

**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN
HỆ CHUYÊN GIA**

ĐỀ TÀI:

**ỨNG DỤNG GIAO TIẾP NGƯỜI MÁY BẰNG NHẬN
DIỆN KHUÔN MẶT VÀ TIẾNG NÓI**

Sinh viên thực hiện	:	NGUYỄN ĐÌNH MẠNH
Giảng viên hướng dẫn	:	NGUYỄN THỊ THANH TÂN
Ngành	:	CÔNG NGHỆ THÔNG TIN
Chuyên ngành	:	CÔNG NGHỆ PHẦN MỀM
Lớp	:	D14CNPM8

Hà Nội, tháng 12 năm 2022

PHIẾU CHẤM ĐIỂM

Sinh viên thực hiện:

Họ và tên	Chữ Ký	Ghi chú
Nguyễn Đình Mạnh		

Họ và tên giảng viên	Chữ ký	Ghi chú
Giảng viên chấm 1:		
Giảng viên chấm 2:		

MỤC LỤC

	Trang
MỤC LỤC.....	1
Lời mở đầu.....	2
CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN.....	3
1.Hệ chuyên gia.....	3
1.1 Tổng quan về nhận diện khuôn mặt.....	5
1.2 Phân tích bài toán.....	5
1.3 Các công cụ.....	5
1.3.1. PyThon.....	5
1.3.2 Thư viện OpenCV.....	6
1.3.3 Visual Studio Code.....	7
1.3.4 Thư viện Haar Cascade.....	7
1.3.5 DB Browser (SQLite)	7
1.4 Tổng quan về trợ lý ảo.....	8
CHƯƠNG 2 THUẬT TOÁN NHẬN DẠNG.....	10
2.1 Thị giác máy tính.....	10
2.2 Phát hiện khuôn mặt.....	10
2.2.1 Vấn đề.....	10
2.1.2 Phát hiện khuôn mặt sử dụng Haar Cascade.....	10
CHƯƠNG 3 XÂY DỰNG VÀ ĐÁNH GIÁ HỆ THỐNG.....	13
3.1 Biểu đồ hệ thống.....	13
3.2 Biểu đồ UseCase.....	13
3.3 Code.....	13
KẾT LUẬN.....	17
TÀI LIỆU THAM KHẢO.....	18

LỜI MỞ ĐẦU

Khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa người với người, và cũng mang một lượng thông tin giàu có, chẳng hạn có thể xác định giới tính, tuổi tác, trạng thái cảm xúc của người đó,... hơn nữa khảo sát chuyển động của các đường nét trên khuôn mặt có thể biết được người đó đang muốn nói gì. Trong hệ thống nhận dạng người thì quá trình nhận dạng khuôn mặt được đánh giá là bước khó khăn và quan trọng nhất so với các bước còn lại của hệ thống. Do đó, nhận dạng khuôn mặt là điều quan trọng và cần thiết.

Nhận dạng khuôn mặt người là một công nghệ được ứng dụng rộng rãi trong đời sống hằng ngày của con người như các hệ thống giám sát, quản lý vào ra, tìm kiếm thông tin người nổi tiếng,... có rất nhiều phương pháp nhận dạng khuôn mặt đề nâng cao hiệu suất tuy nhiên dù ít hay nhiều những phương pháp này đang vấp phải những thử thách về độ sáng, hướng nghiêng, kích thước ảnh, hay ảnh hưởng của tham số môi trường.

Cùng với sự phát triển của xã hội, vấn đề an ninh, bảo mật đang được yêu cầu khắt khe tại mọi quốc gia trên thế giới. Các hệ thống nhận dạng con người, đồ vật... được ra đời và phát triển với độ tin cậy ngày càng cao. Với cách tiếp cận đối tượng nhận dạng theo phương pháp này, chúng ta có thể thu nhập được nhiều thông tin từ đối tượng hơn, mà không cần tác động nhiều đến đối tượng cũng vẫn đảm bảo tính chính xác, an toàn, thuận tiện.

Trong phạm vi bài báo cáo này em xin được trình bày quá trình thực hiện **ứng dụng giao tiếp người máy bằng nhận diện khuôn mặt và tiếng nói**. Cuối cùng, mặc dù đã cố gắng rất nhiều nhưng do thời gian có hạn, khả năng dịch và hiểu tài liệu chưa tốt nên nội dung báo cáo này không thể tránh khỏi những thiếu sót, rất mong được sự chỉ bảo, góp ý của các thầy cô.

