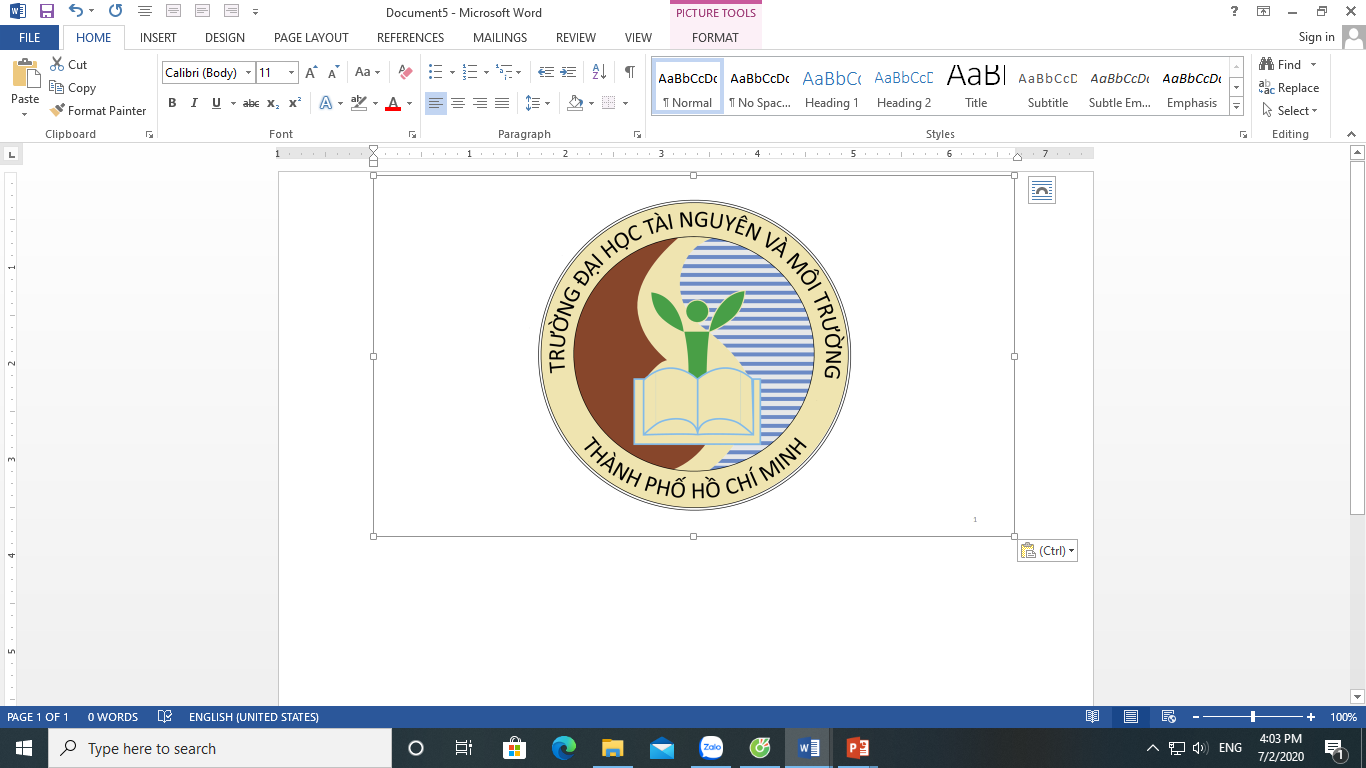
**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**ĐỒ ÁN MÔN HỌC**

