SAVITRIBAI PHULE PUNEUNIVERSITY

A MINI PROJECT REPORT ON

AGE AND GENDER CLASSIFICATION USING DEEP LEARNING

Submitted by

Name: Parth Mangalkar	**Roll no: A-71**
Name: Aryaman Nasare	**Roll no: B-04**

CLASS: BE **DIV: A**

Under the Guidance of

Ms. Jyoti Raghatwan

DEPARTMENT OF COMPUTER ENGINEERING

RMD SINHGAD SCHOOL OF ENGINEERING

WARJE, PUNE 411058

2022 - 2023

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

RMD SINHGAD SCHOOL OF ENGINEERING

WARJE, PUNE 411058

CERTIFICATE

This is to certify that the project report entitles

“Age and Gender Classification Using Deep Learning”

Submitted by

Name: Parth Mangalkar	PRN No :72035923G

is a bonafide work carried out by them under the supervision of Ms. Jyoti Raghatwan. And it is submittedtowards the partial fulfillment of the requirement of University of Pune for Fourth Year.

Ms. Jyoti Raghatwan
Guide
Department of Computer Engineering

Prof. Vina M. Lomte
Head,
Department of Computer Engineering

Dr. V. V. Dixit
Principal,
RMD Sinhgad School of Engineering

Certificate by Guide

This is to certify that Mr. Parth Mangalkar has completed the MINI Project work under my guidance andsupervision and that, I have verified the work for its originality in documentation, problem statement, implementation and results presented in the Project. Any reproduction of other necessary work is with theprior permission and has given due ownership and included in the references.

Signature of Guide

Ms. Jyoti Raghatwan

ACKNOWLEDGEMENT

It is our pleasure to acknowledge sense of gratitude to all those who helped us in making thisseminar.

We thank our Mini Project Guide **Ms. Jyoti Raghatwan** for helping us and providing all necessary information regarding our project.

We are also thankful to **Prof. Vina M. Lomte (Head - Department of Computer Engineering)** for providing us the required facilities and helping us while carrying out this seminar work.

Finally, we wish to thank all our teachers and friends for their constructive comments, suggestionsand criticism and all those directly or indirectly helped us in completing this seminar.

NAME OF THE STUDENT: PARTH MANGALKAR

ABSTRACT

This machine learning mini-project involves building a gender and age detector using deep learning on the Adience dataset. The objective is to accurately predict the gender and age range of a person from a single image of their face. The project uses a convolutional neural network (CNN) architecture with 3 convolutional layers, 2 fully connected layers, and a final output layer of SoftMax type.

The Adience dataset contains 26,580 photos of 2,284 subjects in eight age ranges, and the models have been trained on this dataset. The project involves detecting faces, classifying into male/female and one of the eight age ranges, and putting the results on the image. The implementation involves using OpenCV, TensorFlow, Caffe, and PyTorch libraries.

INTROUDCTION:

This machine learning mini-project aims to build a gender and age detector that can identify the gender and age of a person in a picture using deep learning on the Adience dataset. The project uses models trained by Tal Hassner and Gil Levi and predicts gender as 'Male' or 'Female' and age in one of eight ranges from 0-2 to 60-100 years. The project uses computer vision techniques and the OpenCV library, along with a convolutional neural network architecture with three convolutional layers and two fully connected layers, and a SoftMax output layer. The Adience dataset, which contains over 26,000 photos of 2,284 subjects in eight age ranges, is used to train the models. The project involves detecting faces, classifying into gender and age, and putting the results on the image for display. The steps for executing the project include downloading the Adience dataset, creating files for age and gender classification, initializing protocol buffers and models, and capturing video stream to classify on a webcam's stream.

1.1 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.

1.2 Problem Statement

• To detect if a person is a male or female and also their age.

SOFTWARE:

2.1 Hardware Specifications:

- Memory: 8 GB
- Processor: AMD Ryzen 5 3500x 6-core
- Graphics: Nvidia GeForce GT 710
- OS type: 64-bit
- Disk: 40 GB

2.2 Software Specifications:

- Operating system: Windows 10
- Python 3.7
- OpenCV2

THEORY:

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.

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 to generate symbolic or numerical information which can then be used to make decisions.

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.

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.

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.

DATASET:

For this python project, we'll use the Adience dataset. 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.

ALGORITHM:

1. Download Dataset

- Adience dataset

2. Create files

A directory you'll call gad. The contents of this directory 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.

3. 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.

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

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

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

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

8. 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.

9. 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

10. 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.