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1. Hệ chuyên gia

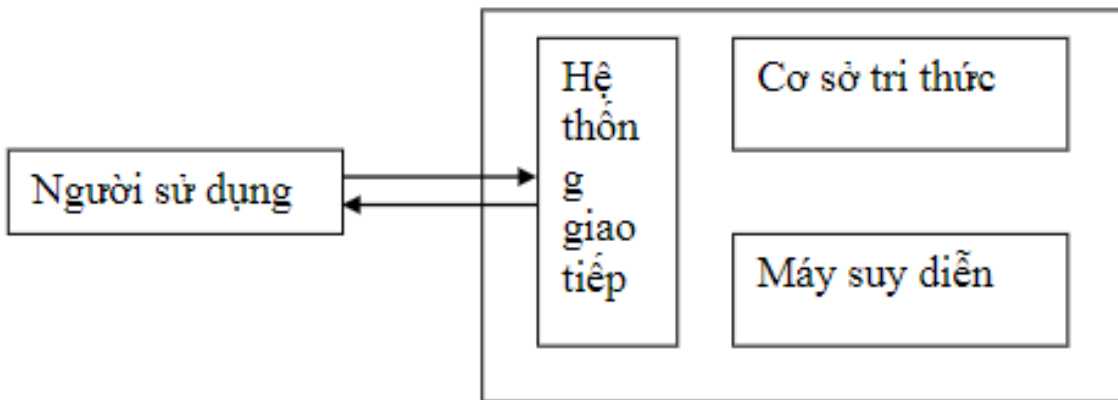
Hệ chuyên gia, còn gọi là hệ thống dựa tri thức hoặc hệ thống chuyên gia, là một chương trình máy tính chứa một số tri thức đặc thù của một hoặc nhiều chuyên gia con người về một chủ đề cụ thể nào đó. Các chương trình thuộc loại này đã được phát triển từ các thập niên 1960 và 1970, và trở thành ứng dụng thương mại từ thập niên 1980. Dạng phổ biến nhất của hệ chuyên gia là một chương trình gồm một tập luật phân tích thông tin (thường được cung cấp bởi người sử dụng hệ thống) về một lớp vấn đề cụ thể, cũng như đưa ra các phân tích về các vấn đề đó, và tùy theo thiết kế chương trình mà đưa lời khuyên về trình tự các hành động cần thực hiện để giải quyết vấn đề. Đây là một hệ thống sử dụng các khả năng lập luận để đạt tới các kết luận. Nhiều hệ chuyên gia đã được thiết kế và xây dựng để phục vụ các lĩnh vực kế toán, y học, điều khiển tiến trình (process control), dịch vụ tư vấn tài chính (financial service), tài nguyên con người (human resources), v.v..

Hệ chuyên gia là một hệ thống chương trình máy tính chứa các thông tin, tri thức và các quá trình suy luận về một lĩnh vực cụ thể nào đó để giải quyết các vấn đề khó hoặc hóc búa đòi hỏi sự tinh thông đầy đủ của các chuyên gia con người đối với các giải pháp của họ. Nói một cách khác hệ chuyên gia là dựa trên tri thức của các chuyên gia con người giỏi nhất trong lĩnh vực quan tâm.

- Tri thức của hệ chuyên gia bao gồm các sự kiện và các luật. Các sự kiện được cấu thành bởi một số nhiều các thông tin, được thu thập rộng rãi, công khai và được sự đồng tình của các chuyên gia con người trong lĩnh vực. Các luật biểu thị sự quyết đoán chuyên môn của các chuyên gia trong lĩnh vực.
- Mức độ hiệu quả của một hệ chuyên gia phụ thuộc vào kích thước và chất lượng của cơ sở tri thức mà hệ đó có được.
- Mỗi hệ chuyên gia chỉ đặc trưng cho một lĩnh vực vấn đề nào đó, như y học, tài chính, khoa học hay công nghệ, vv..., mà không phải là cho bất cứ một lĩnh vực vấn đề nào.

Ví dụ : hệ chuyên gia về lĩnh vực y học để phát hiện các căn bệnh lây nhiễm sẽ có nhiều tri thức về một số triệu chứng lây bệnh, lĩnh vực tri thức y học bao gồm các căn bệnh, triệu chứng và chữa trị.

Hoạt động của một hệ chuyên gia dựa trên tri thức được minh họa như sau:



1.1 Tổng quan về nhận diện khuôn mặt

Hơn một thập kỉ qua có rất nhiều công trình nghiên cứu về bài toán xác định khuôn mặt người từ ảnh đen trắng, xám đến ảnh màu như ngày hôm nay. Các nhà nghiên cứu đi từ bài toán đơn giản, mỗi ảnh chỉ có một mặt người nhìn thẳng vào thiết bị thu hình và đầu ở tư thế thẳng đứng trong ảnh đen trắng. Cho đến ngày hôm nay bài toán mở rộng cho ảnh màu, có nhiều khuôn mặt trong cùng một ảnh, có nhiều tư thế thay đổi trong ảnh. Không những thế mà còn mở rộng cả phạm vi từ môi trường xung quanh khá đơn giản cho đến môi trường xung quanh rất phức tạp nhằm đáp ứng nhu cầu của con người.

Trong những năm gần đây các ứng dụng về trí tuệ nhân tạo ngày càng phát triển và được đánh giá cao. Một lĩnh vực đang được quan tâm của trí tuệ nhân tạo nhằm tạo ra các ứng dụng thông minh, có tính người đó là nhận dạng. Trong đề tài này em chọn đối tượng là khuôn mặt.

1.2 Phân tích bài toán

Bài toán nhận diện khuôn mặt(Face Recognition) bao gồm các bước: lấy data từ camera, train data, nhận diện khuôn mặt.

1.3 Các công cụ

Để giải quyết bài toán nhận diện khuôn mặt này em sử dụng thư viện opencv, haarcascades, ngôn ngữ python trên phần mềm Visual Studio Code.