**AN TOÀN BẢO MẬT MẠNG MÁY TÍNH VÀ HỆ THỐNG**

**NGHIÊN CỨU PHƯƠNG PHÁP NHẬN DIỆN KÝ TỰ HÌNH ẢNH VÀ XÂY DỰNG ỨNG DỤNG NHẬN DIỆN CHỨNG MINH NHÂN DÂN**

Giảng viênhướng dẫn: **ThS. Phạm Trọng Huynh**

Sinh viên thực hiện: **Nguyễn Tiến Bách MSSV: 0850080060**

**Ngô Phúc Thịnh MSSV: 0850080106**

**Hoàng Quang Tuấn MSSV: 0850080112**

Lớp : **08\_CNPM**

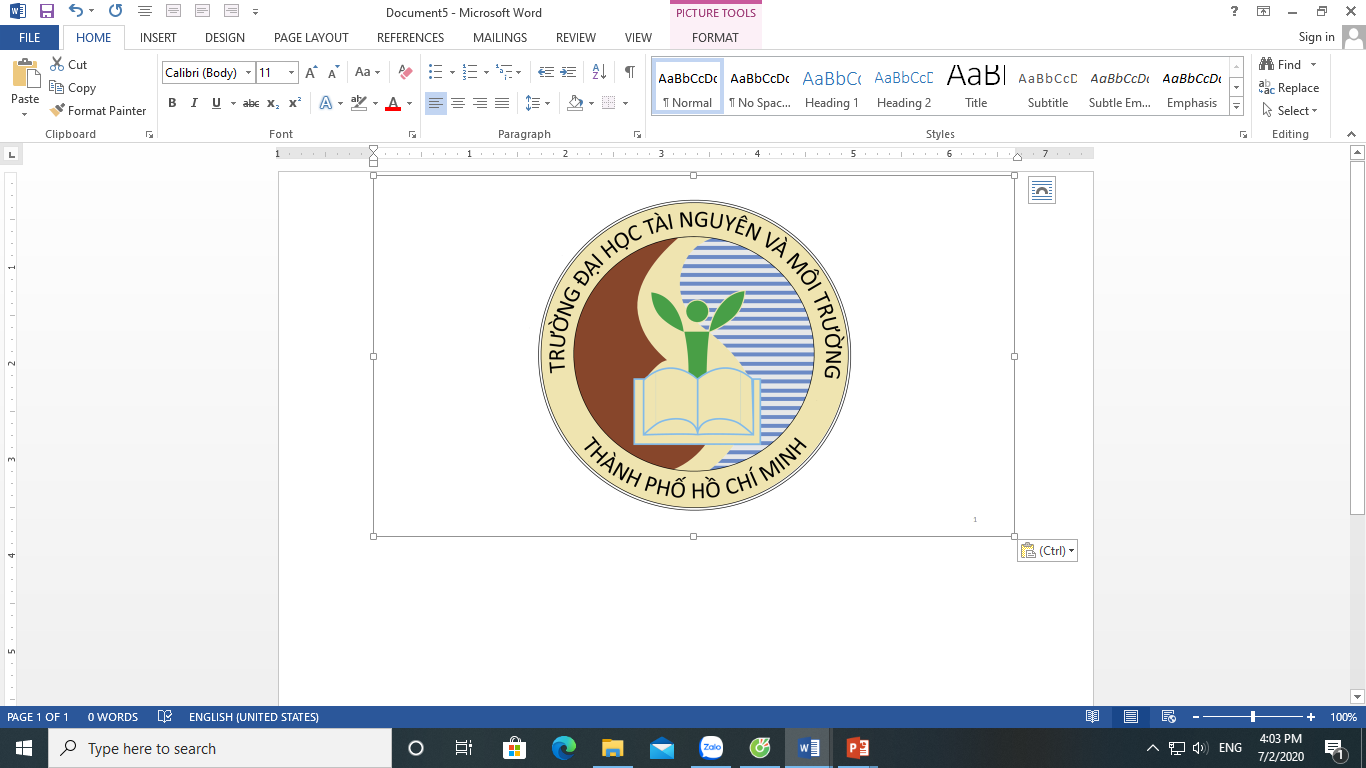
Khóa : **08**

**TP. Hồ Chí Minh, tháng 4 năm 2023**

**BỘ TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**ĐỒ ÁN MÔN HỌC**

**AN TOÀN BẢO MẬT MẠNG MÁY TÍNH VÀ HỆ THỐNG**

**NGHIÊN CỨU PHƯƠNG PHÁP NHẬN DIỆN KÝ TỰ HÌNH ẢNH VÀ XÂY DỰNG ỨNG DỤNG NHẬN DIỆN CHỨNG MINH NHÂN DÂN**

Giảng viênhướng dẫn: **ThS. Phạm Trọng Huynh**

Sinh viên thực hiện:  **Nguyễn Tiến Bách MSSV: 0850080060**

**Ngô Phúc Thịnh MSSV 0850080106**

**Hoàng Quang Tuấn MSSV: 0850080112**

Lớp : **08\_CNPM**

Khóa : **08**

**TP. Hồ Chí Minh, tháng 4 năm 2023**

**MỞ ĐẦU**

Ngày nay, tin học đã có những bước tiến nhanh chóng về ứng dụng của nó trong mọi lĩnh vực của cuộc sống trên phạm vi toàn thế giới nói chung và Việt Nam nói riêng, nó là ngành công nghệ kỹ thuật không thể thiếu trong việc áp dụng vào các hoạt động xã hội như: các hoạt động quản lý hệ thống, kinh doanh, trong hoạt động nghiên cứu,…Cùng với đó thì ta phải bảo mật những thông tin của mình để tránh bị mất mát thông tin, thông tin không bị sao chép, đánh cắp,…đồng thời dữ liệu đó khi đến tay người nhận một cách nhanh chóng và chính xác.

Việc sử dụng các hệ mật mã để bảo mật dữ liệu khi truyền tin trên mạng là rất cần thiết. Khoa học mật mã là ngành đã ra đời từ lâu và trải qua nhiều giai đoạn phát triển, từ một môn khoa học thực nghiệm đã nhanh chóng trở thành môn khoa học logic đỉnh cao và ngày càng hội tụ nhiều những kiến thức tinh túy của loài người

**LỜI CẢM ƠN**

Sau quá trình học tập và rèn luyện tại khoa Hệ Thống Thông Tin và Viễn Thám của Trường Đại Học Tài Nguyên Và Môi Trường TPHCM chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành đồ án môn học của mình.

Chúng em xin gửi lời cảm ơn chân thành đến thầy **ThS. Phạm Trọng Huynh** đã quan tâm hướng dẫn truyền đạt học những kiến thức và kinh nghiệm cho chúng em trong suốt thời gian học học tập bộ môn **Bảo mật Mạng máy tính và Hệ thống**

Mặc dù đã rất cố gắng hoàn thành công việc nhưng do thời gian có hạn và kĩ năng chưa cao nên việc phân tích thiết kế còn nhiều thiếu sót do đó trong quá trình làm đồ án môn không tránh khỏi được những sai sót, chúng em mong nhận được sự góp ý của thầy và các bạn để được hoàn thiện hơn.

**NHẬN XÉT**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

………….., ngày….tháng….năm……

NGƯỜI NHẬN XÉT

*(ký tên)*

**MỤC LỤC**

[Chương 1: TỔNG QUAN ĐỀ TÀI 1](#_Toc3125)

[1.1. Định nghĩa vấn đề 1](#_Toc19480)

[1.2. Lý do chọn đề tài 1](#_Toc3768)

[1.3. Nhận dạng ký tự quang học (OCR) 1](#_Toc4525)

[1.4. Tầm quan trọng của OCR 2](#_Toc29501)

[1.5. OCR hoạt động như thế nào 2](#_Toc28276)

[1.6. OCR có những loại nào 3](#_Toc16269)

[1.6.1. Phần mềm nhận dạng ký tự quang học đơn giản 3](#_Toc23733)

[1.6.2. Phần mềm nhận dạng ký tự thông minh 4](#_Toc2512)

[1.6.3. Nhận dạng từ thông minh 4](#_Toc4857)

[1.6.4. Nhận dạng ký hiệu quang học 4](#_Toc11135)

[1.7. Những lợi ích của OCR 4](#_Toc21917)

[1.8. Phương pháp triển khai đề tài 5](#_Toc3364)

[Chương 2: CƠ SỞ LÝ THUYẾT 7](#_Toc21018)

[2.1. Ngôn ngữ lập trình Python 7](#_Toc22791)

[2.1.1. Khái niệm 7](#_Toc14724)

[2.1.2. Quá trình phát triển 8](#_Toc5163)

[2.1.4. Đặc tính của Python 10](#_Toc15375)

[2.1.5. Tại sao ngôn ngữ Python lại phổ biến 11](#_Toc22601)

[2.2. Thư viện OpenCV 13](#_Toc21909)

[2.2.1. OpenCV là gì? 13](#_Toc32439)

[2.2.2. Lịch sử của OpenCV 14](#_Toc7025)

[2.2.3. Chức năng có trong thư viện OpenCV 14](#_Toc19282)

[2.2.4. Các module được dùng trong OpenCV 14](#_Toc32310)

[2.3. Giới thiệu về Tesseract 15](#_Toc24546)

[2.3.1. Công nghệ và cách thức hoạt động 16](#_Toc17924)

[2.4. Bài toán nhận diện khuôn mặt 17](#_Toc17310)

[2.4.1. Tìm kiếm khuôn mặt (Face Detection) 17](#_Toc1666)

[2.4.2. Trích rút đặc trưng (Feature Extraction) 18](#_Toc26541)

[2.5. Giới thiệu Anaconda 19](#_Toc28628)

[2.5.1. So sánh Anaconda và Python 19](#_Toc6365)

[Chương 3: THIẾT KẾ DỰ ÁN 22](#_Toc19450)

[3.1. Cài đặt các công cụ 22](#_Toc4071)

[3.1.1. Cài đặt Anaconda 22](#_Toc13443)

[3.1.2. Cài đặt OpenCV 27](#_Toc7414)

[3.1.3. Cài đặt python 29](#_Toc10871)

[3.1.4. Cài đặt Tesseract 32](#_Toc4177)

[3.2. Demo 33](#_Toc19169)

[Chương 4: KẾT LUẬN 39](#_Toc20813)

[4.1. Tổng kết 39](#_Toc27748)

[4.1.1. Những điểm đã làm được 39](#_Toc24063)

[4.1.2. Những điểm chưa làm được 39](#_Toc5821)

[4.2. Hướng phát triển: 39](#_Toc322)

[TÀI LIỆU THAM KHẢO 40](#_Toc2857)

