**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**🙠🕮🙢**



**BÁO CÁO NIÊN LUẬN CƠ SỞ-CNTT**

**ĐỀ TÀI**

**NHẬN DẠNG KHUÔN MẶT VỚI PYTHON VÀ OPENCV**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:**  **TS. Phạm Thế Phi** | **SINH VIÊN THỰC HIỆN:**  **Trần Lê Tuyết Hương**  **MSSV B1809130**  **Lớp: DI18V7A1** |

***Cần Thơ, 11/2021***

**🙠🕮🙢**

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

**🙠🖎🙢**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

Cần Thơ, ngày … tháng … năm 2021

**Giảng viên hướng dẫn**

**TS. Phạm Thế Phi**

# LỜI CẢM ƠN

**🙠🖎🙢**

Trong thời gian làm niên luận cơ sở, em đã nhận được rất nhiều sự giúp đở đến từ thầy cô và bạn bè. Đầu tiên em xin gửi lời cảm ơn đến thầy cô Khoa Công nghệ thông tin và Truyền thông nói chung, thầy cô Trường Đại học Cần Thơ nói riêng đã tạo điều kiện thuận lợi cho chúng em học tập để có được những kiến thức về các môn đại cương, cơ sở cũng như các môn chuyên nghành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập.

Xin cảm ơn giảng viên hướng dẫn – TS. Thầy Phạm Thế Phi đã hướng dẫn tận tình, chi tiết và chia sẻ nhiều kiến thức để chúng em có thể vận dụng chúng vào bài niên luận cơ sở này.

Nhưng do kiến thức và kĩ năng còn hạn chế, phương pháp tiếp cận, phân tích, giải quyết vấn đề còn chưa được thấu đáo, chi tiết nên không tránh khỏi những thiếu sót trong quá trình nghiên cứu và trình bày. Rất kính mong được sự đóng góp ý kiến từ Thầy để bài niên luận được hoàn chỉnh hơn.

Em xin chân thành cảm ơn gia đình, bạn bè đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suôt quá trình học tập và làm niên luận cơ sở. Cuối cùng, em xin kính chúc Thầy nhiều sức khỏe, thành công và hạnh phúc.

Vĩnh Long, ngày 27 tháng 11 năm 2021

**Người viết**

Trần Lê Tuyết Hương

# MỤC LỤC

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 2](#_Toc89441232)

[LỜI CẢM ƠN 3](#_Toc89441233)

[MỤC LỤC 4](#_Toc89441234)

[DANH MỤC HÌNH ẢNH 5](#_Toc89441235)

[TÓM TẮT 6](#_Toc89441236)

[PHẦN GIỚI THIỆU 7](#_Toc89441237)

[**1.** **Lý do chọn đề tài** 7](#_Toc89441238)

[**2.** **Mục đích chọn đề tài** 7](#_Toc89441239)

[**3.** **Đối tượng và phạm vi nghiên cứu** 7](#_Toc89441240)

[**4.** **Phương pháp nghiên cứu** 8](#_Toc89441241)

[**5.** **Nội dung nghiên cứu:** 8](#_Toc89441242)

[PHẦN NỘI DUNG 9](#_Toc89441243)

[CHƯƠNG 1: ĐẶC TẢ YÊU CẦU 9](#_Toc89441244)

[**1.** **Các chức năng:** 9](#_Toc89441245)

[**2.** **Mô tả các chức năng:** 9](#_Toc89441246)

[CHƯƠNG 2: THIẾT KẾ GIẢI PHÁP 10](#_Toc89441247)

[**1.** **Giới thiệu công nghệ và nền tảng lý thuyết.** 10](#_Toc89441248)

[**1.1** **Tổng quan về thị giác máy tính (Computer vision)** 10](#_Toc89441249)

[**1.2 Thư viện OpenCV** 10](#_Toc89441250)

[**1.3 Python** 11](#_Toc89441251)

[CHƯƠNG 3: CÀI ĐẶT GIẢI PHÁP 12](#_Toc89441252)