1.3.1. PyThon

Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

Các đặc điểm của Python:

- Ngữ pháp đơn giản, dễ đọc.
- Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented)
- Hỗ trợ module và hỗ trợ gói (package)
- Xử lý lỗi bằng ngoại lệ (Exception)
- Kiểu dữ liệu động ở mức cao.
- Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
- Có khả năng tương tác với các module khác viết trên C/C++ (Hoặc Java cho Jython, hoặc .Net cho IronPython).

- Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

1.3.2 Thư viện OpenCV

OpenCV (Open Computer Vision library) do Intel phát triển, được giới thiệu năm 1999 và hoàn thiện thành phiên bản 1.0 năm 2006. Thư viện opencv – gồm khoảng 500 hàm — được viết bằng ngôn ngữ lập trình C và tương thích với các hệ điều hành Windows, Linux, Mac OS... đóng vai trò xác lập chuẩn giao tiếp, dữ liệu, thuật toán cho lĩnh vực CV và tạo điều kiện cho mọi người tham gia nghiên cứu và phát triển ứng dụng

Trước OpenCV không có một công cụ chuẩn nào cho lĩnh vực xử lý ảnh. Các đoạn code đơn lẻ do các nhà nghiên cứu tự viết thường không thống nhất và không ổn định. Các bộ công cụ thương mại như Matlab, Simulink,...v.v.. lại có giá cao chỉ thích hợp cho các công ty phát triển các ứng dụng lớn. Ngoài ra còn có các giải pháp kèm theo thiết bị phần cứng mà phần lớn là mã đóng và được thiết kế riêng cho từng thiết bị, rất khó khăn cho việc mở rộng ứng dụng.

OpenCV là công cụ hữu ích cho những người bước đầu làm quen với xử lý ảnh số vì các ưu điểm sau:

- OpenCV là công cụ chuyên dụng: được Intel phát triển theo hướng tối ưu hóa cho các ứng dụng xử lý và phân tích ảnh, với cấu trúc dữ liệu hợp lý, thư viện tạo giao diện, truy xuất thiết bị phần cứng được tích hợp sẵn. OpenCV thích hợp để phát triển nhanh ứng dụng
- OpenCV là công cụ mã nguồn mở: Không chỉ là công cụ miễn phí, việc được xây dựng trên mã nguồn mở giúp OpenCV trở thành công cụ thích hợp cho nghiên cứu và phát triển, với khả năng thay đổi và mở rộng các mô hình, thuật toán
- OpenCV đã được sử dụng rộng rãi: Từ năm 1999 đến nay, OpenCV đã thu hút được một lượng lớn người dùng, trong đó có các công ty lớn như Microsoft, IBM, Sony, Siemens, Google và các nhóm nghiên cứu ở Stanford, MIT, CMU, Cambridge,... Nhiều forum hỗ trợ và cộng đồng người dùng đã được thành lập, tạo nên kênh thông tin rộng lớn, hữu ích cho việc tham khảo, tra cứu

1.3.3 Visual Studio Code

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Visual Studio Code là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json (ví dụ như tập tin tùy chỉnh của người dùng). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tốn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.

1.3.4 Thư viện Haar Cascade

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 của họ với khẳng định “Phát hiện đối tượng một cách nhanh chóng bằng cách sử dụng tầng (Cascade) tăng cường các tính năng đơn giản”.

Triển khai ban đầu được sử dụng để phát hiện khuôn mặt chính diện và các đặc điểm như Mắt, Mũi và Miệng. Tuy nhiên, có nhiều đặc trưng Haar được đào tạo trước đó trong GitHub của họ cho các đối tượng khác cũng như cho toàn bộ cơ thể, thân trên, thân dưới, nụ cười và nhiều đồ vật khác.

Nói một cách dễ hiểu hơn, Haar Cascade là gì? 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 (Cascade) và tạo thành một cỗ máy nhận diện khuôn mặt hoàn chỉnh.

1.3.5 DB Browser (SQLite)

SQLite Database Browser hay còn được gọi với tên DB Browser for SQLite là một công cụ mã nguồn mở, được thiết kế trực quan và chỉnh sửa các tệp cơ sở dữ liệu tương thích với SQLite. Đây là công cụ dành cho người dùng và nhà phát triển muốn tạo ra cơ sở dữ liệu, tìm kiếm và chỉnh sửa dữ liệu.

1.4 Tổng quan về trợ lý ảo

Trợ lý ảo (Virtual Assistant) hay còn gọi là trợ lý AI hay trợ lý kỹ thuật số, là chương trình ứng dụng được thiết kế để "hiểu" các lệnh thoại bằng ngôn ngữ tự nhiên và thực hiện các tác vụ cho người dùng

