SOFT COMPUTING

(CSE-2009)

Project Report

Age and Gender Detection using CNN

Varad Vijay Temghare
(19BCI7045)
Aditya Vikram Lahoti
(19BCE7259)
Shreepad Labshetwar
(19BCE7648)

Submitted to:
Dr. Hari Seetha

Abstract :

Age and gender detection is having numerous applications and is like the building block to various other applications. The number of applications are increasing day by day especially since the rise of social media. There are many ways for age and gender detection but the accuracy is pretty high when CNN (Convolutional Neural Network) is used. In this project I attempt to close the gap between automatic face recognition capabilities and those of age and gender estimation methods.

1) Introduction :

Age and gender play fundamental roles in social interactions in our daily life. There are reserved different salutations and grammar rules for men or women, and different vocabularies are used when addressing elders compared to young people. The ability to automatically estimate the age and gender accurately and reliably from face images is still not as desired for commercial applications.

Age Classification :

The problem of automatically extracting age related attributes from facial images has received increasing attention in recent years and many methods have been put forth for the same. Early methods for age estimation are based on calculating ratios between different measurements of facial features. There are various methods with its pros and cons for the classification of age like Gaussian Mixture Models (GMM), Hidden-Markov-Model, Gabor Image descriptors etc.

Gender Classification :

One of the early methods for gender classification used a neural network trained on a small set of near-frontal face images. The other methods include SVM classifiers, AdaBoost, Webers Local texture Descriptor etc.

CNN Architecture :

CNNs are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification,

medical image analysis, computer vision and natural language processing. There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and
fully-connected (FC) layers. When these layers are stacked, a CNN
architecture will be formed

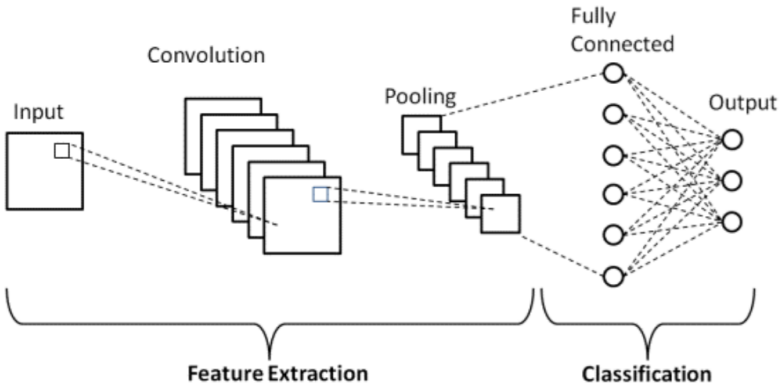


Fig 1. CNN Architecture

2) Literature Review :

1. Basic Convolutional Neural Network :

   A deep Convolutional Neural Network(CNN) is trained on face images to output the gender and label. The architecture in [2] proposed a 3 convolution blocks with varying number of filters and filter sizes. The results obtained for gender was around 86.8% with over sampling of the adience dataset and for age with 1-off error was around 84.7%. These results seem slightly less accurate than SVM from [1] but the advantage of CNNs or rather any deep learning method involves the accuracy to depict a generalised form as long as over fitting is avoided.

2. Cellular and embedded vision applications benefit from efficient models called MobileNets. With a simplified architecture, MobileNets generate lightweight deep neural networks using depthwise separable convolutions. Simple global hyperparameters that balance latency and accuracy are introduced. Given the problem constraints, these

hyper-parameters help model builders select the appropriate model size for their application. It outperforms other popular models on ImageNet classification in comprehensive resource and accuracy tradeoff trials. MobileNets are then demonstrated in a variety of applications and use cases, including object identification, fine-grain classification, facial attributes and large-scale geolocation.

3. Deep learning refers to the shining branch of machine learning that is based on learning levels of representations. Convolutional Neural Networks (CNN) is one kind of deep neural network. It can study concurrently. In this article, we gave a detailed analysis of the process of CNN algorithm both the forward process and back propagation. Then we applied the particular convolutional neural network to implement the typical face recognition problem by Python.

3) Problem Definition and Proposed model :

Automatic prediction of age and gender from face images has drawn a lot of attention recently, due to its wide applications in various facial analysis problems. However, due to the large variations of face images (such as variation in lighting, pose, scale, occlusion), the existing models are still behind the desired accuracy level. There is a need to try and reach the desired accuracy.

