

# 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.

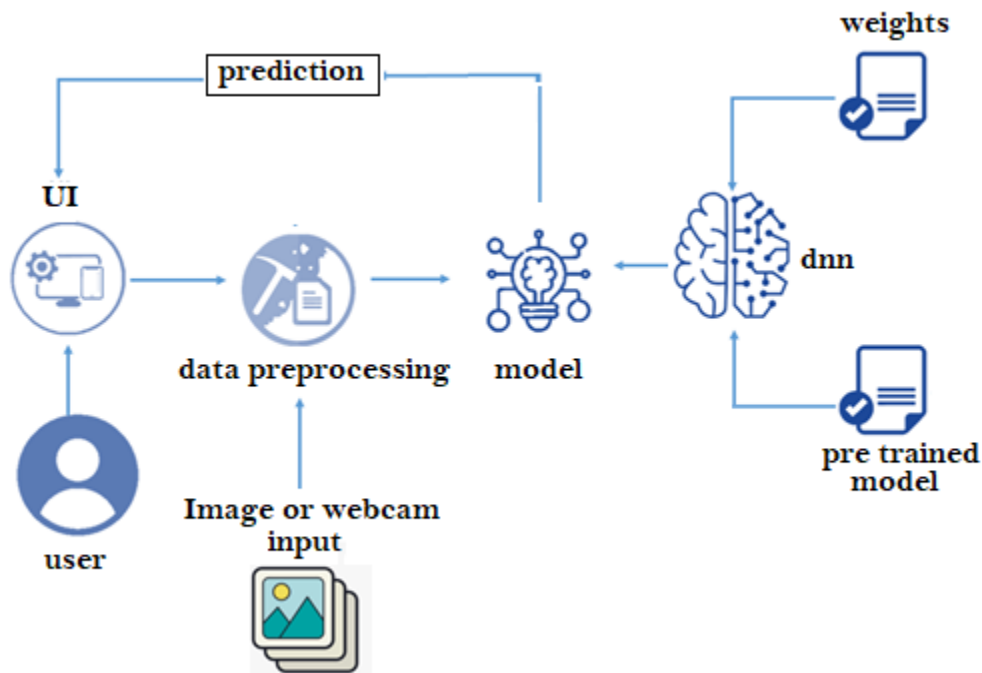
## **b. Purpose:-**

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.

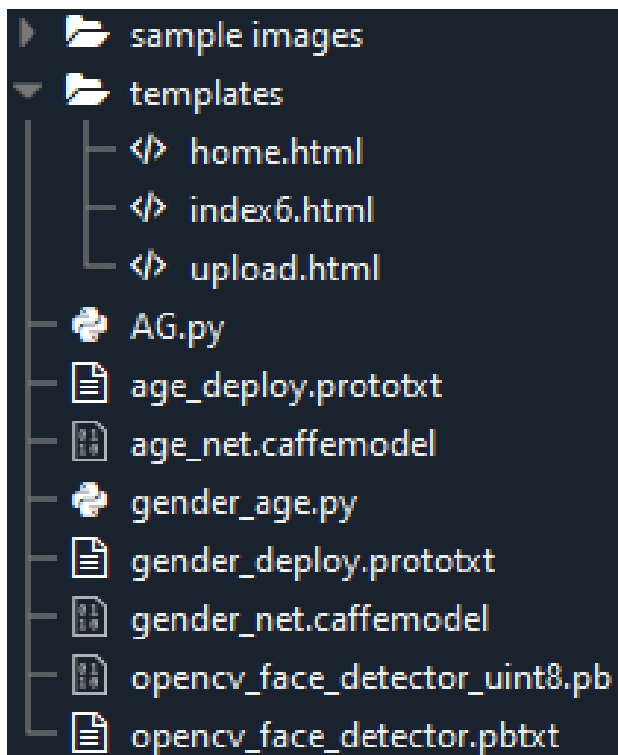
# Age And Gender Detection Using Deep Learning With IBM Cloud