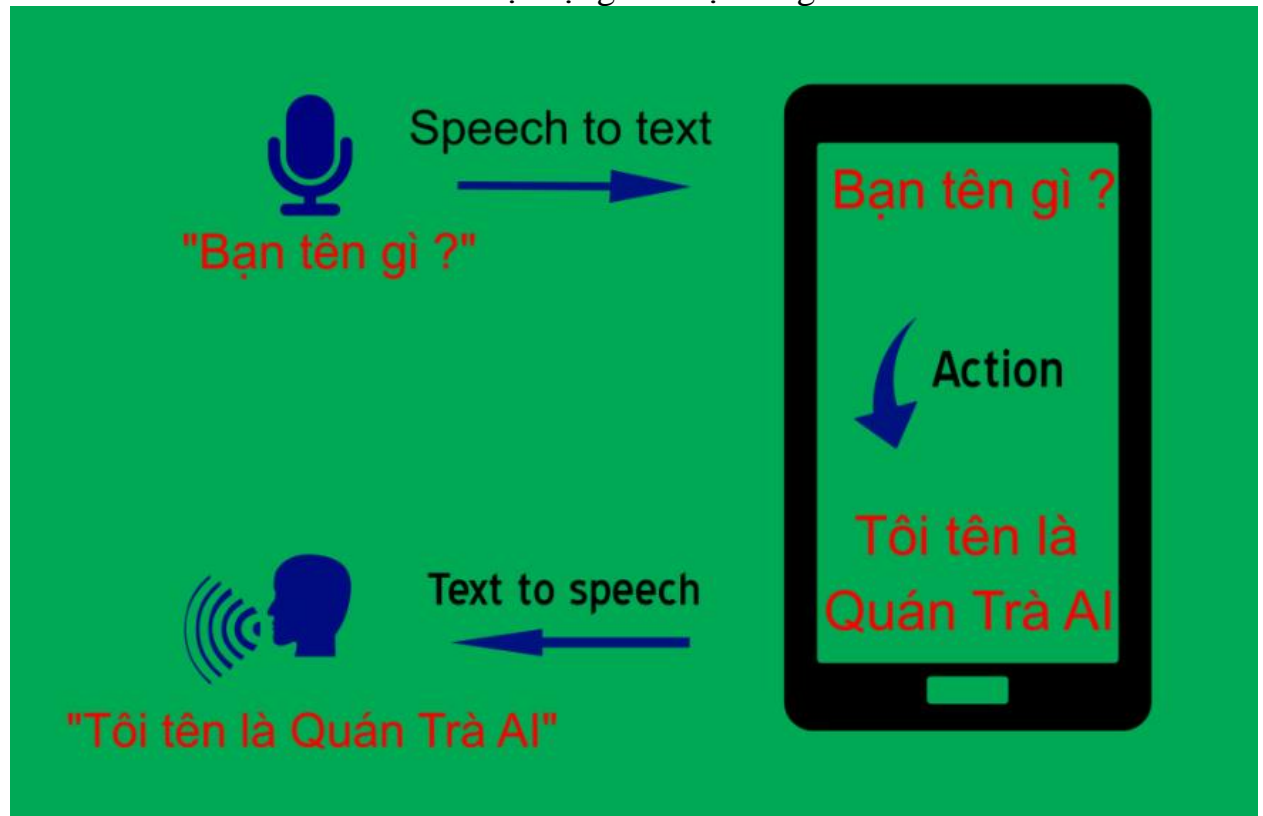
Để xây dựng một hệ thống trợ lý ảo gồm có 3 bước:

Bước thứ nhất: Nhận dạng giọng nói (speech to text) hay nói cách khác là chuyển giọng nói về dạng văn bản.

Bước thứ hai: Đưa ra hành động (action) có nghĩa là với văn bản đó thì sẽ đưa ra câu trả lời là gì (đưa ra hành động cho câu hỏi trên).

Bước thứ ba: Chuyển văn bản về giọng nói (text to speech). Ngược với bước thứ nhất, sau khi có được câu trả lời bằng văn bản ở bước 2 thì ta cần chuyển về giọng nói.

Xem hình sau để hiểu về cách hoạt động của hệ thống:



Code mẫu:

chuyển từ text sang nói

```
texttonoi.py > ...
1  import speech_recognition as sr
2  from gtts import gTTS
3  import playsound
4  # import os
5  # import time
6
7  def speak(text):
8      tts = gTTS(text=text, lang='vi')
9      filename = 'voice.mp3'
10     tts.save(filename)
11     playsound.playsound(filename)
12
13     speak("xin chào super")
```

Chuyển từ nói sang text

```
noitotext.py > ...
1  import speech_recognition as sr
2
3  r = sr.Recognizer()
4
5  with sr.Microphone() as source:
6      audio_data = r.record(source, duration=5)
7      print("Recognizing...")
8      try:
9          text = r.recognize_google(audio_data, language="vi")
10     except:
11         text = ""
12     print(text)
```

CHƯƠNG 2 THUẬT TOÁN NHẬN DẠNG

2.1 Thị giác máy tính

Thị giác máy tính (Computer Vision) là một trong những lĩnh vực hot nhất của khoa học máy tính và nghiên cứu trí tuệ nhân tạo. Dù chúng vẫn chưa thể cạnh tranh với sức mạnh thị giác của mắt người, đã có rất nhiều ứng dụng hữu ích được tạo ra khai thác tiềm năng của chúng.

Đối với máy tính, hình ảnh này giống như tất cả các hình ảnh khác, đó là một mảng các pixel, các giá trị số đại diện cho các sắc độ của màu đỏ, xanh lá cây và xanh dương. Một trong những thách thức mà các nhà khoa học máy tính phải vật lộn từ những năm 1950s là tạo ra những cỗ máy có thể hiểu được hình ảnh và video như con người. Lĩnh vực thị giác máy tính từ đó đã trở thành một trong những lĩnh vực nghiên cứu hot nhất về khoa học máy tính và trí tuệ nhân tạo.

