TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

**KHOA ĐIỆN TỬ**

**Bộ môn:** **TIN HỌC CÔNG NGHIỆP**



**BÁO CÁO TIỂU LUẬN**

MÔN HỌC: TRÍ TUỆ NHÂN TẠO (AI)

Sinh viên: 1. Dương Hồng Khiêm

Mssv: K185480106044

Lớp: 54KTMT

Giáo viên hướng dẫn: Nguyễn Tuấn Linh

**Thái Nguyên – 20.…**

1. **Giới thiệu đề tài**

* Tên đề tài: Nhận diện khuôn mặt.
* Yêu cầu bài toán:
* Sử dụng mysql lưu thông tin người
* Sử dụng thư viện nhận dạng mặt người có sẵn trong opencv
* Đọc ảnh qua camera nhận dạng tên, tuổi, giới tính.
* Bài toán nhận diện khuôn mặt sử dụng dữ liệu là ảnh được lấy từ video, tách thành các ảnh và dùng ảnh để train mô hình.
* Bài toán sử dụng thuật toán Haar Cascade có sẵn của opencv.
* **Haar Cascade là gì?**

+ Haar Cascade là một thuật toán được tạo ra dựa trên những tính năng đó để phát hiện đối tượng (có thể là khuôn mặt, mắt, tay, đồ vật,…) được đề xuất vào năm 2001 bởi Paul Viola và Michael Jones trong bài báo cáo của họ với khẳng định “Phát hiện đối tượng một cách nhanh chóng bằng cách sử dụng tầng (Cascade) tang cường các tính năng đơn giản”.

+ Nói một cách dễ hiểu hơn: Haar Cascade là một lớp model có thể giúp chúng ta nhận diện khuôn mặt (Haar Cascade face detection) Haar Cascade sử dụng các tầng Haar và sau đó sử dụng thật nhiều đặc trưng đó qua nhiều lượt (Casade) và tạo thành một cỗ máy nhận diện khuôn mặt hoàn chỉnh.

* **Haar Cascade hoạt động như thế nào?**

+ Trong bài báo cáo của tác giả *Paul Viola và Michael Jones* đã trình bày một phương pháp mới và nhanh hơn để xử lý hình ảnh và phát hiện khuôn mặt bằng cách sử dụng các đặc điểm hình chữ nhật như hình dưới đây. Các đặc điểm hình chữ nhật tương tự như nhận được sử dụng để phát hiện các đặc điểm khác nhau của khuôn mặt như mắt và các nốt như trong hình minh họa.



+ Các tính năng hình chữ nhật được chạy lần lượt trên hình ảnh và tổng số pixel nằm trong phần màu trắng được trừ cho tổng số pixel nằm trong phần màu đen.

**Trong đó:**

* Là các bộ lọc bắt các cạnh trong ảnh.
* Bắt các đường thẳng trong ảnh.
* Về đặc trưng 4 hình vuông.

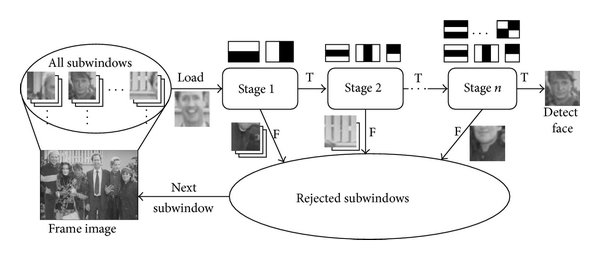
Trong hình minh họa bên dưới, đặc điểm hình chữ nhật đầu tiên đang tính toán cho sự khác biệt về cường độ giữa vùng mắt và vùng má trên khuôn mặt. Và đặc điểm hình chữ nhật thứ hai là **đo sự chênh lệch** về cường độ giữa vùng mắt và sống mũi. Bộ lọc Haar chỉ có thể nhìn cụ thể vào một trong cửa sổ để nhận diện.