**DANH MỤC HÌNH ẢNH**

*Hình 2.1 Tesseract*

*Hình 2.2 Quá trình Tesseract*

*Hình 2.3 Anaconda*

*Hình 3.1 Download Anaconda*

*Hình 3.2 Bắt đầu tải về*

*Hình 3.3 Giao diện sau khi tải xong*

*Hình 3.4 Tiến hành cài đặt*

*Hình 3.5 Đọc điều khoản và tiếp tục cài đặt*

*Hình 3.6 Chọn Next để tiếp tục*

*Hình 3.7 Install*

*Hình 3.8 Tiếp tục*

*Hình 3.9 Kết thúc*

*Hình 3.10 Kiểm tra*

*Hình 3.11 Cài đặt thành công*

*Hình 3.12 Kiểm tra version Python*

*Hình 3.13 pip search “opencv”*

*Hình 3.14 Install opencv-python*

*Hình 3.15 Kiểm tra cài đặt opencv*

*Hình 3.16 Cài đặt python*

*Hình 3.17 Bắt đầu cài đặt*

*Hình 3.18 Chọn checkbox và tiếp tục install*

*Hình 3.19 Đang cài đặt*

*Hình 3.20 Cài đặt thành công*

*Hình 3.21 Kiểm tra thư mục*

*Hình 3.22 Kiểm tra dòng lệnh Print*

*Hình 3.23 Cài đặt Tesseract bằng pip*

*Hình 3.24 Gọi API của Tesseract*

*Hình 3.25 Truyền dữ liệu ảnh đầu vào*

*Hình 3.26 Chuyển đổi màu ảnh sang trắng đen*

*Hình 3.27 Kết quả*

*Hình 3.28 Lưu thông tin dữ liệu*

# **Chương 1: TỔNG QUAN ĐỀ TÀI**

## Định nghĩa vấn đề

Ngày nay do tốc độ phát triển của các ngành khoa học kỹ thuật, đặc biệt là sự bùng nổ của lĩnh vực công nghệ thông tin. Cùng với sự phát triển của Internet, thì các công nghệ trí tuệ nhân tạo cũng là phần không thể thiếu trong xã hội ngày nay, nó là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh, phát hiện các đối tượng, tạo ảnh, siêu phân giải hình ảnh và nhiều hơn vậy. Dựa vào sự phát triển đó nhóm em chọn đề tài “Nghiên cứu phương pháp nhận diện ký tự hình ảnh và xây dựng ứng dụng nhận diện chứng minh nhân dân” để học và hiểu rõ hơn về các quy trình, hoạt động của ứng dụng và đồng thời cung tìm hiểu kĩ hơn về các thư viện sử dụng trong ứng dụng này

## Lý do chọn đề tài

Thấy được tầm quan trọng của hệ thống máy tính và phần mềm để định vị các đối tượng trong một hình ảnh và xác định từng đối tượng đối với nhu cầu của tất cả mọi người và hầu hết trên khắp cả nước. Việc sử dụng ứng dụng nhận diện chứng minh nhân dân thì giúp cho mọi người việc truy cập thông tin để xác nhận hoặc bắt các tình nghi phạm tội một cách nhanh chóng, thuận tiện và tiết kiệm thời gian hơn. Mặc dù ứng dụng nhận diện chứng minh nhân dân đã xuất hiện nhiều nhưng chúng em vẫn quyết định lựa chọn và đi sâu nghiên cứu rõ hơn về đề tài “Nghiên cứu phương pháp nhận diện ký tự hình ảnh và xây dựng ứng dụng nhận diện chứng minh nhân dân”. Chúng em cũng muốn được hiểu hơn và đồng thời thử sức với đề tài này

## Nhận dạng ký tự quang học (OCR)

- Nhận dạng ký tự quang học (OCR) là quá trình chuyển đổi một hình ảnh văn bản thành định dạng văn bản mà máy có thể đọc được. Ví dụ: nếu bạn quét một biểu mẫu hoặc biên lai, máy tính của bạn sẽ lưu bản quét đó dưới dạng tệp hình ảnh. Bạn không thể sử dụng trình soạn thảo văn bản để chỉnh sửa, tìm kiếm hoặc đếm số từ trong tệp hình ảnh. Tuy nhiên, bạn có thể sử dụng OCR để chuyển đổi hình ảnh thành tài liệu văn bản, trong đó phần nội dung sẽ được lưu trữ dưới dạng dữ liệu văn bản.[2]

* 1. **Tầm quan trọng của OCR**

- Hầu hết các luồng công việc kinh doanh sẽ bao gồm việc nhận thông tin từ các phương tiện truyền thông dạng bản in. Các biểu mẫu, hóa đơn dạng giấy, bản quét tài liệu pháp lý và bản in hợp đồng đều là một phần trong quy trình kinh doanh. Khối lượng giấy tờ lớn như vậy làm mất rất nhiều thời gian và không gian để lưu trữ và quản lý. Mặc dù quản lý tài liệu không cần giấy tờ là cách thức hiệu quả nhất nhưng việc quét tài liệu thành hình ảnh sẽ tạo ra nhiều thách thức. Quá trình này đòi hỏi phải có thao tác can thiệp thủ công và cũng có thể khá buồn tẻ và lề mề

- Hơn nữa, việc số hóa nội dung tài liệu này sẽ tạo ra các tệp hình ảnh với văn bản ẩn bên trong. Không thể xử lý văn bản trong hình ảnh bằng phần mềm xử lý văn bản theo cách giống như đối với tài liệu văn bản. Công nghệ OCR giải quyết vấn đề này bằng cách chuyển đổi hình ảnh văn bản thành dữ liệu văn bản mà phần mềm kinh doanh khác có thể phân tích. Sau đó, bạn có thể sử dụng dữ liệu để tiến hành phân tích, hợp lý hóa hoạt động, tự động hóa các quy trình và cải thiện năng suất.[2]

* 1. **OCR hoạt động như thế nào**

Quá trình hoạt động của OCR gồm các bước sau:

- Thu nhận hình ảnh: Một máy quét sẽ đọc tài liệu và chuyển đổi chúng thành dữ liệu nhị phân. Phần mềm OCR phân tích hình ảnh đã quét và phân loại vùng sáng làm nền và vùng tối làm văn bản

- Tiền xử lý ảnh: Trước tiên, phần mềm OCR sẽ làm sạch hình ảnh và loại bỏ các lỗi để chuẩn bị cho bước đọc. Sau đây là một số kỹ thuật làm sạch của phần mềm OCR:

+ Chỉnh thẳng hoặc nghiêng nhẹ tài liệu đã quét để khắc phục lỗi về căn chỉnh trong quá trình quét

+ Khử nhiễu đốm hoặc loại bỏ mọi đốm ảnh kỹ thuật số hay làm mịn các viền của hình ảnh văn bản.

+ Làm sạch đường viền khung và đường thẳng trong hình ảnh

+ Nhận dạng chữ viết cho công nghệ OCR đa ngôn ngữ

- Nhận dạng văn bản: Hai loại thuật toán OCR hoặc quy trình phần mềm chính mà phần mềm OCR sử dụng để nhận dạng văn bản được gọi là so khớp mẫu và trích xuất đặc điểm

- So khớp mẫu: Cách thức hoạt động của so khớp mẫu là tách biệt một hình ảnh ký tự, được gọi là hình dạng chữ và so sánh với một hình dạng chữ tương tự được lưu trữ. Tính năng nhận dạng mẫu chỉ hoạt động hiệu quả khi hình dạng chữ được lưu trữ có phông chữ và tỷ lệ tương tự với hình dạng chữ đầu vào. Phương thức này hoạt động tốt đối với hình ảnh quét từ tài liệu được đánh máy bằng phông chữ đã biết

- Trích xuất đặc điểm: Trích xuất đặc điểm sẽ chia nhỏ hoặc phân tách hình dạng chữ thành các đặc điểm như nét thẳng, nét vòng khép kín, hướng nét và giao điểm nét. Sau đó, hệ thống sử dụng các đặc điểm này để tìm kết quả phù hợp nhất hoặc kết quả gần đúng nhất trong số các hình dạng chữ khác nhau được lưu trữ.