Nhiều thập kỷ sau, chúng ta đã đạt được tiến bộ lớn trong việc tạo ra các phần mềm có thể hiểu và mô tả nội dung của dữ liệu một cách trực quan. Nhưng chúng ta cũng đã nhận ra rằng cần phải đi xa đến mức nào trước khi có thể hiểu và tái tạo một trong những chức năng cơ bản của bộ não con người.

2.2 Phát hiện khuôn mặt

Phát hiện khuôn mặt (Face Detection) là một kỹ thuật máy tính để xác định được các vị trí và các kích thước của khuôn mặt người trong bức ảnh bất kỳ. Kỹ thuật này nhận biết các đặc trưng của khuôn mặt.

2.2.1 Vấn đề

Như ở não của con người chúng ta có những sợi thần kinh để làm tất cả các công việc nhận diện khuôn mặt và tưởng tượng ra các khuôn mặt trong các vật thể hàng ngày.

Máy tính thì lại không có khả năng bậc cao kiểu đó ít nhất là hiện tại (trong tương lai có thể có). Nên chúng ta cần xây dựng một quy trình (hệ thống) nơi chúng ta giải quyết từng bước của nhận diện khuôn mặt một cách riêng biệt và chuyển kết quả hiện tại đó cho bước tiếp theo.

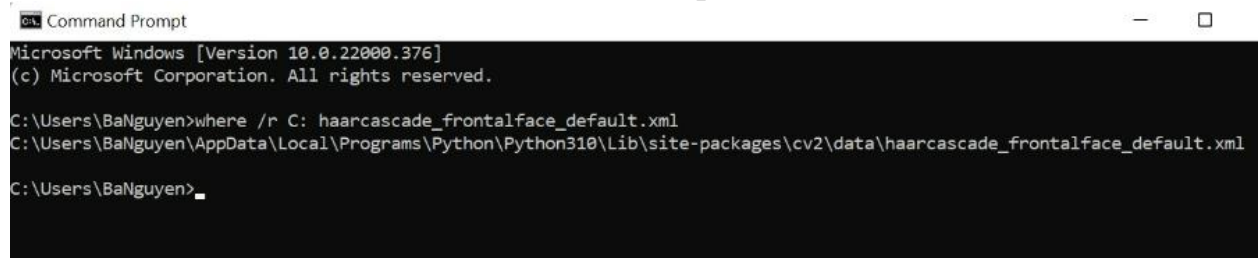
2.1.2 Phát hiện khuôn mặt sử dụng Haar Cascade

Ở đây các nhà nghiên cứu đã huấn luyện tạo ra các file phát hiện khuôn mặt và đôi mắt ('haarcascade_frontalface_default.xml', 'haarcascade_eye.xml'). Chúng ta chỉ cần truy cập vào trang chia sẻ để tải về 2 file này, bỏ vào thư mục chứa file chạy code python là có thể sử dụng được. Nếu các bạn muốn huấn luyện bộ phân loại đối tượng của riêng mình cho bất kỳ đối tượng nào như ô tô, máy bay, ... các bạn có thể sử dụng OpenCV để thực hiện điều này.

Đầu tiên chúng ta cần load các bộ phân loại XML cần thiết, sau đó load hình ảnh (hoặc video) đầu vào ở thang độ xám:

```
import numpy as np
import cv2
face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
img = cv2.imread('kids.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Trường hợp nếu không muốn tải về các file “.xml”, OpenCV đã cung cấp sẵn các file này, công việc của chúng ta là chỉ ra đường dẫn của file. Chúng ta có thể tìm kiếm file haarcascade trên cmd với câu lệnh như sau: where /r C: haarcascade_frontalface_default.xml. Kết quả tìm được:



```
Command Prompt
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

C:\Users\BaNguyen>where /r C: haarcascade_frontalface_default.xml
C:\Users\BaNguyen\AppData\Local\Programs\Python\Python310\Lib\site-packages\cv2\data\haarcascade_frontalface_default.xml

C:\Users\BaNguyen>
```

Tiếp đó, chúng ta cần thay đường dẫn trong câu lệnh số 4 và số 5 như sau:

```
face_cascade = cv2.CascadeClassifier(r'C:\Users\BaNguyen\AppData\Local\
Programs\Python\Python310\Lib\site-packages\cv2\data\
haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier(r'C:\Users\BaNguyen\AppData\Local\
Programs\Python\Python310\Lib\site-packages\cv2\data\
haarcascade_eye.xml')
```

Bây giờ chúng ta sẽ tiến hành tìm kiếm khuôn mặt trong hình ảnh. Nếu có khuôn mặt được tìm thấy, hàm tìm kiếm sẽ trả về vị trí của khuôn mặt được phát hiện dưới dạng Rect(x, y, w, h). Khi có được thông tin vị trí này, chúng ta có thể tạo ROI cho khuôn mặt và áp dụng tính năng phát hiện đôi mắt trên ROI này (vì mắt luôn ở trong khuôn mặt):

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
```

