Age And Gender Detection Using Deep Learning With IBM Cloud

Introduction :-

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.

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.

a. Overview:-

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.

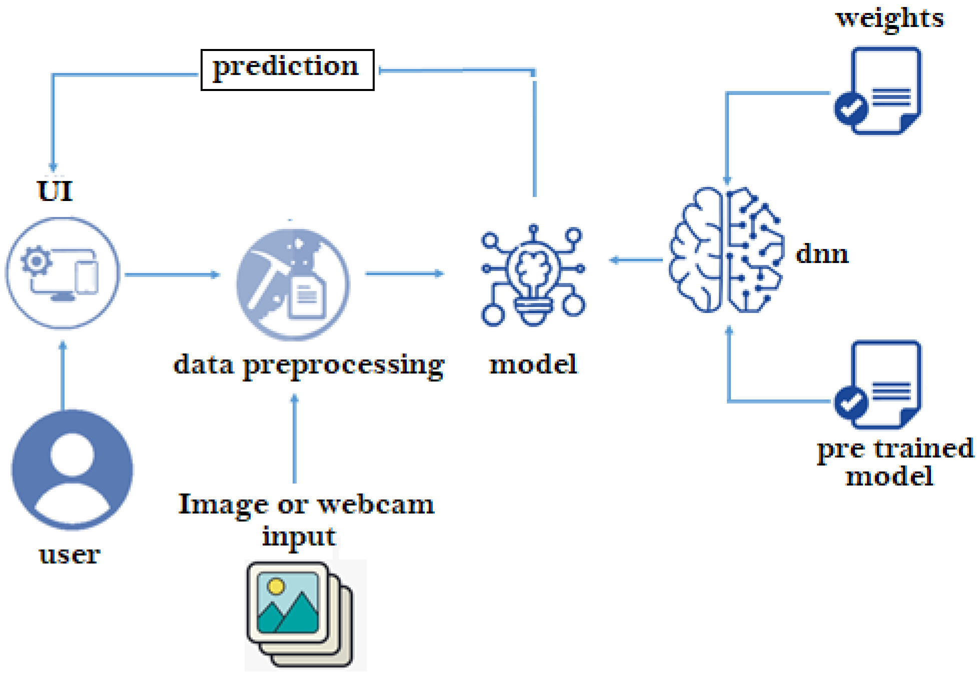
b. Purpose:-

The project aims to build a web application that can detect the age and gender of a person using Deep Learning which can facilitate a sales person to understand their customers better.

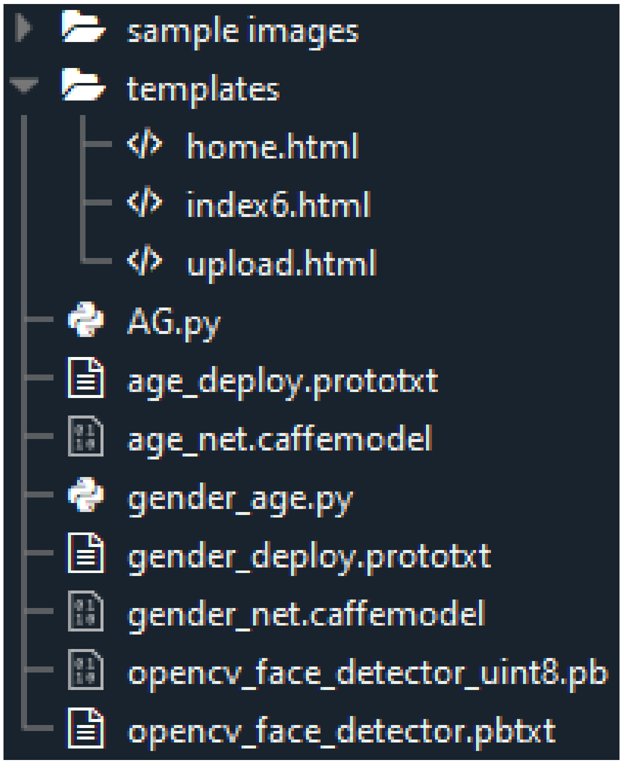
Age And Gender Detection Using Deep Learning With IBM Cloud

Theoretical Analysis:

a. Block diagram :-



b. Hardware / Software designing:-



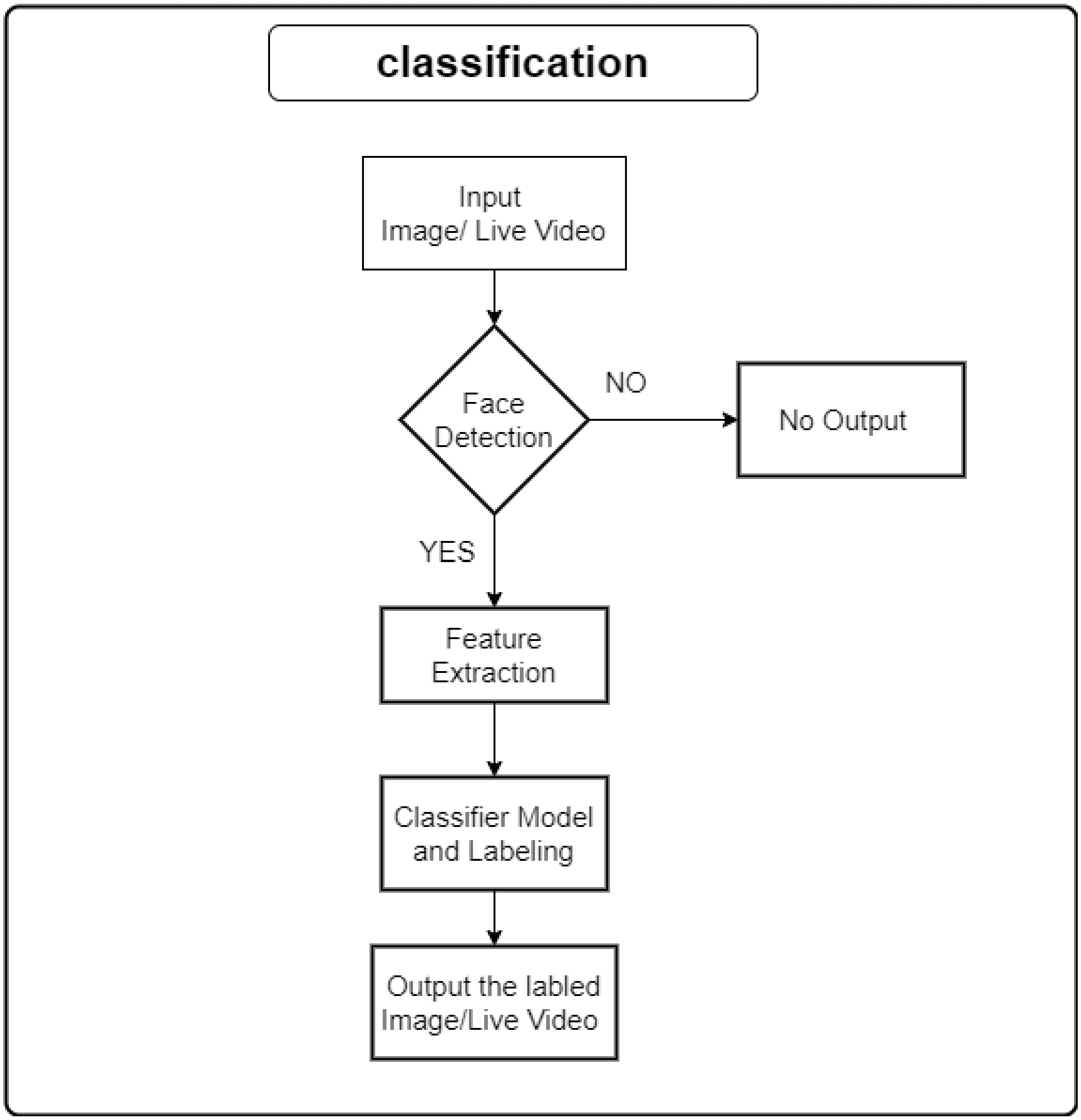
```
sample images
templates
    home.html
    index6.html
    upload.html
AG.py
age_deploy.prototxt
age_net.caffemodel
gender_age.py
gender_deploy.prototxt
gender_net.caffemodel
opencv_face_detector_uint8.pb
opencv_face_detector.pbtxt
```

we are building a Flask Application which 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.Sample images folder contains the sample images which we give manually in the code by giving the path of the image.

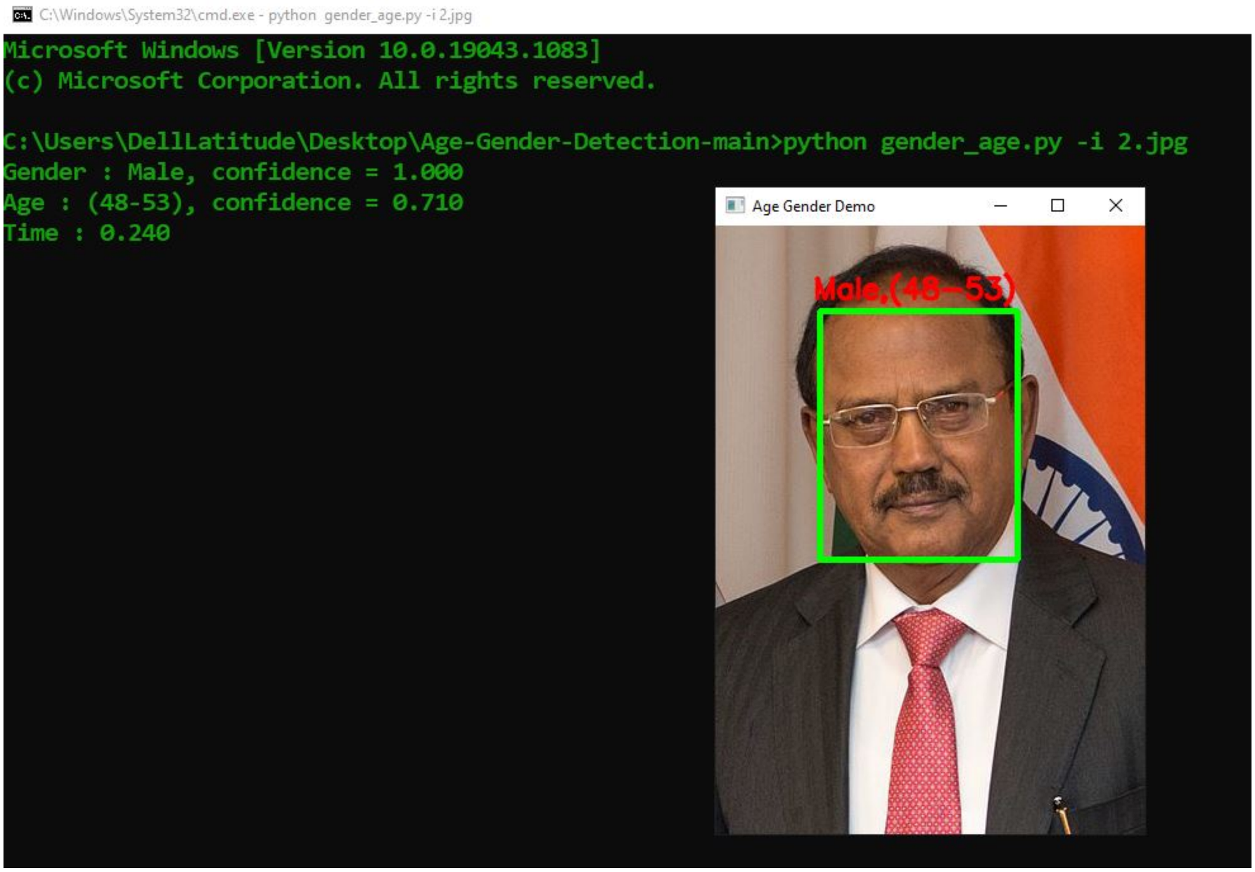
Age And Gender Detection Using Deep Learning With IBM Cloud

Flowchart :-



Age And Gender Detection Using Deep Learning With IBM Cloud

Result:-



C:\Windows\System32\cmd.exe - python gender_age.py -i 2.jpg
Age Gender Demo
Male,(48-53)

```
Microsoft Windows [Version 10.0.19043.1083]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DellLatitude\Desktop\Age-Gender-Detection-main>python gender_age.py -i 2.jpg
Gender : Male, confidence = 1.000
Age : (48-53), confidence = 0.710
Time : 0.240
```

Age And Gender Detection Using Deep Learning With IBM Cloud

Applications:-

- Security and Video Surveillance,
- Electronic Customer Relationship Management,
- Biometrics,
- Electronic Vending machines,
- Human-Computer interaction,
- Entertainment,
- Cosmetology,
- Forensic art etc..