## Theoretical Analysis:

### a. Block diagram :-



### b. Hardware / Software designing:-

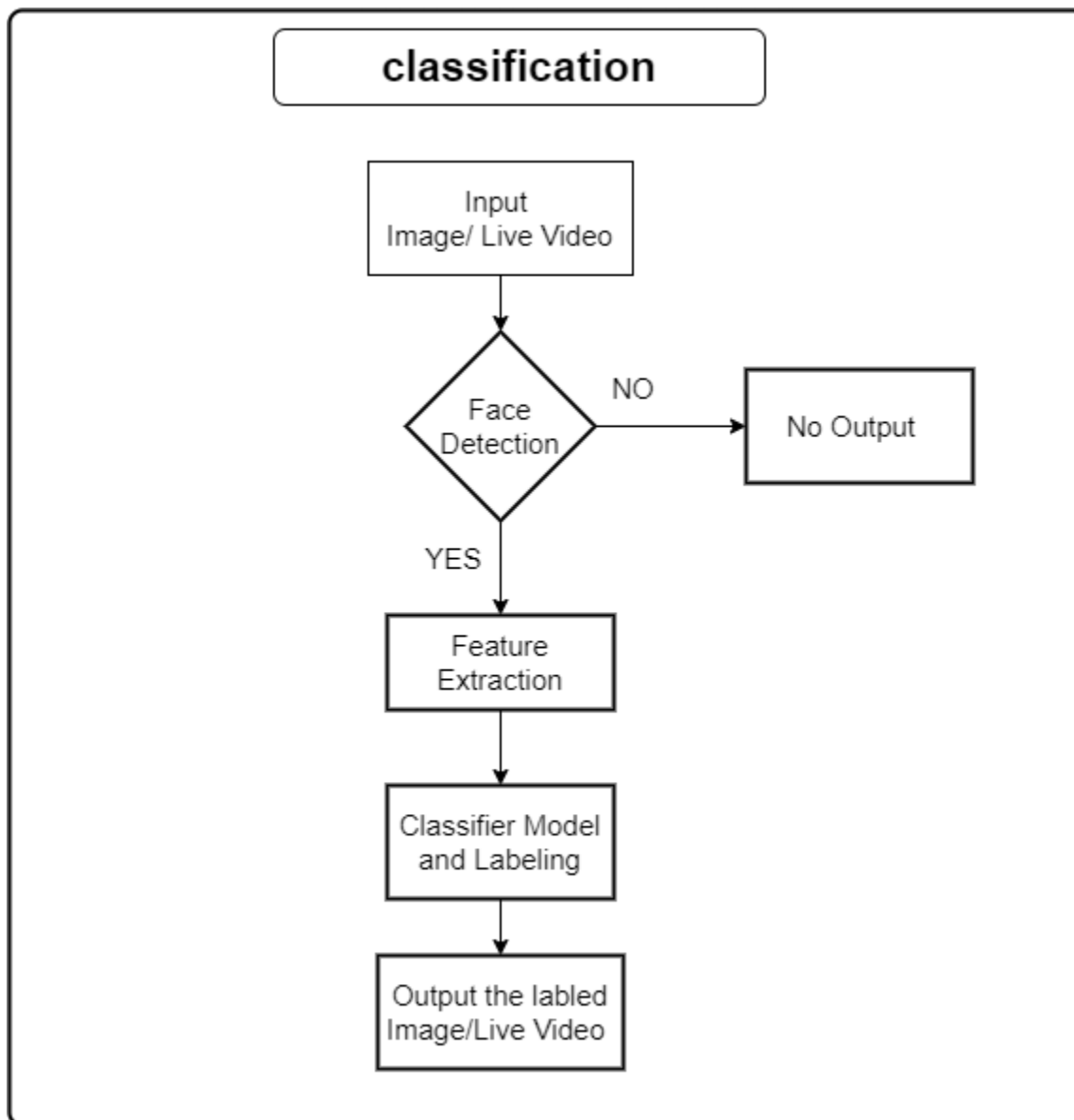


we are building a Flask Application which needs HTML pages stored in the templates folder and a python script AG.py which we run from prompt whereas gender\_age.py should run from the python IDE.

