

# Face Recognition

END-TERM REPORT

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

By:

S.no	Name of the Student	Registration No	Roll No
1.	Kanigolla Naga Venkata Bala Likhith	11903700	48
2.	Daggupati Sai Giri Vardhan	11918609	40
3.	Avula Pavan Kumar	11906950	71

**Courses Code: INT213**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

## **Objective**

In this Python Project, I had used Deep Learning to accurately identify the gender and age of a person from a single image of a face. We can also identify the gender and age of the person in live through the web camera. It can be used to find the gender and age of the persons. It can be used in the CCTV Cameras and we can also use this at some institutions. Like in our real-life situations we will find some of the areas like amusement parks in which there will some age restrictions to enter there. At that place it can be used, and it will say the age. So that human work will be reduced. We can implement to the machine like if below the age of 12 I not allowed then it will make a sound so that they will be stopped. Similarly, we can use this in many areas.

## **Introduction**

A real time face recognition system can identify or verifying a person from a video frame. To recognize the face in a frame, first you need to detect whether the face is present in the frame. If it is present, mark it as a region of interest (ROI), extract the ROI and process it for facial recognition. Combining people tracking with age and gender detection is a good idea for many and many applications in real life scenarios such as caffe store management to gather the information of customers for further analysis, or in/out people control for security purposes in buildings.

## ***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 argparse?***

Argparse is a complete argument processing library. Arguments can trigger different actions, specified by the action argument to `add_argument()`. Supported actions include storing the argument (singly, or as part of a list), storing a constant value when the argument is encountered (including special handling for true/false values for boolean switches), counting the number of times an argument is seen, and calling a callback.

## **Feasibility Analysis**

'Face Recognition' System can play an important role in situations where gender and/or age based restrictions are imposed due to regulatory, security perspectives (but not just limited to these). Below are some of the Use Cases where this System can benefit

- Amusement Parks: Due to Security reasons some areas of the Park and/or some entertainment rides are restricted for gender/age. System that can take live feed and estimate/predict the age and/or gender can greatly help to implement Security policies.
- Covid-19 type Situation: Various restrictions are imposed in India (and around the world too) to contain the spread of the virus. One of the guidelines is to restrict the impact on kids and old age people by imposing entry restrictions for various places (like religious, travel etc.).

## **Software Requirement Analysis**

Python- 3.7 and Above

Python Libraries Required:

OpenCV

Argparse

Math - (Inbuilt Library)

## **Installation**

```
pip install opencv-python  
pip install argparse
```

## **Other Files Needed**

```
1.age_net.caffemodel  
2.gender_net.caffemodel  
3.opencv_face_detector_uint8.pb  
4.opencv_face_detector.pbtxt  
5.age_deploy.prototxt  
6.gender_deploy.prototxt
```

**Download the files at**

<https://drive.google.com/drive/folders/1gmi5jv>

## Code and its Explanation

Importing the required libraries

```
import cv2
import math
import argparse
```

Now we are writing a definition in which we will keep a frame to detect the face using the opencv library and some ui like while it detects the face it comes in the rectangle shape similarly we will define all the things. We will also define the frame length and height. In one line we can design as per our own way we style that frame. Here we used a color of parrot green. You can change that at (52,235,52) if you want any other. And we are also detecting whether the face is present or not in the given frame.

```
def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn=frame.copy()
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

    net.setInput(blob)
    detections=net.forward()
    faceBoxes=[]
    for i in range(detections.shape[2]):
        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)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (52,235,52), int(round(frameHeight/150)), 8)

    return frameOpencvDnn,faceBoxes
```

We use the argparse library to create an argument parser so we can get the image argument from the command prompt. We can use this in the two type one is by passing a image and passing the image through command prompt. I saved the python file as detectface.py. another one is detecting face by camera.

Usage : {programname} --image {imagefilewithextention}

Example: detectface.py --image sample.jpg

```
parser=argparse.ArgumentParser()
parser.add_argument('--image')
args=parser.parse_args()
```

## Including the required files

Now we are including the all the files which I said in below in the variables for further usage. 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.

```
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"
```

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

```
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)']
genderList=['Male','Female']
```

Now, use the readNet() method to load the networks. now here we will use the all the required thing which was stored in the variables in previous steps.

