**A PYTHON GENDER AND AGE DETECTION PROGRAM**

We tackled the problem of classifying a person’s age and gender based on an image. Potential uses of this application include uses in dating sites and social media, access to age-restricted content and for checkout at grocery stores when buying alcohol at the self-checkout.

To build a gender and age detector that can approximately guess the gender and age of the person (face) in a picture using [***Deep Learning***](https://data-flair.training/blogs/deep-learning/) on the Adience dataset.

Adience dataset, this dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It has a total of 26,580 photos of 2,284 subjects in eight age ranges (as mentioned above) and is about 1GB in size.

OpenCV is a library of programming functions mainly aimed at real-time computer vision.

We use the argparse library to create an argument parser so we can get the image argument from the command prompt. We make it parse the argument holding the path to the image to classify gender and age for.

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.

Once you start treating age prediction as a regression problem, it becomes significantly harder for a model to accurately predict a single value representing that person’s image.

However, if you treat it as a classification problem, defining buckets/age brackets for the model, our age predictor model becomes easier to train, often yielding substantially higher accuracy than regression-based prediction alone.

**The process of visual age prediction is difficult, and I’d consider it subjective when either a computer or a person tries to guess someone’s age.**

In order to evaluate an age detector, you cannot rely on the person’s actual age. Instead, you need to measure the accuracy between the predicted age and the perceived age.

**CODE:**

#A Gender and Age Detection program by Kritika Jain and Mudrika Kapoor

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), (0,255,0), int(round(frameHeight/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)

**OUTPUT:**