prototxt 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 :-**



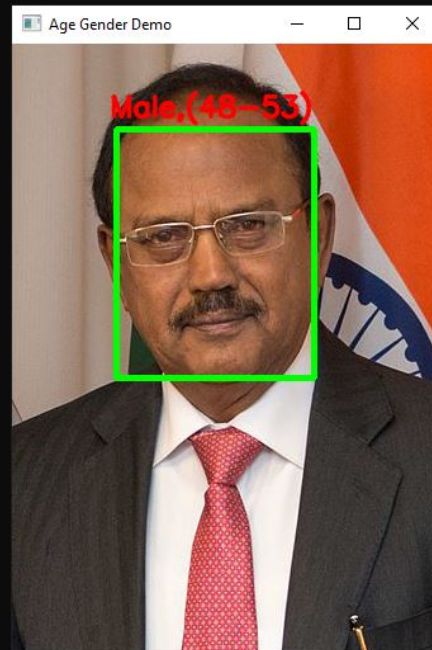
# Age And Gender Detection Using Deep Learning With IBM Cloud

## Result:-

C:\Windows\System32\cmd.exe - python gender\_age.py -i 2.jpg

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](#)

```
1 # USAGE
2 import cv2 as cv #opencv
3 import time #time
4 from flask import Flask,request, render_template
5 import os
6 from werkzeug.utils import secure_filename
7
8 #face Detection
9 faceProto = "./Datasets/opencv_face_detector.pbtxt"
10 faceModel = "./Datasets/opencv_face_detector_uint8.pb"
11
12 #Age Prediction
13 ageProto =  "./Datasets/age_deploy.prototxt" #weight file
    training data
14 ageModel =  "./Datasets/age_net.caffemodel" #model file
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
15
16 #Gender Detetion
17 genderProto = "./Datasets/gender_deploy.prototxt"
18 genderModel = "./Datasets/gender_net.caffemodel"
19
20 MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744,
    114.895847746)
21 ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)',
    '(38-43)', '(48-53)', '(60-100)'] #age list
22 genderList = ['Male', 'Female'] # Gender List
23
24 # Load network
25 ageNet = cv.dnn.readNetFromCaffe(ageProto,ageModel)#Age
    #dnn-deep neural network is a pre trained model
26 genderNet =
    cv.dnn.readNetFromCaffe(genderProto,genderModel)#Gender
27 faceNet = cv.dnn.readNet(faceModel,faceProto)#Face
28
29
30 def getFaceBox(net, frame, conf_threshold=0.7):
31     frameOpencvDnn = frame.copy()
32     frameHeight = frameOpencvDnn.shape[0]
33     frameWidth = frameOpencvDnn.shape[1]
34     blob = cv.dnn.blobFromImage(frameOpencvDnn, 1.0, (300,
    300), [104, 117, 123], True, False)
35
36     net.setInput(blob)
37     detections = net.forward()#stores the face data
38     bboxes = []
39     for i in range(detections.shape[2]): #drawing the
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
rectangles
40     confidence = detections[0, 0, i, 2]
41     if confidence > conf_threshold:
42         x1 = int(detections[0, 0, i, 3] * frameWidth)
43         y1 = int(detections[0, 0, i, 4] * frameHeight)
44         x2 = int(detections[0, 0, i, 5] * frameWidth)
45         y2 = int(detections[0, 0, i, 6] * frameHeight)
46         bboxes.append([x1, y1, x2, y2])
47         cv.rectangle(frameOpencvDnn, (x1, y1), (x2, y2),
48             (0, 255, 0), int(round(frameHeight/150)), 8)
49     return frameOpencvDnn, bboxes
50 app=Flask(__name__,template_folder="templates")
51 @app.route('/', methods=['GET'])
52 def index():
53     return render_template('home.html')
54 @app.route('/home', methods=['GET'])
55 def about():
56     return render_template('home.html')
57 @app.route('/image1',methods=['GET','POST'])
58 def image1():
59     return render_template("index6.html")
60
61 @app.route('/predict',methods=['GET','POST'])
62 def image():
63     if request.method == 'POST':
64         print("inside image")
65         f = request.files['image']
66
67         basepath = os.path.dirname(__file__)
68         file_path = os.path.join(basepath, 'uploads',
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
secure_filename(f.filename))
69     f.save(file_path)
70     print(file_path)
71     cap = cv.VideoCapture(file_path)
72     padding = 20
73     while cv.waitKey(1) < 0:
74         # Read frame
75         t = time.time()
76         hasFrame, frame = cap.read()
77         if not hasFrame:
78             cv.waitKey()
79             break
80         frameFace, bboxes = getFaceBox(faceNet, frame)
81         if not bboxes:
82             print("No face Detected, Checking next frame")
83             continue
84
85         for bbox in bboxes:
86             # print(bbox)
87
88             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)]
89
90             blob = cv.dnn.blobFromImage(face, 1.0, (227,
227), MODEL_MEAN_VALUES, swapRB=False)
91             genderNet.setInput(blob)
92             genderPreds = genderNet.forward()
93             gender = genderList[genderPreds[0].argmax()]
94             ageNet.setInput(blob)
```