Appendix

a. Source code

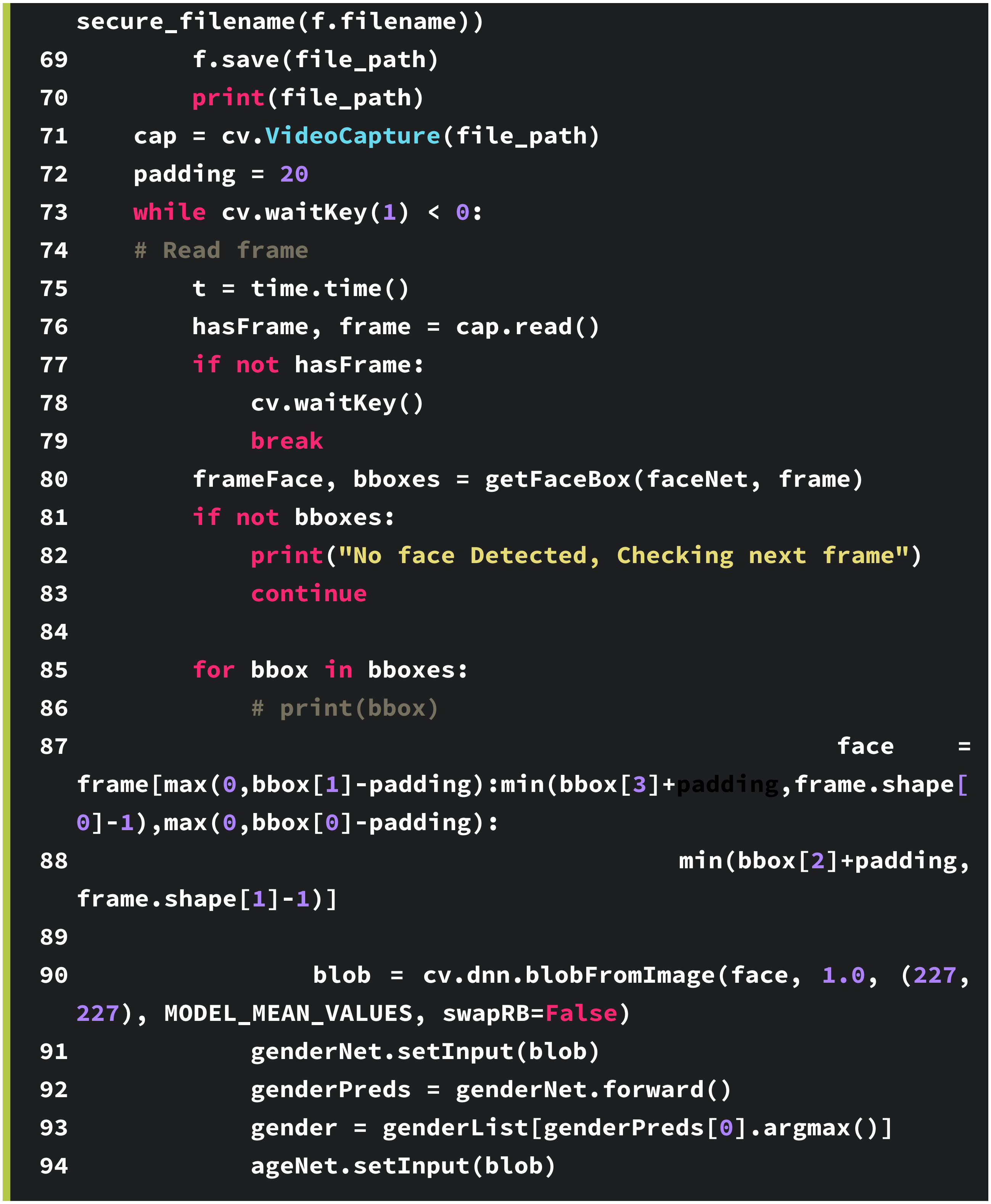
AG.py

```
# USAGE
import cv2 as cv #opencv
import time #time
from flask import Flask,request, render_template
import os
from werkzeug.utils import secure_filename

#face Detection
faceProto = "./Datasets/opencv_face_detector.pbtxt"
faceModel = "./Datasets/opencv_face_detector_uint8.pb"

#Age Predition
ageProto = "./Datasets/age_deploy.prototxt" #weight file training data
ageModel = "./Datasets/age_net.caffemodel" #model file
```