Objectives :

● To have a better understanding about CNN

● To reduce the gap between automatic face recognition capabilities and those of age and gender estimation methods.

● To get desired accuracy in the prediction of age and gender

Model :

In this project I will use Deep Learning to identify the gender and age of a person from a single image of a face. The models trained by Tal Hassner and Gil Levi will be used. The predicted gender may be one of 'Male' and 'Female', and the predicted age will be falling in a range. Figure 2 shows the model representation of the project. This project will have three convolutional layers and two fully connected layers and a final output layer (as shown in Fig 3.). With this model first the face will be detected, then classified as Male or Female, after which it will be classified into one of the age groups and then the results will be displayed.

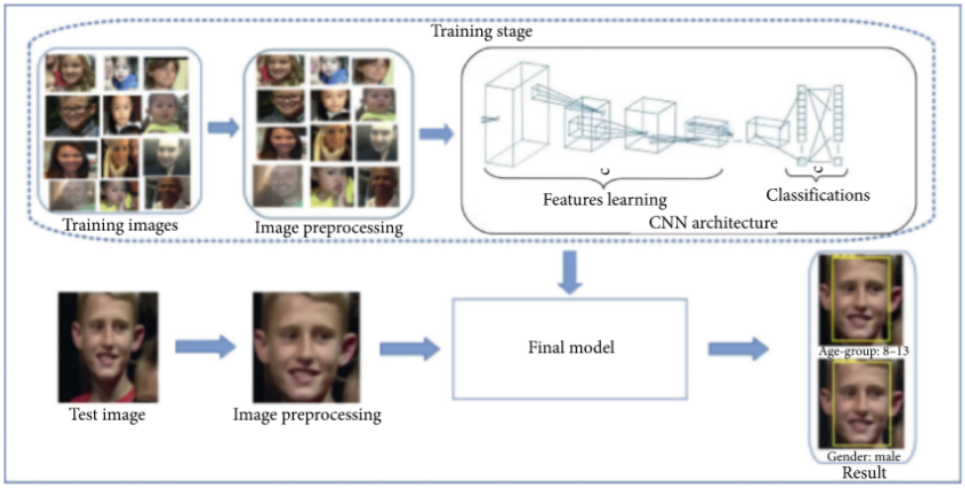


Fig 2. Model representation

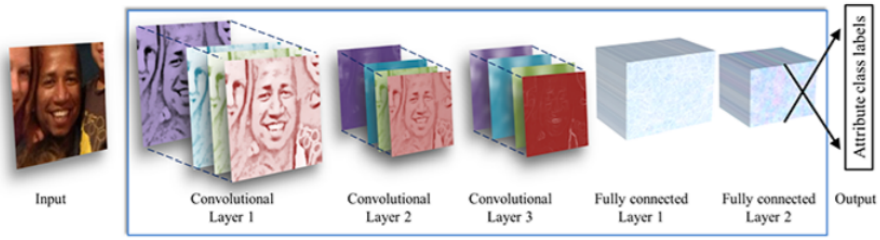


Fig 3. CNN Architecture

4) Implementation :

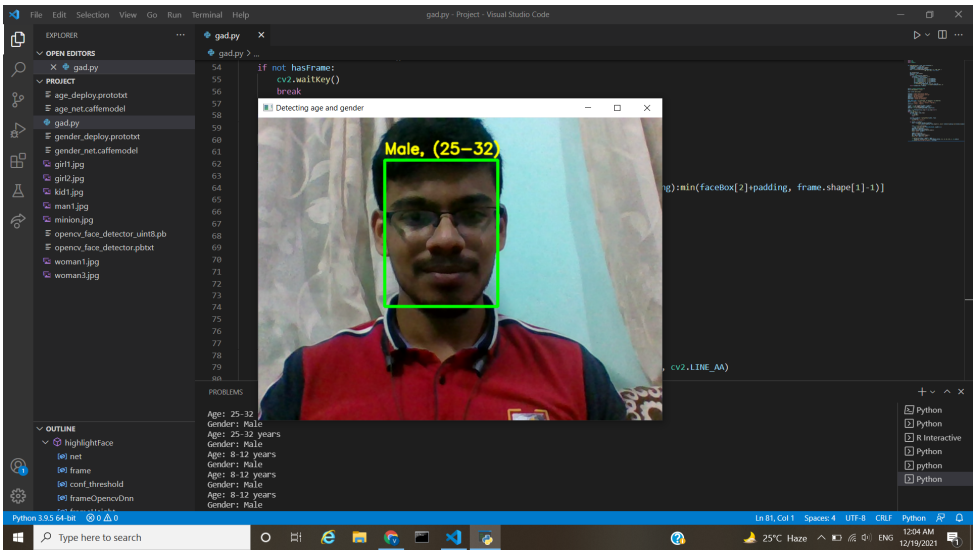
4.1 Hardware Details :