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

Now capture video stream in case you'd like to classify on a webcam's stream. 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.

```
video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
while cv2.waitKey(1)<0:
    hasFrame,frame=video.read()
    if not hasFrame:
        cv2.waitKey()
```

```
break
```

We 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.

```
resultImg, faceBoxes = highlightFace(faceNet, frame)
if not faceBoxes:
    print("No face detected")
```

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. Then, we do the same thing for age.

```
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()]
print(f'Gender: {gender}')

ageNet.setInput(blob)
agePreds = ageNet.forward()
age = ageList[agePreds[0].argmax()]
print(f'Age: {age[1:-1]} years')
```

Now let's add the gender and age texts to the resulting image and display it with `imshow()`.

```
cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-
10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 255), 2, cv2.LINE_AA)
cv2.imshow("Detecting age and gender", resultImg)
```

## Complete Code:

```
import cv2
import math
import argparse

def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn = frame.copy()
    frameHeight = frameOpencvDnn.shape[0]
    frameWidth = frameOpencvDnn.shape[1]
    blob = cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

    net.setInput(blob)
    detections = net.forward()
```

```

faceBoxes=[]
for i in range(detections.shape[2]):
    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)
        faceBoxes.append([x1,y1,x2,y2])
        cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (52,235,52), int(round(frame
Height/150)), 8)
    return frameOpencvDnn,faceBoxes

parser=argparse.ArgumentParser()
parser.add_argument('--image')

args=parser.parse_args()

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)', '(25-32)', '(38-43)', '(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)

video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
while cv2.waitKey(1)<0:
    hasFrame,frame=video.read()
    if not hasFrame:
        cv2.waitKey()
        break

    resultImg,faceBoxes=highlightFace(faceNet,frame)
    if not faceBoxes:
        print("No face detected")

    for faceBox in faceBoxes:
        face=frame[max(0,faceBox[1]-padding):
                    min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
                    :min(faceBox[2]+padding, frame.shape[1]-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()]
print(f'Gender: {gender}')

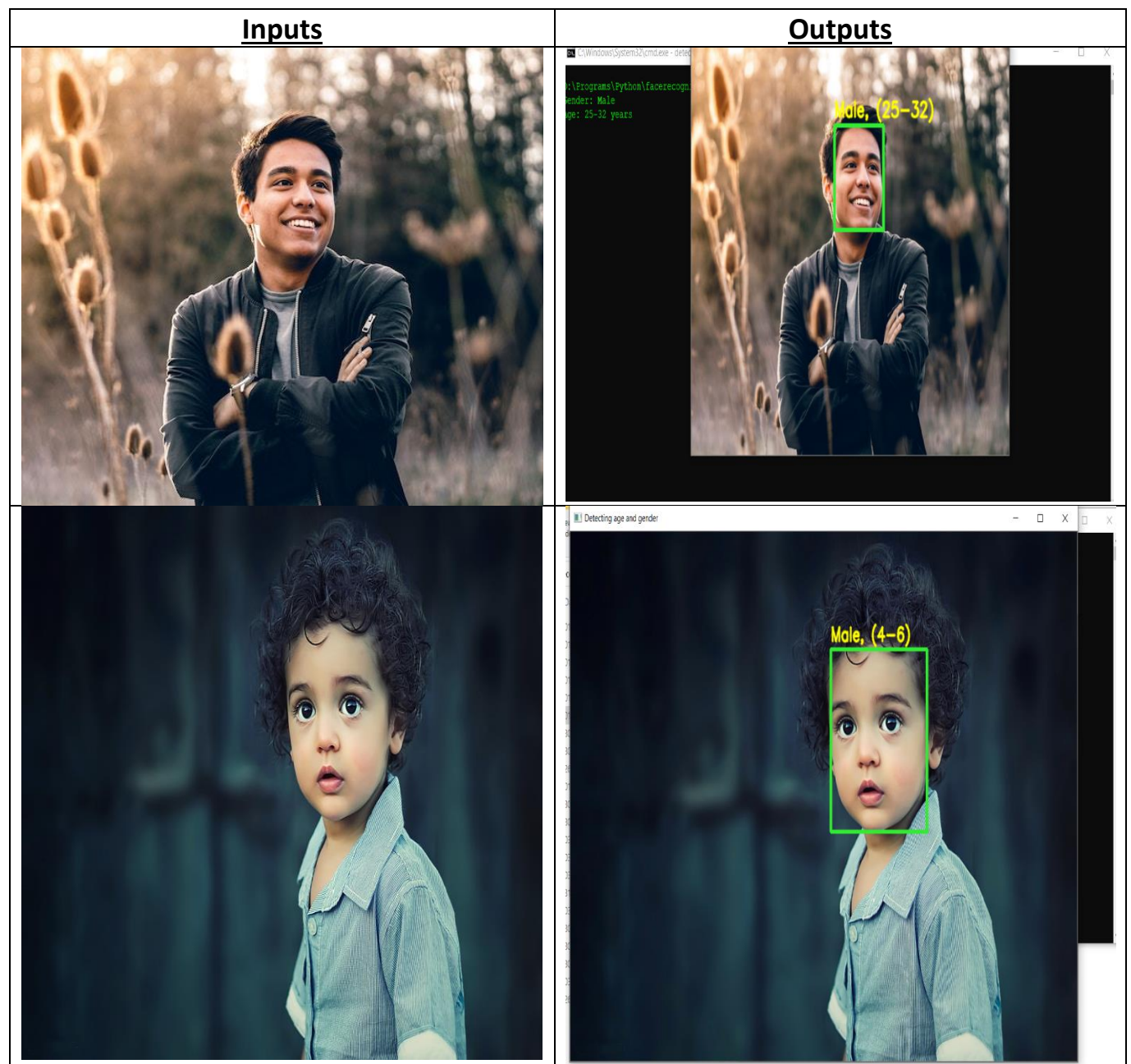
ageNet.setInput(blob)
agePreds=ageNet.forward()
age=ageList[agePreds[0].argmax()]
print(f'Age: {age[1:-1]} years')

cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-
10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE_AA)
cv2.imshow("Detecting age and gender", resultImg)

