KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Course Activity Report on

"Gender and Age Detection"

Submitted in the partial fulfillment for the academic requirement of

6th Semester B.E Computer Science & Engineering

In

Artificial Intelligence & Machine Learning Lab

Submitted by

NAME OF THE CANDIDATES	USN
1. MEGHAN GHIVARI	2GI18CS071
2. NIKHIL TALAWAR	2GI18CS079
3. PAVAN BAGEWADI	2GI18CS088
4. PRATHIK GOWDA	2GI18CS096

Course project report and ppt content

1. Title
2. Problem statement for that the project
3. Objectives of Defined Problem statement
4. Design / Algorithm/Flowchart/Methodology
5. Implementation details/Function/Procedures/Classes and Objects (Language/Tools)
6. Working model of the final solution
7. Report and Oral Presentation skill

Marks allocation:

	Batch No. : **6**					
1.	Project Title: **GENDER & AGE DETECTION**	Marks Range	USN			
			2GI18CS071	2GI18CS079	2GI18CS088	2GI18CS0096
2.	Problem statement (PO2)	0-1				
3.	Objectives of Defined Problem statement (PO1,PO2)	0-2				
4.	Design / Algorithm/Flowchart/Methodology (PO3)	0-3				
5.	Implementation details/Function/Procedures/Classes and Objects (Language/Tools) (PO1,PO3,PO4,PO5)	0-4				
6.	Working model of the final solution (PO3,PO12)	0-5				
7.	Report and Oral presentation skill (PO9,PO10)	0-5				
	Total	20				

*** 20 marks is converted to 10 marks for CGPA calculation**

1.Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2.Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and Engineering sciences.
3.Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4.Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5.Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6.The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7.Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8.Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9.Individual and team work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

10.Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Introduction

First introducing you with the terminologies used in this advanced python project of gender and age detection –

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, analyzing, 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.

Gender and Age Detection – About the Project

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.

Project Overview

- We used Adience dataset for age and gender classification.
- Trained CNN (Convolutional Neural Networks) using Adience dataset for age and gender prediction.
- Created a python application using OpenCV deep learning module to perform real time age and gender detection.

The Dataset

For this python project, we'll use the Adience dataset; the dataset is available in the public domain and you can find it here. 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.

Adience Dataset

It attempts to capture all the variations in appearance, noise, pose, lighting and more, that can be expected of images taken without careful preparation or posing.

- Total number of Genders : 2(Male,Female)
- Total number of subjects : 2,284
- Number of age groups : 8

  (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60-)
- Gender labels : Yes
- Subject labels : Yes

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

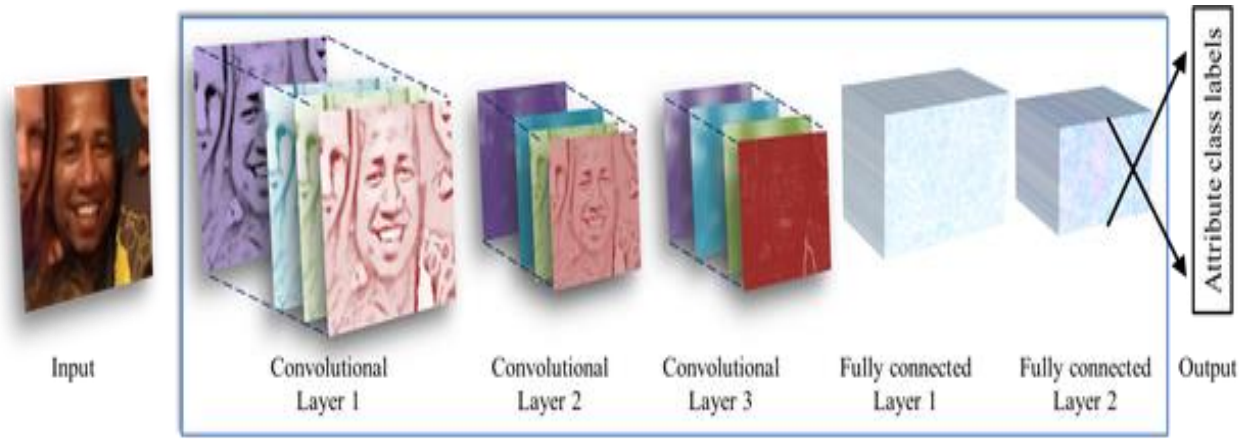
  It has 2 fully connected layers, each with 512 nodes, and a final output layer of softmax type.