[**1.** **Cài đặt ngôn ngữ lập trình Python:** 12](#_Toc89441253)

[**2.** **Cài đặt thư viện:** 15](#_Toc89441254)

[CHƯƠNG 4: XỬ LÝ VÀ NHẬN DẠNG HÌNH ẢNH TRÊN MÀN HÌNH 16](#_Toc89441255)

[**1. Haar-cascade Detection** 16](#_Toc89441256)

[**2. Xử lý hình ảnh** 16](#_Toc89441257)

[**3. Nhận dạng hình ảnh** 16](#_Toc89441258)

[PHẦN KẾT LUẬN 19](#_Toc89441259)

[TÀI LIỆU THAM KHẢO 20](#_Toc89441260)

# DANH MỤC HÌNH ẢNH

Hình 1: Truy cập vào trang web Python. 12

Hình 2: Danh mục download cho hệ điều hành tương thích 12

Hình 3: Bắt đầu cài đặt 13

Hình 4: Tùy chỉnh cài đặt. 13 Hình 5: Tùy chỉnh cài đặt. 14 Hình 6: Hoàn thành cài đặt. 14 Hình 7: Hình ảnh đầu vào qua camera. 16 Hình 8: Mảng chứa dữ liệu sau khi huấn luyện. 17 Hình 9:  Hình ảnh được chuyển sang thang độ xám. 18 Hình 10: Hình ảnh sau khi huấn luyện xong mô hình. 18

# TÓM TẮT

**🙠🖎🙢**

Trong nghiên cứu này, em đã học cách tạo ra chương trình có thể nhận dạng được khuôn mặt với dữ liệu đầu vào là ảnh chụp từ camera bằng cách sử dụng thư viện OpenCV và Python.

Để tạo ra chương trình nhận dạng khuôn mặt, em đã áp dụng ba bước đơn giản:

* Bước 1: Tạo dataSet để nhận dạng và dùng camera để chụp hình dữ liệu.
* Bước 2: Tạo dataTraining sử dụng để training(huấn luyện) dữ liệu.
* Bước 3: Tạo lớp detector để nhận dạng và hiển thị thông tin.

Ngoài ra, em sử dụng cơ sở dữ liệu để lưu trữ và tinh chỉnh chương trình để thu được kết quả tốt nhất.

# PHẦN GIỚI THIỆU

## **Lý do chọn đề tài**

Với sự phát triển không ngừng của khoa học và công nghệ, đặc biệt trong lĩnh vực trí tuệ nhân tạo (AI) - các bài toán nhận dạng được nhiều người áp dụng như nhận dạng chữ viết, nhận dạng dấu vân tay, nhận dạng âm thanh,… Một trong những bài toán được nhiều người quan tâm nhất hiện nay đó là nhận dạng khuôn mặt.

Nhận dạng khuôn mặt đóng vai trò quan trọng trong nhiều lĩnh vực đời sống hằng ngày của con người như các hệ thống giám sát, chấm công, quản lý vào ra, điểm danh sinh viên,… đặc biệt là về an ninh, bảo mật.

Và hiện nay, nhận dạng khuôn mặt không còn quá xa lạ với chúng ta nữa. Có rất nhiều phương pháp được áp dụng như sử dụng OpenCV, Haar-like, SVM, Adaboost,… Trong nghiên cứu này em sẽ sử dụng phương pháp nhận dạng khuôn mặt đơn giản Haar-cascade Detection .

## **Mục đích chọn đề tài**

* Tìm hiểu về thư viện OpenCV.
* Tìm hiểu các bài toán nhận dạng khuôn mặt.
* Nghiên cứu về phương pháp Haar-cascade Detection.
* Có thể nhận dạng được khuôn mặt qua camera.

## **Đối tượng và phạm vi nghiên cứu**

1. Đối tượng:

- Các phương pháp, thuật toán phục vụ cho việc phát hiện và nhận dạng khuôn mặt người qua camera.

- Bộ thư viện xử lý ảnh OpenCV.

- Bộ cơ sở dữ liệu, cụ thể trong nghiên cứu này em sử dụng SQLite Expert.