# Age And Gender Detection Using Deep Learning With IBM Cloud

```
95         agePreds = ageNet.forward()
96         age = ageList[agePreds[0].argmax()]
97         label = "{},{ {}".format(gender, age)
98         cv.putText(frameFace, label, (bbox[0]-5,
bbox[1]-10), cv.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0, 255),
99                 2, cv.LINE_AA)
100         cv.imshow("Age Gender Demo", frameFace)
101         if cv.waitKey(1) & 0xFF == ord('q'):
102             break
103
104         # Release handle to the webcam
105     cap.release()
106     cv.destroyAllWindows()
107
108     return render_template("index6.html")
109
110
111
112
113
114 @app.route('/upload', methods=['GET', 'POST'])
115 def predict():
116
117     # Load images.
118     cap = cv.VideoCapture(0)
119     padding = 20
120     while cv.waitKey(1) < 0:
121         # Read frame
122         t = time.time()
123         hasFrame, frame = cap.read()
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
124         if not hasFrame:
125             cv.waitKey()
126             break
127         frameFace, bboxes = getFaceBox(faceNet, frame)
128         if not bboxes:
129             print("No face Detected, Checking next frame")
130             continue
131
132         for bbox in bboxes:
133             # print(bbox)
134
135             face =
136             frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[
137                 0]-1),max(0,bbox[0]-padding):min(bbox[2]+padding,
138                 frame.shape[1]-1)]
139
140             blob = cv.dnn.blobFromImage(face, 1.0, (227,
141                 227), MODEL_MEAN_VALUES, swapRB=False)
142             genderNet.setInput(blob)
143             genderPreds = genderNet.forward()
144             gender = genderList[genderPreds[0].argmax()]
145
146             #print("Gender : {}, confidence =
147             {:.3f}".format(gender, genderPreds[0].max()))
148
149             ageNet.setInput(blob)
150             agePreds = ageNet.forward()
151             age = ageList[agePreds[0].argmax()]
152
153             # print("Age : {}, confidence = {:.3f}".format(age,
154             agePreds[0].max()))
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
148
149         label = "{}{}".format(gender, age)
150         cv.putText(frameFace, label, (bbox[0]-5,
    bbox[1]-10), cv.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0, 255), 2,
    cv.LINE_AA)
151         cv.imshow("Age Gender Demo", frameFace)
152         #name = args.i
153         #cv.imwrite('./detected/'+name, frameFace)
154         #print("Time : {:.3f}".format(time.time() - t))
155         if cv.waitKey(1) & 0xFF == ord('q'):
156             break
157
158         # Release handle to the webcam
159     cap.release()
160     cv.destroyAllWindows()
161
162     return render_template("home.html")
163
164
165 if __name__ == '__main__':
166     app.run(host='0.0.0.0', port=8000, debug=False)
167
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

## home.html:-

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Home</title>
5 <style>
6 .navbar
7 {
8   margin: 0px;
9   padding:20px;
10  background-color:white;
11  opacity:0.6;
12  color:black;
13  font-family:'Roboto',sans-serif;
14  font-style: italic;
15  border-radius:20px;
16  font-size:25px;
17 }
18 a
19 {
20  color:grey;
21  float:right;
22  text-decoration:none;
23  font-style:normal;
24  padding-right:20px;
25 }
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
26a:hover{
27background-color:black;
28color:white;
29border-radius:15px;0
30font-size:30px;
31padding-left:10px;
32}
33p
34{
35color:white;
36font-style:italic;
37font-size:30px;
38}
39body
40{
41background-image:
    url("https://cdn.dribbble.com/users/3496409/screenshots/7749099/scanner.gif");
42background-size: cover;
43}
44</style>
45</head>
46<body>
47<div class="navbar">
48<a href="/upload" >Real Time Analysis</a>
49<a href="/image1" >Predict</a>
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
50<a href="/home">Home</a>
51<br>
52</div>
53<br>
54<center><b><font      color="white"      size="15"
font-family="Comic Sans MS" >Age and Gender
detection using Deep Learning</font></b></center>
55<div>
56<br>
57<center>
58<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.
59In 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.
60</p>
61</center>
62</div>
63</body>
64</html>
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

index6.html:-