These networks have 3 types of layers: Input layer, hidden layer and output layer. In these networks, data moves from the input layer through the hidden nodes and to the output nodes.

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 same network architecture is used for both age and gender classification. The proposed network comprises of only three convolutional layers and two fully-connected layers with a small number of neurons. This architecture is relatively shallow, compared to the much larger architectures applied. A schematic illustration of the network is below:



The network contains three convolutional layers, each followed by a ReLU operation and a pooling layer. The first two layers also follow an LRN layer [14]. The first Convolutional Layer contains 96 filters of 7×7 pixels, the second Convolutional Layer contains 256 filters of 5×5 pixels, The third and final Convolutional Layer contains 384 filters of 3 × 3 pixels. Finally, two fully-connected layers are added, each containing 512 neurons and each followed by a ReLU operation and a dropout layer.

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.

Steps for practicing gender and age detection python project

1. D**ownload the zip file from the given GitHub link**.

Unzip it and put its contents in a directory you'll call gad.
The contents of this zip are:

- opencv_face_detector.pbtxt
- opencv_face_detector_uint8.pb
- age_deploy.prototxt
- age_net.caffemodel
- gender_deploy.prototxt
- gender_net.caffemodel
- a few pictures to try the project on

For face detection, we have a .pb file- this is a protobuf file (protocol buffer); it holds the graph definition and the trained weights of the model. We can use this to run the trained model. And while a .pb file holds the protobuf in binary format, one with the .pbtxt extension holds it in text format. These are TensorFlow files. For age and gender, the .prototxt files describe the network configuration and the .caffemodel file defines the internal states of the parameters of the layers.

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.

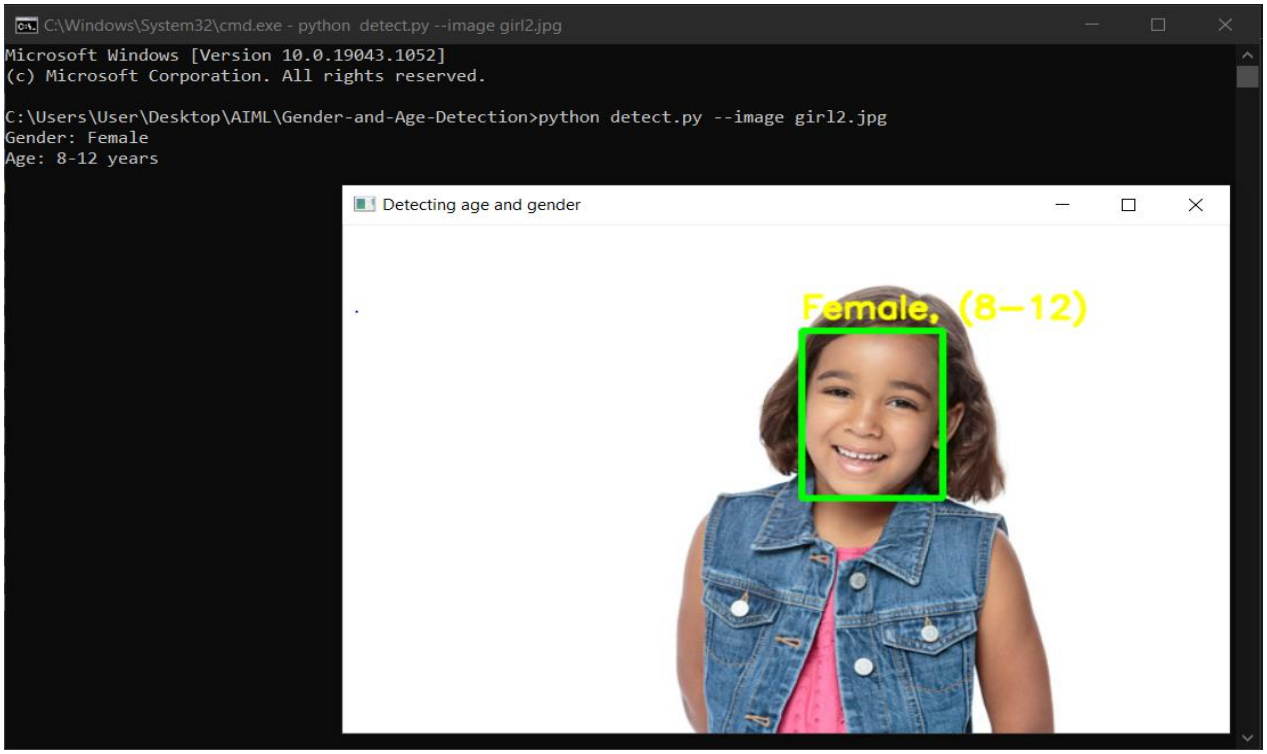
9. But if there are indeed faceBoxes, for each of those, we define the face, create a 4-dimensional blob from the image. In doing this, we scale it, resize it, and pass in the mean values.