1. Phạm vi nghiên cứu:

- Tập trung tìm hiểu về nhận dạng khuôn mặt.

- Việc xử lý ảnh, nhận dạng khuôn mặt phải thỏa mãn: ánh sáng bình thường, góc ảnh trực diện hoặc góc nghiêng không quá lớn, không bị che khuất, ảnh có chất lượng cao.

## **Phương pháp nghiên cứu**

* Dữ liệu đầu vào là ảnh chụp từ camera.
* Được xử lý và lưu trữ lại thông qua ảnh xám.
* Lấy ảnh xám để huấn luyện với thông tin nhận được.
* Hiển thị kết quả ra màn hình.

## **Nội dung nghiên cứu:**

* Tạo cơ sở dữ liệu để lưu trữ thông tin.
* Tiến hành xây dựng thuật toán nhận dạng.

# PHẦN NỘI DUNG

# CHƯƠNG 1: ĐẶC TẢ YÊU CẦU

## **Các chức năng:**

* Nhập từ bàn phím thông tin của đối tượng.
* Hiển thị nhận dạng khuôn mặt ra màn hình.

## **Mô tả các chức năng:**

* Lấy thông tin từ bàn phím lưu vào cơ sở dữ liệu.
* Truyền đường dẫn chứa file ảnh đã huấn luyện vào trong phần mềm, sau đó phần mềm sẽ hiển thị thông tin lên cửa sổ nhận dạng.
* **Trình tự thực hiện:**
* Nhập thông tin của đối tượng từ bàn phím, cửa sổ camera sẽ hiện lên và chụp ảnh lại.
* Sử dụng mô hình để huấn luyện các ảnh vừa chụp được.
* Sử dụng mô hình nhận dạng khuôn mặt.
* Xuất lên màn hình kết quả sau khi nhận dạng.
* **Kết quả:** Nhận dạng được khuôn mặt có trong camera.

# CHƯƠNG 2: THIẾT KẾ GIẢI PHÁP

## **Giới thiệu công nghệ và nền tảng lý thuyết.**

### **1.1** **Tổng quan về thị giác máy tính (Computer vision)**

Thị giác máy tính là thuật ngữ mô tả một tập hợp các công nghệ cho phép các thiết bị máy tính, phần mềm, robot hoặc bất kỳ thiết bị nào thu nhận, phân tích và xử lý hình ảnh. Các nguồn hình ảnh khác nhau có thể vô cùng đa dạng. Chúng có thể là hình ảnh, video, dữ liệu 3D, dữ liệu từ máy quét y tế hoặc công nghiệp,.v.v. Mục đích là để cung cấp cho các thiết bị này khả năng “nhìn” và phản ứng tùy thuộc vào thông tin chúng nhận được. Trong sự phức tạp và các trường hợp sử dụng của mình, thị giác máy tính thường được so sánh với nhận dạng giọng nói.

Lưu ý rằng không nên có sự nhầm lẫn giữa thị giác máy tính và xử lý hình ảnh. Xử lý hình ảnh là phân tích hình ảnh kỹ thuật số hoặc thực hiện các thuật toán, bao gồm việc phân loại, trích xuất, chỉnh sửa hoặc lọc… Xử lý hình ảnh liên quan đến các công nghệ và phương pháp được sử dụng để gia tăng hình ảnh về khía cạnh thông tin trong khi thị giác máy tính hướng đến các hành động thực tế.

Mặc dù rõ ràng là ứng dụng đầu tiên của thị giác máy tính là quản lý hình ảnh nhưng nó cũng có thể được sử dụng để thực hiện các hoạt động khác nhau bao gồm nhận dạng đối tượng hoặc phát hiện sự kiện.

### **1.2 Thư viện OpenCV**

OpenCV (Open Computer Vision) là à một thư viện mã nguồn mở phục vụ cho việc nghiên cứu hay phát triển về thị giác máy tính, có hơn 500 hàm khác nhau, tối ưu hóa và xử lý ứng dụng trong thời gian thực, giúp cho việc xây dựng các ứng dụng xử lý ảnh, thị giác máy tính … một cách nhanh hơn. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, iOS.