```
1
2 <html lang="en">
3
4 <head>
5 <meta charset="utf-8">
6 _____ <meta http-equiv="X-UA-Compatible"
  content="IE=edge">
7 _____ <meta name="viewport"
  content="width=device-width, initial-scale=0.6">
8 _____ <script
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.
  3.1/jquery.min.js"></script>
9
10 _____ <link
  href="https://fonts.googleapis.com/icon?family=Mater
  ial+Icons" rel="stylesheet">
11 _____ <meta charset="UTF-8">
12 _____ <title>Predict</title>
13 _____ <link
  href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bo
  otstrap.min.css" rel="stylesheet">
14
15 _____ <script
  src="https://cdn.bootcss.com/popper.js/1.12.9/umd/po
  pper.min.js"></script>
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
16 _____<script src=
17 "https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"
   ></script>
18 _____<script src=
19 "https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap
   p.min.js"></script>
20 <link _____ href="{ { _____ url_for('static',
   filename='css/main.css') } }" rel="stylesheet">
21 <style>
22 .bar
23 {
24 margin: 0px;
25 padding:20px;
26 background-color:white;
27 opacity:0.6;
28 color:black;
29 font-family:'Roboto',sans-serif;
30 font-style: italic;
31 border-radius:20px;
32 font-size:25px;
33 }
34 a
35 {
36 color:grey;
37 float:right;
38 text-decoration:none;
```



# Age And Gender Detection Using Deep Learning With IBM Cloud

```
39font-style:normal;
40padding-right:20px;
41}
42a:hover{
43background-color:black;
44color:white;
45border-radius:15px;0
46font-size:30px;
47padding-left:10px;
48}
49
50div1{
51  text-align: center;
52  width: 650spx;
53  height: 750px;
54  border: 5px solid teal;
55  padding: 20px;
56  margin: 10px;
57  position: absolute;
58
59}
60
61body
62{
63  _____ background-image:
  url("https://singularityhub.com/wp-content/uploads/2
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
018/06/artificial-intelligence-confusion-719504626.j
pg");.
64 background-size: cover;
65
66}
67</style>
68</head>
69
70<body>
71
72<div class="bar">
73<a href="/upload" >Real Time Analysis</a>
74<a href="/image1" >Predict</a>
75<a href="/home">Home</a>
76<br>
77</div>
78<br>
79<div1><h1><font color="white" size="6"
font-family="Roboto">Age and Gender
Detection!</h1><br>
80<p><i><font color="white" size="4"
font-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
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
button.</p>
81_<br>
82          <div>
83              <h4>Upload Image Here</h4>
84              <form action = "http://localhost:5000/" id="upload-file" method='post'
enctype="multipart/form-data">
85                  <label for="imageUpload" class="upload-label">
86                      Choose...
87                  </label>
88                  <input type="file" name="image" id="imageUpload" accept=".png,
.jpg, .jpeg,.pdf">
89              </form>
90_<center>
91          <div class="image-section" style="display:none;">
92              <div class="img-preview">
93                  <div id="imagePreview">
94                      </div>
95              </div>
96              <div>
97                  <button type="button" class="btn btn-info btn-lg "
id="btn-predict">Predict!</button>
98              </div>
99          </div>
100          <div class="loader" style="display:none;"></div>
101_</center>
102
103
104          </div>
105          </div1>
106
107_<footer>
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
108 _____<script src=  
109 "{_url_for('static', filename='js/main.js')_}"  
    type="text/javascript"></script>  
110 </footer>  
111  
112  
113 </html>  
114
```

## upload.html:-

```
1  
2 <html lang="en">  
3  
4 <head>  
5 _____<title>Register</title>  
6 _____<link href=
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
7 "https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstr  
  ap.min.css" rel="stylesheet">  
8 <style>  
9 .bar  
10{  
11margin: 0px;  
12padding:20px;  
13background-color:white;  
14opacity:0.6;  
15color:black;  
16font-family: 'Roboto',sans-serif;  
17font-style: italic;  
18border-radius:20px;  
19font-size:25px;  
20}  
21a  
22{  
23color:grey;  
24float:right;  
25text-decoration:none;  
26font-style:normal;  
27padding-right:20px;  
28}  
29a:hover{  
30background-color:black;  
31color:white;
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

