about 2.7MB on disk.

- Create a shallow copy of `frame` and get its height and width.
- Create a blob from the shallow copy.
- Set the input and make a forward pass to the network.
- `faceBoxes` is an empty list now. for each value in 0 to 127, define the confidence (between 0 and 1). Wherever we find the confidence greater than the confidence threshold, which is 0.7, we get the `x1`, `y1`, `x2`, and `y2` coordinates and append a list of those to `faceBoxes`.
- Then, we put up rectangles on the image for each such list of coordinates and return two things: the shallow copy and the list of `faceBoxes`.

9. But if there are indeed `faceBoxes`, for each of those, we define the face, create a 4-dimensional blob from the image. In doing this, we scale it, resize it, and pass in the mean values.

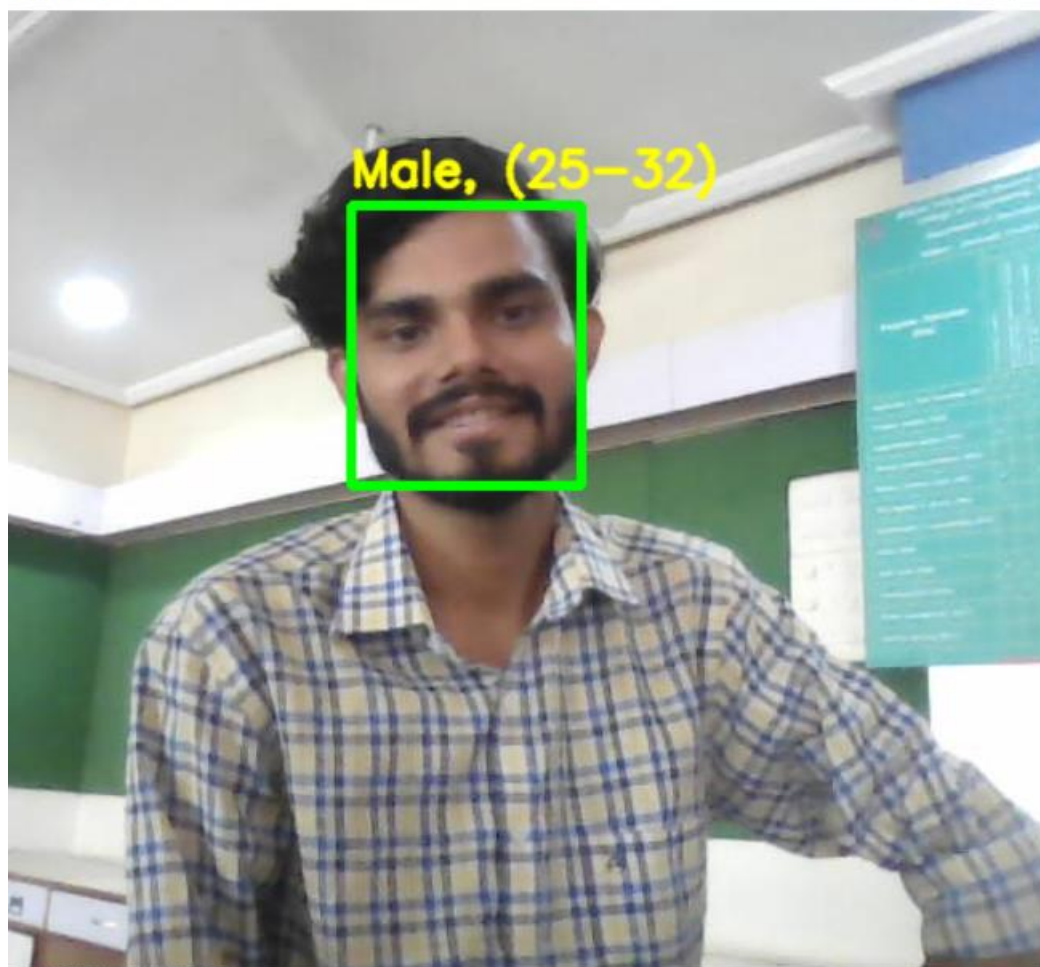
10. We feed the input and give the network a forward pass to get the confidence of the two class. Whichever is higher, that is the gender of the person in the picture.

