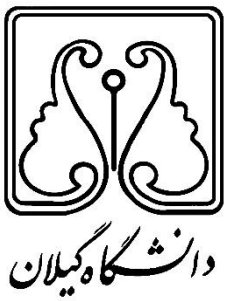

Real-time prediction of age and gender based on ANN

Nikzad, Shadrouy, Motallebi, Nouri

Mr.tourani

Winter 2019

۱- مقدمه

سن و جنس، دو مورد از ویژگی های مهم چهره، نقش بسیار اساسی در تعاملات اجتماعی دارند، شناسایی سن و برآورد جنسیت از تصویر یک چهره؛ یک کار مهم در برنامه های هوشمند مانند کنترل دسترسی، تعامل رایانه و انسان، اجرای قانون، بازاریابی می کند.

۲- پیاده سازی

قدم اول- Face detection with Haar cascades

تشخیص اجسام با استفاده از الگوریتم Haar Cascade ، یک روش بسیار موثر است که اولین بار توسط Paul Viola و Michael Jones در سال ۲۰۰۱ مطرح شد. این الگوریتم بر پایه یادگیری ماشین بوده و بدین صورت عمل می کند که با استفاده از تصاویری که چهره در آنها وجود دارد (تصاویر مثبت) و تصاویری که چهره در آنها وجود ندارد (تصاویر منفی) تابعی را آموزش می دهد تا بتواند مولفه هایی مانند وجود فاصله بین چشم ها را پیدا کند. این الگوریتم منحصر به تشخیص چهره نیست؛ و بیشتر یک آموزش دهنده است و می توان با آموزش دادن تابع دلخواه خودتان هر شی مانند ماشین، میز، مداد و ... را شناسایی کنید .OpenCV مجموعه ای از توابع از پیش آموزش دیده را برای تشخیص چهره، چشمان، لبخند و ... در خود دارد که می توان به سادگی از آنها استفاده کرد.

ابتدا پکیج cv2 را برای OpenCV وارد می کنیم. سپس توابع از پیش آموزش دیده را برای شناسایی صورت و چشم ها بارگذاری می کنیم. این توابع در مسیر opencv/data/haarcascades قرار دارند.

```
def loadPreBuiltFacialDet():
    return openCV.CascadeClassifier('data/haarcascade_frontalface_alt.xml')
```

در قسمت بعدی چون می خواهیم عمل تشخیص چهره را به صورت Real-time انجام دهیم، از دوربین استفاده می کنیم. برای این منظور دوربین را فراخوانی می کنیم و کد را درون یک حلقه می نویسیم.

توجه داشته باشید که برای استفاده از اگوریتمhaar cascade ، حتماً باید تصویر به صورت سیاه و سفید باشد. به همین دلیل هر فریم را پس از دریافت از دوربین از حالت رنگی به سیاه و سفید تبدیل می کنیم.

```
def imageToGray(image):
    return openCV.cvtColor(image, openCV.COLOR_BGR2GRAY)
```

در مرحله بعدی، با استفاده از دستور MultiScale، تابع تشخیص چهره را به تصویر سیاه و سفید خود اعمال می کنیم . اعدادی که در این دستور آمده اند، محدوده اندازه را تعیید می کنند. این تابع یک خروجی با مختصات محل صورت می دهد. سپس با استفاده از خروجی این تابع با یک مستطیل محدوده صورت را نشان می دهیم.

قدم دوم– Gender Recognition with CNN

شناسایی جنسیت با استفاده از امکان های نمایش OpenCV بسیار مشهور است و برخی از شما ممکن است در مورد آن نیز کار کرده یا خوانده باشید. اما، در این مثال، ما از روش دیگری برای شناخت جنسیت استفاده خواهم کرد. در این مثال از مدل های CNN استفاده کرده ایم. ما می خواهیم از بسته dnn OpenCV که مخفف "شبکه های عصبی عمیق" است استفاده کنیم.

