

Face, Age and Genre Detection

By Chriss Jordan Oboa

What is Face, Age and Genre Detection about?

It is a program which is capable to take an image of a person, understand where the face of the person and then pass it through a Machine learning and Deep learning algorithm to detect the gender and the approximate age of that person; it's as simple as that.

What can the Face, Age and Genre Detection be used For?

This project is not only used for fun but can also be used to enhance safety across the world. Those features as Age and Gender classification can be helpful in several real-life situation including security and video surveillance, electronic customer relationship management, biometrics, electronic vending machines, human-computer interaction, entertainment, cosmetology, forensic art, and more.

Features mostly needed for this project

Machine learning and Deep Learning are involved in this project.

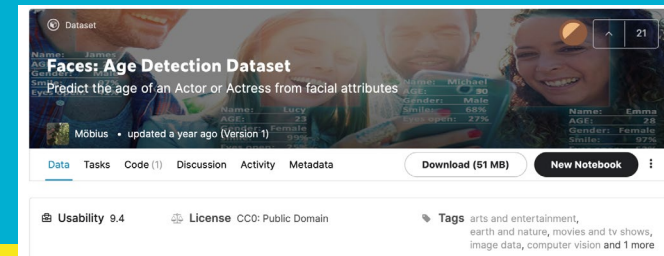
- This project uses OpenCV
- A function named Blob (in which I will give more details in the coming slides)
- Convolutional Neural Network
- A solid Dataset

Problems I ran into

The first issue we ran into was based on the training the model via CNN in which we couldn't have satisfied and efficient enough results to fully accommodate it to the detection part of the project. This project was already a failure based on the results I got when training. The max results was 76% when adding more layers.

As result, we opt to use a great alternative suggestion from Misbah Mohammed, in which and send me a link of his youtube video where he talked about pre-trained model and how to use them.

The second issue was based on the efficacy of detecting the face in which was a little bit off in some cases. We fixed it via tweaking the blob



What are pre-trained models?

Pre-trained models are models developed, deployed and made by researchers.

They are shipped through different types of images which they themselves collected and they also had their own database.

Those images collected are completely different. They had a different brightness, different profile, were taken on a different day, different saturation, different person in which make those used images have different property level.

How did I execute this project?

Three steps were required to fully execute this project:

- Face reduction
- Pre-processing stage
- Age and Gender detection

Execution: Step 1

Step one is simply doing the face reduction. This step is reading the image and passing it to the face detectors. Once the face is detected and there's a small pre-processing stage in which localize and lock the face in to start pre-processing.

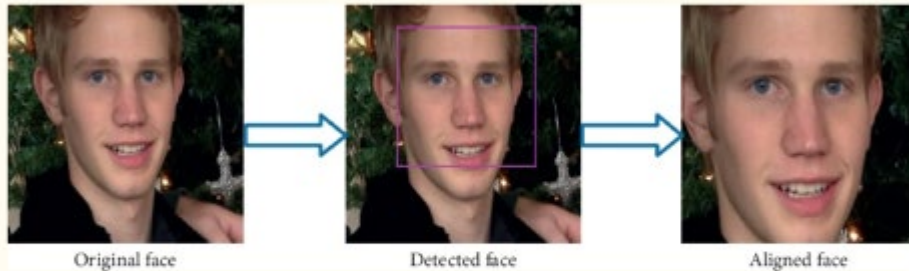


Figure 2

The image preprocessing phase.

Execution: Step 2

Step two is the pre-processing stage: some sort of processing on the image and once that is done then you're sending it to the detector module which is going to detect the age and gender of the person.

This step uses the function called **blob** that I referred on the third slide.

The **blob** from image just brings that image into that basic property so that when you send it to the gender or the age detection model it will give you the right output without much error.

```
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)
```