+ Vì vậy, sử dụng các đặc điểm hình chữ nhật này trên một số hình ảnh, chúng ta có thể tạo ra hang nghìn điểm đặc trưng cho hình ảnh. Tuy nhiên, việc tính toán tổng số pixel trong các vùng trắng và đen trong toàn bộ ảnh có thể là một hoạt động tốn kém, đặc biệt là đối với các ảnh lớn.

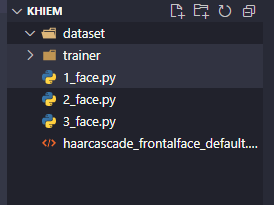
+ Các tác giả cũng đề xuất một phương pháp được gọi là ảnh phân tích có thể đạt được tính toán tương tư bằng cách thực hiện các phép toán chỉ trên bốn điểm ảnh. Có thể sử dụng hang nghìn hình ảnh được gắn nhãn để tạo phân loại nhận diện khuôn mặt.

+ Các bước hoạt động của thuật toán Haar Cascade.



* **Bước 1:** Hình ảnh (đã được gửi đến bộ phân loại) được **chia thành các phần nhỏ**(hoặc các cửa sổ con như trong hình minh họa).
* **Bước 2:** Chúng tôi đặt N không có bộ dò theo cách xếp tầng trong đó mỗi bộ phát hiện sự kết hợp của các loại **đặc trưng khác nhau** từ các hình ảnh (ví dụ: đường thẳng, cạnh, hình tròn, hình vuông) được truyền qua. Giả sử khi việc trích xuất đối tượng địa lý được thực hiện, mỗi phần phụ được gán một **giá trị tin cậy**.
* **Bước 3:** Hình ảnh (hoặc hình ảnh phụ) có độ tin cậy cao nhất được phát hiện dưới dạng khuôn mặt và được gửi đến **bộ tích lũy** trong khi phần còn lại bị từ chối. Do đó, Cascade tìm nạp khung hình / hình ảnh tiếp theo nếu còn lại và bắt đầu lại quá trình.

1. **Code:**



* Bài làm gồm ba bước:
* Bước 1: tạo dataset để train, kết nối mysql lưu thông tin mặt người.
* Bước 2: lấy ảnh file dataset để train dữ liệu.
* Bước 3: tạo lớp detector nhận diện khuôn mặt đã train.

-B1: kết nối sql và lấy data

import cv2

import mysql.connector

import pyodbc

cam = cv2.VideoCapture(0)#sử dụng camera

detector=cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')#sử dụng thư viện nhận diện có sẵn opencv

#hàm thêm mới và cập nhật dữ liệu được lưu trong mysql

def insertOrUpdate(*Id*,*Name*, *Tuoi*, *Gt*):

    #kết nối với database

    conn = pyodbc.connect("Driver={SQL Server};"

                      "Server=JIN\MSSQLSERVER01;"

                      "Database=DL\_AI;" "UID=sa;""PWD=Jin12122000;""Trusted\_Connection=yes;")

    cmd="SELECT \* FROM People WHERE ID="+str(*Id*)

    cursor=conn.execute(cmd)

    isRecordExist=0

    #dùng for chạy câu lệnh sql thêm mới dữ liệu

    for row in cursor:

        isRecordExist=1

    if(isRecordExist==1):

        cmd="UPDATE People SET Name='"+str(*Name*)+"' WHERE ID="+str(*Id*)

    else:

        cmd="INSERT INTO People(Id,Name,Tuoi,GioiTinh) Values("+str(*Id*)+",'"+str(*Name*)+"'"+","+str(*Tuoi*)+",'"+str(*Gt*)+"')"

    conn.execute(cmd)

    conn.commit()

    conn.close()

#nhập thông tin dl mới đưa về hàm insertOrUpdate

id=input('Nhap id: ')

name=input('Nhap name: ')

tuoi=input('Nhap tuoi: ')

gioiTinh=input('Nhap gioi tinh: ')

insertOrUpdate(id,name, tuoi, gioiTinh)

sampleNum=0

while(True):

    #sử dụng vòng lặp phát hiện mặt trong video

    ret, img = cam.read()#đọc camera

    gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)#chuyển ảnh->ảnh xám

    faces = detector.detectMultiScale(gray, 1.3, 5)#phát hiện đối tượng trong ảnh

    for (x,y,w,h) in faces:

        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)#khoanh vùng đối tượng

        #incrementing sample number

        sampleNum=sampleNum+1

        #saving the captured face in the dataset folder

        cv2.imwrite("dataSet/User."+id +'.'+ str(sampleNum) + ".jpg", gray[y:y+h,x:x+w])#lưu ảnh vào folder

        cv2.imshow('frame',img)#show came

    #wait for 100 miliseconds

    if cv2.waitKey(100) & 0xFF == ord('q'):

        break

    # dùng lại khi lấy được 21 ảnh

    elif sampleNum>20:

        break

cam.release()

* B2: train dữ liệu

import cv2,os

import numpy as np

from PIL import Image

recognizer = cv2.face.LBPHFaceRecognizer\_create()

path='dataSet'

def getImagesAndLabels(*path*):

    #lấy đường dẫn của tất cả tệp trong thư mục

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

    faces=[]

    IDs=[]

    for imagePath in imagePaths:

        #Convert ảnh và add vào mảng faces cùng với ID đối với bàn toán classfication

        # ở đây thì Feature là ảnh còn Label chính là id người dùng

        faceImg=Image.open(imagePath).convert('L')*;*

        faceNp=np.array(faceImg,'uint8')

        #tách lấy ID của ảnh

        ID=int(os.path.split(imagePath)[-1].split('.')[1])

        faces.append(faceNp)

        #print (ID)

        IDs.append(ID)

        cv2.imshow("traning",faceNp)

        cv2.waitKey(10)

    return IDs, faces

Ids,faces=getImagesAndLabels(path)

#trainning

recognizer.train(faces,np.array(Ids))#mặt phát hiện train

recognizer.save('trainer/trainningData.yml')#lưu file train dữ liệu

cv2.destroyAllWindows()

* B3: nhận diện dữ liệu đã train

import cv2

import numpy as np

from PIL import Image

import pickle

import pyodbc

faceDetect=cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')*;*

cam=cv2.VideoCapture(0)*;*

rec=cv2.face.LBPHFaceRecognizer\_create()*;*

rec.read("trainer\\trainningData.yml")

id=0

#thiết lập font chữ

#fontface = cv2.FONT\_HERSHEY\_SIMPLEX

fontface = cv2.FONT\_ITALIC

fontscale = 1

fontcolor = (0,100,0)

#lấy dữ liệu ở db qua ID

def getProfile(*id*):

    conn = pyodbc.connect("Driver={SQL Server};"

                      "Server=JIN\MSSQLSERVER01;"

                      "Database=DL\_AI;"

                      "UID=sa;"

                      "PWD=Jin12122000;"

                      "Trusted\_Connection=yes;")

    cmd="SELECT \* FROM People WHERE ID="+str(*id*)

    cursor=conn.execute(cmd)

    profile=None

    for row in cursor:

        profile=row

    conn.close()

    return profile

while(True):

    #đọc camera

    ret,img=cam.read()*;*

    gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

    faces=faceDetect.detectMultiScale(gray,1.3,5)#phát hiện mặt trong ảnh

    for(x,y,w,h) in faces:

        #hiện thị thông tin khuôn mặt nhận dạng được

        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

        id,conf=rec.predict(gray[y:y+h,x:x+w])

        profile=getProfile(id)

        if(profile!=None):

            cv2.putText(img, "Name: " + str(profile[1]), (x,y+h+30), fontface, fontscale, fontcolor ,2)

            cv2.putText(img, "Tuoi: " + str(profile[2]), (x,y+h+60), fontface, fontscale, fontcolor ,2)

            cv2.putText(img, "Gioi\_tinh: " + str(profile[3]), (x,y+h+90), fontface, fontscale, fontcolor ,2)

        cv2.imshow('Face',img)

    if cv2.waitKey(1)==ord('q'):

        break*;*

cam.release()

cv2.destroyAllWindows()

* **Kết quả**
* Sau khi chạy file 1 để chụp ảnh lưu vào dataset, chạy file 2 để training dữ liệu vừa chụp rồi chạy file 3 để nhận diện khuôn mặt đã được train.