OpenCV có nhiều ứng dụng:

* Nhận dạng hình ảnh
* Xử lý hình ảnh
* Phục hồi hình ảnh và video
* Các ứng dụng khác.

### **1.3 Python**

Python là ngôn ngữ lập trình hướng đối tượng thông dụng dùng để viết các tiện ích hệ thống. Nó cũng được sử dụng như ngôn ngữ kết dính đóng vai trò tích hợp C và C++.

Được tạo ra bởi Guido van Rossum tại Amsterdam năm 1990, Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Python là ngôn ngữ có hình thức khá đơn giản và rõ ràng, do đó tạo nên sự dễ dàng tiếp cânh cho những lập trình viên mới bắt đầu.

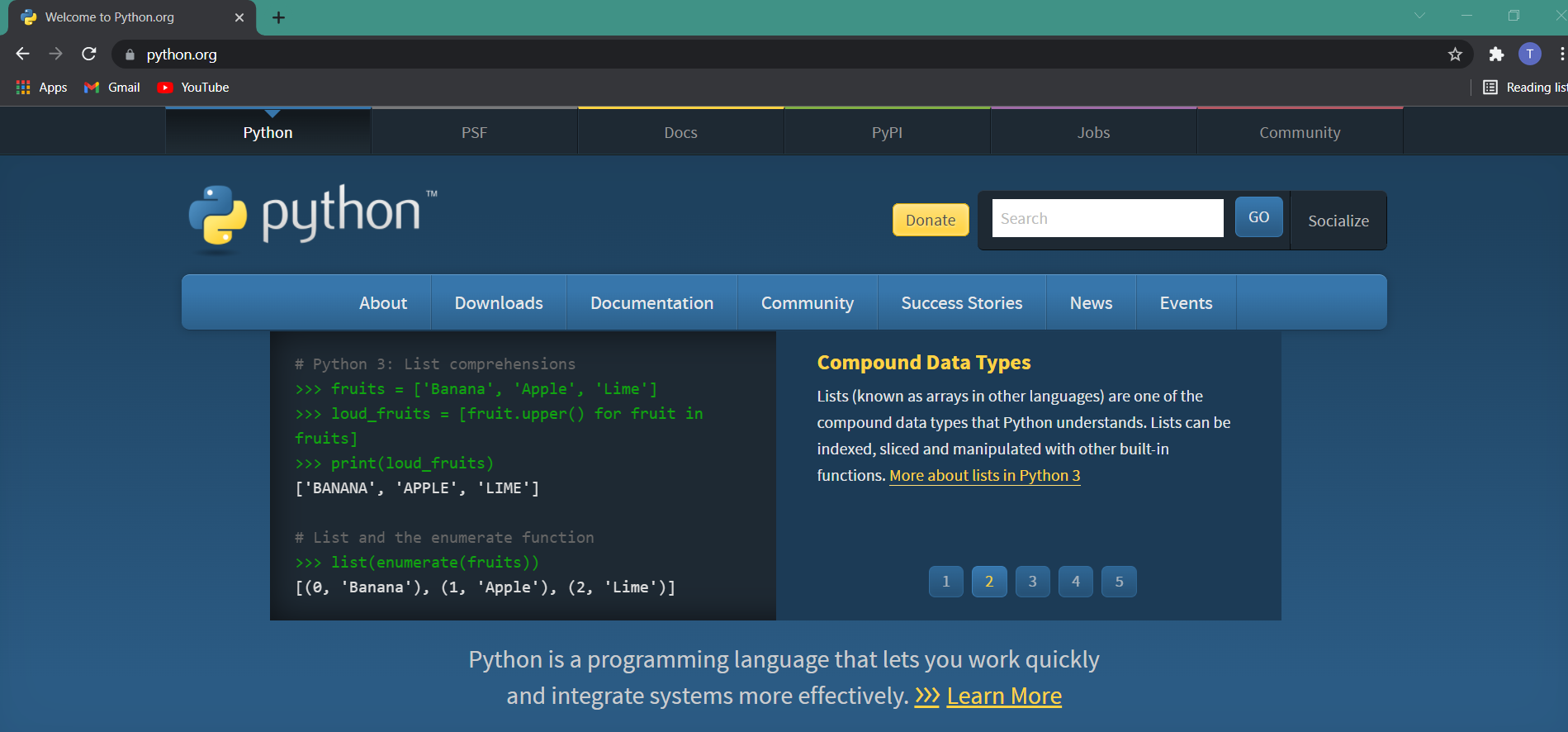
Ban đầu, Python được phát triển để chạy trên nền Unix, nhưng rồi theo thời gian, nó đã được mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.

