



Shamsipour Technical and Vocational College

دانشکده فنی و حرفه ای شهید شمسی پور

تشخیص چهره انسان با پایتون و OpenCV

استاد: فرخ کریمی

محقق: امیر کریمی

ID کانال تلگرام: @robot_ir

مرداد ۹۶



فهرست مطالب

.....	مقدمه	۴
..... Dataset Creator	فصل اول : تولیدکننده مجموعه داده ها	۵
.....	کد کامل بخش اول برنامه	۶
.....	فصل دوم : آموزش به تشخیص دهنده	۸
.....	کد کامل بخش دوم	۹
.....	فصل سوم : تشخیص و شناسایی یک چهره ای که از قبل آموزش داده شده است	۱۰
.....	نتیجه گیری	۱۳
.....	مراجع	۱۴

مقدمه

برای تشخیص و شناسایی چهره روش های متفاوتی وجود دارد ، از بین این روشها می توان به روش مبتنی بر عکس و روش مبتنی بر ویژگی های چهره نام برد.

برنامه ای که در اینجا شرح داده ایم به سه بخش تقسیم شده و از کتابخانه OpenCV و زبان برنامه نویسی Python استفاده کرده ایم.

این گزارش دارای سه فصل می باشد. فصل اول به توضیح بخش اول برنامه پرداخته است ، در این بخش به نحوه ایجاد dataset با یک ID خاص پرداخته شده است.

در فصل دوم به توضیح بخش دوم برنامه یعنی نحوه ایجاد فایل YML پرداخته ایم و نحوه عملکرد برنامه در بخش دوم نیز کاملا توضیح داده شده است.

و در آخر ، در فصل سوم به قسمت پایانی و اصلی برنامه پرداخته شده است و نحوه تشخیص چهره کاملا توضیح داده شده است .

فصل اول

تولیدکننده مجموعه داده ها Dataset Creator

برای تشخیص چهره و شناسایی یک چهره به کمک OpenCV باید سه مرحله را طی کنیم که در این فصل به بخش اول پرداخته ایم. و سعی شده که کاملاً خط به خط کدها توضیح داده شود.

برای تشخیص چهره ابتدا باید یکسری عکس از صورت فرد مورد نظر گرفته شود. و یک ID به آن عکسها اختصاص داده شود.

برای مثال دو نفر را در نظر بگیرید نفر اول با ID شماره ۱ و نفر دوم با ID شماره ۲، تمام عکسهای مربوط به نفر اول با همان ID یعنی ID شماره ۱ در dataset ذخیره میشود و همین امر برای نفر دوم هم صورت میگیرد.

ما از این عکسها برای آموزش تشخیص دهنده استفاده می کنیم تا چهره این دو نفر را تشخیص دهد.

فعلاً در این فصل به دریافت عکسها و اختصاص ID به مجموعه عکسها می پردازیم.

لازم به ذکر است که در اینجا به توضیح بدنه اصلی کد پرداخته ایم و در آخر همین فصل کد کامل را قرار داده ایم.

```
#We detect a face by detectMultiScale and get an object
faces = detector.detectMultiScale(gray,1.3,5)
```

این خط کد همانطور که از کامنت آن پیداست بدنبال یک صورت انسان در تصویر می گردد.

```
#To draw a rectangle(image,point 1,point 2,color,thickness)
cv2.rectangle(img, (x, y), (x+w, y+h), (255,0,0), 2)
```

و این خط هم یک مستطیل در اطراف صورت تشخیص داده شده میکشد.

```
#saving the capture face in the dataset folder and second param is for size of images
cv2.imwrite("imagesdb/image."+ Id +'.'+str(sampleNum) + ".jpg",gray[y:y+h,x:x+w])
```

و تمام عکسهای گرفته شده از صورت را در پوشه imagesdb ذخیره می کند.