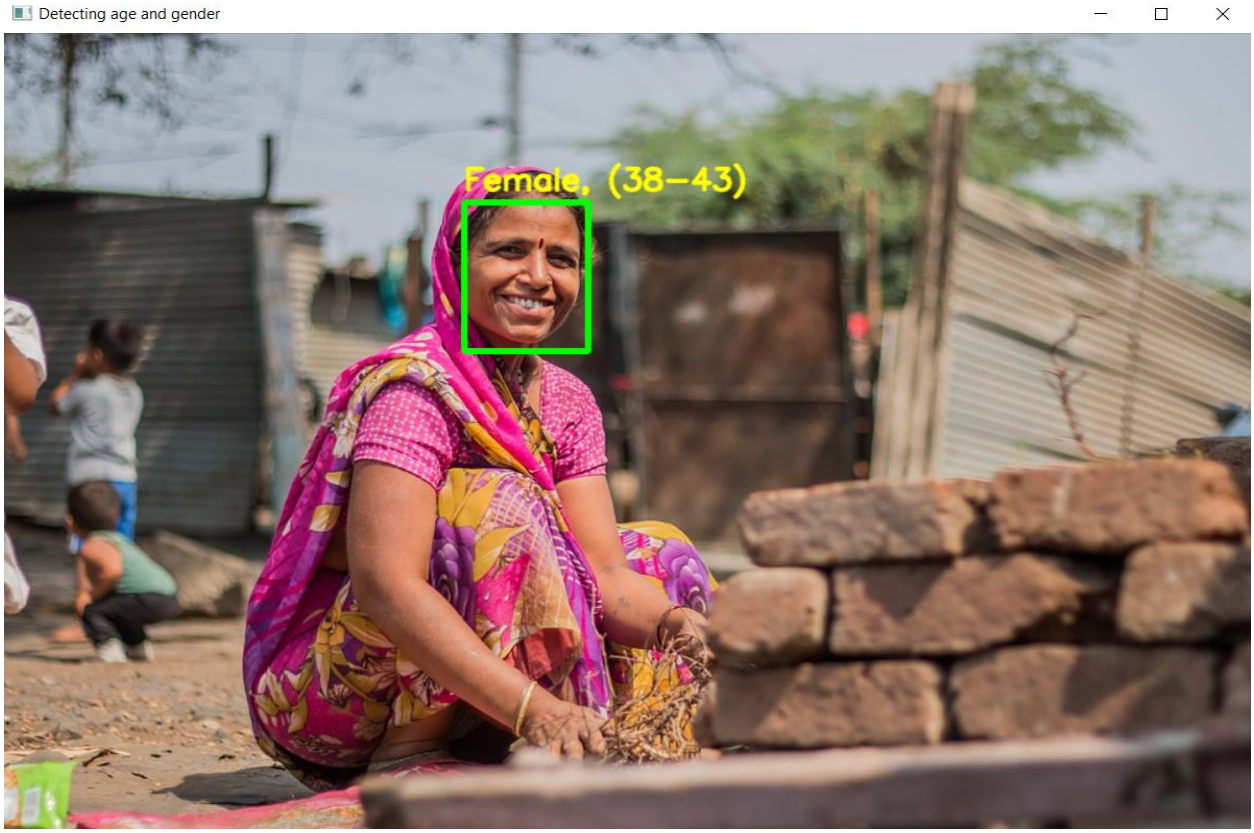
10. We feed the input and give the network a forward pass to get the confidence of the two class. Whichever is higher, that is the gender of the person in the picture.