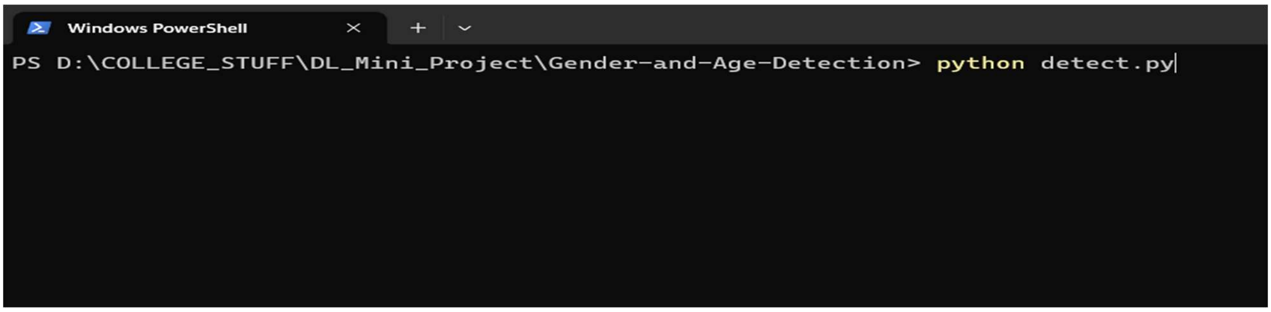
11. We feed the input and give the network a forward pass to get the confidence of the two class. Whichever is higher, that is the gender of the person in the picture.

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

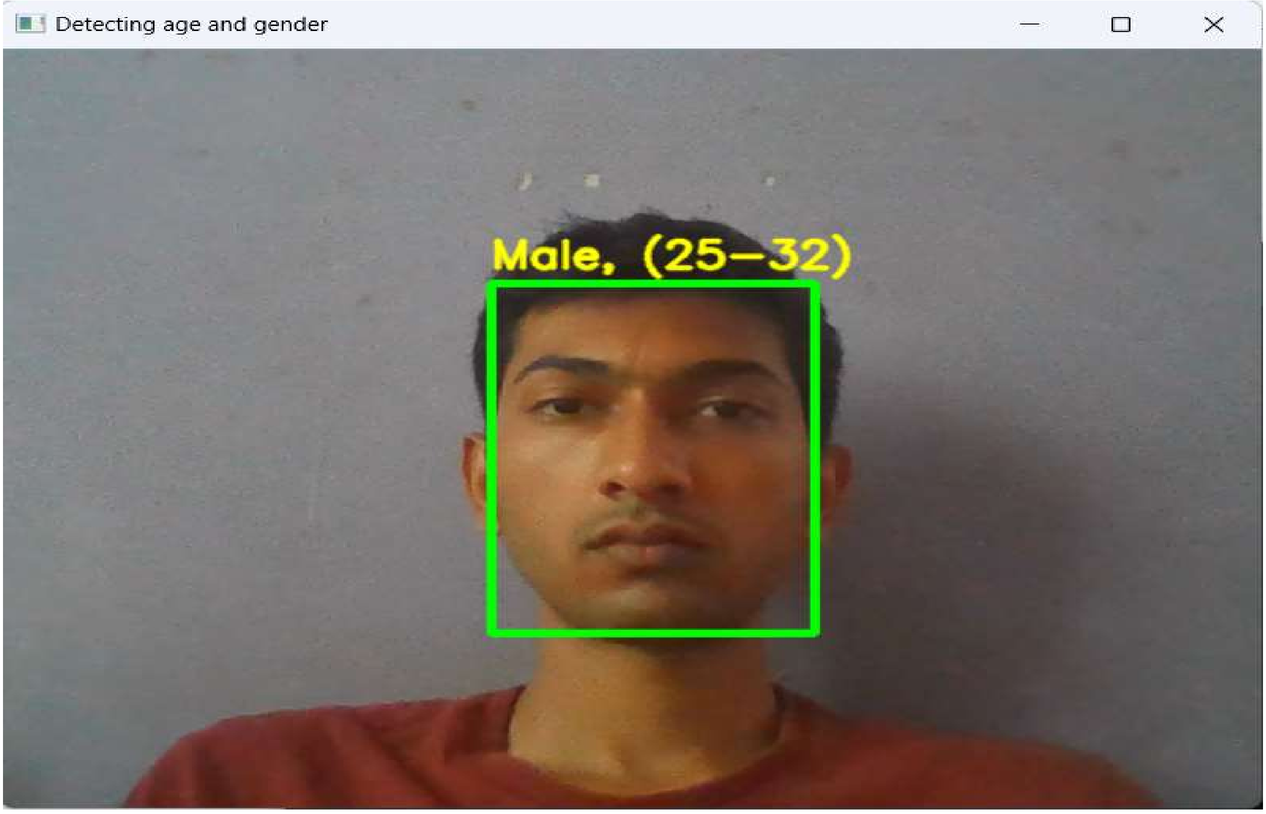
13. We'll add the gender and age texts to the resulting image and display it with