در بسته dnn، OpenCV کلاسی به نام Net ارائه داده است که می تواند برای جمع آوری یک شبکه عصبی مورد استفاده قرار گیرد. علاوه بر این، این بسته ها از وارد کردن مدل های شبکه عصبی از چارچوب های یادگیری عمیق شناخته شده مانند caffe، tensorflowپشتیبانی می کنند. محققان مدل های CNN خود را به عنوان مدل کافه (caffe models) منتشر کرده اند .بنابراین، ما با استفاده از CaffeImporter آن مدل را وارد برنامه خود خواهیم کرد

قدم سوم – Age Recognition with CNN

این تقریباً به قسمت تشخیص جنسیت شبیه است به جز اینکه پرونده مربوط به نمونه اولیه و پرونده مدل کافه "deploy_agenet.prototxt" و "age_net.caffemodel" هستند. علاوه بر این، لایه خروجی CNN (لایه احتمال) در این CNN شامل ۹ مقدار برای ۹ کلاس سنی است.

```
ageList = ['(0, 4)', '(4, 8)', '(8, 15)', '(15, 21)', '(21, 24)',
        '(25, 32)', '(38, 43)', '(48, 53)', '(60, 100)']
```

یک caffe model دارای دو فایل به هم مرتبط هست :

۱- prototxt

تعریف CNN در اینجا آمده است. این پرونده لایه های شبکه عصبی، ورودی ها، خروجی ها و عملکرد هر لایه را تعریف می کند.

۲- caffemodel

این شامل اطلاعات شبکه عصبی آموزش دیده (مدل آموزش دیده) است.

۳- توضیحات کد

قسمت اول- import کردن کتابخانه های مورد نیاز

```
from tkinter import *
import cv2 as openCV
import imutils
import time
import numpy as np
```

قسمت دوم- برای گرفتن ویدیو باید یک شی برای گرفتن ویدیو درست کنیم. الان به عنوان مثال آرگومان را صفر می گذاریم چون میخواهیم از دوربین device استفاده کنیم.

```
capture = openCV.VideoCapture(0)
```

قسمت سوم- ۳ لیست جداگانه برای ذخیره سازی سن و جنسیت

```
MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
genderList = ['Male', 'Female']
gender_list2 = ['Female', 'Female']
ageList = ['(0, 4)', '(4, 8)', '(8, 15)', '(15, 21)', '(21, 24)',
        '(25, 32)', '(38, 43)', '(48, 53)', '(60, 100)']
```

قسمت چهارم- با استفاده از این قسمت ما طول و عرض فیلم خود را تعیین می کنیم.

```
def videoType(a):
    if a == 0:
        capture = openCV.VideoCapture(0)
        capture.set(3, 480)  # set width
        capture.set(4, 640)  # set height
```

قسمت پنجم- این قسمت برای شناسایی face در ویدیو می باشد.

```
def faceDetect(f_cascade ,image, scaleFactor, minNeighbors):
    return f_cascade.detectMultiScale(image, scaleFactor, minNeighbors)
```

آرگومان ها به این صورت می باشند :
Image : اولین ورودی تصویر ماست که به سیاه و سفید تبدیل شده است.
scaleFactor : این عملکرد درک غلط از اندازه را جبران می کند . وقتی یک چهره ظاهراً بزرگتر از دیگری است به دلیل نزدیک تر بودن به دوربین است.
minNeighbors : الگوریتم شناسایی است که از یک پنجره متحرک برای ردیابی اشیاء استفاده می کند. کار آن این است که تعداد اشیایی که در نزدیکی دوربین هستند را، قبل از اینکه بتواند صورت یافت شده را پیدا کند، مشخص می کند.

قسمت ششم- داخل لیست چهره ها می چرخیم و مستطیل ها را بر روی صورت های انسان در این فیلم می کشیم. در اینجا اساساً در حال یافتن چهره، شکستن صورت، اندازه آنها و مستطیل کشیدن دور آن ها هستیم.

```
ef getFaces(faces, image, font):
    for (x, y, w, h)in faces:
        openCV.rectangle(image, (x, y), (x+w, y+h), (255, 255, 0), 2)
        faceImage = image[y:y+h, h:h+w].copy()
```

قسمت هفتم-openCV تابعی را برای تسهیل پردازش تصویر برای طبقه بندی یادگیری عمیق فراهم می کند:
Image : این تصویر ورودی است که می خواهیم قبل از انتقال آن از طریق شبکه عصبی عمیق ما برای طبقه بندی، پردازش کنیم. دارای ۴ ویژگی زیر است.

scale factor : پس از انجام تفریق متوسط ، می توانیم بصورت اختیاری تصاویر خود را براساس فاکتور مقیاس بندی کنیم. این مقدار به مقدار ۱.۰ (به عنوان مثال، بدون مقیاس) پیش فرض است اما می توانیم مقدار دیگری نیز ارائه دهیم.

size : در اینجا ما اندازه فضایی را که از شبکه عصبی Convolutional انتظار دارد ارائه می دهیم. برای اکثر شبکه های عصبی فعلی این حالت ۲۲۴× ۲۲۴، ۲۲۷× ۲۲۷ یا ۲۹۹ × ۲۹۹ است.