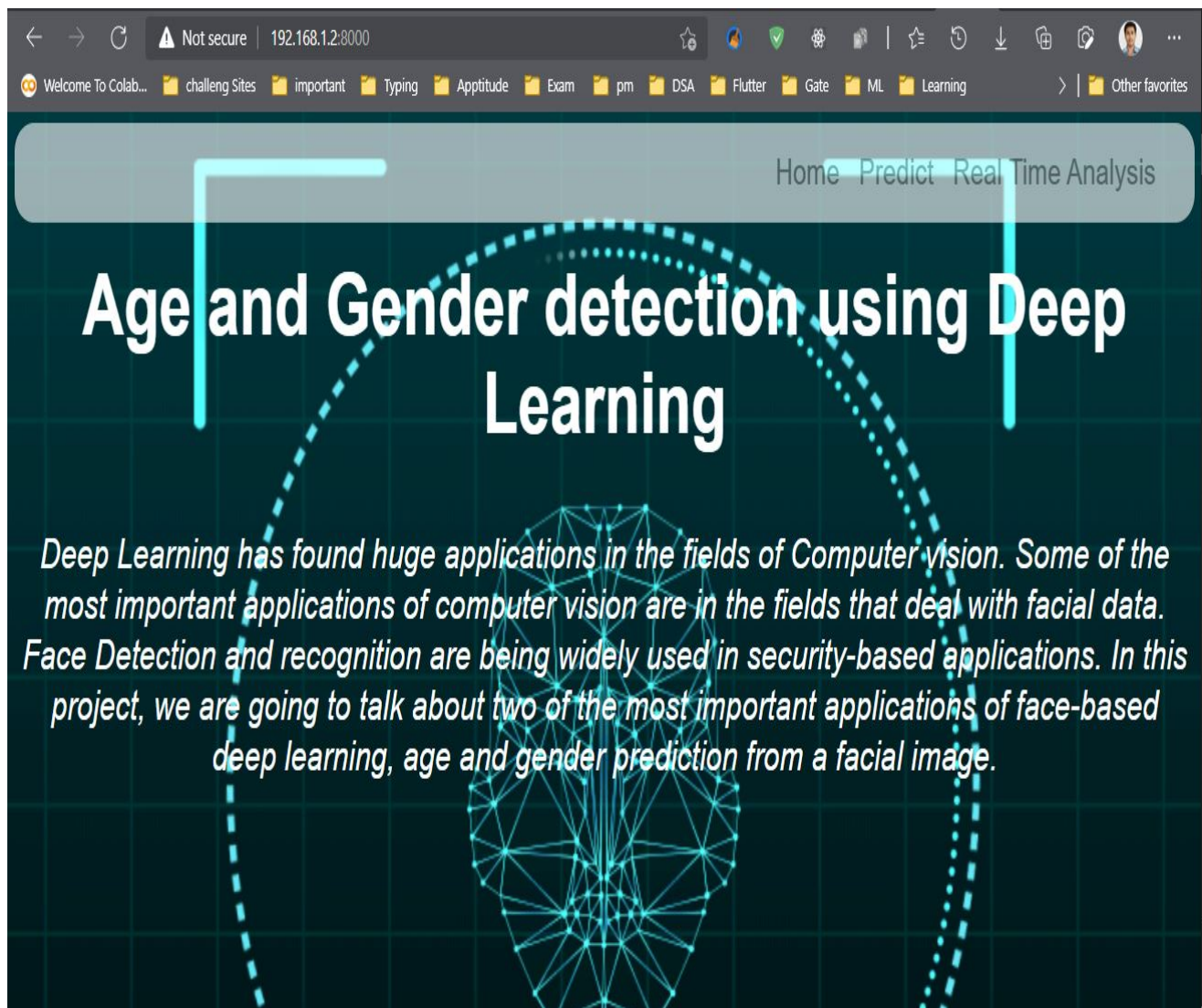
```
32border-radius:15px;0
33font-size:30px;
34padding-left:10px;
35}
36body
37{
38    background-image: url("");
39    background-size: cover;
40}
41p
42{
43color:white;
44font-style:italic;
45font-size:30px;
46}
47</style>
48</head>
49
50<body>
51
52<div class="bar">
53<a href="/upload" >Real Time Analysis</a>
54<a href="/image1" >Predict</a>
55<a href="/home">Home</a>
56<br>
57</div>
```