# CHƯƠNG 3: CÀI ĐẶT GIẢI PHÁP

## **Cài đặt ngôn ngữ lập trình Python:**

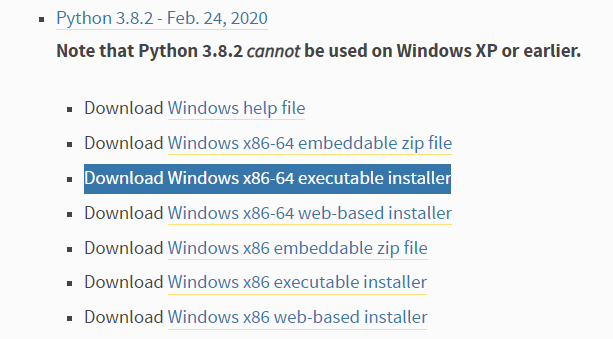
Chương trình được cài đặt trên hệ điều hành windows 10.

**Bước 1:** Truy cập trang web Python



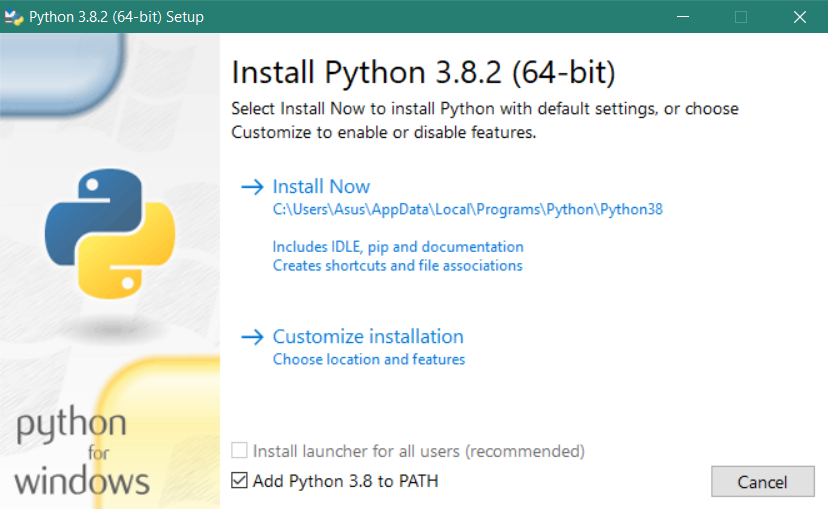
*Hình 1: Truy cập vào trang web python.*