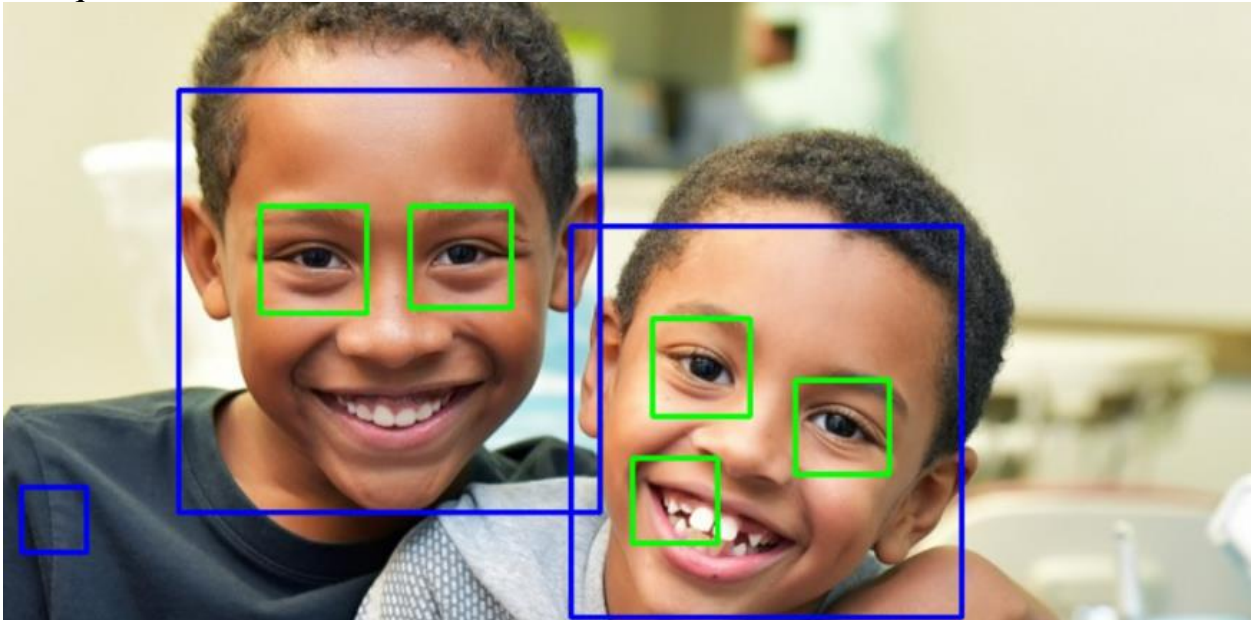
```

cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
roi_gray = gray[y:y+h, x:x+w]
roi_color = img[y:y+h, x:x+w]
eyes = eye_cascade.detectMultiScale(roi_gray)
for (ex,ey,ew,eh) in eyes:
    cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0,255,0), 2)
cv2.imshow('img', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

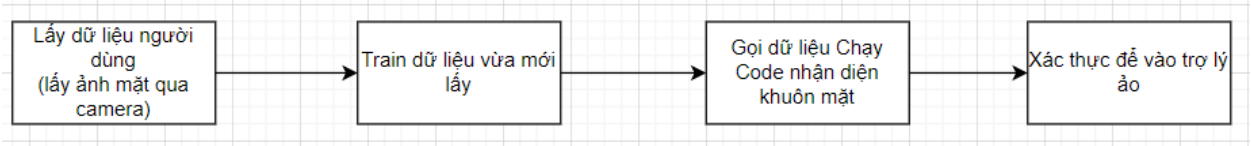
Trong hàm `face_cascade.detectMultiScale`, đối số “1.3” là `scaleFactor` – càng nhỏ thì càng phát hiện tốt hơn, nhưng thuật toán chạy lâu hơn; “5” là `minNeighbors` – càng lớn thì phát hiện được càng ít nhưng khuôn mặt tìm được có độ chính xác và chất lượng cao hơn.

Kết quả nhận được như sau:

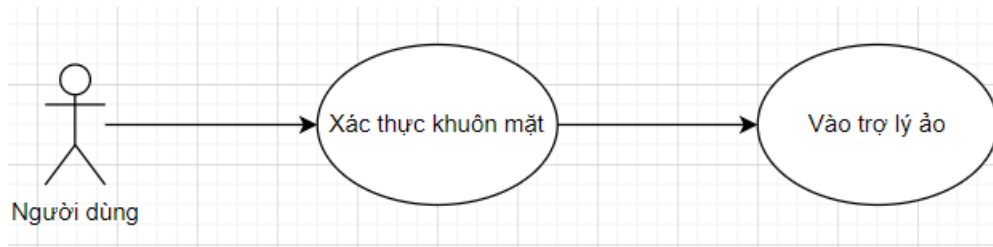


CHƯƠNG 3 XÂY DỰNG VÀ ĐÁNH GIÁ HỆ THỐNG

3.1 Biểu đồ hệ thống



3.2 Biểu đồ UseCase



3.3 Code

```
34
35 id = input("Enter you ID: ")
36 name = input("Enter you Name: ")
37 insertOrUpdate(id, name)
38
39 sampleNum = 0
40 while(True):
41     ret, frame = cap.read()
42
43     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
44
45     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
46
47     for (x, y, w, h) in faces :
48         cv2.rectangle(frame, (x, y), (x+w, y+h) , (0,255,0), 2)
49
50     if not os.path.exists('dataset'):
51         os.makedirs('dataset')
52
53     sampleNum +=1
54
55     cv2.imwrite('dataset/User.'+str(id)+'.'+str(sampleNum)+ '.jpg', gray[y:y+h, x:x+w])
56
57     cv2.imshow('frame', frame)
58     cv2.waitKey(1)
59
60     if sampleNum >99:
61         break;
62
63 cap.release()
64 cv2.destroyAllWindows()
```