- Hậu xử lý: Sau khi phân tích, hệ thống sẽ chuyển đổi dữ liệu văn bản được trích xuất thành tệp trên máy tính. Một số hệ thống OCR có thể tạo các tệp PDF có chú thích bao gồm cả phiên bản trước và sau của tài liệu được quét.[2]

* 1. **OCR có những loại nào**
     1. **Phần mềm nhận dạng ký tự quang học đơn giản**

- Một công cụ OCR đơn giản hoạt động bằng cách lưu trữ nhiều khuôn thức hình ảnh văn bản và phông chữ khác nhau dưới dạng mẫu. Phần mềm OCR sử dụng các thuật toán so khớp mẫu để so sánh các hình ảnh văn bản, theo từng ký tự một, với cơ sở dữ liệu nội bộ. Nếu hệ thống so khớp văn bản theo từng từ một thì sẽ được gọi là nhận dạng từ quang học. Giải pháp này có những hạn chế vì số lượng phông chữ và kiểu chữ viết tay là gần như vô hạn, cũng như không thể ghi lại hay lưu trữ tất cả kiểu loại trong cơ sở dữ liệu được.[2]

* + 1. **Phần mềm nhận dạng ký tự thông minh**

- Các hệ thống OCR hiện đại sử dụng công nghệ nhận dạng ký tự thông minh (ICR) để đọc văn bản hệt như cách con người đọc. Những hệ thống này sử dụng các phương thức nâng cao để đào tạo máy hoạt động giống như con người bằng cách sử dụng phần mềm máy học. Một hệ thống máy học được gọi là mạng nơ-ron phân tích văn bản qua nhiều cấp độ, xử lý hình ảnh lặp đi lặp lại. Hệ thống sẽ tìm kiếm các thuộc tính hình ảnh khác nhau, chẳng hạn như nét cong, nét thẳng, nét giao nhau và nét vòng, đồng thời kết hợp kết quả của tất cả các cấp độ phân tích khác nhau này để cho ra kết quả cuối cùng. Mặc dù ICR thường xử lý hình ảnh theo từng ký tự một nhưng quá trình này vẫn diễn ra nhanh chóng, thu được kết quả chỉ trong vài giây.[2]

* + 1. **Nhận dạng từ thông minh**

- Hệ thống nhận dạng từ thông minh hoạt động theo nguyên tắc giống như ICR, nhưng sẽ xử lý toàn bộ hình ảnh của từ thay vì tiền xử lý hình ảnh thành ký tự

* + 1. **Nhận dạng ký hiệu quang học**

- Nhận dạng ký hiệu quang học xác định logo, hình mờ và các biểu tượng văn bản khác trong tài liệu

* 1. **Những lợi ích của OCR**

- Văn bản có thể kiếm được: Các doanh nghiệp có thể chuyển đổi các tài liệu hiện có và tài liệu mới của họ thành một kho lưu trữ thông tin hoàn toàn có thể tìm kiếm được. Họ cũng có thể xử lý cơ sở dữ liệu văn bản tự động bằng cách sử dụng phần mềm phân tích dữ liệu để xử lý thông tin sâu hơn

- Hiệu quả hoạt động: Bạn có thể cải thiện hiệu quả bằng cách sử dụng phần mềm OCR để tự động tích hợp luồng công việc tài liệu và luồng công việc kỹ thuật số trong doanh nghiệp của bạn. Sau đây là một số ví dụ về những gì phần mềm OCR có thể thực hiện:

+ Quét các biểu mẫu điền tay để xác minh, xem xét, chỉnh sửa và phân tích tự động. Điều này giúp tiết kiệm thời gian cần thiết cho việc xử lý tài liệu và nhập dữ liệu thủ công

+ Tìm các tài liệu cần thiết bằng cách nhanh chóng tìm kiếm một cụm từ trong cơ sở dữ liệu để bạn không phải tự mình lục tìm trong thùng hồ sơ

+ Chuyển đổi ghi chú viết tay thành văn bản, tài liệu có thể chỉnh sửa

- Giải pháp trí tuệ nhân tạo: OCR thường là một phần của các giải pháp trí tuệ nhân tạo khác mà các doanh nghiệp có thể triển khai. Ví dụ: OCR trang bị trên xe ô tô tự lái để quét đọc biển số và biển báo, phát hiện logo thương hiệu trong các bài đăng trên mạng xã hội hoặc xác định bao bì sản phẩm trong hình ảnh quảng cáo. Công nghệ trí tuệ nhân tạo như vậy giúp các doanh nghiệp đưa ra những quyết định về tiếp thị và hoạt động tốt hơn, giảm chi phí và cải thiện trải nghiệm của khách hàng.[2]

* 1. **Phương pháp triển khai đề tài**

Đồ án này sử dụng thư viện OpenCV và Tesseract với ngôn ngữ mô tả là Python cùng với tệp tin XML Haarcascade để nhận diện chứng minh nhân dân realtime khi dùng webcam

- Python là ngôn ngữ thông dụng và phổ biến trong trí tuệ nhân tạo cũng như công nghệ học máy. Python là một ngôn ngữ có tính đơn giản, nhất quán và cho phép truy cập vào các thư viện và frameword hỗ trợ cho AI và học máy

- Thư viện OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. 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 cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần

- Tesseract là một thư viện OCR (Optical Character Recognition - Nhận diện ký tự quang học) mã nguồn mở, được phát triển bởi nhóm Google Research. Nó có khả năng chuyển đổi các tài liệu văn bản được quét hoặc chụp thành các đối tượng ký tự có thể xử lý bằng máy tính.[2]

**Chương 2: CƠ SỞ LÝ THUYẾT**

**2.1. Ngôn ngữ lập trình Python**

**2.1.1. Khái niệm**

**-** Python là ngôn ngữ lập trình hướng đối tượng bậc cao, dùng để phát triển website và nhiều ứng dụng khác nhau. Hiện nay, python đang được sử dụng rất rộng rãi trong các ứng dụng về trí tuệ nhân tạo, học máy, học sâu. Python được tạo ra bởi Guido và Rossum và được phát triển trong một dự án mã nguồn mở

- Với cú pháp cực kì đơn giản và dễ hiểu, dễ sử dụng, Python là sự lựa chọn hoàn hảo cho những ai lần đầu tiên học lập trình. Tuy nhiên, đây cũng là ngôn ngữ nổi tiếng về sự chặt chẽ, nhanh, mạnh và có mặt ở mọi hệ điều hành

- Python có 100 thư viện xây dựng sẵn để thực hiện các thuật toán Machine Learning và Deep Learning khác nhau. Vì vậy, mỗi khi bạn muốn chạy một thuật toán trên một tập dữ liệu, tất cả những gì bạn phải làm là cài đặt và tải các gói cần thiết bằng một lệnh duy nhất. Ví dụ về các thư viện được xây dựng trước bao gồm numPy, Keras, Tensolflow, Pytorch, ...

