Project Report

Title: Age and Gender Detection using Deep Learning

1 INTRODUCTION

1.1 Overview

Deep Learning has found huge applications in the fields of Computer vision. Some of the most important applications of computer vision are in the fields that deal with facial data. Face Detection and recognition are being widely used in security-based applications.

In this guided project, we are going to talk about two of the most important applications of face-based deep learning, age and gender prediction from a facial image.

To build this project we will be using OpenCV and Caffe model where the weights will be taken from pre-trained models each for facial detection, age and gender. The prediction will result on a real-time basis as well as on a manual basis.

1.2 Purpose

By the end of this project we will:

- Know fundamental concepts and techniques of Neural Network.
- Will gain a broad understanding of image data.
- Know how to draw bounding boxes according to your requirement
- Know how to build a web application using the Flask framework.

2 LITERATURE SURVEY

2.1 Existing problem

Several methods for solving this age and gender detection by AI are prevailing already just like by using CNN and RNN.

2.2 Proposed solution

- User interacts with the UI (User Interface) and clicks on image or video cam option
- The image in the code or the webcam input will pass through the program

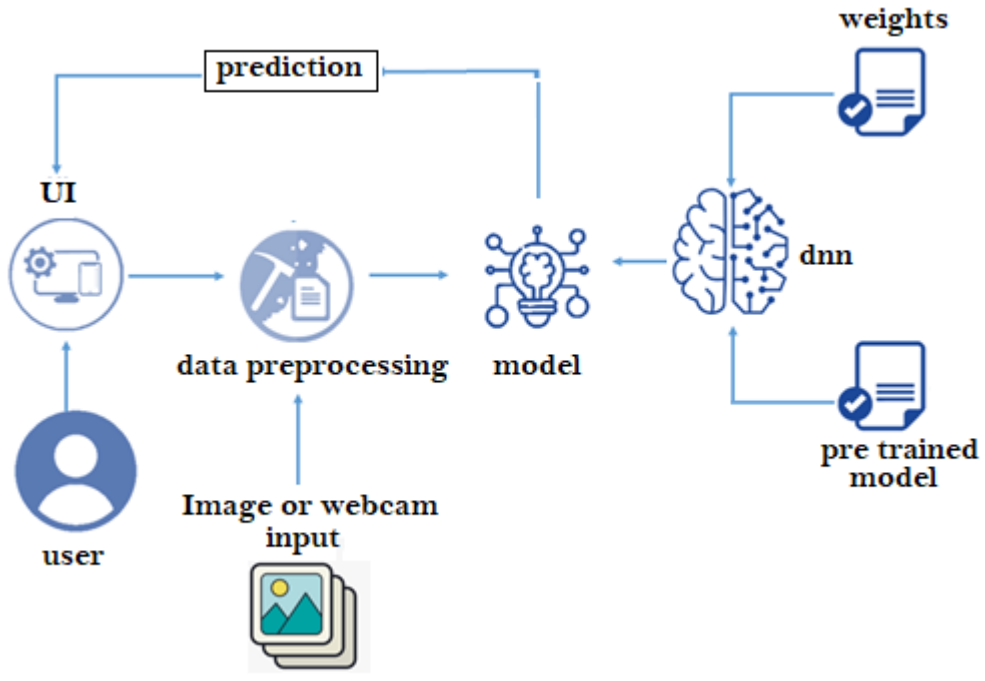
- Once any person's face is detected the model will give the gender along with approx. age of that person.

To accomplish this, we have to complete all the activities and tasks listed below

- Importing required libraries
- Importing necessary files
    - Importing weight and training data
    - Declaring a default list of values
- Load the network
    - Declaring the pre-trained dnn
- Capturing the data
    - Getting the bounding box
    - Applying Blob operation
    - Getting the required values
    - Printing the results
- Application Building
    - Create an HTML file
    - Build Python Code

3 THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

No hardware required but do require software.