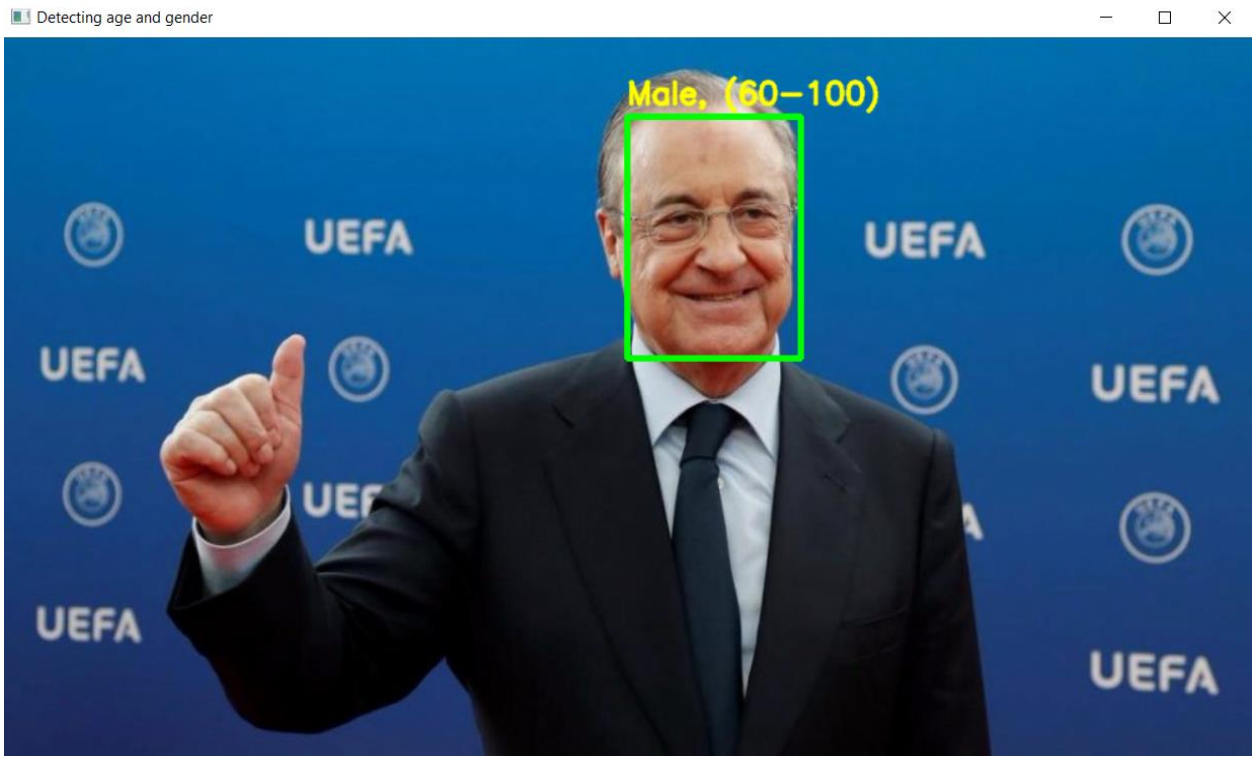
11. Then, we do the same thing for age.