In the project, the software utilises webcam to take live input ie. run time input to predict Gender and Age of the person. Good amount of light is essential to predict the information with great accuracy, lack of it may lead to decrease in accuracy of the model.

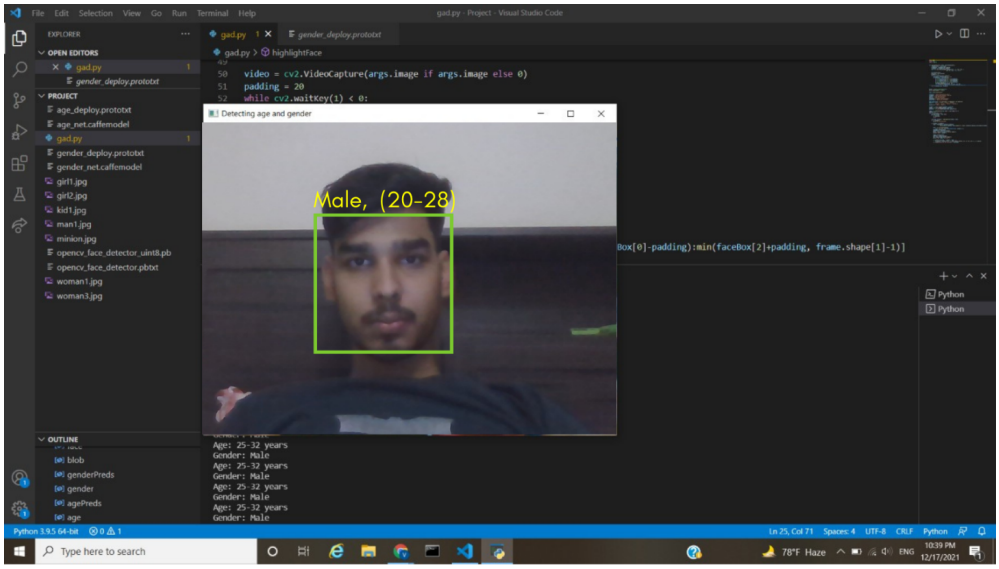
4.2 Software Details :

The software works on CNN algorithms in the python language, where it takes input as live video (in this case webcam), and predicts accurate age and gender of the person facing towards the camera. The software runs exceptionally well in ideal conditions.