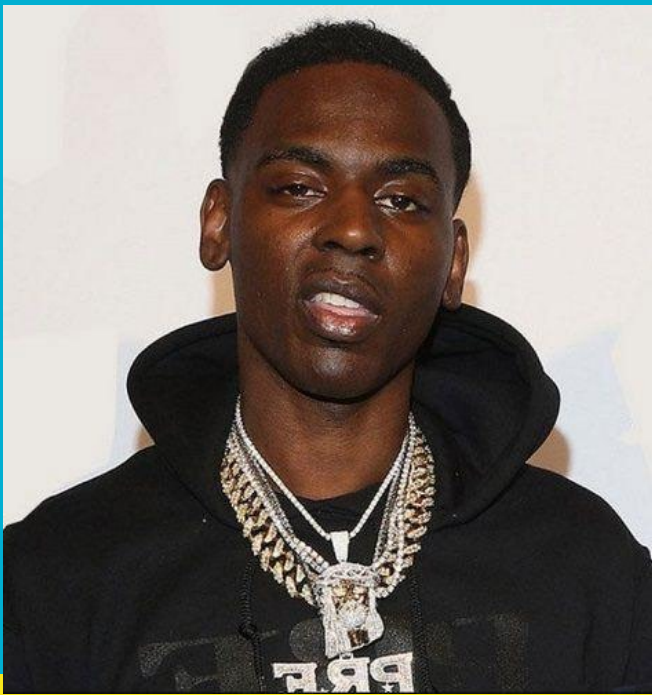
Execution: Step 3

This step is the part where Age and Gender detection received the pre-processed image, then run it into the age and detector frame in which uses the pretrained model based on Convolution Neural Network made by researchers, provided to us by Misbah Mohammed.

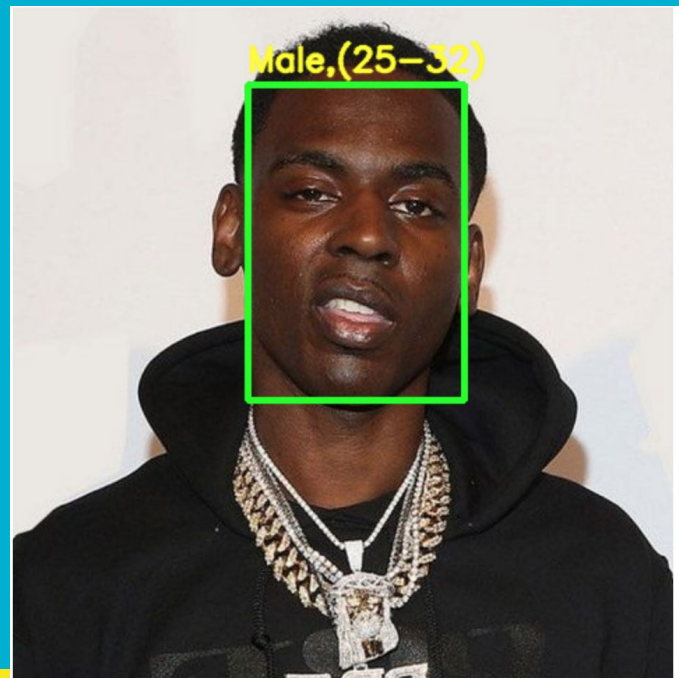
```
[ ] input = cv.imread("image4.jpg")  
    output = age_gender_detector(input)  
    cv2_imshow(output)
```