# Age And Gender Detection Using Deep Learning With IBM Cloud

```
58 <nav class="navbar navbar-dark bg-dark">  
59 <div class="container">  
60 <a class="navbar-brand" href="#">Age and  
   Gender Recognition Using Deep Learning</a>  
61 </div>  
62
```

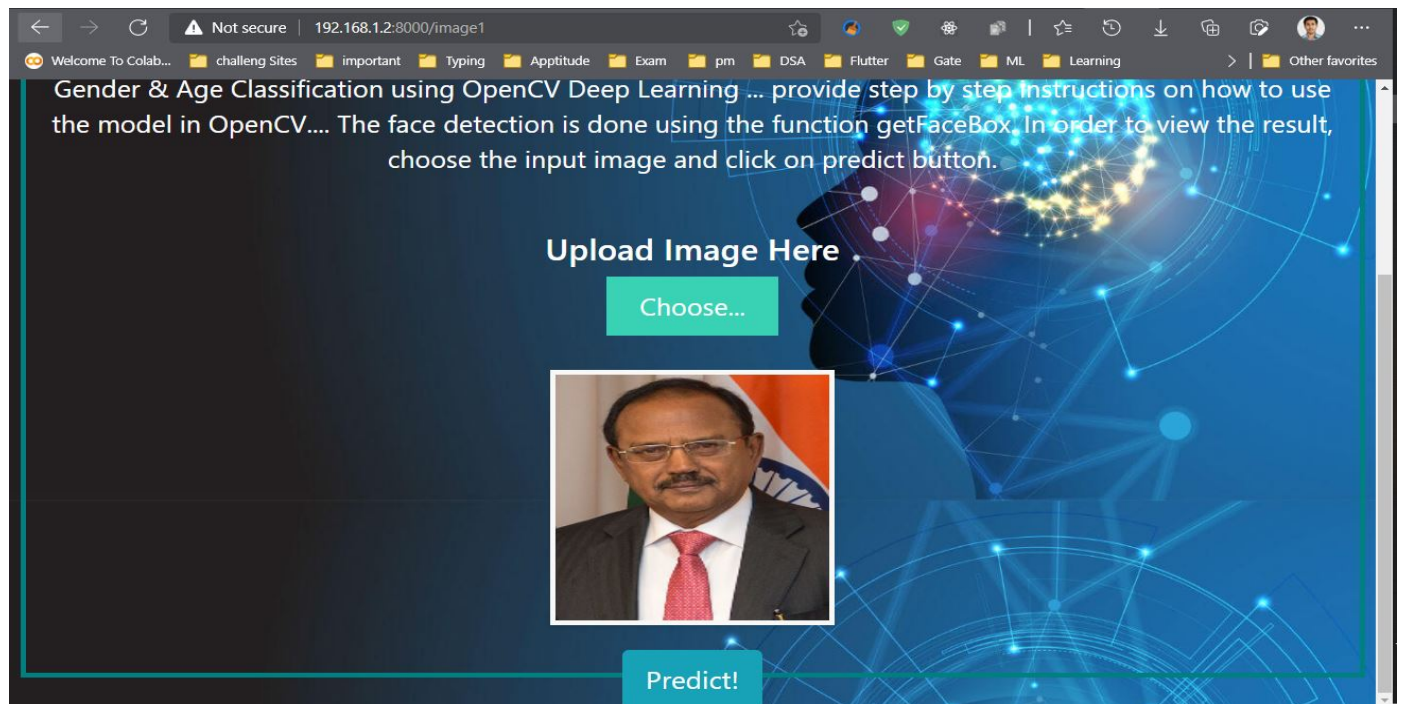
b. UI output Screenshot.

# Age And Gender Detection Using Deep Learning With IBM Cloud





# Age And Gender Detection Using Deep Learning With IBM Cloud



# Age And Gender Detection Using Deep Learning With IBM Cloud