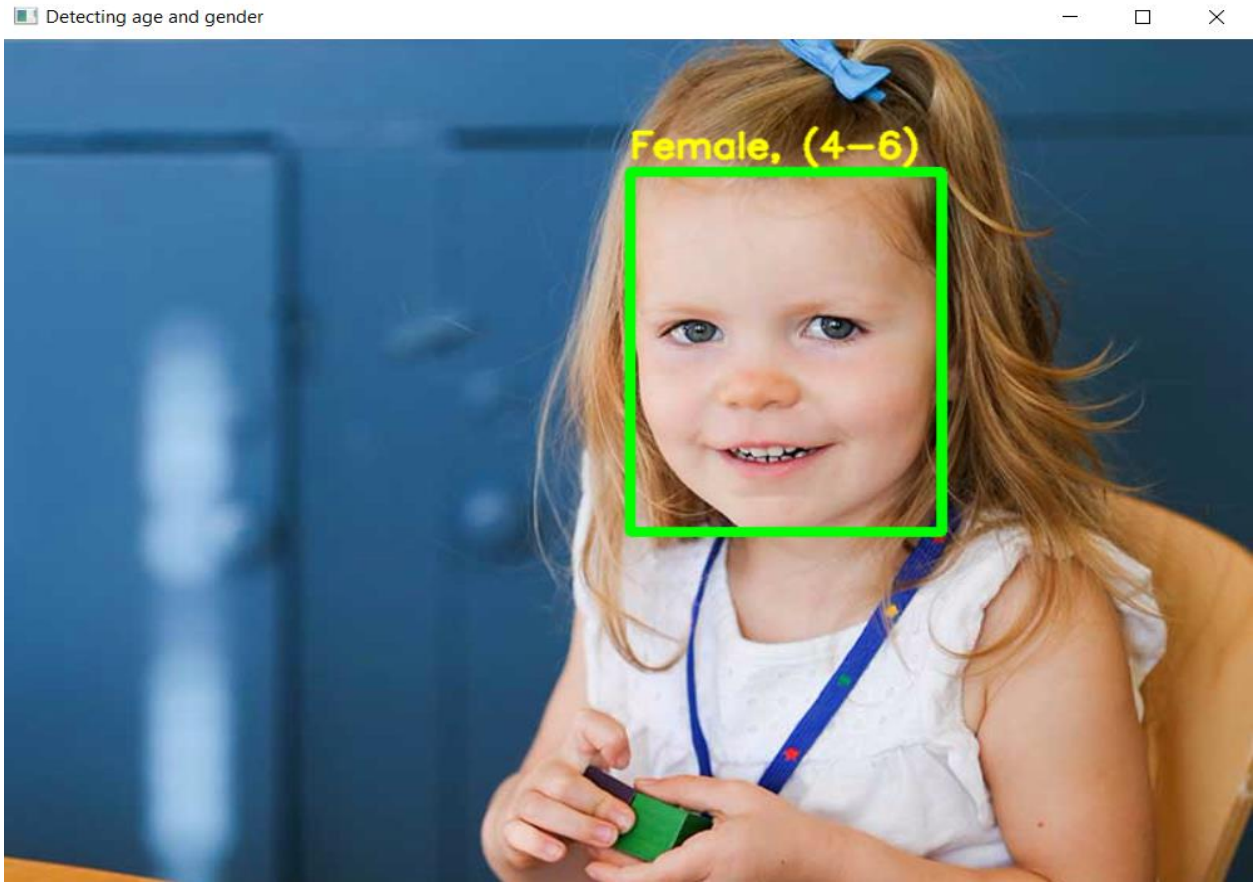
12. We'll add the gender and age texts to the resulting image and display it with imshow().