Results

Before



After

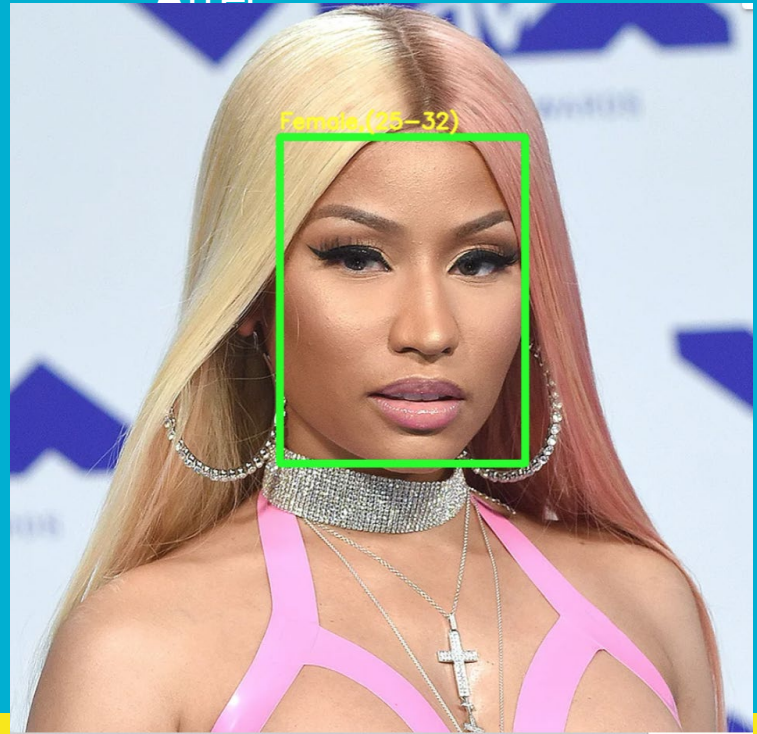


Results

Before

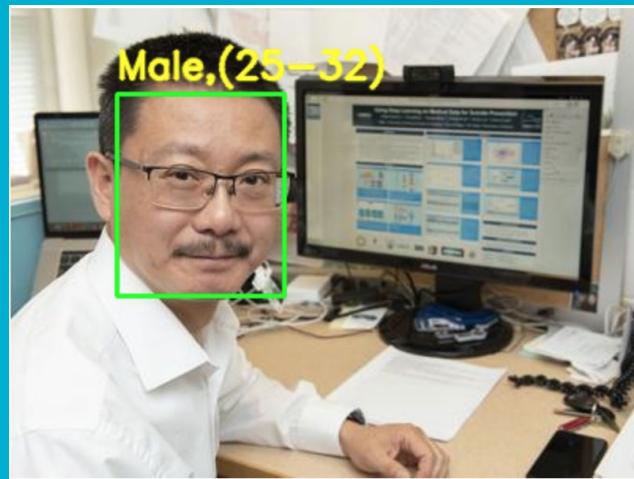


After



Improvement needed

Some of the result were not quite satisfying and are often due to the image quality or even the angle and the shadow of the images in which tend to affect the final results when detecting.

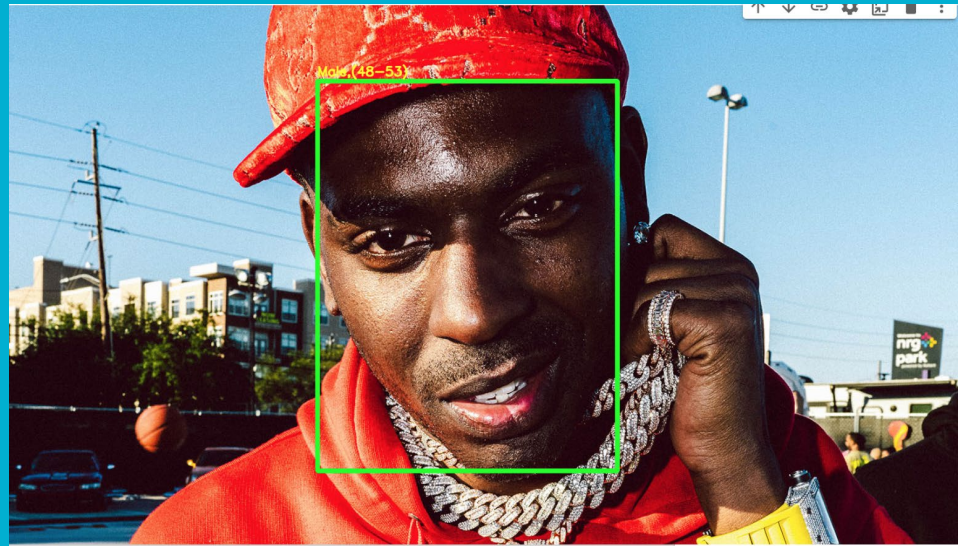


One more example

Before



After



Another way to improve the accuracy

A better accuracy can be also achieve by tweaking the size highlighted in the image below.

```
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, bbox[0]-padding), min(bbox[2]+padding, frame.shape[1]-1), min(bbox[0]+padding, frame.shape[0]-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)  
        agePreds = ageNet.forward()  
        age = ageList[agePreds[0].argmax()]
```

For videos

Unprocessed videos

<https://youtu.be/cyKbISwOMck>

<https://youtu.be/9SkTR2nrQ6Q>

<https://youtu.be/LLGWRtoK1Xs>

Processed videos

<https://www.youtube.com/watch?v=bEmZr0OixBA>

<https://www.youtube.com/watch?v=w35K7P77gxU>

<https://www.youtube.com/watch?v=Eps8HoSNc1A>