In order to develop this project, we need to install the following software/packages

Anaconda Navigator :

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook,
QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

Deep Learning Concepts :

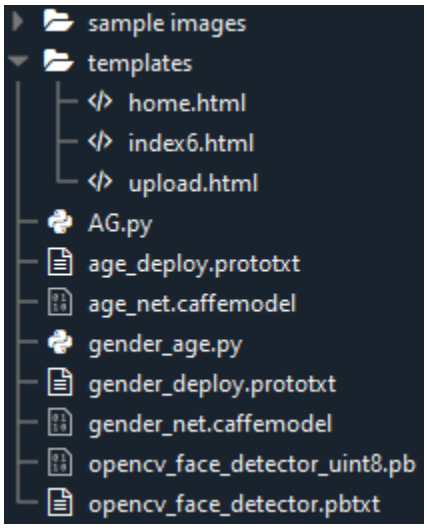
Opencv – It is an Open Source Computer Vision Library which is mainly used for image processing, video capture and analysis including features like face detection and object detection.

Caffemodel- Caffe is a deep learning framework. To get a Caffe model you need to train the network. That prototxt file is only to deploy the model and cannot be used to train it.

4 EXPERIMENTAL INVESTIGATIONS

We were able to detect the approximate Age and Gender of the image provided which were either real-time or uploaded files.