**Bước 2:** Vào mục Downloads chọn Windows và download python 3.8.2



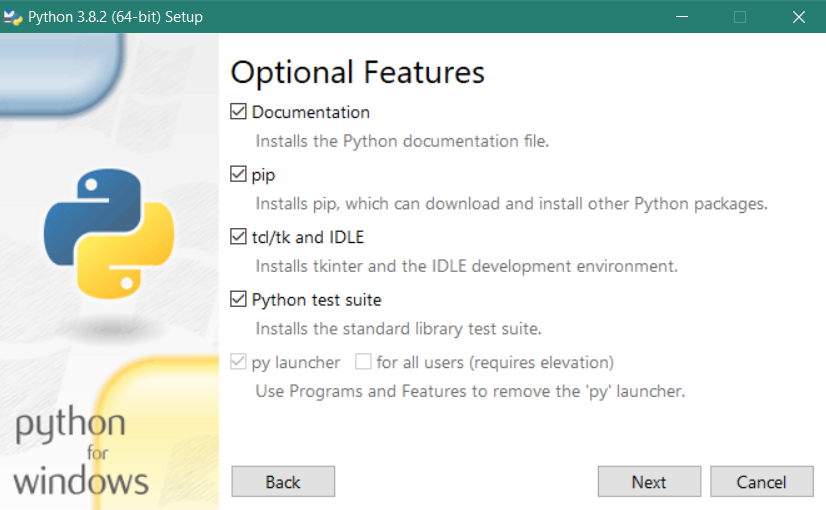
*Hình 2: Danh mục download cho hệ điều hành tương thích.*

**Bước 3:** Cài đặt

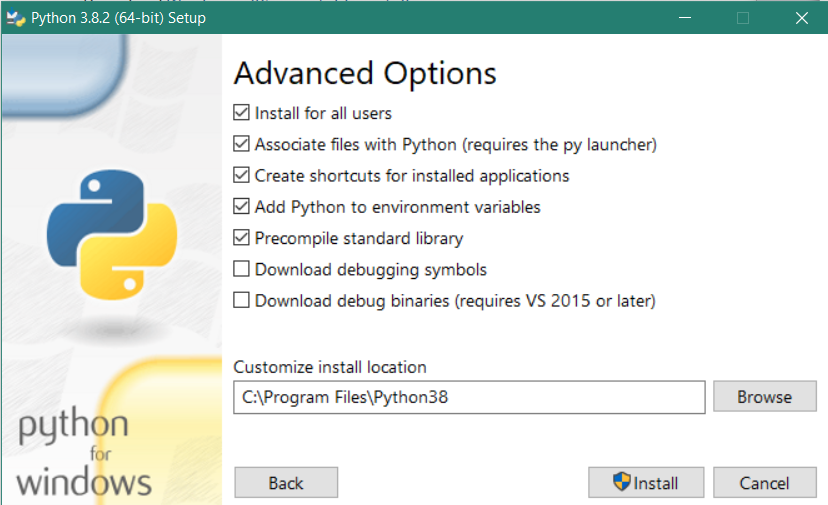


*Hình 3: Bắt đầu cài đặt*

**Bước 4:** Tùy chỉnh cài đặt

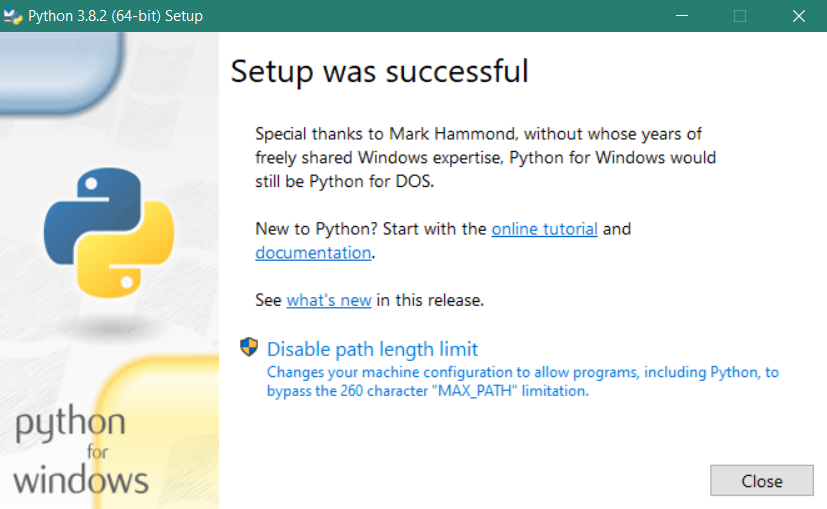


*Hình 4: Tùy chỉnh cài đặt.*



*Hình 5: Tùy chỉnh cài đặt.*

**Bước 5:** Cài đặt hoàn tất



*Hình 6: Hoàn thành cài đặt.*

## **Cài đặt thư viện:**

Để cài đặt thư viện ta dùng công cụ PIP đã được cài đặt chung với python trong phần tùy chỉnh cài đặt phía trên. Và để dùng công cụ PIP ta mở Command Prompt (cmd) trên windows.