کد کامل بخش اول برنامه

```
import cv2
#Start webcam
cam = cv2.VideoCapture(0)

#haarcascade_frontalface_default.xml : it is a file that OpenCV stores
#all its pre-trained Haar classifiers to detect various objects, body parts,etc.
#Now, next line loads our Haar cascade from disk
detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

#Get an ID
Id = raw_input('Enter ID for Recognition')
#We use 'sampleNum' for imageNames and for break from the loop
sampleNum = 0

while(True):
    ret,img = cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    #We detect a face by detectMultiScale and get an object
    faces = detector.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        #To draw a rectangle(image,point 1,point 2,color,thickness)
        cv2.rectangle(img, (x, y), (x+w, y+h), (255,0,0), 2)

        #incrementing sample number
        sampleNum = sampleNum+1
        #saving the capture face in the dataset folder and second param is for size of images
        cv2.imwrite("imagesdb/image."+ Id +'.'+str(sampleNum) + ".jpg",gray[y:y+h,x:x+w])

        cv2.imshow('frame',img)
        #wait for 100 milliseconds
        if cv2.waitKey(100) & 0xFF == ord('q'):
            break

        #break if the sample number is more than 20
        elif sampleNum > 25:
            break
    cam.release()
    cv2.destroyAllWindows()
```

و در کل همانطور که مشاهده می کنید کد بالا ۲۵ عکس را از طریق webcam گرفته و در یک پوشه تحت عنوان imagesdb (که باید در مسیر جاری برنامه ایجاد کنید) ذخیره میکند.

توضیحات ریزتر در برنامه به شکل کامنت قرار داده شده و در صورت نیاز می توانید از تابع help() هم استفاده کنید .

فصل دوم

آموزش به تشخیص دهنده

در این فصل به نحوه آموزش Recognizer یا همان تشخیص دهنده می پردازیم. می خواهیم به Recognizer آموزش دهیم که چطور یک صورت خاص را از دیگر صورت ها تشخیص دهد برای اینکار ما باید تمامی عکس های گرفته شده در فایل قبلی را دریافت و به فایل YML تبدیل کنیم.

- ابتدا به توضیح بدنه اصلی کد می پردازیم.

```
imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
```

تمام آدرس های فایل ها را دریافت می کند و در متغیر imagePaths

```
#now looping though all the paths and loading the Ids and images
for imagePath in imagePaths:
    #loading the image and converting it to gray
    pilImage = Image.open(imagePath).convert('L')
    #Now we are converting the PIL image into numpy array, be careful about 'dtype="uint8"'
    imageNp = np.array(pilImage,dtype="uint8")
    #getting the Id from the image
    Id = int(os.path.split(imagePath)[-1].split(".")[1])
    #extract the face from the training image sample
    faces = detector.detectMultiScale(imageNp)
    #if a face is there then append that in the list as well as Id of it
    for (x,y,w,h) in faces :
        faceSamples.append(imageNp[y:y+h,x:x+w])
        Ids.append(Id)
return faceSamples,Ids
```

در اینجا ابتدا عکس ها بارگذاری میشوند و سپس به gray تبدیل میشوند و در آخر به یک آرایه تبدیل میشود. و در خط هشتم ID هر عکس جدا میشود و در متغیر Id قرار می گیرد. در خط دهم ، صورت تشخیص داده شده استخراج می شود. و در حلقه بعدی اگر صورت وجود داشته باشد در واقع یک صورت انسان تشخیص داده شده باشد. در دو لیست Ids و faceSamples قرار میگیرد.

کد کامل بخش دوم

```

import cv2,os #we will need the os to access the file list in our dataset folder
import numpy as np
from PIL import Image
#We have to use .face. in openCV 3..
#Now we need to initialize the recognizer and the face detector
recognizer = cv2.face.createLBPHFaceRecognizer()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

#So I am going to name this function "getImagesAndLabels"
#we need the path of the dataset folder so we will provide the folder path as argument.
# So the function will be like this
def getImagesAndLabels(path) :
    #get the path of all the files in the folder
    imagePath = [os.path.join(path,f) for f in os.listdir(path)]
    #create empty face list
    faceSamples = []
    #create empty ID list
    Ids = []
    #now looping through all the paths and loading the Ids and images
    for imagePath in imagePath:
        #loading the image and converting it to gray
        pilImage = Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array, be careful about 'dtype="uint8"'
        imageNp = np.array(pilImage,dtype="uint8")
        #getting the Id from the image
        Id = int(os.path.splitext(imagePath)[-1].split(".")[1])
        #extract the face from the training image sample
        faces = detector.detectMultiScale(imageNp)
        #if a face is there then append that in the list as well as Id of it
        for (x,y,w,h) in faces :
            faceSamples.append(imageNp[y:y+h,x:x+w])
            Ids.append(Id)
    return faceSamples,Ids

faces,Ids = getImagesAndLabels('imagesdb')
recognizer.train(faces,np.array(Ids))
recognizer.save('dbgenerator/dbgenerator.yml')

```

نکته : توجه کنید که باید در مسیر جاری برنامه یک پوشه با نام dbgenerator ایجاد کنید .