```

1  recognizer:
2  ---
3  opencv_SoftFarsi
4  threshold: 1.7976931348623157e+308
5  radius: 1
6  neighbors: 8
7  grid_x: 8
8  grid_y: 8
9  histogram:
10 ---
11 ---
12 ---
13 ---
14 ---
15 ---
16 ---
17 ---
18 ---
19 ---
20 ---
21 ---
22 ---
23 ---
24 ---
25 ---
26 ---
27 ---
28 ---
29 ---
30 ---
31 ---
32 ---
33 ---
34 ---
35 ---
36 ---
37 ---
38 ---
39 ---
40 ---
41 ---
42 ---
43 ---
44 ---
45 ---
46 ---
47 ---
48 ---
49 ---
50 ---
51 ---
52 ---
53 ---
54 ---
55 ---
56 ---
57 ---
58 ---
59 ---
60 ---
61 ---
62 ---
63 ---
64 ---
65 ---
66 ---
67 ---
68 ---
69 ---
70 ---
71 ---
72 ---
73 ---
74 ---
75 ---
76 ---
77 ---
78 ---
79 ---
80 ---
81 ---
82 ---
83 ---
84 ---
85 ---
86 ---
87 ---
88 ---
89 ---
90 ---
91 ---
92 ---
93 ---
94 ---
95 ---
96 ---
97 ---
98 ---
99 ---
100 ---

```

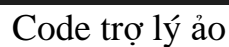
Lấy dữ liệu ảnh

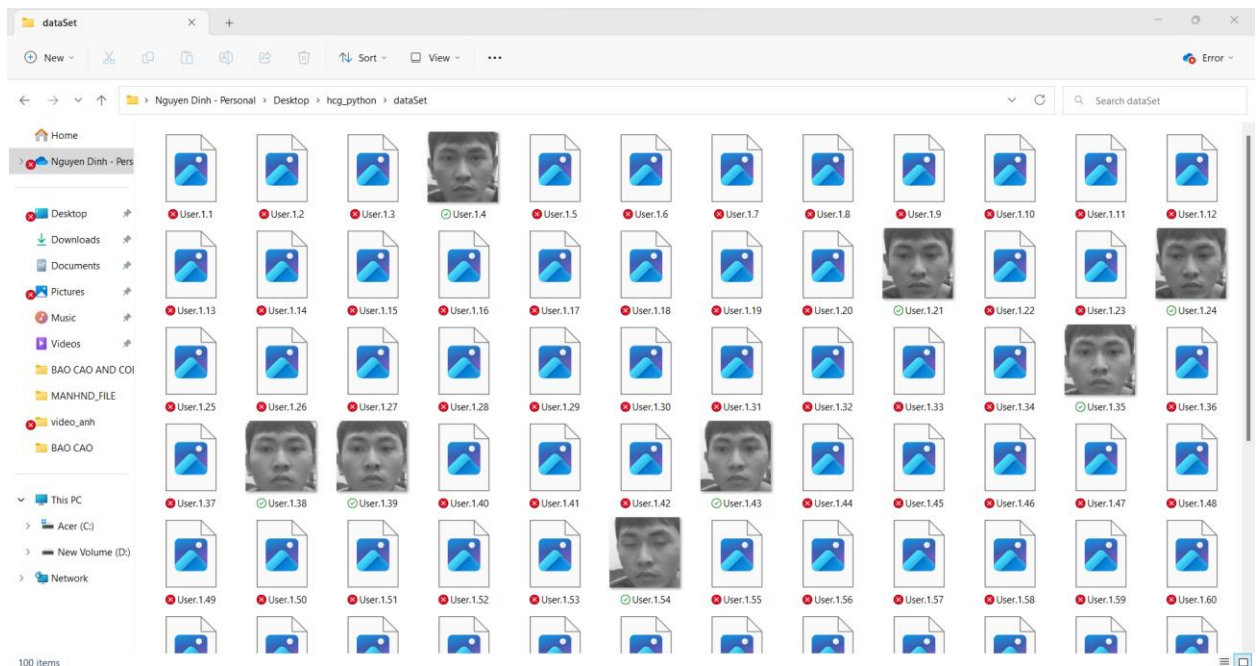
```

1  import cv2
2  import numpy as np
3  import os
4  from PIL import Image
5
6  recognizer = cv2.face.LBPHFaceRecognizer_create()
7
8  path = 'dataSet'
9
10 def getImagesWithId(path):
11     imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
12
13     # print(imagePaths)
14
15     faces = []
16     ids = []
17
18     for imagePath in imagePaths:
19         facing = image.open(imagePath).convert('L')
20
21         faceObj = np.array(facing, 'uint8')
22
23         print(faceObj)
24
25         id = int(imagePath.split("\\\\")[1].split("\\.")[1])
26
27         faces.append(faceObj)
28         ids.append(id)
29
30     cv2.imshow('training', faceObj)
31     cv2.waitKey(10)
32
33     return faces, ids
34
35 faces, ids = getImagesWithId(path)
36
37 recognizer.train(faces, np.array(ids))
38
39 if not os.path.exists('recognizer'):
40     os.makedirs('recognizer')
41
42 recognizer.save('recognizer/trainingdata.yml')
43
44 cv2.destroyAllWindows()
45
46 # getImagesWithId(path)