**Thư viện numpy:**

- Cài đặt: pip install numpy

**Thư viện OpenCV:**

- Cài đặt: pip install opencv-contrib-python

**Thư viện pillow:**

- Cài đặt: pip install pillow.

Ngoài ra, còn phải cài đặt thêm SQLite. Tạo cơ sở dữ liệu “data” với bảng “People”.

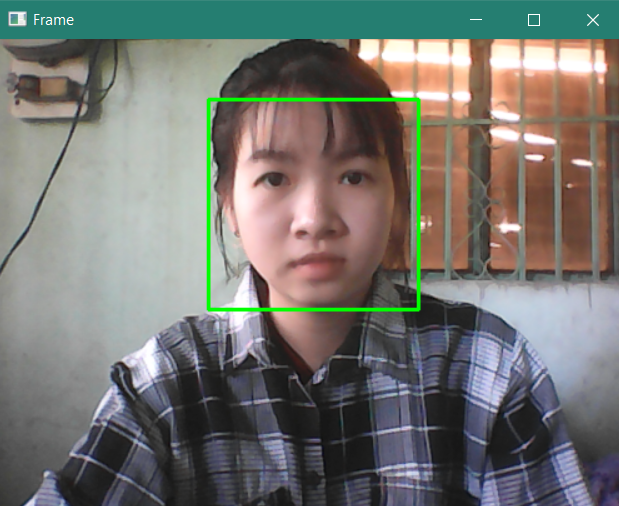
# CHƯƠNG 4: XỬ LÝ VÀ NHẬN DẠNG HÌNH ẢNH TRÊN MÀN HÌNH

## **1. Haar-cascade Detection**

Haar-cascade về cơ bản là sử dụng các đặc trưng loại Haar và sau đó sử dụng thật nhiều đặc trưng đó qua nhiều lượt (cascade) để tạo thành một cỗ máy nhận diện hoàn chỉnh.

## **2. Xử lý hình ảnh**

Điều đầu tiên chúng ta sẽ sử dụng hình ảnh được chụp từ camera làm dữ liệu đầu vào và mục tiêu cuối cùng là nhận ra được khuôn mặt trên camera.

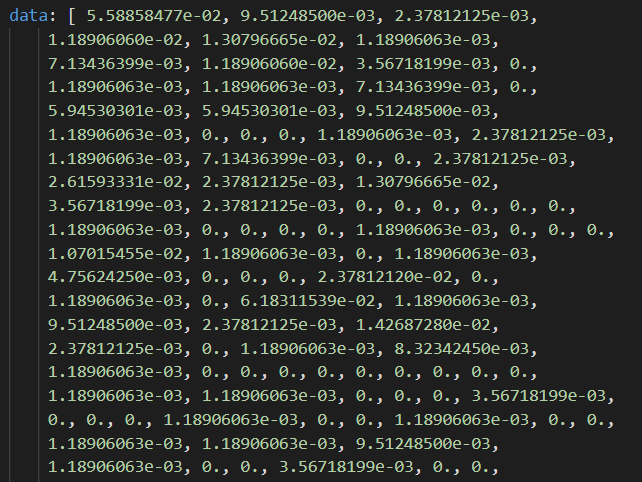


*Hình 7: Hình ảnh đầu vào qua camera.*

## **3. Nhận dạng hình ảnh**

Quá trình nhận dạng trải qua 2 bước:

* Bước 1: **Huấn luyện dữ liệu.**Ta sẽ load đường dẫn và file trong thư mục dataSet, tách “ID” từ tên file ảnh, chuyển ảnh và thêm vào mảng “faces” cùng với “ID”. Sau đó huấn luyện và lưu vào file .yml.



*Hình 8: Mảng chứa dữ liệu sau khi huấn luyện.*

Tiếp theo là xử lý trước hình ảnh.