- Python có thể chạy trên nhiều nền tảng bao gồm Windows, MacOS, Linux, ... Trong khi chuyển mã từ nền tảng này sang nền tảng khác, bạn có thể sử dụng các gói như Pylnstaller sẽ xử lý mọi vấn đề phụ thuộc.

- Python có một cộng đồng người dùng khổng lồ luôn hữu ích khi chùng ta gặp phải vấn đề mã hóa. Ngoài lượng người hâm mộ đông đảo, Python có nhiều cộng đồng, nhóm và diễn đàn nơi các lập trình viên đăng các lỗi của họ và giúp đỡ lẫn nhau.

- Triển khai các ứng dụng trí tuệ nhân tạo liên quan đến hàng tấn thuật toán. Nhờ hỗ trợ của Pythons cho các gói được xác định trước, chúng ta không cần phải code các thuật toán. Và để làm cho mọi thứ dễ dàng hơn**, [học Python](https://mindx.edu.vn/blog/post/hoc-lap-trinh-python)** cung cấp phương pháp “kiểm tra khi code” giúp làm giảm số lượng code bạn phải gõ.[6]

**2.1.2. Quá trình phát triển**

Sự phát triển của Python có thể chia thành các giai đoạn

- Python 1: bao gồm các bản phát hành 1.x. Giai đoạn này, kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995, Guido làm việc tại CWI (*Centrum voor Wiskunde en Informatica* - Trung tâm Toán-Tin học tại Amsterdam, Hà Lan). Vì vậy, các phiên bản Python đầu tiên đều do CWI phát hành. Phiên bản cuối cùng phát hành tại CWI là 1.2. Vào năm 1995, Guido chuyển sang CNRI (*Corporation for National Research Initiatives*) ở Reston, Virginia. Tại đây, ông phát hành một số phiên bản khác. Python 1.6 là phiên bản cuối cùng phát hành tại CNRI. Sau bản phát hành 1.6, Guido rời bỏ CNRI để làm việc với các lập trình viên chuyên viết phần mềm thương mại. Tại đây, ông có ý tưởng sử dụng Python với các phần mềm tuân theo chuẩn GPL. Sau đó, CNRI và FSF (*Free Software Foundation* - Tổ chức phần mềm tự do) đã cùng nhau hợp tác để làm bản quyền Python phù hợp với GPL. Cùng năm đó, Guido được nhận Giải thưởng FSF vì Sự phát triển Phần mềm tự do (*Award for the Advancement of Free Software*). Phiên bản 1.6.1 ra đời sau đó là phiên bản đầu tiên tuân theo bản quyền GPL. Tuy nhiên, bản này hoàn toàn giống bản 1.6, trừ một số sửa lỗi cần thiết.

- Python 2: vào năm 2000, Guido và nhóm phát triển Python dời đến BeOpen.com và thành lập BeOpen PythonLabs team. Phiên bản Python 2.0 được phát hành tại đây. Sau khi phát hành Python 2.0, Guido và các thành viên PythonLabs gia nhập Digital Creations. Python 2.1 ra đời kế thừa từ Python 1.6.1 và Python 2.0. Bản quyền của phiên bản này được đổi thành Python Software Foundation License. Từ thời điểm này trở đi, Python thuộc sở hữu của Python Software Foundation (PSF), một tổ chức phi lợi nhuận được thành lập theo mẫu Apache Software Foundation

- Python 3, còn gọi là Python 3000 hoặc Py3K: Dòng 3.x sẽ không hoàn toàn tương thích với dòng 2.x, tuy vậy có công cụ hỗ trợ chuyển đổi từ các phiên bản 2.x sang 3.x. Nguyên tắc chủ đạo để phát triển Python 3.x là "bỏ cách làm việc cũ nhằm hạn chế trùng lặp về mặt chức năng của Python". Trong PEP (*Python Enhancement Proposal*) có mô tả chi tiết các thay đổi trong Python.Các đặc điểm mới của Python 3.0 sẽ được trình bày phần cuối bài này.

**2.1.3. Ứng dụng của Python**

- Python thường được sử dụng để phát triển trang web và phần mềm, tự động hóa tác vụ, phân tích dữ liệu và trực quan hóa dữ liệu. Vì tương đối dễ học, Python đã được nhiều người không phải là lập trình viên như kế toán và nhà khoa học áp dụng cho nhiều công việc hàng ngày, chẳng hạn như tổ chức tài chính.

****Phân tích dữ liệu và học máy****

- Python đã trở thành một yếu tố chính trong khoa học dữ liệu, cho phép các nhà phân tích dữ liệu và các chuyên gia khác sử dụng ngôn ngữ này để thực hiện các phép tính thống kê phức tạp, tạo trực quan hóa dữ liệu, xây dựng thuật toán [học máy](https://glints.com/vn/blog/hoc-machine-learning/" \t "https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/_blank), thao tác và phân tích dữ liệu cũng như hoàn thành các nhiệm vụ khác liên quan đến dữ liệu.

- Python có thể xây dựng nhiều dạng trực quan hóa dữ liệu khác nhau, chẳng hạn như biểu đồ đường và thanh, biểu đồ hình tròn, biểu đồ 3D. Python cũng có một số thư viện cho phép các lập trình viên viết chương trình để phân tích dữ liệu và học máy nhanh hơn và hiệu quả hơn, như TensorFlow và Keras.

****Phát triển web****

- Python thường được sử dụng để phát triển back-end của trang web hoặc ứng dụng những phần mà người dùng không nhìn thấy. Vai trò của Python trong phát triển web có thể bao gồm gửi dữ liệu đến và đi từ máy chủ, xử lý dữ liệu và giao tiếp với cơ sở dữ liệu, định tuyến URL và đảm bảo tính bảo mật. Python cung cấp một số khuôn khổ để phát triển web. Những cái thường được sử dụng bao gồm Django và Flask.

- Một số công việc phát triển web sử dụng Python bao gồm kỹ sư phụ trợ, nhà phát triển Python, kỹ sư phần mềm và kỹ sư DevOps.

****Tự động hoá và phát triển phần mềm****

- Nếu bạn thấy mình thực hiện một nhiệm vụ lặp đi lặp lại, bạn có thể làm việc hiệu quả hơn bằng cách tự động hóa nó bằng Python. Quá trình viết code được sử dụng để xây dựng các quy trình tự động này được gọi là viết script. Trong thế giới mã hóa, tự động hóa có thể được sử dụng để kiểm tra lỗi trên nhiều tệp, chuyển đổi tệp, thực hiện phép toán đơn giản và loại bỏ các bản sao trong dữ liệu.

- Python thậm chí có thể được sử dụng bởi những người mới bắt đầu để tự động hóa các tác vụ đơn giản trên máy tính—chẳng hạn như đổi tên tệp, tìm và tải xuống nội dung trực tuyến hoặc gửi email hoặc văn bản theo khoảng thời gian mong muốn. Trong phát triển phần mềm, Python có thể hỗ trợ các tác vụ như kiểm soát bản dựng, theo dõi lỗi và thử nghiệm. Với Python, các nhà phát triển phần mềm có thể tự động kiểm tra các sản phẩm hoặc tính năng mới. Một số công cụ Python được sử dụng để kiểm thử phần mềm bao gồm Green và Requestium.