4.3 Screenshots :



Screenshot - 1



Screenshot - 2

4.4 Sample code:

```
# Import required modules
import cv2 as cv
import math
import time
from google.colab.patches import cv2_imshow
# import argparse

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()
```

```
bboxes = []
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)
bboxes.append([x1, y1, x2, y2])
cv.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0), int(round(frameHeight/150)), 8)
return frameOpencvDnn, bboxes

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

ageProto = "modelNweight/age_deploy.prototxt"

ageModel = "modelNweight/age_net.caffemodel"

genderProto = "modelNweight/gender_deploy.prototxt"
genderModel = "modelNweight/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']

# Load network
ageNet = cv.dnn.readNet(ageModel, ageProto)
genderNet = cv.dnn.readNet(genderModel, genderProto)
faceNet = cv.dnn.readNet(faceModel, faceProto)

padding = 20

def age_gender_detector(frame):
# Read frame
t = time.time()
frameFace, bboxes = getFaceBox(faceNet, frame)
```

```
for bbox in bboxes:
# print(bbox)
face =
frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[0]-1),max(0,b
box[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 Output : {}".format(genderPreds))
                                    print("Gender : {}, conf = {:.3f}".format(gender,
genderPreds[0].max()))

ageNet.setInput(blob)
agePreds = ageNet.forward()
age = ageList[agePreds[0].argmax()]
print("Age Output : {}".format(agePreds))
                                                print("Age : {}, conf = {:.3f}".format(age, agePreds[0].max()))
label = "{},{}".format(gender, age)
                                    cv.putText(frameFace, label, (bbox[0], bbox[1]-10),
cv.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 255), 2, cv.LINE_AA) return
frameFace
input = cv.imread("image3.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
input = cv.imread("image6.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
input = cv.imread("image2.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
import cv2
cap = cv2.VideoCapture('video.mp4')
ret, frame = cap.read()
frame_height, frame_width, _ = frame.shape
```

```
out =
cv2.VideoWriter('output.avi',cv2.VideoWriter_fourcc('M','J','P','G'), 10,
(frame_width,frame_height))
print("Processing Video...")
while cap.isOpened():
ret, frame = cap.read()
if not ret:
out.release()
break
output = age_gender_detector(frame)
out.write(output)
out.release()
print("Done processing video")
```

Methodology :

The network comprises only three convolutional layers and two fully-connected layers with a small number of neurons. All three color channels are processed directly by the network. Images are first rescaled to 256 × 256 and a crop of 227 × 227 is fed to the network. The three subsequent convolutional layers are then defined as follows:

1. 96 filters of size 3×7×7 pixels are applied to the input in the first convolutional layer, followed by a rectified linear operator (ReLU), a max pooling layer taking the maximal value of 3 × 3 regions
2. The 96 × 28 × 28 output of the previous layer is then processed by the second convolutional layer, containing 256 filters of size 96 × 5 × 5 pixels.
3. Finally, the third and last convolutional layer operates on the 256 × 14 × 14 blob by applying a set of 384 filters of size 256 × 3 × 3 pixels, followed by ReLU and a max pooling layer.
4. Finally, the output of the last fully connected layer is fed to a soft-max layer that assigns a probability for each class. The prediction itself is made by taking the class with the maximal probability for the given test image.

Applications :

Age and gender detection has many significant real-world applications. Computer-based age estimation is useful in situations where one's age is to be determined. There are several application areas for age estimation including the following:

1. **Age simulation :**

Characterization of facial appearance at different ages could be effectively used in simulating or modeling one's age at a particular point in time. Estimated ages at different times could help in learning the aging pattern of an individual, which could assist in simulating facial appearance of the individual at some unseen age.

2. **Security and Surveillance :**

Age estimation can be used in surveillance and monitoring of alcohol and cigarette vending machines and bars for preventing minors from accessing alcoholic drinks and cigarettes and restricting children access to adult websites and movies. Age estimation can also be used in health-care systems like robotic nurses and doctors expert systems for customized medical services.

3. **Biometrics :**

Age and gender detection can be applied in age-invariant face recognition , iris recognition, hand geometry recognition, and fingerprint recognition in order to improve accuracy of hard (primary) biometric system

4. **Employment :**

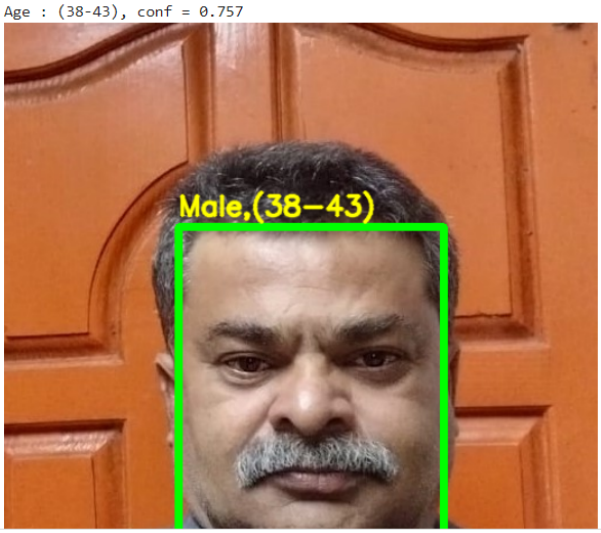
Some government employment like the military and police consider one's age as a requirement. Age estimation systems could be used to determine age of the recruits during the recruitment process.