Age And Gender Detection Using Deep Learning With IBM Cloud

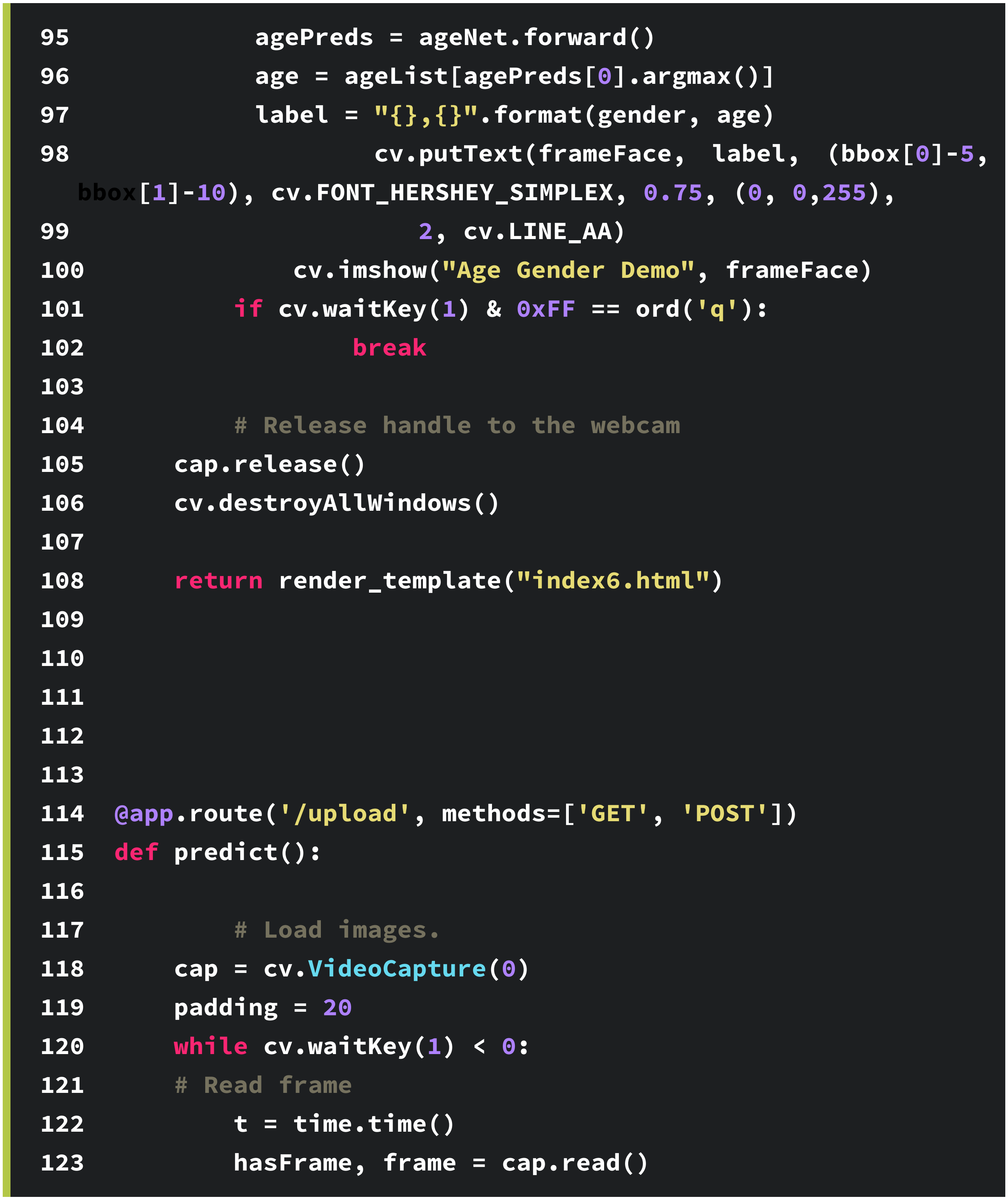
```

#Gender Detetion
genderProto = "./Datasets/gender_deploy.prototxt"
genderModel = "./Datasets/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)'] #age list
genderList = ['Male', 'Female'] # Gender List

# Load network
ageNet = cv.dnn.readNetFromCaffe(ageProto,ageModel)#Age #dnn-deep neural network is a pre trained model
genderNet = cv.dnn.readNetFromCaffe(genderProto,genderModel)#Gender
faceNet = cv.dnn.readNet(faceModel,faceProto)#Face


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()#stores the face data
    bboxes = []
    for i in range(detections.shape[2]): #drawing the
```

```
rectangles
        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

app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/image1',methods=['GET','POST'])
def image1():
    return render_template("index6.html")

@app.route('/predict',methods=['GET','POST'])
def image():
    if request.method == 'POST':
        print("inside image")
        f = request.files['image']

        basepath = os.path.dirname(__file__)
              file_path = os.path.join(basepath, 'uploads',
```



```
secure_filename(f.filename))
        f.save(file_path)
        print(file_path)
    cap = cv.VideoCapture(file_path)
    padding = 20
    while cv.waitKey(1) < 0:
    # Read frame
        t = time.time()
        hasFrame, frame = cap.read()
        if not hasFrame:
            cv.waitKey()
            break
        frameFace, bboxes = getFaceBox(faceNet, frame)
        if not bboxes:
            print("No face Detected, Checking next frame")
            continue

        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)
```

Age And Gender Detection Using Deep Learning With IBM Cloud

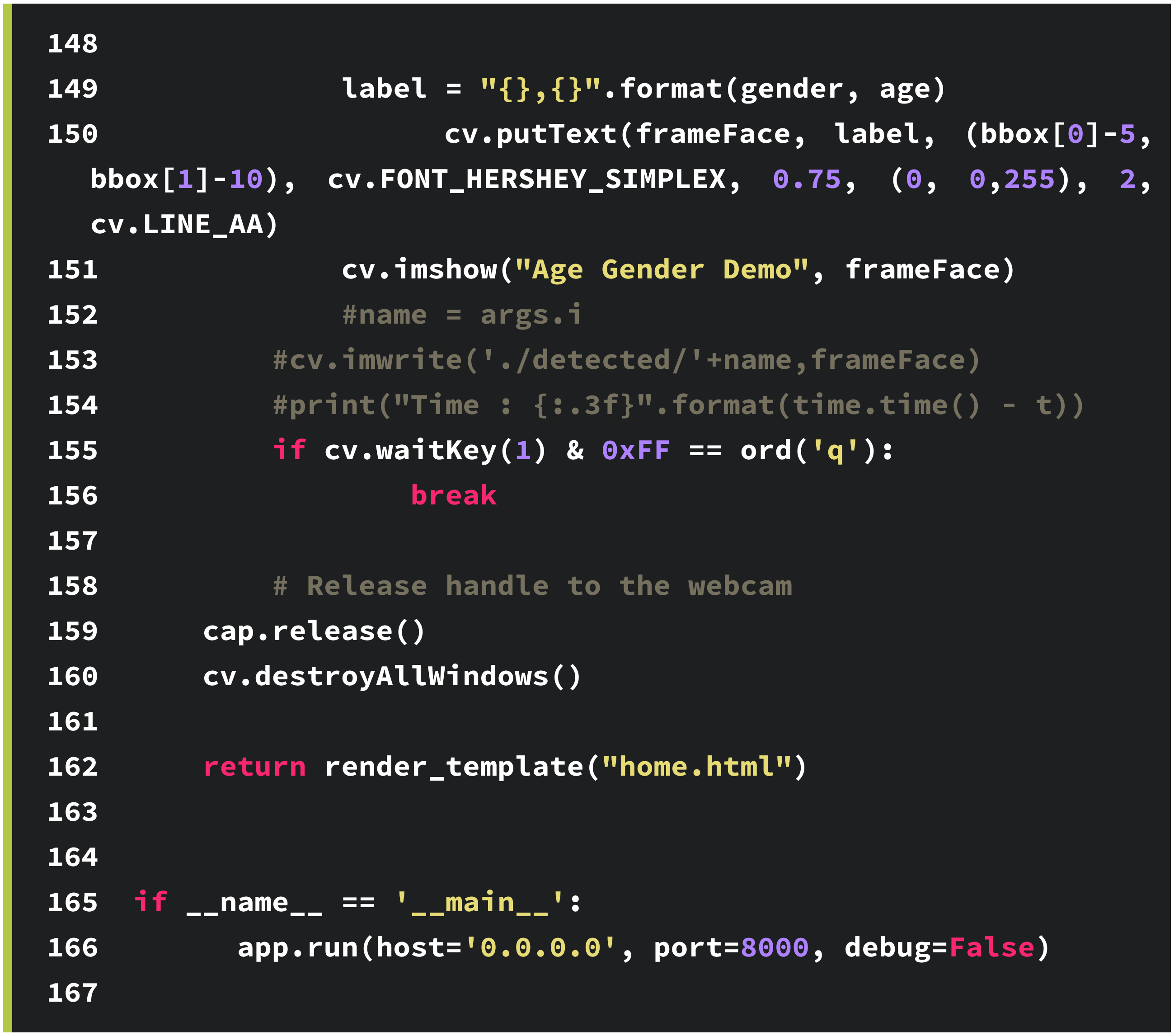


```
         agePreds = ageNet.forward()
         age = ageList[agePreds[0].argmax()]
         label = "{},{}".format(gender, age)
                 cv.putText(frameFace,  label,  (bbox[0]-5, bbox[1]-10), cv.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0,255),
                    2, cv.LINE_AA)
            cv.imshow("Age Gender Demo", frameFace)
        if cv.waitKey(1) & 0xFF == ord('q'):
                break

        # Release handle to the webcam
    cap.release()
    cv.destroyAllWindows()

    return render_template("index6.html")





@app.route('/upload', methods=['GET', 'POST'])
def predict():

        # Load images.
    cap = cv.VideoCapture(0)
    padding = 20
    while cv.waitKey(1) < 0:
    # Read frame
        t = time.time()
        hasFrame, frame = cap.read()
```