11. Then, we do the same thing for age.

12. We'll add the gender and age texts to the resulting image and display it with `imshow()`.

a gender

— □ ×



Software Used:

Visual Studio: -

Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

There are 3 editions of Microsoft Visual Studio as follows:

- 1. Community:** It is a free version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than \$1 Million(US Dollars) then you are not permitted to use this edition. In a non-enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem(Access to thousands of extensions) and Languages(You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.
- 2. Professional:** It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility(Professional developer tools for building any application type), Productivity(Powerful features such as CodeLens improve your team's productivity), Collaboration(Agile project planning tools, charts, etc.) and Subscriber benefits like Microsoft software, plus Azure, Pluralsight, etc.
- 3. Enterprise:** It is an integrated, end to end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high-quality software.

Result with Analysis

Analysis of the code: -

- First, we have imported three libraries - cv2 library, math library & argparse library.
- Secondly, we have loaded our dataset - Adience dataset.
- In this Python Project, we will use Deep Learning to accurately identify the gender and age of a person from a single image of a face. We will use the models trained by [Tal Hassner and Gil Levi](#).
- The predicted gender may be one of 'Male' and 'Female', and the predicted age may be one of the following ranges- (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100) (8 nodes in the final SoftMax layer).
- It is very difficult to accurately guess an exact age from a single image because of factors like makeup, lighting, obstructions, and facial expressions.
- And so, we make this a classification problem instead of making it one of regression.

Source Code: -

```
# Group 5 PBL Project
#Members
#1.Kushagra Goel(ECE-1)
#2.Mansi Lal(ECE-1)
#3.Rohit Kumar Gupta(ECE-1)

import cv2
import math
import argparse

def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn=frame.copy()
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123],
True, False)

    net.setInput(blob)
    detections=net.forward()
    faceBoxes=[]
    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)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0),
int(round(frameHeight/150)), 8)
        return frameOpencvDnn,faceBoxes

parser=argparse.ArgumentParser()
parser.add_argument('--image')

args=parser.parse_args()

# face model
faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
# Age model
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
# Gender model
genderProto="gender_deploy.prototxt"
genderModel="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']

#To load face model
faceNet=cv2.dnn.readNet(faceModel,faceProto)
#To load Age model
ageNet=cv2.dnn.readNet(ageModel,ageProto)
#To load Gender model
genderNet=cv2.dnn.readNet(genderModel,genderProto)

video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
while cv2.waitKey(1)<0:
    hasFrame,frame=video.read()
    if not hasFrame:
        cv2.waitKey()
        break

    resultImg,faceBoxes=highlightFace(faceNet,frame)
    if not faceBoxes:

```

```

print("No face detected")

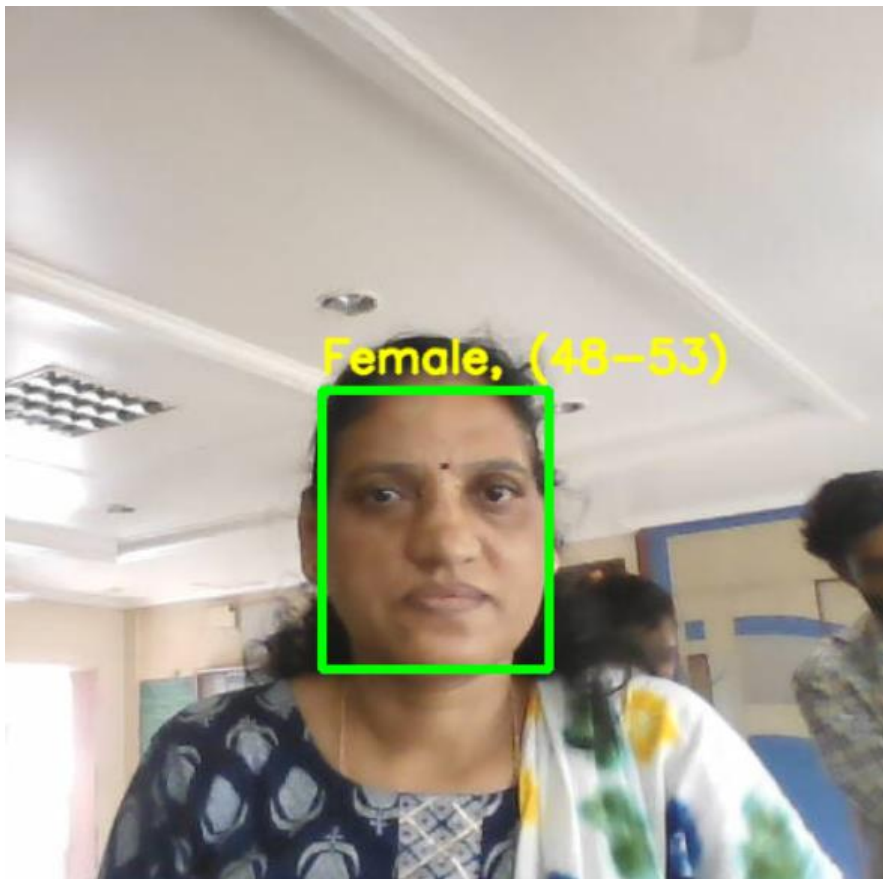
for faceBox in faceBoxes:
    face=frame[max(0,faceBox[1]-padding):
              min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
              :min(faceBox[2]+padding, frame.shape[1]-1)]

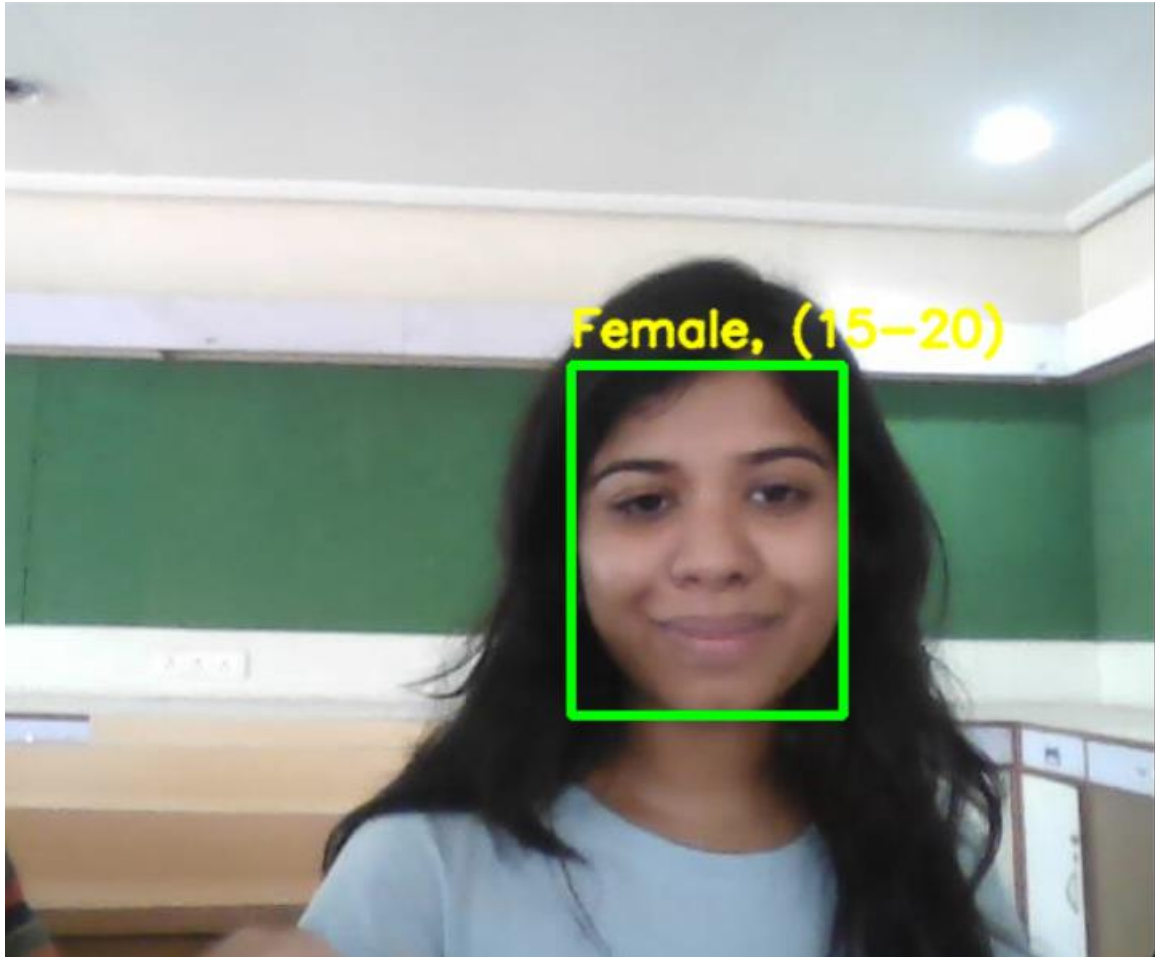
    blob=cv2.dnn.blobFromImage(face, 1.0, (227,227), MODEL_MEAN_VALUES,
swapRB=False)
    genderNet.setInput(blob)
    genderPreds=genderNet.forward()
    gender=genderList[genderPreds[0].argmax()]
    print(f'Gender: {gender}')


    ageNet.setInput(blob)
    agePreds=ageNet.forward()