**2.1.4. Đặc tính của Python**

- Python đang trở nên phổ biến trong cộng đồng lập trình nhờ có các đặc tính sau:

* Ngôn ngữ thông dịch: Python được xử lý trong thời gian chạy bởi Trình thông dịch Python.
* Ngôn ngữ hướng đối tượng: Nó hỗ trợ các tính năng và kỹ thuật lập trình hướng đối tượng.
* Ngôn ngữ lập trình tương tác: Người dùng có thể tương tác trực tiếp với trình thông dịch python để viết chương trình.
* Ngôn ngữ dễ học: Python rất dễ học, đặc biệt là cho người mới bắt đầu.
* Cú pháp đơn giản: Việc hình thành cú pháp Python rất đơn giản và dễ hiểu, điều này cũng làm cho nó trở nên phổ biến.
* Dễ đọc: Mã nguồn Python được xác định rõ ràng và có thể nhìn thấy bằng mắt.
* Di động: Mã Python có thể chạy trên nhiều nền tảng phần cứng có cùng giao diện.
* Có thể mở rộng: Người dùng có thể thêm các mô-đun cấp thấp vào trình thông dịch Python.
* Có thể cải tiến: Python cung cấp một cấu trúc cải tiến để hỗ trợ các chương trình lớn sau đó là shell-script.

### **2.1.5. Tại sao ngôn ngữ Python lại phổ biến**

- Python phổ biến vì nhiều lý do. Dưới đây là một cái nhìn sâu sắc hơn về những gì làm cho nó trở nên linh hoạt và dễ sử dụng cho các lập trình viên:

* Nó có cú pháp đơn giản bắt chước ngôn ngữ tự nhiên nên dễ đọc và dễ hiểu hơn. Điều này giúp xây dựng các dự án nhanh hơn và cải thiện chúng tốt hơn.
* Nó linh hoạt. Python có thể được sử dụng cho nhiều tác vụ khác nhau, từ phát triển web đến học máy.
* Nó thân thiện với người mới bắt đầu, khiến Python trở nên phổ biến đối với các lập trình viên mới vào nghề.
* Python là mã nguồn mở, có nghĩa là nó được sử dụng và phân phối miễn phí, kể cả cho mục đích thương mại.
* Kho lưu trữ các mô-đun và thư viện của Python—các gói mã mà người dùng bên thứ ba đã tạo để mở rộng khả năng của Python—rất rộng lớn và đang phát triển.
* Python có một cộng đồng lớn và tích cực đóng góp vào kho mô-đun và thư viện của Python, đồng thời hoạt động như một nguồn tài nguyên hữu ích cho các lập trình viên khác. Cộng đồng hỗ trợ rộng lớn có nghĩa là nếu các lập trình viên gặp khó khăn, việc tìm ra giải pháp là tương đối dễ dàng; ai đó chắc chắn cũng đã gặp phải vấn đề tương tự trước đây.

****Lợi thế của Python****

- Python là một ngôn ngữ dễ đọc và đơn giản để hiểu cho các nhà phát triển chưa bao giờ viết code. Do đó, cộng đồng người dùng Python không ngừng phát triển và lớn mạnh. Có rất nhiều học giả và giáo sư trong cộng đồng người dùng Python. Vì vậy, khi xảy ra sự cố, nhà phát triển có thể tập trung vào vấn đề đó và nhận trợ giúp từ những người khác trong cộng đồng mà không phải lo lắng về sự phức tạp của ngôn ngữ.

- Python là một ngôn ngữ lập trình miễn phí và mở. Giấy phép [nguồn mở](https://glints.com/vn/blog/open-source-la-gi/" \t "https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/_blank) được OSI phê chuẩn mà Python được phát triển theo đó làm cho Python trở thành ngôn ngữ tự do sử dụng và phân phối, kể cả cho mục đích thương mại. Nó sẽ làm giảm chi phí của bạn để bảo trì. Trong khi các nhà phát triển có thể chia sẻ, sao chép và thay đổi nó. Đối với cộng đồng Python, đây là cơ hội để chia sẻ kiến thức với các chuyên gia cấp dưới.

- Các lập trình viên nói rằng Python rất dễ sử dụng. Mặc dù khi xây dựng các ứng dụng hoặc trò chơi dành cho thiết bị di động, [C++](https://glints.com/vn/blog/lap-trinh-cpp-la-gi/" \t "https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/_blank) hoặc bất kỳ ngôn ngữ kịch bản điển hình nào khác có thể dễ sử dụng hơn, nhưng Python sẽ tốt hơn để vì nó dễ dàng xây dựng các ứng dụng phía máy chủ, tự động hóa hệ thống xây dựng và thu thập dữ liệu thử nghiệm.

- Python có nhiều lựa chọn thư viện và framework, đây là một trong những lợi thế lớn nhất của Python. Từ NumPy đến TensorFlow, thư viện Python được sử dụng cho mọi thứ từ trực quan hóa dữ liệu, học máy, khoa học dữ liệu, xử lý ngôn ngữ tự nhiên và phân tích dữ liệu phức tạp.

- Nó sở hữu một thư viện lớn với khả năng quản lý bộ nhớ và thiết kế trống giúp tăng năng suất của nhà phát triển Python. Nhờ đó, các nhà phát triển có thể quản lý cơ sở dữ liệu, tài liệu, trình duyệt web; thực hiện kiểm tra đơn vị và nhiều chức năng khác. Ngoài ra, Python có thể được sử dụng để phát triển nhiều tác vụ, chẳng hạn như phát triển ứng dụng web và máy tính để bàn, phát triển các hệ thống tính toán phức tạp, hệ thống quản lý hỗ trợ cuộc sống, [Internet of Things (IoT)](https://glints.com/vn/blog/nganh-iot-la-gi/" \t "https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/_blank), trò chơi, v.v.

**2.2. Thư viện OpenCV**

**2.2.1. OpenCV là gì?**

- OpenCV viết tắt của Open Source Computer Vision được xem là một trong những thư viện mã nguồn mở hàng đầu cho việc xử lý ảnh theo thời gian thực. OpenCV là thư viện miễn phí và hỗ trợ đa nền tảng cung cấp theo giấy phép BSD mã nguồn mở.

- OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các learning machine khác

- Thư viện OpenCV được phát hành với giấy phép BDS. Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV sở hữu giao diện thiên thiện với mọi loại ngôn ngữ lập trình, ví dụ như C++, C, Python hay Java… Ngoài ra, nó cũng dễ dàng tương thích với các hệ điều hành khác nhau, bao gồm từ Windows, Linux, Mac OS, iOS cho đến cả Android.

### **2.2.2. Lịch sử của OpenCV**

- OpenCV chính thức được ra mắt đầu tiên vào năm 1999, OpenCV là thư viện mã nguồn mở miễn phí cho cả học thuật và thương mại

- OpenCV hỗ trợ đa nền tảng gồm Windows, Linux, Mac OS, iOS và Android.

- OpenCV hỗ trợ ngôn ngữ lập trình C/ C++, Python và Java.

- OpenCV được viết bằng C / C ++ và tích hợp OpenCL.

- Thư viện có hơn 2500 thuật toán được tối ưu hóa.