Age And Gender Detection Using Deep Learning With IBM Cloud

```
        if not hasFrame:
            cv.waitKey()
            break
        frameFace, bboxes = getFaceBox(faceNet, frame)
        if not bboxes:
            print("No face Detected, Checking next frame")
            continue

        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()]

            #print("Gender : {}, confidence = {:.3f}".format(gender, genderPreds[0].max()))

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

        # print("Age : {}, confidence = {:.3f}".format(age, agePreds[0].max()))
```

Age And Gender Detection Using Deep Learning With IBM Cloud



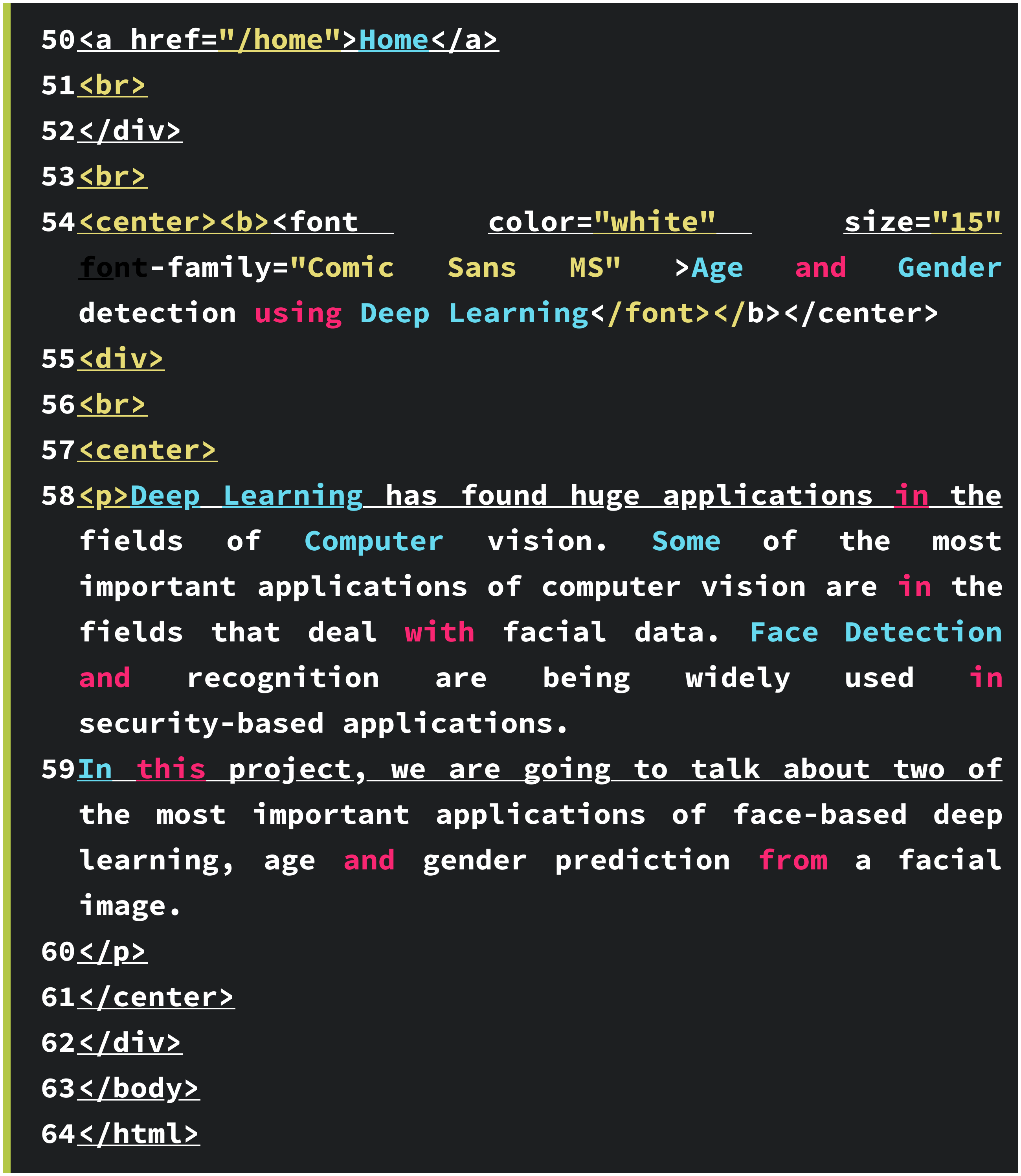
```

            label = "{},{}".format(gender, age)
                  cv.putText(frameFace, label, (bbox[0]-5, bbox[1]-10), cv.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0,255), 2, cv.LINE_AA)
            cv.imshow("Age Gender Demo", frameFace)
            #name = args.i
        #cv.imwrite('./detected/'+name,frameFace)
        #print("Time : {:.3f}".format(time.time() - t))
        if cv.waitKey(1) & 0xFF == ord('q'):
                break

        # Release handle to the webcam
    cap.release()
    cv.destroyAllWindows()

    return render_template("home.html")


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