mean : اینها مقادیر میانگین تفریق ما هستند. آنها می توانند یک سه ضلعی از RGB باشند یا می توانند یک مقدار واحد باشند که در این صورت مقدار عرضه شده از هر کانال تصویر کم می شود. اگر تفریق متوسط را انجام می دهید، اطمینان حاصل کنید که ترتیب ۳ تاپل را در(R ، G، B) مرتب باشد. به خصوص هنگام استفاده از رفتار پیش فرض.swapRB = True

swapRB: openCV فرض می کند که تصاویر به ترتیب کانال BGR هستند. با این حال، میانگین مقدار فرض می کند که ما از دستور RGB استفاده می کنیم. برای برطرف کردن این اختلاف می توانیم با تنظیم این مقدار درTrue ، کانال های R و B را در تصویر جابجا کنیم. به طور پیش فرض، OpenCV این تعویض کانال را برای ما انجام می دهد .

```
blob = openCV.dnn.blobFromImage(faceImage, 1, (227, 227), MODEL_MEAN_VALUES, swap
RB=False)
```

قسمت هشتم- نشخیص سن و جنسیت

```
        genderNet.setInput(blob)
        genderPredicion = genderNet.forward()
        gender = genderList[genderPredicion[0].argmax()]
        print("Gender : " + gender)

        ageNet.setInput(blob)
        agePredictions = ageNet.forward()
        age = ageList[agePredictions[0].argmax()]
        print("Age Range: " + age)
```

قسمت نهم- حال باید متن را با استفاده از ماژول() putText که در openCV موجود است؛ در قاب خروجی خود قرار دهیم.

```
overlay_text = "%s %s" % (gender, age)
        openCV.putText(image, overlay_text, (x, y), font, 1, (255, 255, 255), 2,
openCV.LINE_AA)

    openCV.imshow('detection', image)
```

قسمت دهم- تابعی تعریف کردیم برای بارگذاری caffemodel و prototxt روی هر دو ویژگی age detector و gender detector

اینها اساساً مدلهای CNN از قبل آموزش دیده هستند که تشخیص را انجام می دهند.

```
def caffeModelInitialize():

    ageNet = openCV.dnn.readNetFromCaffe(
        'data/deploy_age.prototxt',
        'data/age_net.caffemodel')

    genderNet = openCV.dnn.readNetFromCaffe(
        'data/deploy_gender.prototxt',
        'data/gender_net.caffemodel')

    return(ageNet, genderNet)
```

قسمت یازدهم- این قسمت تابع های تعریف شده قرار داده شده است، که در کامنت ها توضیح داده شده اند.

```
def read_from_camera(ageNet, genderNet):
    font = openCV.FONT_HERSHEY_SIMPLEX

    while True:
        #check wheather the capture read was true or false
        ret, image = capture.read()

        #Load the pre-built model for facial detection.
        f_cascade = loadPreBuiltFacialDet()

        #Convert the image to gray image as OpenCV face detector expects gray ima
ges.
        gray = imageToGray(image)

        #Function to detect objects(faces)
        faces = faceDetect(f_cascade, gray, 1.1, 5)

        if(len(faces) > 0):
            print("Found {} faces".format(str(len(faces))))

        getFaces(faces, image, font)
```

```
#0xFF is a hexadecimal constant which is 11111111 in binary.
        if openCV.waitKey(1) & 0xFF == ord('q'):
            break
```

در آخر هم if زیر را تعریف کرده ایم تا برنامه در طول اجرا بسته نشود.

قسمت دوازدهم- این قسمت برای رابط کاربری برنامه و مربوط به کتابخانه ی tkinter است که در ابتدای برنامه import کردیم.

```
root = Tk()
    menu = Menu(root)
    root.config(menu=menu)
    filemenu = Menu(menu)
    menu.add_cascade(label='File', menu=filemenu)
    filemenu.add_command(label='New')
    filemenu.add_command(label='Open...')
    filemenu.add_separator()
    filemenu.add_command(label='Exit', command=root.quit)
    helpmenu = Menu(menu)
    menu.add_cascade(label='Help', menu=helpmenu)
    helpmenu.add_command(label='About')

    root.title('Age & Gender Detection')
    button1 = Button(root, text='Start Detection', width=25, command=create_windo
w)
    button1.pack()

    button2 = Button(root, text="Close", width=25, command=root.destroy)
    button2.pack()

    mainloop()
```

پایان