```


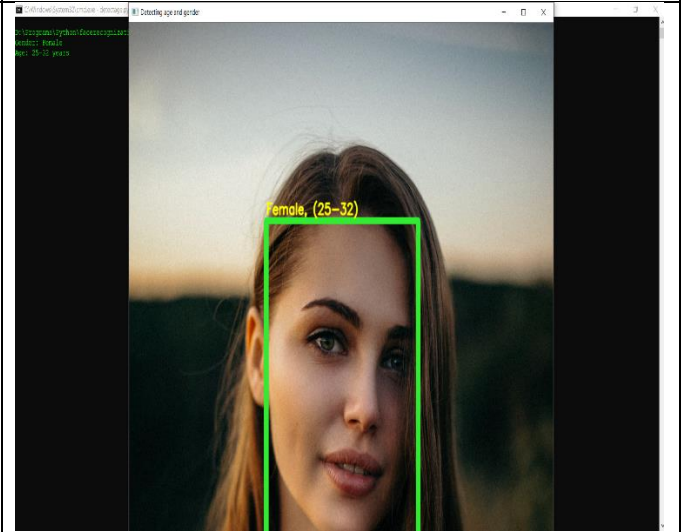

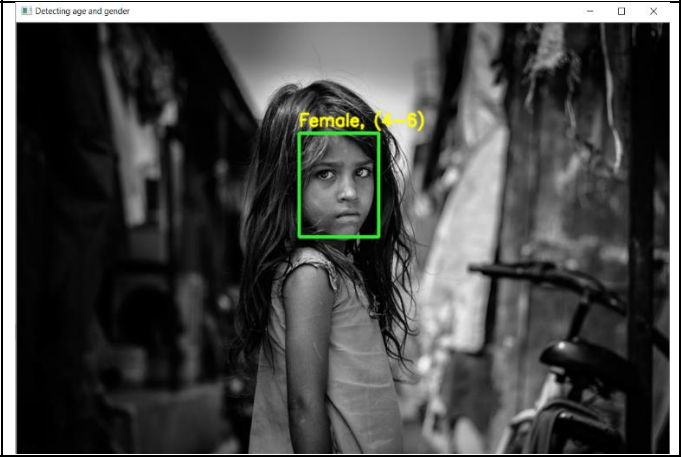
## Outputs

### Male:








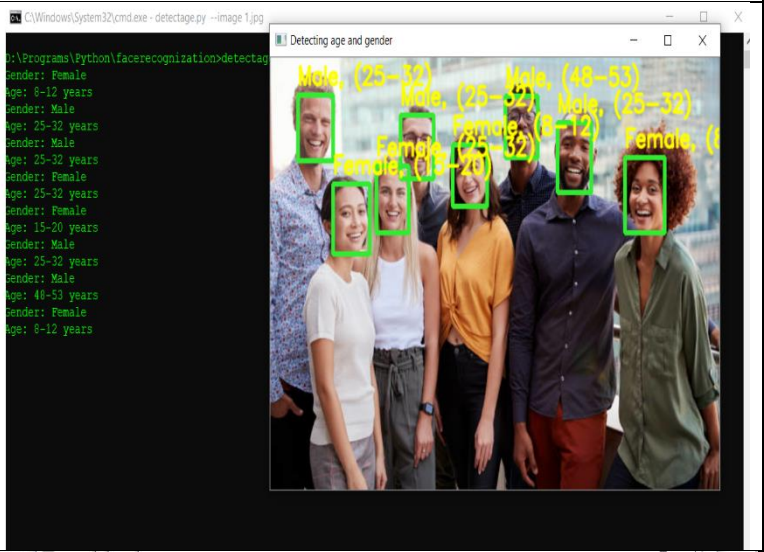

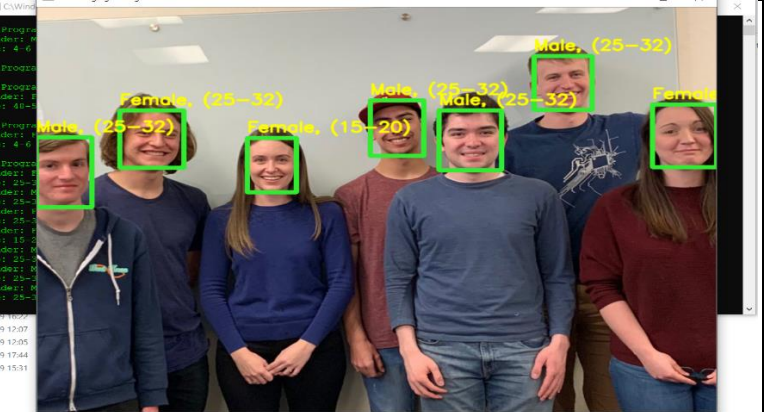
**Female:**

Inputs	Outputs
	
	


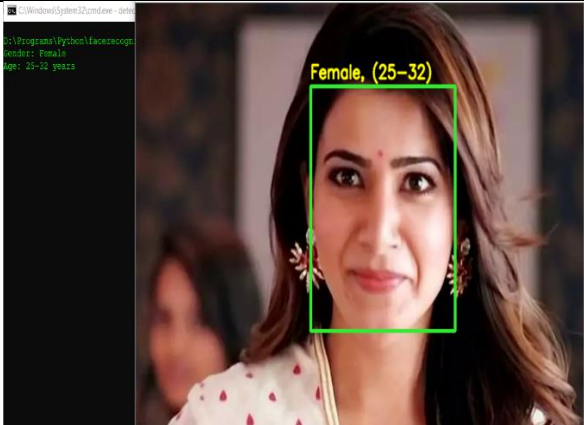

**Animals or Animated:**

Inputs	Output
	No Face Detected
	No Face Detected

Group Photos:

Inputs	Outputs
	
	

Accuracy:

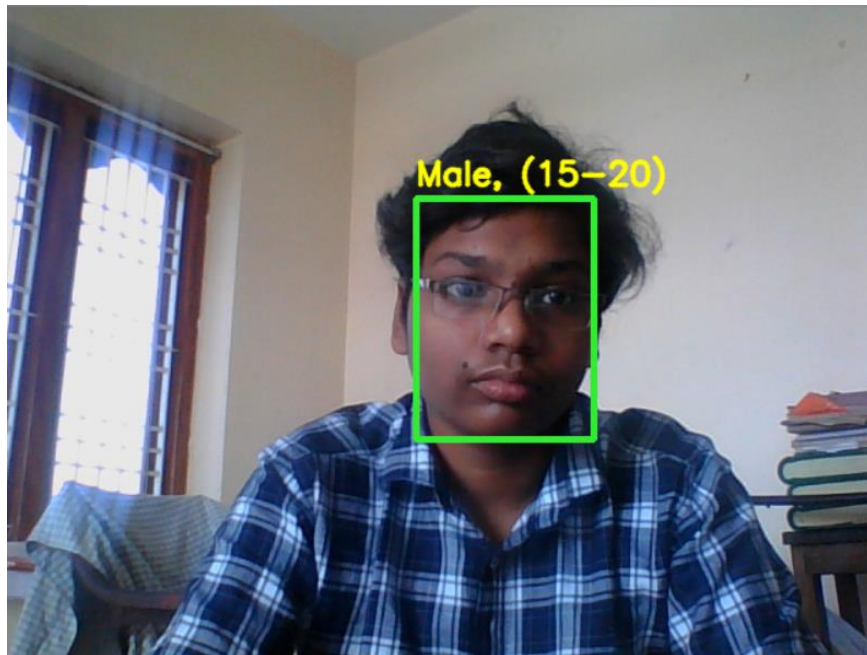
Inputs	Outputs
	
<div><div>Samantha Akkineni / Age</div><div>33 years 28 April 1987</div><div></div></div>	