```

Age And Gender Detection Using Deep Learning With IBM Cloud

home.html:-

```
<!DOCTYPE html>
<html>
<head>
<title>Home</title>
<style>
.navbar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
```

```
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
p
{
color:white;
font-style:italic;
font-size:30px;
}
body
{
background-image: url("https://cdn.dribbble.com/users/3496409/screenshots/7749099/scanner.gif");
background-size: cover;
}
</style>
</head>
<body>
<div class="navbar">
<a href="/upload" >Real Time Analysis</a>
<a href="/image1" >Predict</a>
```



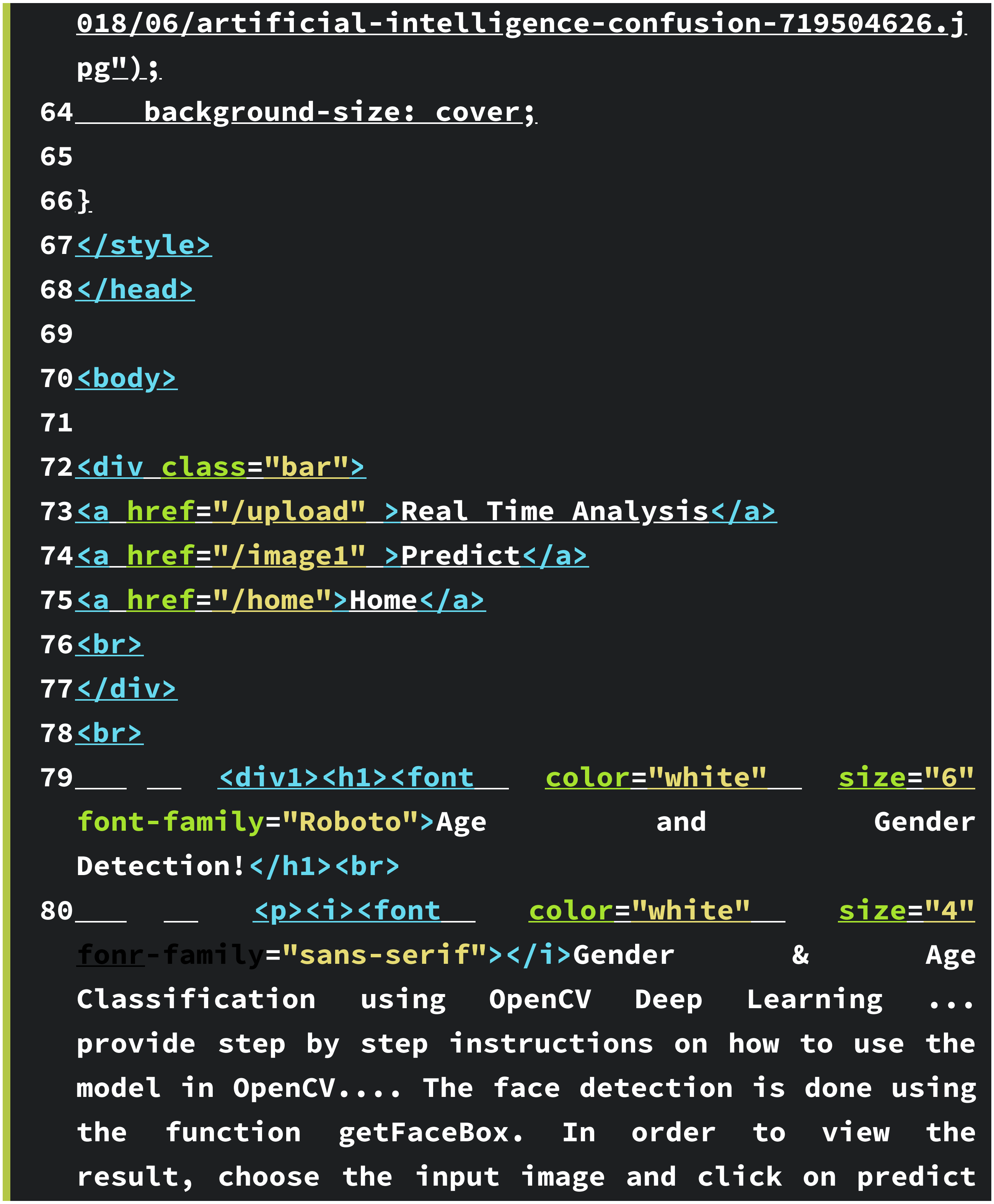
```
<a href="/home">Home</a>
<br>
</div>
<br>
<center><b><font color="white" size="15" font-family="Comic Sans MS" >Age and Gender detection using Deep Learning</font></b></center>
<div>
<br>
<center>
<p>Deep Learning has found huge applications in the fields of Computer vision. Some of the most important 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 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.
</p>
</center>
</div>
</body>
</html>
```