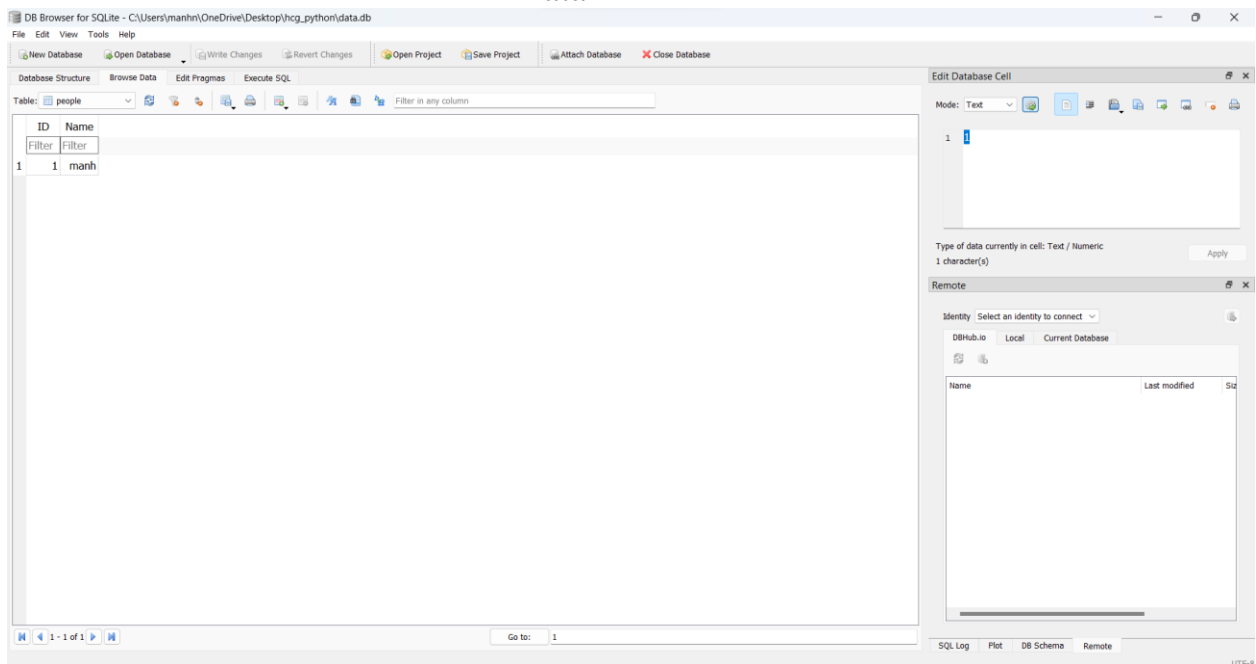
```

Train dữ liệu





Data ảnh



Data người dùng

KẾT LUẬN

Hệ chuyên gia hiện nay vẫn là vấn đề hết sức quan trọng với các tổ chức phát triển phần mềm. Trong quá trình thực hiện báo cáo của mình do thời gian nghiên cứu và kinh nghiệm bản thân còn hạn chế nên còn một số phần của báo cáo nghiên cứu chưa được sâu.

Sau một thời gian thực hiện báo cáo dưới sự hướng dẫn của cô Nguyễn Thị Thanh Tân báo cáo của em đã thực hiện tốt được các mục tiêu đề ra và đạt được những kết quả sau:

Kết quả đạt được:

Trình bày đầy đủ và chính xác các vấn đề tổng quan về phần mềm, công nghệ phần mềm các vấn đề liên quan đến hệ chuyên gia:

- Giới thiệu hệ chuyên gia, nhận diện khuôn mặt dùng Haarcascades, trợ lý ảo
- Áp dụng các kiến thức đã nghiên cứu để thực hiện đề tài ứng dụng giao tiếp người máy bằng nhận diện khuôn mặt và tiếng nói **Hạn chế:**

Mặc dù đã cố gắng hết sức trong thời gian thực hiện đề tài nhưng với kinh nghiệm còn hạn chế nên báo cáo môn học không thể tránh khỏi những thiếu sót.

Trong thời gian tới em sẽ tiếp tục nghiên cứu sâu hơn về các vấn đề của hệ chuyên gia, và đặc biệt là ứng dụng giao tiếp người máy bằng nhận diện khuôn mặt và tiếng nói, để có thể vận dụng các ứng dụng lớn hơn trong thực tế công việc trong tương lai nhằm góp một phần nhỏ bé vào công cuộc chuyên nghiệp hoá hệ chuyên gia ở Việt Nam.

TÀI LIỆU THAM KHẢO

- [1]. Biswas Asit K and John Kolars (1997). Core and Periphery: A Comprehensive Approach to Middle Eastern Water. Oxford University Press, UK.
- [2]. NGHỆ THUẬT THỊ GIÁC & NHỮNG VẤN ĐỀ CƠ BẢN: CÁC YẾU TỐ THỊ GIÁC, NGUYÊN LÝ THỊ GIÁC, TƯ DUY THỊ GIÁC VÀ BỐ CỤC THỊ GIÁC
- [1]. Learning OpenCV 5 Computer Vision with Python: Tackle computer vision and machine learning with the newest tools, techniques and algorithms, 4th Edition