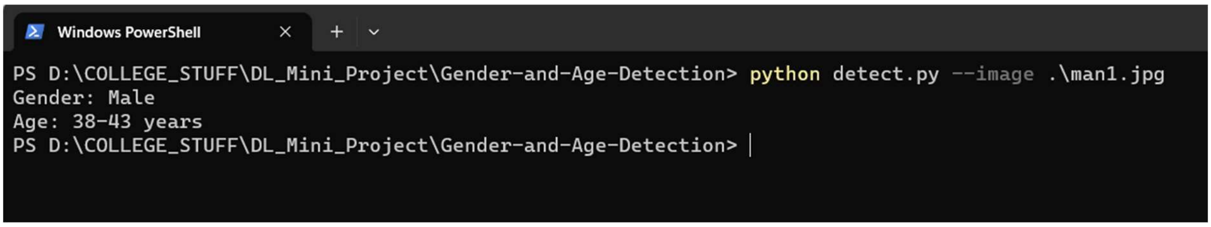
imshow().

OUTPUT:

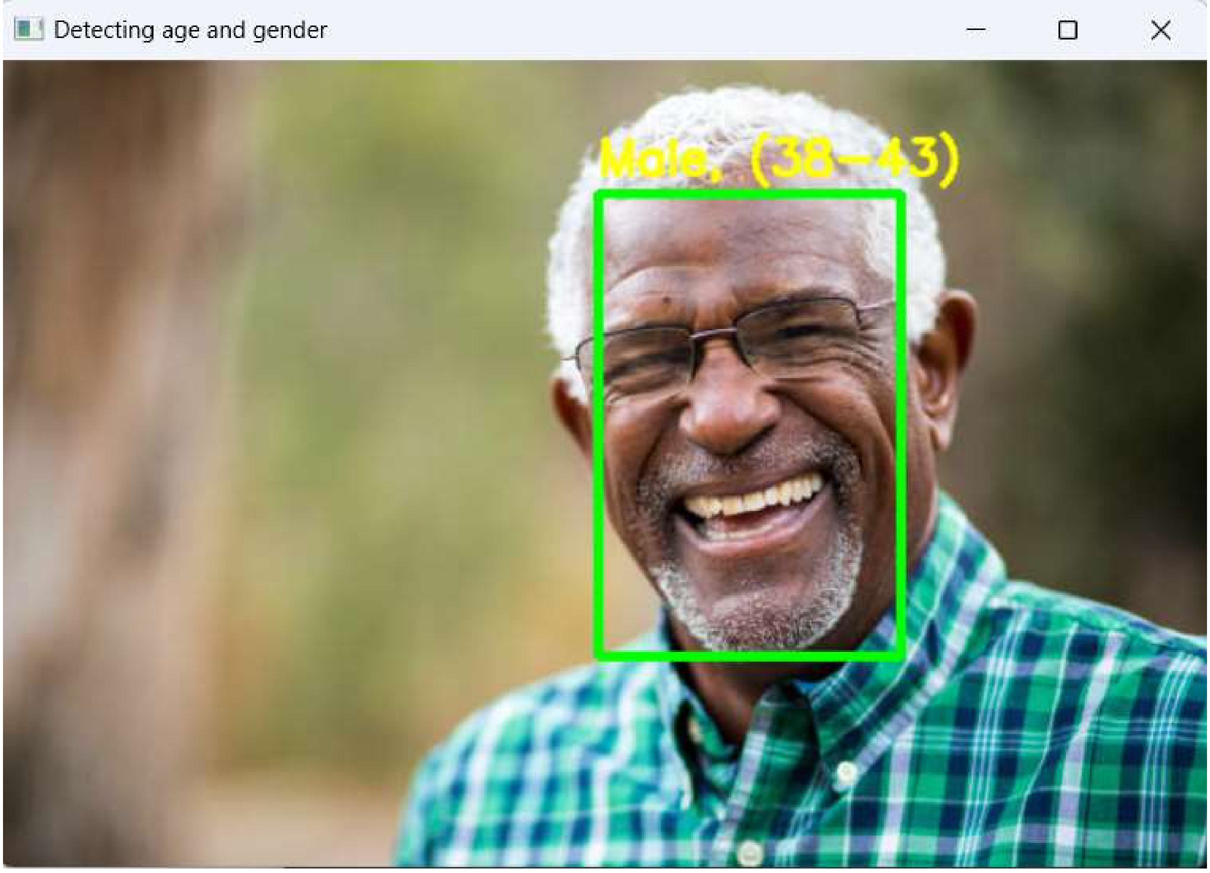


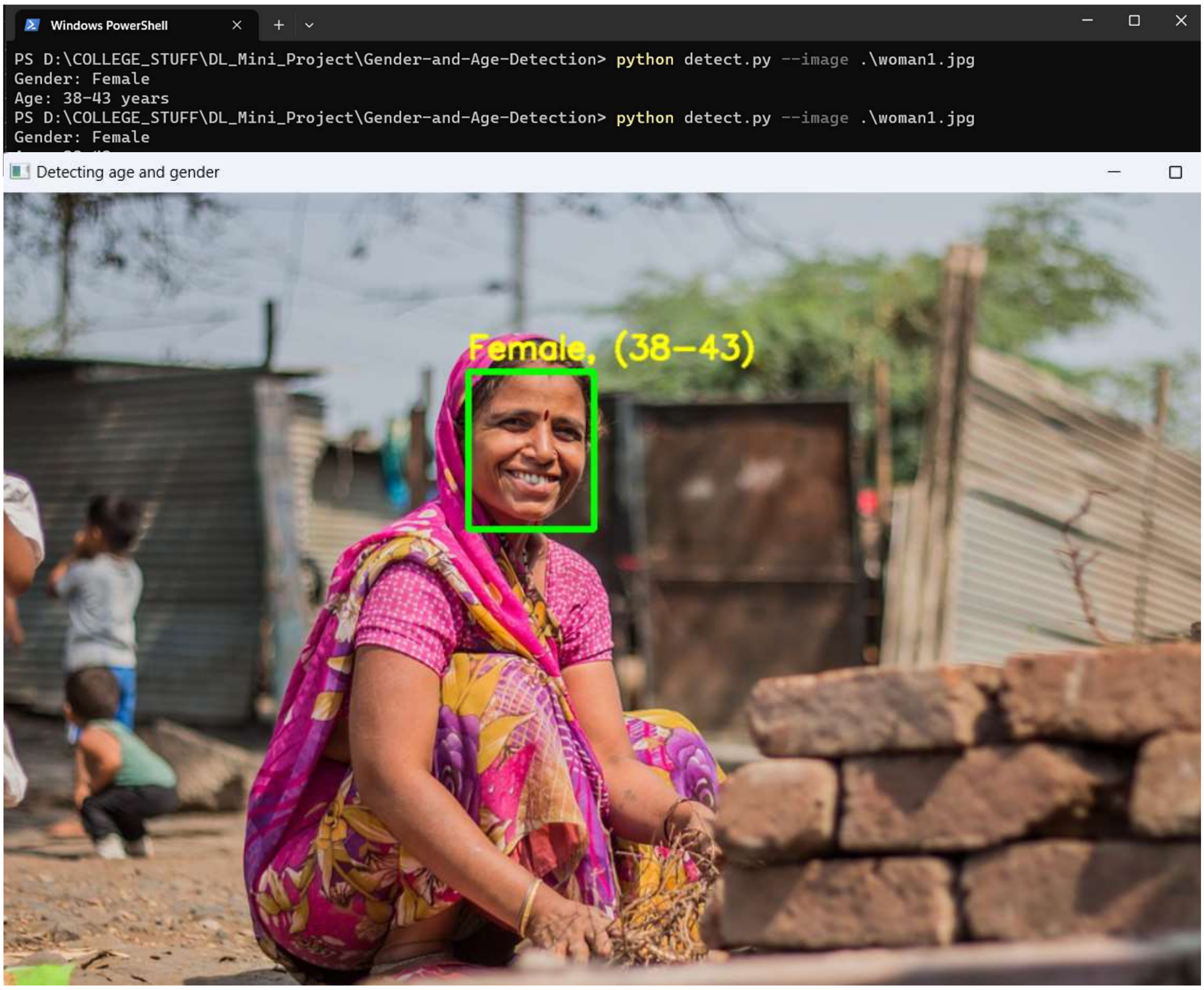
```
PS D:\COLLEGE_STUFF\DL_Mini_Project\Gender-and-Age-Detection> python detect.py
```





```
PS D:\COLLEGE_STUFF\DL_Mini_Project\Gender-and-Age-Detection> python detect.py --image .\man1.jpg
Gender: Male
Age: 38-43 years
PS D:\COLLEGE_STUFF\DL_Mini_Project\Gender-and-Age-Detection> |
```





```
PS D:\COLLEGE_STUFF\DL_Mini_Project\Gender-and-Age-Detection> python detect.py --image .\woman1.jpg
Gender: Female
Age: 38-43 years
PS D:\COLLEGE_STUFF\DL_Mini_Project\Gender-and-Age-Detection> python detect.py --image .\woman1.jpg
Gender: Female
```

CONCLUSION:

Thus, we have successfully studied and implemented machine learning model for age and gender classification using OpenCV.