Age And Gender Detection Using Deep Learning With IBM Cloud

index6.html:-

```

<html lang="en">

<head>
<meta charset="utf-8">
              <meta http-equiv="X-UA-Compatible" content="IE=edge">
              <meta name="viewport" content="width=device-width, initial-scale=0.6">
              <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

              <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <meta charset="UTF-8">
    <title>Predict</title>
              <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

              <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src=
"https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"
></script>
    <script src=
"https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstra
p.min.js"></script>
<link href="{{ url_for('static',
filename='css/main.css') }}" rel="stylesheet">
<style>
.bar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
```

```
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}

div1{
  text-align: center;
  width: 650spx;
  height: 750px;
  border: 5px solid teal;
  padding: 20px;
  margin: 10px;
  position: absolute;

}

body
{
                background-image: url("https://singularityhub.com/wp-content/uploads/2
```



```
018/06/artificial-intelligence-confusion-719504626.jpg");
    background-size: cover;

}
</style>
</head>

<body>

<div class="bar">
<a href="/upload" >Real Time Analysis</a>
<a href="/image1" >Predict</a>
<a href="/home">Home</a>
<br>
</div>
<br>
        <div1><h1><font color="white" size="6" font-family="Roboto">Age and Gender Detection!</h1><br>
        <p><i><font color="white" size="4" fonr-family="sans-serif"></i>Gender & Age Classification using OpenCV Deep Learning ... provide step by step instructions on how to use the model in OpenCV.... The face detection is done using the function getFaceBox. In order to view the result, choose the input image and click on predict
```

```
button.</p>
<br>
                    <div>
                        <h4>Upload Image Here</h4>
                <form action = "http://localhost:5000/" id="upload-file" method="post"
enctype="multipart/form-data">
                    <label for="imageUpload" class="upload-label">
                        Choose...
                    </label>
                    <input type="file" name="image" id="imageUpload" accept=".png,
.jpg, .jpeg,.pdf">
                </form>
                <center>
                <div class="image-section" style="display:none;">
                    <div class="img-preview">
                        <div id="imagePreview">
                        </div>
                    </div>
                    <div>
                        <button type="button" class="btn btn-info btn-lg "
id="btn-predict">Predict!</button>
                    </div>
                </div>
                  <div class="loader" style="display:none;"></div>
                </center>


            </div>
            </div1>

        <footer>
```

```
    <script src=
"{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>


</html>

```

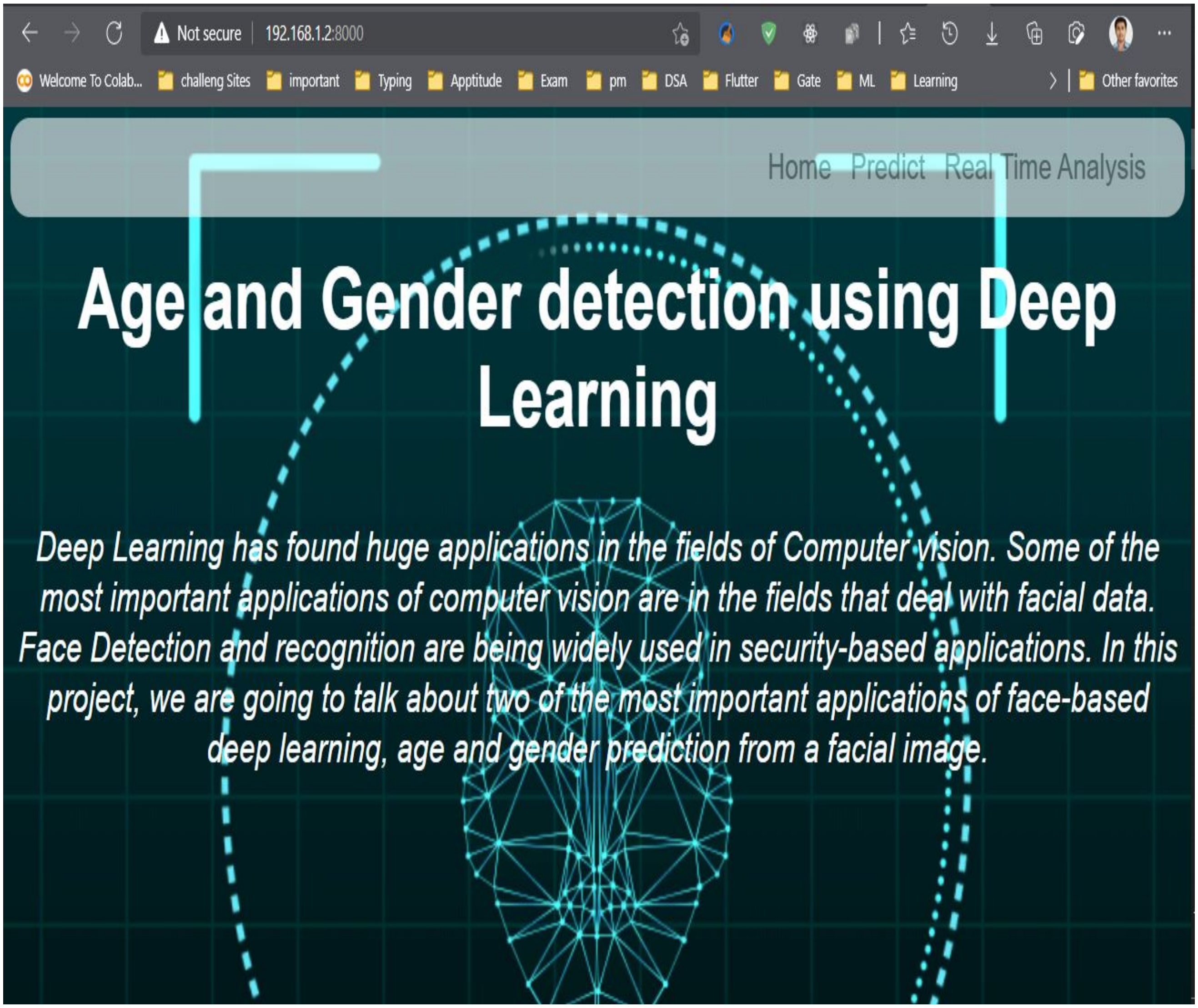
upload.html:-

```

<html lang="en">

<head>
    <title>Register</title>
    <link href=
"https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
<style>
.bar
{
margin:  0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style:  italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
body
{
    background-image: url("");
    background-size: cover;
}
p
{
color:white;
font-style:italic;
font-size:30px;
}
</style>
</head>

<body>

<div class="bar">
<a href="/upload" >Real Time Analysis</a>
<a href="/image1" >Predict</a>
<a href="/home">Home</a>
<br>
</div>
```

```
    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">Age and Gender Recognition Using Deep Learning</a>
        </div>

```

b. UI output Screenshot.