5 FLOWCHART

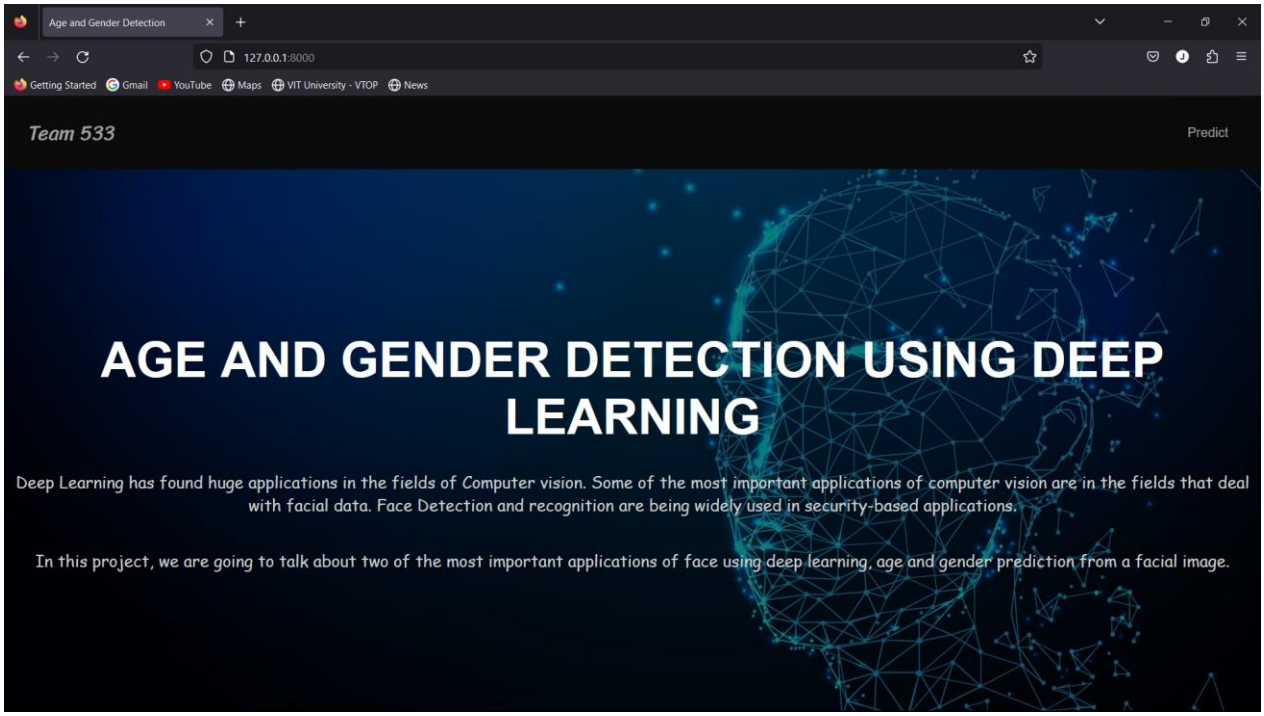


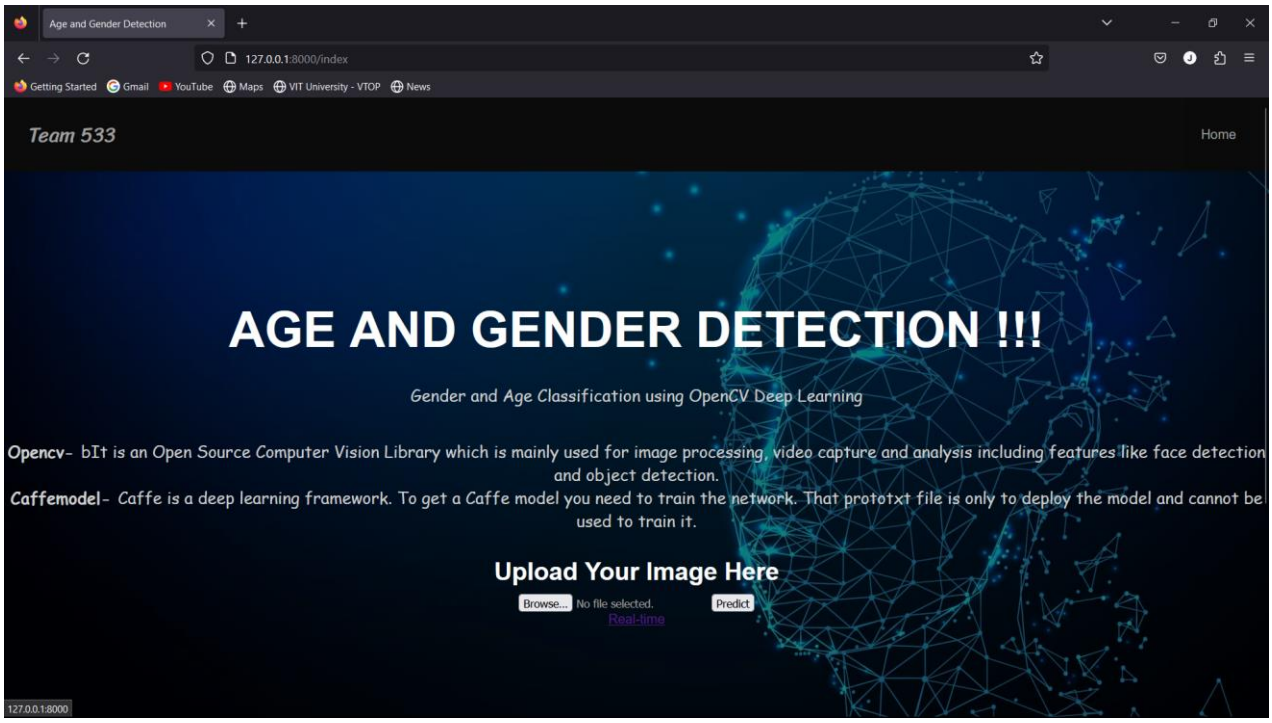
- We are building a Flask Application that needs HTML pages stored in the templates folder and a python script AG.py which we run from prompt whereas gender_age.py should run from the python IDE.
- .prototext and .caffemodel denote the pre-trained model weights.
- The sample images folder contains the sample images which we give manually in the code by giving the path of the image.