بعد اجرا این کد یک فایل با نام dbgenerator.yml در پوشه فوق ایجاد می شود.

توجه کنید که در واقع با اجرا این کد به تشخیص دهنده یک صورت خاص را آموزش داده ایم ، در فصل بعدی با استفاده از این فایل ، تصویر یک صورت خاص را تشخیص می دهیم.

فصل سوم

تشخیص و شناسایی یک صورتی که از قبل آموزش داده شده است

در این فصل با استفاده از فایل YML ایجاد شده در فصل قبل به تشخیص صورت موردنظر می پردازیم.

- بدنه اصلی کد

```
Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
if(conf < 50):
    if (Id == 1):
        Id = "Amir"
    elif (Id == 2):
        Id= "Hosein"
    elif (Id==6):
        Id = "Ali"
else:
    Id="Unknown"
```

تابع بکار رفته در خط اول در ابتدا متغیر Id را مقدار دهی می کند بوسیله Id تشخیص داده شده و سپس متغیر conf را مقدار دهی می کند. هر چقدر مقدار conf نزدیکتر به صفر باشد به این معناست که صورت تشخیص داده شده اختلاف کمتری با dataset دارد.

در خطوط بعدی دو شرط تودرتو داریم که به این صورت عمل میکند که اگر اختلاف کمتر از پنجاه بود وارد شرط بعدی شده و بررسی میکند که صورت تشخیص داده شده با کدام ID مطابقت دارد.

```
cv2.putText(im, str(Id), (x, y+h), font, 1, fontColor)
```

و اما در این خط یک اسم متناسب با ID را در خروجی نمایش می دهد.

کد کامل بخش سوم

```
import cv2
import numpy as np

#Now we need to initialize the recognizer and the face detector
recognizer = cv2.face.createLBPHFaceRecognizer()
recognizer.load('dbgenerator/dbgenerator.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

cam = cv2.VideoCapture(0)

#Set font and Color
font = cv2.FONT_HERSHEY_SIMPLEX
fontColor = (255,255,255)
while True:
    #Read from the camera and convert to gray
    ret,im = cam.read()
    gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    #Extract the face from the camera
    faces = faceCascade.detectMultiScale(gray,1.2,5)

    for(x,y,w,h) in faces:
        cv2.rectangle(im, (x, y), (x+w, y+h), (255,0,0), 2)
        #We have a high value if the face conforms with the person
        Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
        if(conf < 50):
            if (Id == 1):
                Id = "Amir"
            elif (Id == 2):
                Id= "Hosein"
            elif (Id==6):
                Id = "Ali"
            else:
                Id="Unknown"
        #Put a text in the button of the image
        cv2.putText(im, str(Id), (x, y+h), font, 1, fontColor)
    cv2.imshow('im',im)
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cv2.destroyAllWindows()
```

بعضی از خطوط توضیح داده نشده است زیرا یا قبلا اشاره کرده بودیم یا ساده بوده اند.

نکته : با کمک تابع help() می توانید به جزئیات بیشتری نسبت به کد دست پیدا کنید.

نتیجه گیری

برای تشخیص و شناسایی یک صورت ابتدا باید طبق توضیحات ارائه شده در فصل اول از صورت مورد نظر تعدادی عکس گرفته شود سپس توسط تکه کدی که در فصل دوم شرح داده ایم عکس ها را به یک فایل با پسوند YML تبدیل کنیم.

و در نهایت با استفاده از کدی که در فصل سوم شرح داده ایم صورت فرد مورد نظر را تشخیص می دهیم.

مراجع

http://docs.opencv.org/master/d9/df8/tutorial_root.html

<http://opencv-python-tutroals.readthedocs.io/>

<https://www.packtpub.com/application-development/opencv-3-%E2%80%93-getting-started-image-processing-video>

<https://www.packtpub.com/application-development/opencv-3-example-video>

<https://www.packtpub.com/application-development/opencv-example>