5) Results and discussion:

The images with proper viewing angle, lighting, no filter, proper resolution provide a good accuracy. This implies that better alignment may provide an additional boost in performance. There are instances of gender and age misclassification. These show that many of the mistakes made by the algorithm are mostly due to challenging viewing conditions of some of the images. Most notable are mistakes caused by blur or low resolution. Gender estimation mistakes also frequently occur for images of babies or very young children where obvious gender attributes are not yet visible. Additionally, the algorithm can also be used for recorded videos with an additional code snippet which has also been added at the end of the report.

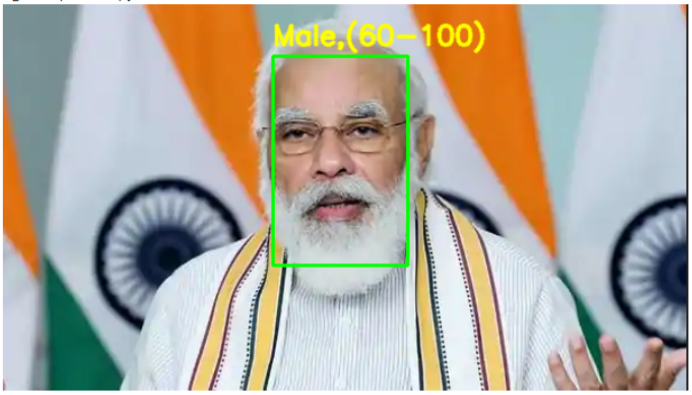
```
input = cv.imread("/content/AJIT KAKA.jpeg")
output = age_gender_detector(input)
cv2_imshow(output)
```

```
Gender : Male, conf = 1.000
Age Output : [[6.4095017e-05 5.8928083e-05 7.3670375e-04 4.2998563e-02 1.9395812e-01
  7.5662148e-01 4.3375259e-03 1.2244957e-03]]
Age : (38-43), conf = 0.757
```



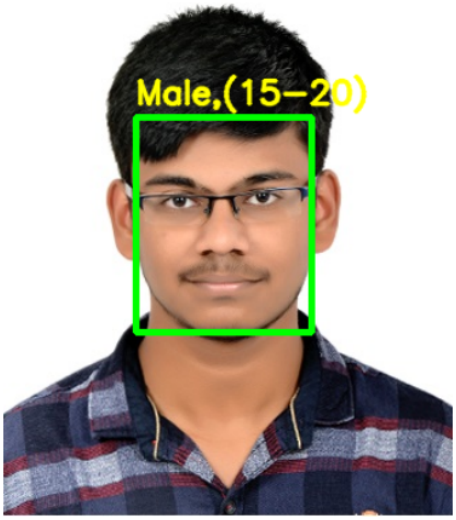
```
input = cv.imread("image2.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
```

```
Gender : Male, conf = 1.000
Age Output : [[1.9950876e-03 5.0318660e-05 8.4574794e-04 1.4283879e-04 7.5796200e-04
  1.0425225e-01 4.2547666e-02 8.4940809e-01]]
Age : (60-100), conf = 0.849
```




```
input = cv.imread("/content/varad pic1.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
```

```
Gender : Male, conf = 1.000
Age Output : [[1.6592719e-05 3.5802153e-05 8.7093245e-03 8.8398057e-01 5.5915505e-02
  5.0943002e-02 3.1760318e-04 8.1571881e-05]]
Age : (15-20), conf = 0.884
```