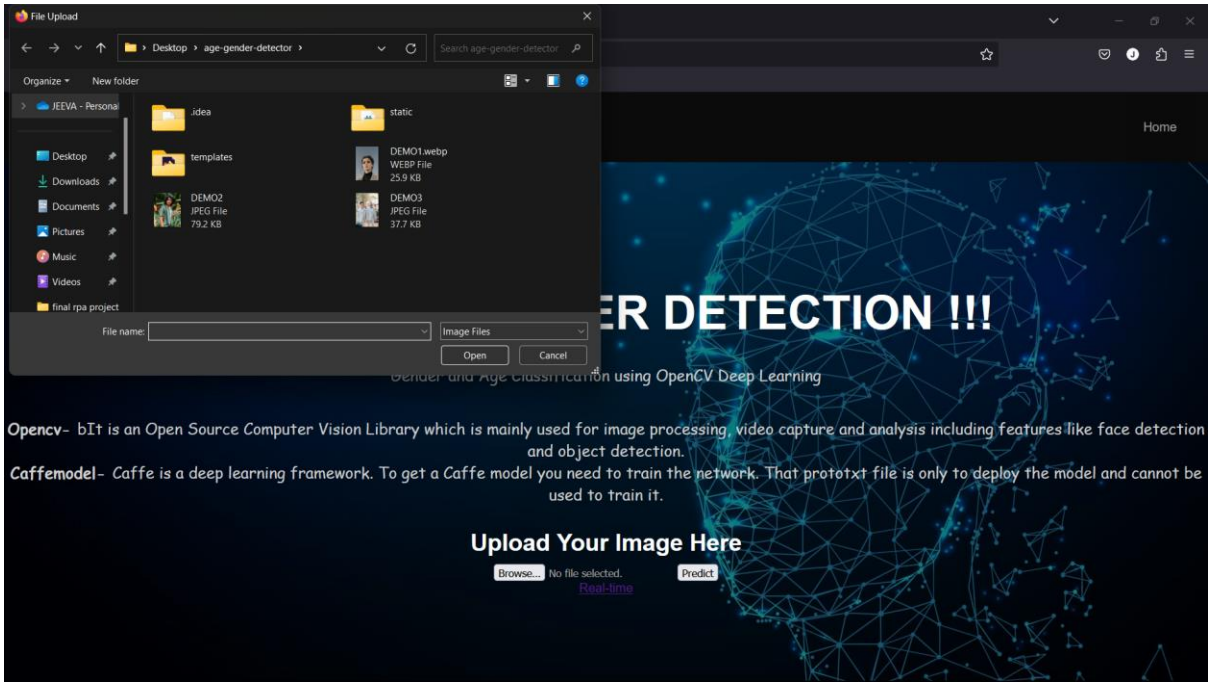
6 RESULT

Some of the screenshots that define the result of our project:

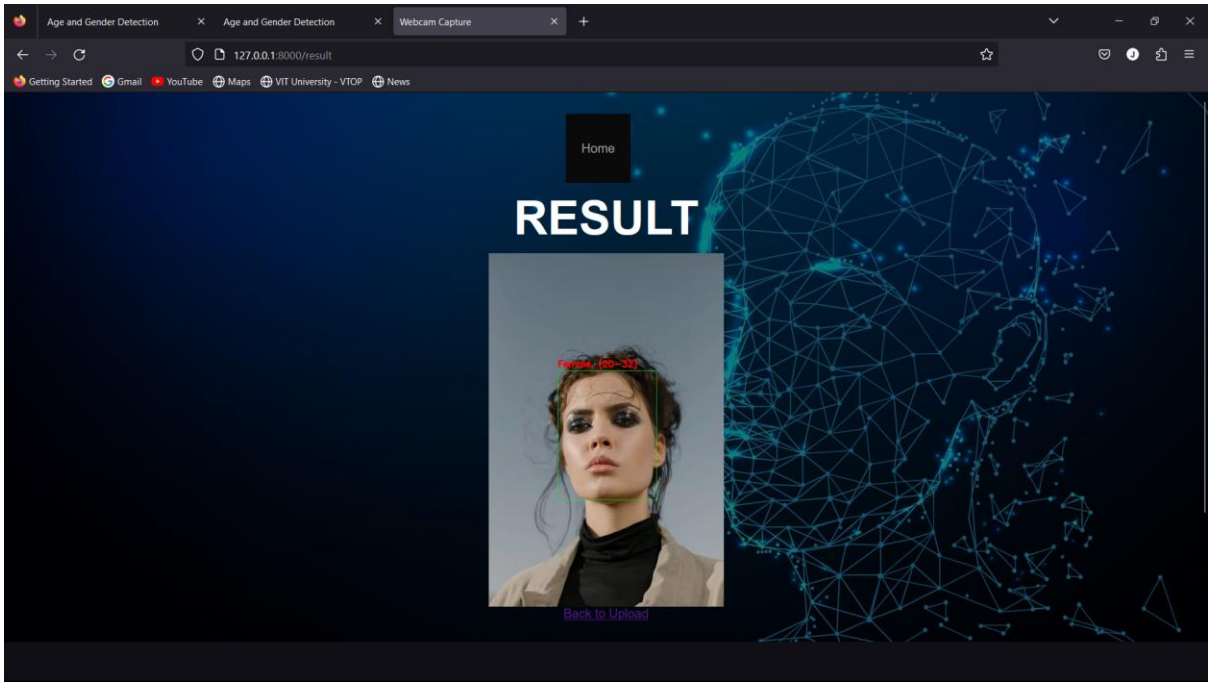
Website:

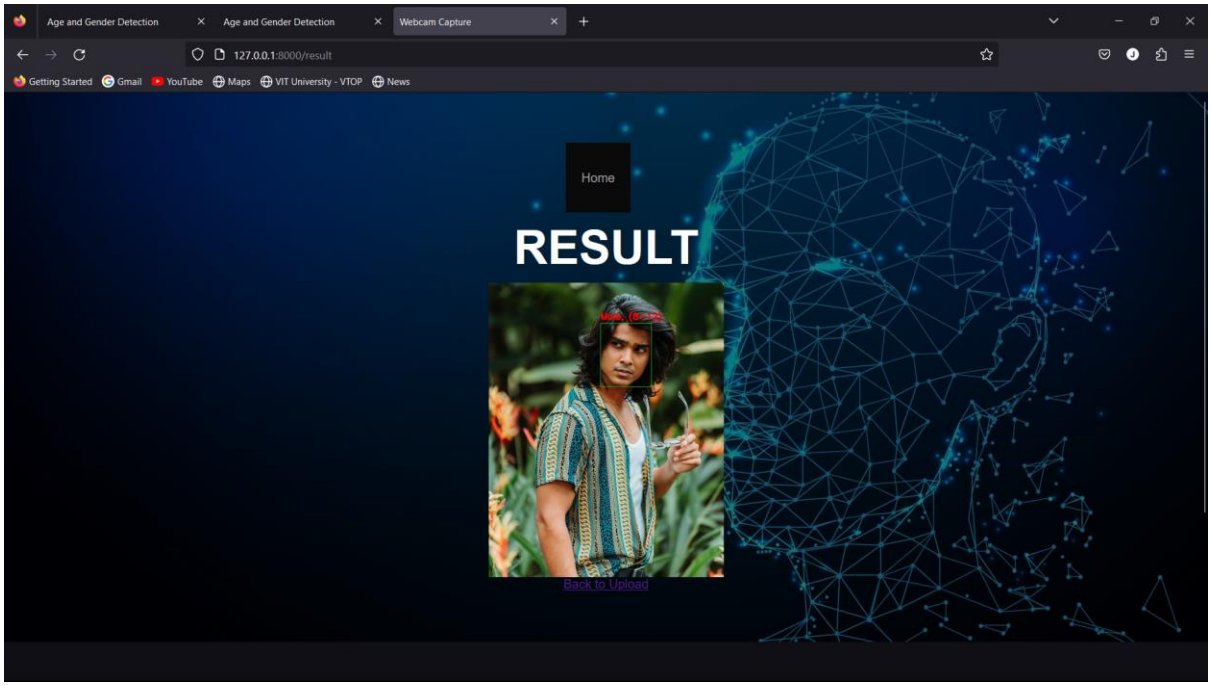


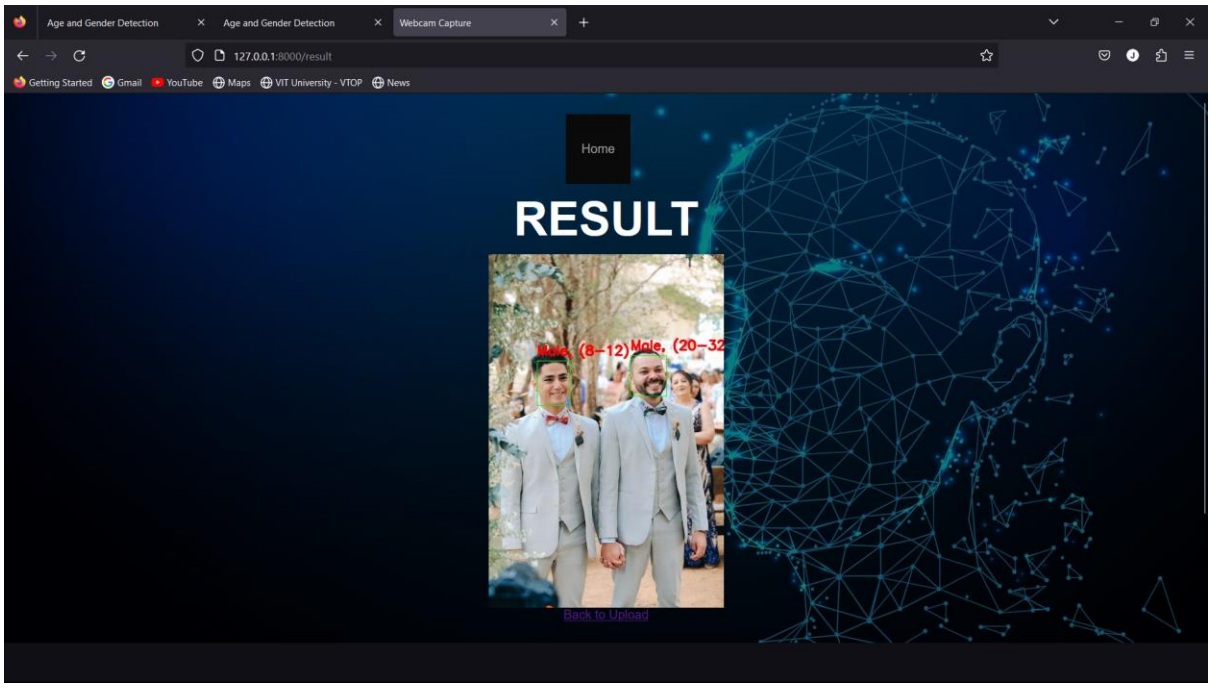




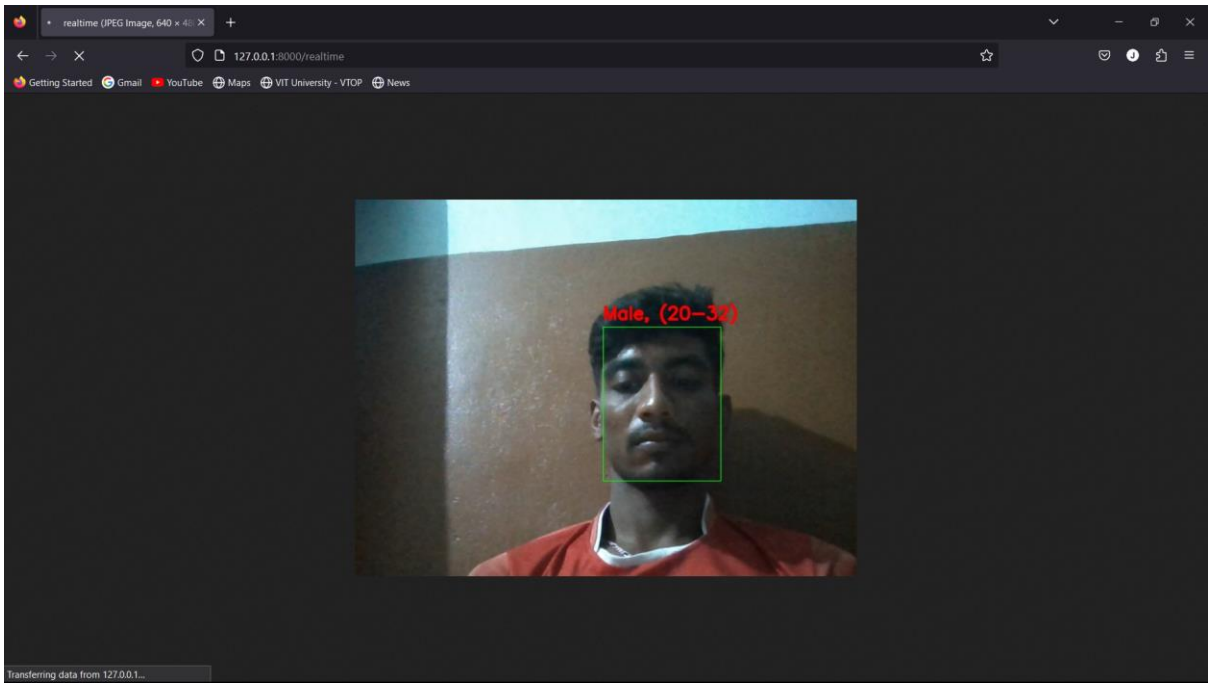
Output (uploaded images):




Age and Gender Detection
Age and Gender Detection
Webcam Capture
127.0.0.1:8000/result
Getting Started
Gmail
YouTube
Maps
VIT University - VTOP
News
Home
RESULT
Back to Upload


Age and Gender Detection
Age and Gender Detection
Webcam Capture
127.0.0.1:8000/result
Getting Started
Gmail
YouTube
Maps
VIT University - VTOP
News
Home
RESULT
Back to Upload

Real-time capturing:



Output (when face cannot be detected or no image is selected):

---

7 ADVANTAGES & DISADVANTAGES

Advantage:

Deep learning has a wide range of applications, including image and speech recognition, natural language processing, and computer vision. One of the main advantages of deep learning is that it can automatically learn features from the data, which means that it doesn't require the features to be hand-engineered.

Age and gender detection is essential for authentication, human-computer interaction, behaviour analysis, product recommendation based on user preferences, and many other areas.