    age=ageList[agePreds[0].argmax()]
    print(f'Age: {age[1:-1]} years')

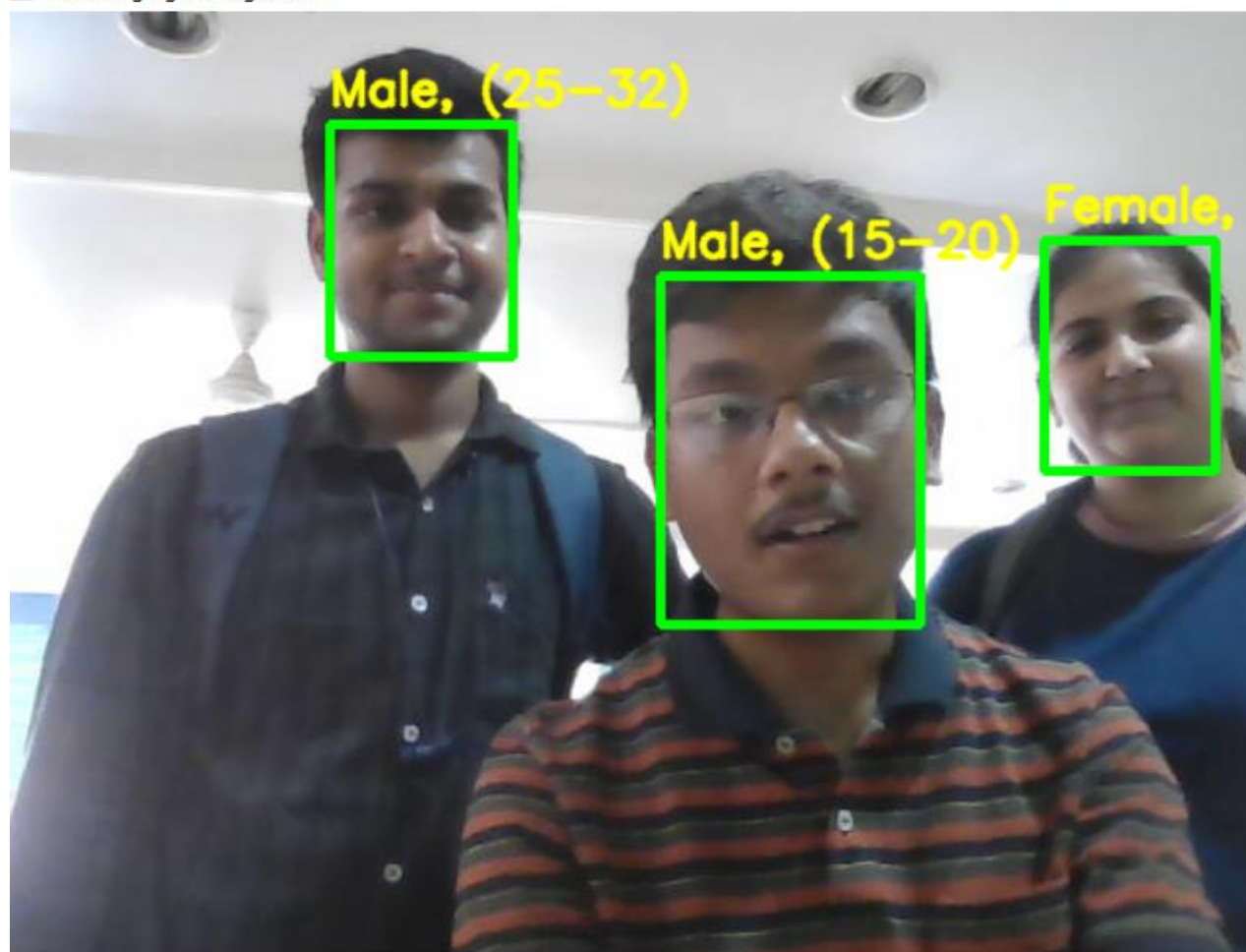
    cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_ITALIC, 0.8, (0,255,255), 2, cv2.LINE_AA)
    cv2.imshow("Detecting age and gender", resultImg)

```





 Detecting age and gender



Project Outcome: -

In this project, we learned about OpenCV and CNN and used pre-defined caffe models to detect age and Gender which implemented our project using Python.

CO4: Correlate results to solution approach followed (Analysis).

CO5: Assess the solution approach (Evaluation).