### **2.2.3. Chức năng có trong thư viện OpenCV**

Thư viện OpenCV bao gồm một số tính năng nổi bật như:

- Bộ công cụ hỗ trợ 2D và 3D

- Nhận diện khuôn mặt

- Nhận diện cử chỉ

- Nhận dạng chuyển động, đối tượng, hành vi

- Tương tác giữa con người và máy tính

- Điều khiển Robot

- Hỗ trợ thực tế tăng cường

**2.2.4. Các module được dùng trong OpenCV**

OpenCV có cấu trúc module, tức là nó bao gồm cả những thư viện liên kết tĩnh lẫn thư viện liên kết động. Nắm rõ các module của OpenCV sẽ giúp bạn đọc hoàn toàn thấu hiểu OpenCV là gì.

- Core functionality (core): Module này sở hữu cơ chế rất nhỏ gọn. Nó được dùng để định hình các cấu trúc của cơ sở dữ liệu cơ bản, bao gồm cả những mảng đa chiều. Ngoài ra nó còn xác định các chức năng của những module đi kèm khác nữa.

- Image Processing (imgproc): Đây là module được dùng cho quá trình xử lý hình ảnh. Nó cho phép người dùng thực hiện các hoạt động như lọc hình ảnh tuyến tính và phi tuyến, thực hiện phép biến hình, thay đổi không gian màu, xây dựng biểu đồ và rất nhiều thao tác khác liên quan.

- Video Analysis (video): Giống như tên gọi của nó, module này cho phép phân tích các video. Kết quả được trả về bao gồm các ước tính chuyển động, thực hiện tách nền và các phép toán theo dõi vật thể

- Camera Calibration and 3D Reconstruction (calib3d): Module này cung cấp các thuật toán hình học đa chiều cơ bản và hiệu chuẩn máy ảnh single và stereo. Ngoài ra nó còn đưa ra các dự đoán kiểu dáng của đối tượng và sử dụng thuật toán thư tín âm thanh nổi cùng các yếu tố tái tạo 3D.

- 2D Features Framework (features2d): Module này giúp phát hiện các tính năng nổi trội của bộ nhận diện, bộ truy xuất thông số và thông số đối chọi.

- Ngoài ra còn có rất nhiều module khác với đa dạng tính năng, ví dụ như: FLANN, Google test wrapper…

**2.3. Giới thiệu về Tesseract**

- Tesseract là một Công cụ nhận dạng văn bản (OCR) nguồn mở, có sẵn theo giấy phép Apache 2.0. Nó có thể được sử dụng trực tiếp hoặc (dành cho lập trình viên) sử dụng API để trích xuất văn bản in từ hình ảnh. Nó hỗ trợ nhiều loại ngôn ngữ. Tesseract không có GUI tích hợp, nhưng có một số có sẵn từ [trang bên thứ 3](https://github.com/tesseract-ocr/tesseract/wiki/User-Projects-%E2%80%93-3rdParty?ref=nanonets.com) . Tesseract tương thích với nhiều ngôn ngữ lập trình và khuôn khổ thông qua các hàm bao có thể tìm thấy [tại đây](https://github.com/tesseract-ocr/tesseract/wiki/AddOns?ref=nanonets.com) . Nó có thể được sử dụng với phân tích bố cục hiện có để nhận dạng văn bản trong một tài liệu lớn hoặc có thể được sử dụng cùng với trình phát hiện văn bản bên ngoài để nhận dạng văn bản từ hình ảnh của một dòng văn bản.



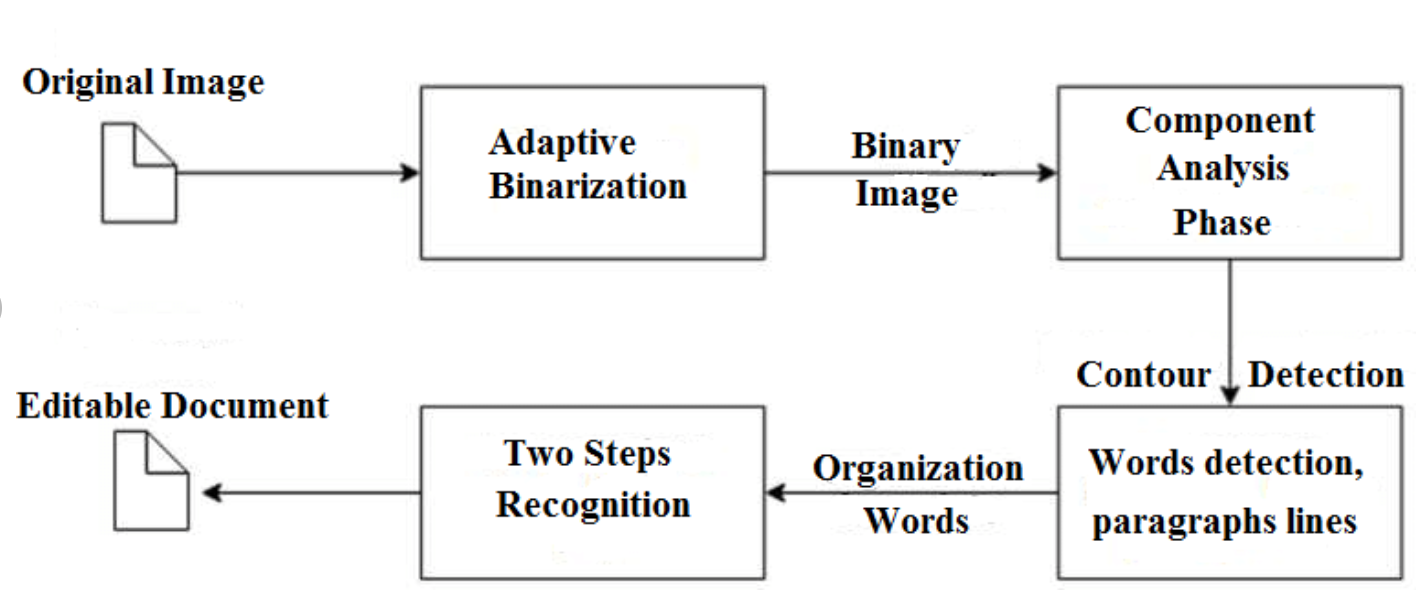
*Hình 2.1 Tesseract*

- Tesseract 4.00 bao gồm một hệ thống con mạng thần kinh mới được định cấu hình làm trình nhận dạng dòng văn bản. Nó có nguồn gốc từ triển khai [LSTM dựa trên Python của OCRopus](https://github.com/tmbdev/ocropy?ref=nanonets.com) nhưng đã được thiết kế lại cho Tesseract trong C++. Hệ thống mạng thần kinh trong Tesseract có trước TensorFlow nhưng tương thích với nó, vì có một ngôn ngữ mô tả mạng được gọi là Ngôn ngữ đặc tả đồ thị biến đổi (VGSL), ngôn ngữ này cũng có sẵn cho TensorFlow.

- Để nhận dạng một hình ảnh chứa một ký tự đơn lẻ, chúng tôi thường sử dụng Mạng thần kinh chuyển đổi (CNN). Văn bản có độ dài tùy ý là một chuỗi ký tự và các vấn đề như vậy được giải quyết bằng RNN và LSTM là một dạng RNN phổ biến. Đọc bài đăng này để tìm hiểu thêm về [LSTM](http://colah.github.io/posts/2015-08-Understanding-LSTMs/?ref=nanonets.com) . [1]

### **2.3.1. Công nghệ và cách thức hoạt động**

- Các LSTM rất giỏi trong việc học các trình tự nhưng chậm lại rất nhiều khi số lượng trạng thái quá lớn. Có những kết quả thực nghiệm cho thấy tốt hơn là yêu cầu một LSTM học một chuỗi dài hơn là một chuỗi ngắn gồm nhiều lớp. Tesseract được phát triển từ mô hình OCRopus trong Python, một nhánh của LSMT trong C++, được gọi là CLSTM. CLSTM là một triển khai của mô hình mạng thần kinh hồi quy LSTM trong C++, sử dụng thư viện Eigen để tính toán số.