Disadvantage:

High computational cost: Training deep learning models requires significant computational resources, including powerful GPUs and large amounts of memory.

Overfitting: Over fitting occurs when a model is trained too well on the training data and performs poorly on new, unseen data.

8 APPLICATIONS

Age and gender detection is essential for authentication, human-computer interaction, behavior analysis, product recommendation based on user preferences, and many other areas. Many companies needed age and gender data capture, but few solutions were available. Significant developments have been made in the past few years to meet this need. In the last decade, artificial intelligence classification systems have been used instead of manual

classification systems for age and gender detection. With the introduction of artificial intelligence, the success rate of solving the problem has increased.

Age detection: Age detection can be used to place ads in the types of media most consumed by your target audience.

Gender detection: Gender detection can be used to determine whether a social media platform is more likely to show your product to men or women.

9 CONCLUSION

Age detection and gender detection ai have many different uses. For example, in-store and front-of-shop age detection and gender detection are very effective even in retail businesses. By using these technologies, businesses can identify customer segments, change or develop marketing strategies according to this segment, and change their product range.

For example, some software can detect the age range of users passing by a business and suggest advertisements that will affect this age range. It is possible to use age detection and gender detection not only for sales purposes but also for security purposes because algorithms can make more accurate age and gender determinations than humans.

The only negative aspect of these algorithms is that they have extremely complex work principles, as mentioned above, and they also require large data sets. This means a loss of both time and money. If you cannot find specialists in the field, it is challenging to learn all these technologies. In addition, software developed in this regard can be highly costly.