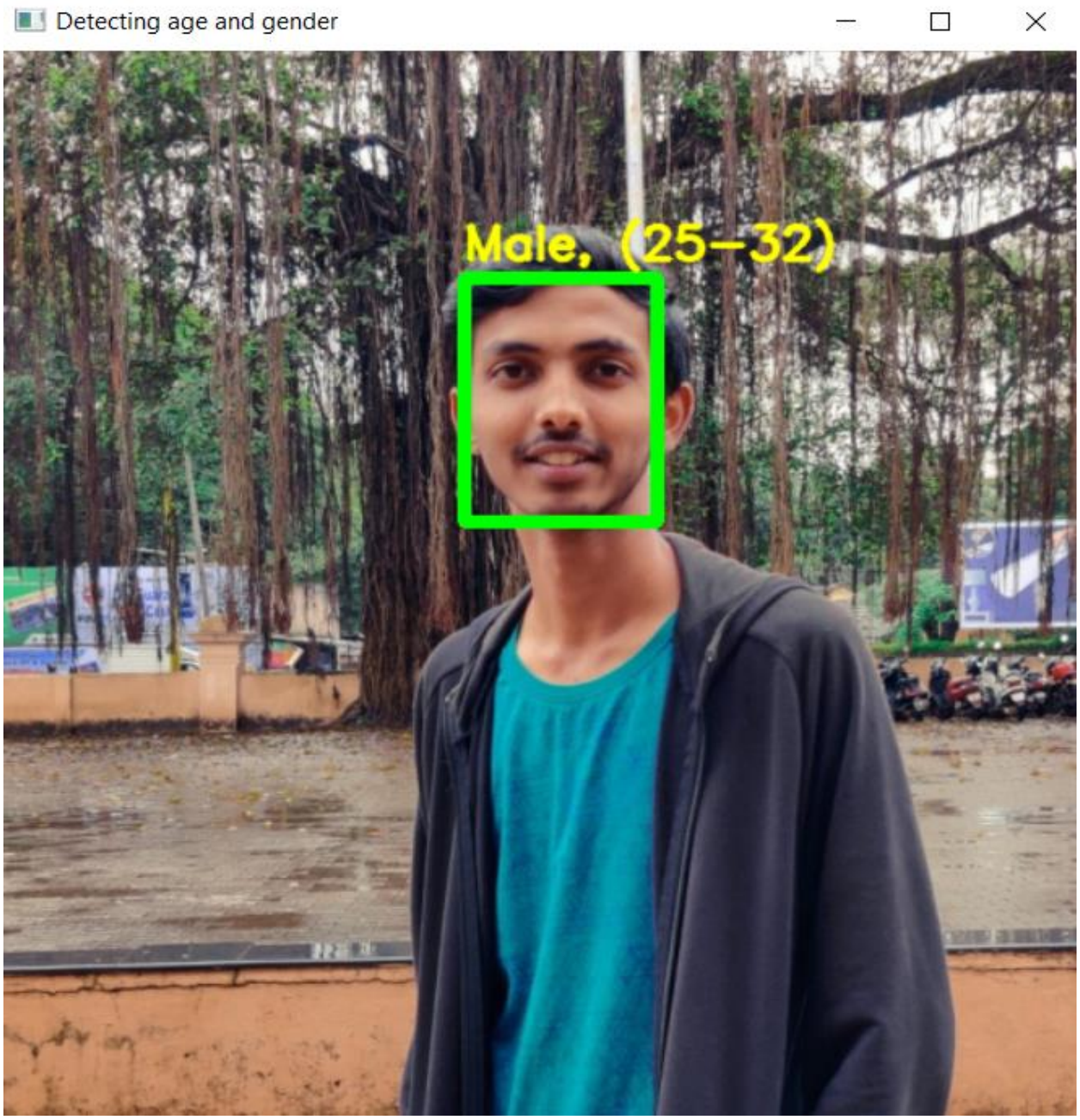
Sample output images:






Detecting age and gender
Male, (60–100)
UEFA
UEFA
UEFA
UEFA
UEFA
UEFA


Detecting age and gender
Female, (4–6)



Applications

- Applications for this technology have a broad scope and the potential to make a large impact. This could be used to aid assisted vision devices for those with deteriorating, or lost, eyesight.

- Social media websites like Facebook could use the information about the age and gender of the people to better infer the context of the image.
- In human machine interaction.

Conclusion

- CNN can be used to provide improved age and gender classification results, even considering the much smaller size of contemporary unconstrained image sets labeled for age and gender.
- The simplicity of the model implies that more elaborate systems using more training data may well be capable of substantially improving results beyond these results.
- One can also try to use a regression model instead of classification for Age Prediction if enough data is available.

Project Demonstration Link:

- https://drive.google.com/file/d/1Ufc9-Bfm1Ji119lwdKv-EK_XxZfXMGf9/view?usp=sharing

Project GitHub Link:

- https://github.com/paone04/project

References

- https://gilscvblog.com/2015/11/19/age-and-gender-classification-using-deep-convolutional-neural-networks/
- https://www.kdnuggets.com/2019/04/predict-age-gender-using-convolutional-neural-network-opencv.html