**Live Webcam Captures:**

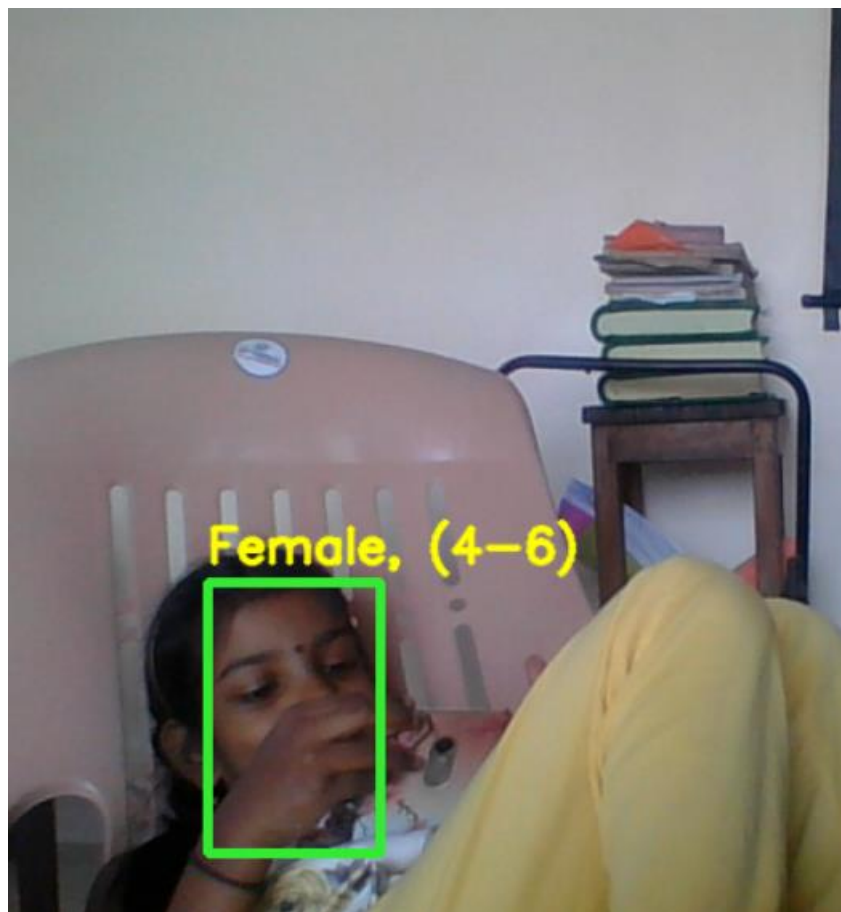
**Original Age: 18 Years**

**Detected Age: 15-20 Years**

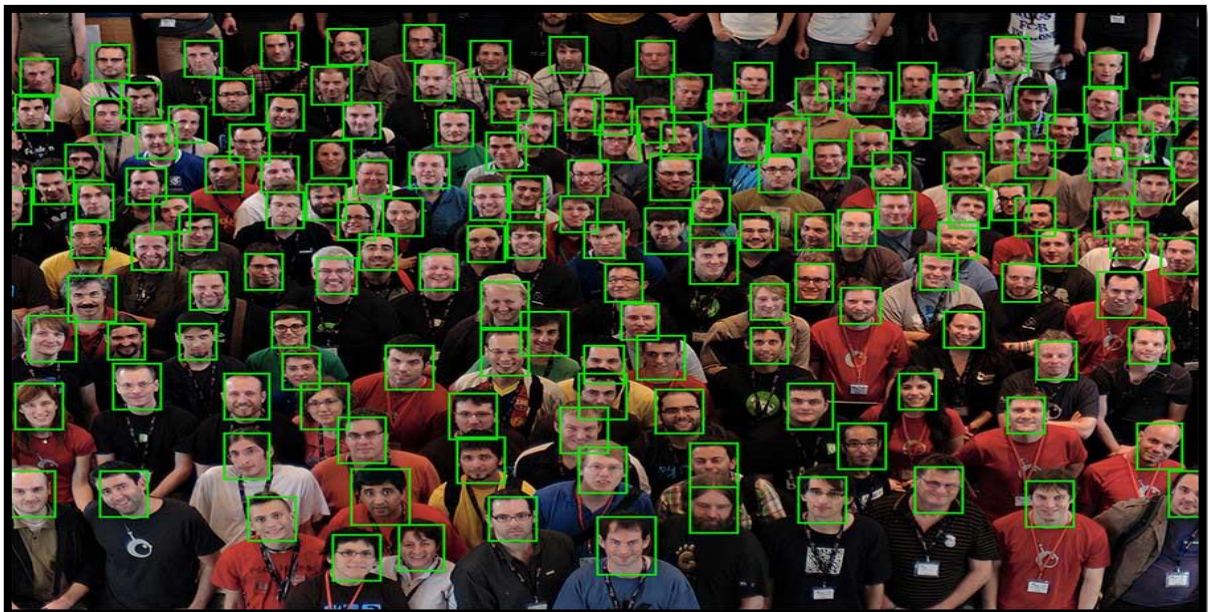
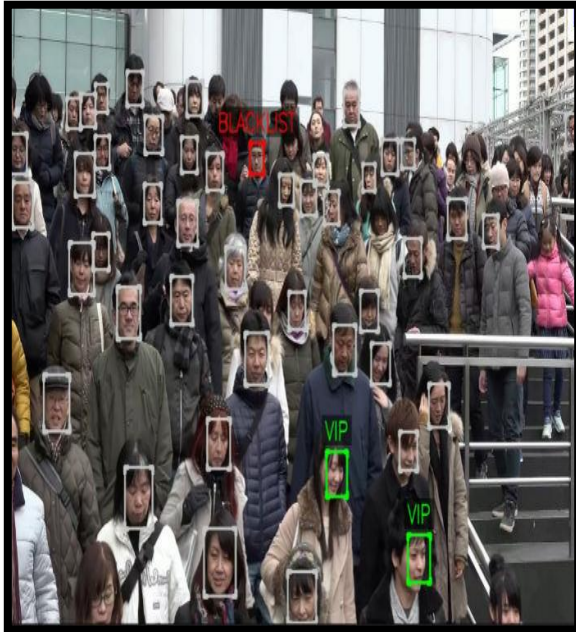


**Original Age: 6 Years**

**Detected Age: 4-6 Years**



## Real life Usage:



**Conclusion:**

We Finally got the end Product “Face Recognition”. This is helpful in finding the age and gender of the person and It was already using in the some real life in some countries. Above are the some example pictures How it was using. Mainly it was used in the cctv cameras. This can be very helpful in finding the criminals.

**References:**

<https://pypi.org/project/face-recognition/>

<https://www.geeksforgeeks.org/>

<https://stackoverflow.com/>

[www.tutorialspoint.com](http://www.tutorialspoint.com)

[www.reddit.com](http://www.reddit.com)

[www.google.co.in](http://www.google.co.in)

[www.quora.com](http://www.quora.com)

<https://realpython.com/face-recognition-with-python/>