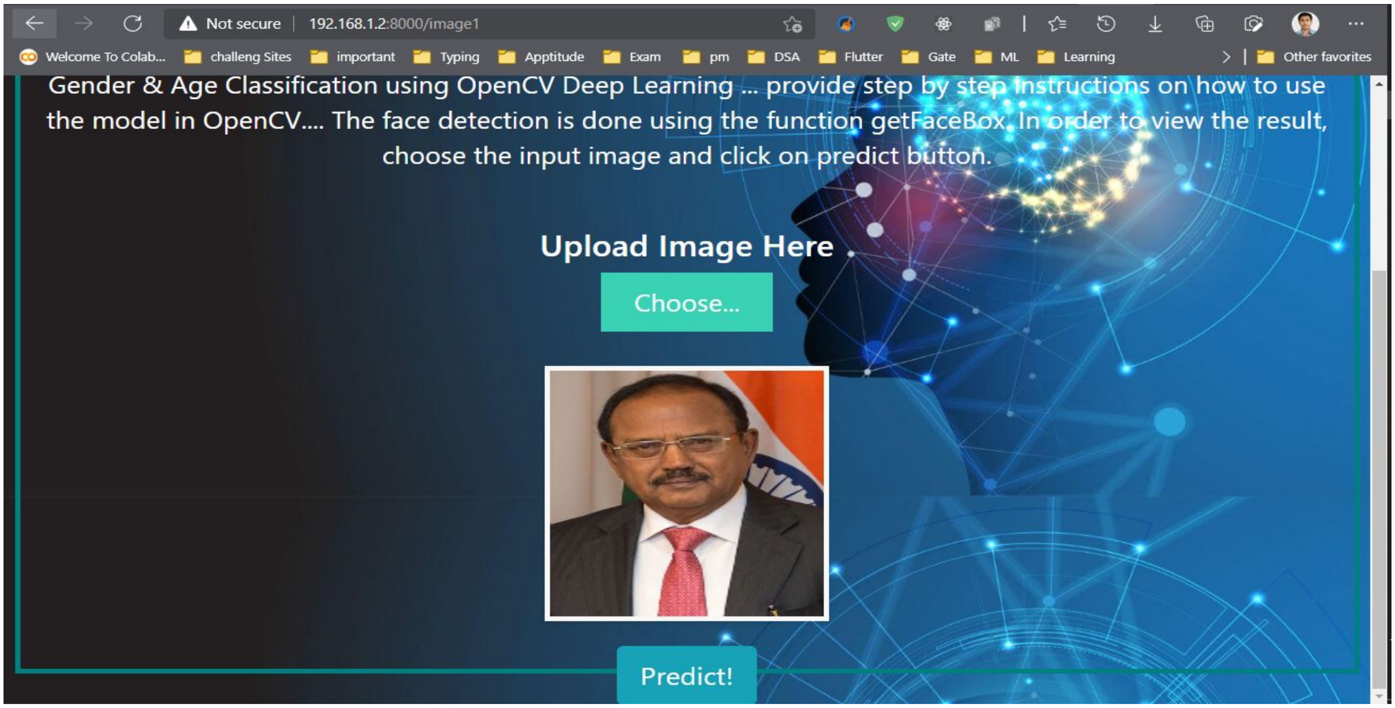
Age And Gender Detection Using Deep Learning With IBM Cloud



Age and Gender detection using Deep Learning

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 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.

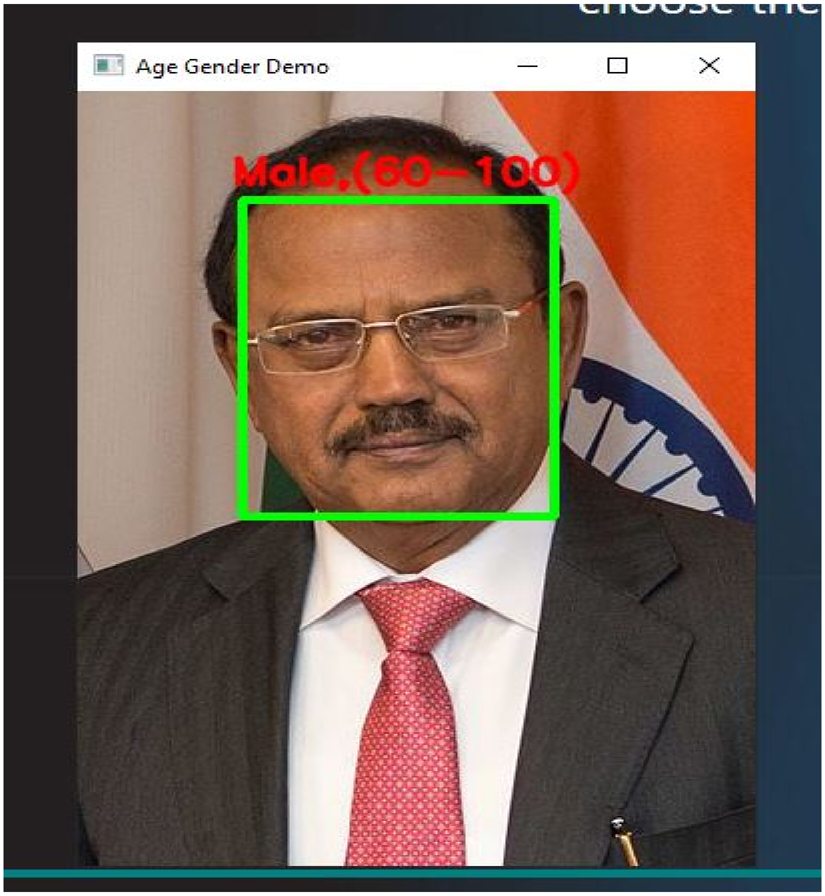
Age And Gender Detection Using Deep Learning With IBM Cloud



Gender & Age Classification using OpenCV Deep Learning ... provide step by step instructions on how to use the model in OpenCV.... The face detection is done using the function getFaceBox. In order to view the result, choose the input image and click on predict button.

Upload Image Here .

Age And Gender Detection Using Deep Learning With IBM Cloud


choose the
Age Gender Demo
Male,(60–100)