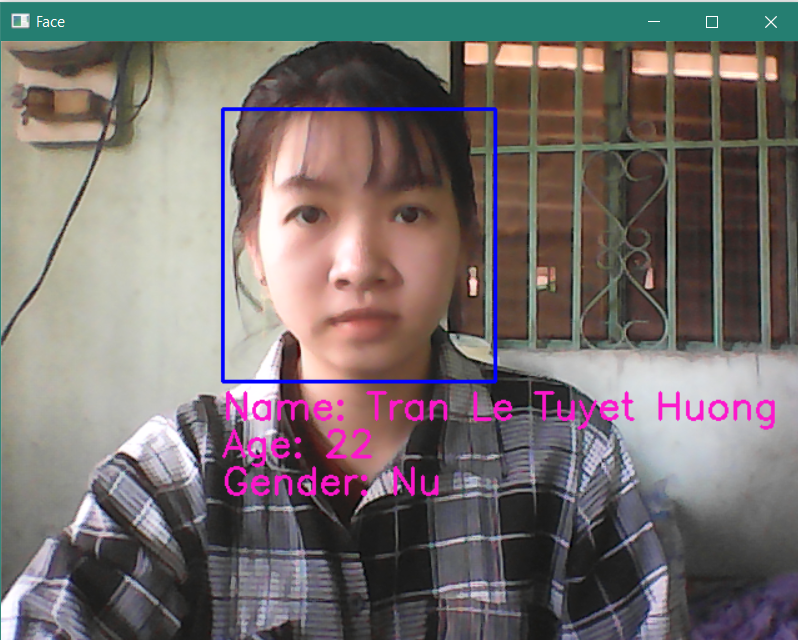
* Thay đổi kích thước nó.
* Chuyển đổi hình ảnh sang thang độ xám.

Sau khi áp dụng các bước tiền xử lý này, tôi được hình như sau:



*Hình 9: Hình ảnh được chuyển sang thang độ xám.*

* Bước 2: **Nhận diện khuôn mặt đã huấn luyện**. Lấy thông tin user từ cơ sở dữ liệu, sử dụng camera để nhận diện, hiển thị thông tin user bằng text.



*Hình 10: Hình ảnh sau khi huấn luyện xong mô hình.*

# PHẦN KẾT LUẬN

Bằng Python và thư viện OpenCV chúng ta đã có thể nhận dạng được khuôn mặt qua camera. Hiệu năng phần mềm tương đối tốt, thời gian phát hiện nhanh đối với mặt người chụp thẳng, không có vật cản.

**Kết quả đạt được:**

- Triển khai được chương trình nhận dạng khuôn mặt bằng thư viện OpenCV và Python.

**Hướng phát triển trong tương lai:**

- Thu thập nhiều hơn hình ảnh thực tế để nhận dạng khuôn mặt người chính xác hơn.

- Cải thiện độ chính xác.

# TÀI LIỆU THAM KHẢO

1. TS. Đỗ Thanh Nghị , TS. Phạm Nguyên Khang. Giáo trình Nguyên lý máy học.
2. TS. Đặng Quốc Bảo, ThS. Lê Minh Thư, et al. Giáo trình Cơ sở dữ liệu.
3. REINIUS, Staffan. Object recognition using the OpenCV Haar cascade-classifier on the iOS platform. 2013.
4. NGUYỄN, Thanh Hải, et al. Giải pháp điểm danh sinh viên bằng nhận diện gương mặt với đặc trưng Haar-Like kết hợp thuật toán rừng ngẫu nhiên. 2020.
5. LÊ, Thị Thu Hằng. Tìm hiểu đặc trưng Haar và bài toán phát hiện mặt người trong ảnh. 2012. PhD Thesis. Đại học Dân lập Hải Phòng.
6. PYTHON, Introducing. Python. 2019.
7. <https://www.pyimagesearch.com/2017/02/13/recognizing-digits-with-opencv-and-python/>
8. <https://www.python.org/downloads/>
9. <https://numpy.org>