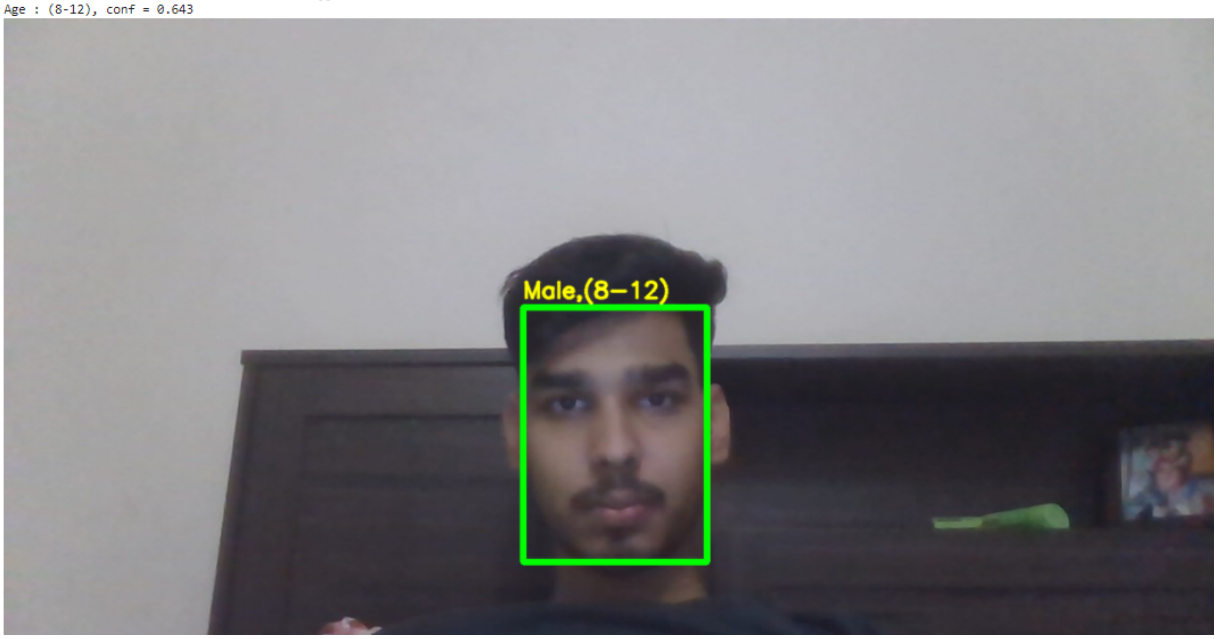
```
[14] input = cv.imread("/content/sdfghjs.jpg")
     output = age_gender_detector(input)
     cv2_imshow(output)
```

```
Gender : Female, conf = 0.970
Age Output : [[7.1143234e-08 3.7007783e-06 1.4115403e-03 6.0432497e-03 9.8783714e-01
  4.6808794e-03 1.4902158e-05 8.4661551e-06]]
Age : (25-32), conf = 0.988
Gender : Female, conf = 0.999
Age Output : [[1.5495528e-08 3.0875523e-07 1.2920689e-06 1.3812208e-04 9.9896026e-01
  8.4707711e-04 4.8211150e-05 4.7517206e-06]]
Age : (25-32), conf = 0.999
```

✓ 0s completed at 1:28 AM

```
input = cv.imread("/content/WIN_20211218_23_21_42_Pro.jpg")
output = age_gender_detector(input)
cv2_imshow(output)
```

```
Gender : Male, conf = 1.000
Age Output : [[6.9791800e-05 2.2114657e-03 6.4295834e-01 2.9472607e-01 5.1613614e-02
  7.6142680e-03 7.7064452e-04 3.5791963e-05]]
Age : (8-12), conf = 0.643
```



6) Conclusion :

Though many previous methods have addressed the problems of age and gender classification, this algorithm is a simple one which works just as efficiently. The easy availability of image collections provides modern machine learning based systems with lots of training data, though this data is not always suitably labeled for supervised learning. Two important conclusions can be made from our results. First, CNN can be used to provide improved age and gender classification results, even considering the much smaller size of contemporary image sets labeled for age and gender. Second, the simplicity of this model implies that more elaborate systems using more training data may well be capable of substantially improving results beyond those reported here.

7) Scope for future :

Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when compared to the tremendous leaps in performance recently reported for the related task of face recognition.

8)References :

● Naveen K B , Dr. Vimuktha Evangeleen Salis, 2019, Survey of Techniques to Estimate the Age and Gender of A Person using Face Images, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 08, Issue 06 (June 2019)

●MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam

●Implementation of Training Convolutional Neural Networks : Tianyi Liu, Shuangsang Fang, Yuehui Zhao, Peng Wang, Jun Zhang
https://arxiv.org/abs/1506.01195v2

● T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. Proc. Conf. Comput. Vision Pattern Recognition, 2015.

● Trainable Convolution Filters and Their Application to Face Recognition Ritwik Kumar , Arunava Banerjee , Baba C. Vemuri , Hanspeter Pfister