10 FUTURE SCOPE

Since, the only negative aspect of these algorithms is that they have extremely complex work principles, as mentioned above, and they also require large data sets. This means a loss of both time and money. If you cannot find specialists in the field, it is challenging to learn all these technologies. In addition, software developed in this regard can be highly costly.

Hence, the solution for this disadvantage can be considered as an advantage in future. If this is done, it can be expanded in vast culture and can help people in many ways.

11 BIBILOGRAPHY

References :

- SmartInternz Guided Project
- https://github.com/smahesh29/Gender-and-Age-Detection
- https://www.hindawi.com/journals/cmmm/2022/1413597/

APPENDIX

A. Source Code

```
import cv2
import os
import numpy as np
from flask import Flask, request, render_template, Response, redirect, url_for
from werkzeug.utils import secure_filename

app = Flask(__name__, template_folder="templates")
app.config['UPLOAD_FOLDER'] = 'static/uploads'

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

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

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)', '(20-32)', '(32-43)', '(48-
53)', '(60-100)']
genderList = ['Male', 'Female']

faceNet = cv2.dnn.readNet(faceModel, faceProto)
ageNet = cv2.dnn.readNet(ageModel, ageProto)
genderNet = cv2.dnn.readNet(genderModel, genderProto)

def getFaceBox(faceNet, frame):
    frameHeight = frame.shape[0]
    frameWidth = frame.shape[1]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (227, 227), [104, 117, 123],
swapRB=False)
    faceNet.setInput(blob)
    detections = faceNet.forward()
    faceBoxes = []
    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > 0.7:
            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(frame, (x1, y1), (x2, y2), (0, 255, 0), 1)
    return frame, faceBoxes

@app.route('/', methods=['GET'])
def home():
```

```
    return render_template('home.html')

@app.route('/index', methods=['GET'])
def index():
    return render_template('index.html')

@app.route('/result', methods=['GET', 'POST'])
def result():
    if request.method == 'POST':
        if 'image' in request.files:
            # Uploading image
            file = request.files['image']
            if file.filename == '':
                return 'No selected image file'

            if file:
                filename = secure_filename(file.filename)
                file_path = os.path.join(app.config['UPLOAD_FOLDER'],
filename)
                file.save(file_path)

                frame = cv2.imread(file_path)
                frame, faceBoxes = getFaceBox(faceNet, frame)

                if not faceBoxes:
                    return "No face detected in the image"

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

                    ageNet.setInput(blob)
                    agePreds = ageNet.forward()
                    age = ageList[agePreds[0].argmax()]

                    label = "{}, {}".format(gender, age)
                    cv2.putText(frame, label, (faceBox[0], faceBox[1] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.75,
                                (0, 0, 255), 2, cv2.LINE_AA)
```

```
                    cv2.rectangle(frame, (faceBox[0], faceBox[1]),
(faceBox[2], faceBox[3]), (0, 255, 0), 1)

                result_path = os.path.join(app.config['UPLOAD_FOLDER'],
'result.jpg')
                cv2.imwrite(result_path, frame)

                return render_template("result.html", image_path=result_path)

    return redirect(url_for('home'))

def generate_frames():
    cap = cv2.VideoCapture(0)
    while True:
        success, frame = cap.read()
        if not success:
            break
        else:
            frame, faceBoxes = getFaceBox(faceNet, frame)

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

                ageNet.setInput(blob)
                agePreds = ageNet.forward()
                age = ageList[agePreds[0].argmax()]

                label = "{}, {}".format(gender, age)
                cv2.putText(frame, label, (faceBox[0], faceBox[1] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.75,
                            (0, 0, 255), 2, cv2.LINE_AA)
                cv2.rectangle(frame, (faceBox[0], faceBox[1]), (faceBox[2],
faceBox[3]), (0, 255, 0), 1)

            ret, buffer = cv2.imencode('.jpg', frame)
            frame = buffer.tobytes()
            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
```

```
@app.route('/realtime')
def realtime():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace;
boundary=frame')

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=True)
```

END OF PROJECT REPORT