*Hình 2.2 Quá trình Tesseract*

## **2.4. Bài toán nhận diện khuôn mặt**

Nhận dạng là một bài toán quan trọng trong lĩnh vực Computer Vision, thuật toán nhận dạng khuôn mặt là một bài toán khá phức tạp, nó đòi hỏi một loạt các vấn đề cần thực hiện:

- Việc làm đầu tiên đó là cần phải tìm kiếm tất cả những khuôn mặt có trong bức hình

- Tập trung vào từng khuôn mặt chắc chắn rằng bạn có thể nhận ra cùng một người từ các góc nhìn hoặc điều kiện sáng tối khác nhau.

- Lựa chọn những đặc trưng đặc trưng trên từng khuôn mặt -VD như đôi mắt to, khuôn mặt dài...

- So sánh những đặc trưng này với những người khác để chúng ta có thể biết được tên của họ.[1]

**2.4.1. Tìm kiếm khuôn mặt (Face Detection)**

- Có thể thấy rõ đầu tiên chúng ta cần xác định vị trí của khuôn mặt có trong bức hình trước khi xác định người đó là ai.

- Để tìm kiếm khuôn mặt trong một bức hình, ta cần làm cho bức hình trở thành ảnh đen trắng. Về cơ bản thì một bức ảnh màu không có tác dụng gì lắm cho việc xác định khuôn mặt.

- Sau đó, vẽ các vector chỉ hướng tối dần đi của các điểm ảnh của bức ảnh đen trắng đó. Tạo ra các khung tỉ lệ để xác định các đặc điểm cơ bản của khuôn mặt thông qua các vecto này.[1]

### **2.4.2. Trích rút đặc trưng (Feature Extraction)**

- Sau khi phát hiện ra khuôn mặt trong bức ảnh, chúng ta tiến hành trích rút những đặc trưng của khuôn mặt. Bước này trích xuất ra một vector đặc trưng đại diện cho một khuôn mặt. Nó phải đảm bảo được tính duy nhất của một khuôn mặt.

- Có những khuôn mặt thì nghiêng sang trái, nghiêng sang phải, khiến bạn chỉ nhìn thấy một phần khuôn mặt của họ, hay có những bức hình thì lại hơi nghiêng đầu, khiến cho bức ảnh bị lệch khỏi khung nhìn. Do đó, chúng ta chúng ta cần phải cố gắng *làm cong* bức hình để đôi mắt, bờ môi luôn ở vị trí mẫu trong ảnh. Điều này sẽ làm cho việc so sánh khuôn mặt ở các bước tiếp theo dễ dàng hơn nhiều.

- Để làm điều này, chúng ta sẽ sử dụng một thuật toán được gọi là face landmark estimation. Ý tưởng cơ bản của thuật toán là tìm ra 68 điểm cụ thể (được gọi là mốc) tồn tại trên mỗi khuôn mặt - đầu cằm, cạnh bên ngoài của mỗi mắt, cạnh bên trong của mỗi lông mày ... Sau đó, chúng ta sẽ dùng một thuật toán machine learning để training để có thể tìm thấy 68 điểm cụ thể trên từng khuôn mặt.[1]

## **2.5. Giới thiệu Anaconda**



*Hình 2.3 Anaconda*

- Anaconda là một nền tảng phân phối miễn phí của ngôn ngữ lập trình Python và R cho tính toán khoa học (khoa học dữ liệu, machine learning, xử lý dữ liệu lớn, phân tích dự đoán, v.v.), nhằm mục đích đơn giản hóa việc quản lý và triển khai gói. Nó có trên cả Windows, MacOS và Linux

- Anaconda chứa tất cả các gói (công cụ) phổ biến nhất mà một nhà khoa học dữ liệu cần. Các package trong Anaconda được quản lý bởi trình quản lý riêng của nền tảng này là **conda.** Ta thường dùng conda để tạo môi trường cô lập các dự án của mình, nhằm sử dụng các phiên bản Python khác nhau hoặc các phiên bản package khác nhau, cũng như dùng nó để cài đặt, gỡ cài đặt và cập nhật các package riêng trong từng dự án.[3]

**2.5.1. So sánh Anaconda và Python**

- **Sự khác biệt chính giữa Anaconda và Python là**: Anaconda bản chất là một nền tảng phân phối các ngôn ngữ lập trình **Python**và **R** cho các tác vụ khoa học dữ liệu và học máy trong khi Python đơn giản một ngôn ngữ lập trình.

- Anaconda được viết bằng Python, tuy nhiên, Conda là trình quản lý gói của bất kỳ package nào có thể được sử dụng trong môi trường của Anaconda trong khi pip, gói của trình quản lý Python, chỉ sử dụng cho các package được viết bằng Python.

- Anaconda được sử dụng chủ yếu cho khoa học dữ liệu và học máy, trong khi Python là một ngôn ngữ lập trình cũng được sử dụng cho nhiều mục đích khác nhau như lập trình web, desktop, mobile,....

- Khi bạn tải Anaconda, nó sẽ đi kèm với khoảng 600 module Python mở rộng, vốn không nằm trong bộ module chuẩn của Python. Các module này chủ yếu dùng cho các công việc liên quan đến khoa học dữ liệu. Bên cạnh đó, ta còn có Anaconda Prompt (chương trình command line), Spyder (code editor) và Anaconda Navigator (giúp mở Jupyter notebook một cách nhanh chóng).[3]

**2.5.2. Lợi ích khi sử dụng Anaconda**

**Anaconda không chỉ có Python**

- Anaconda không chỉ gồm có Python, mà còn có hơn 600 package mở rộng đi kèm, đó là các package cơ bản cho các tác vụ xử lý khoa học dữ liệu như NumPy, Pandas,.... Bạn không cần phải nhập các lệnh trên terminal để tải về nữa, chỉ cần tải Anaconda là có thể sử dụng ngay các package đó, giúp tiết kiệm khá nhiều thời gian cho chúng ta.

**Anaconda cài đặt mà không có quyền của quản trị**

- Bạn có thể cài Anaconda từ hầu hết máy tính mà không cần lo tới quyền admin, như là sử dụng máy tính ở thư viện trường,.... Điều này khiến cho Anaconda có thể cài đặt ở bất cứ nơi đâu, miễn là có máy tính.

**Anaconda hoạt động trên mọi nền tảng (Windows, Linux, MacOS)**

- Nếu bạn sử dụng MacOS, thì hầu hết các bản cài đặt MacOS đều đi kèm với Python. Vấn đề là phiên bản Python đi kèm với MacOS đã cũ (thường là Python legacy, Python 2) và phiên bản Python đi kèm với MacOS bị khóa, cần quyền quản trị để chạy. Vì phiên bản Python được cài sẵn trong MacOS yêu cầu đặc quyền của quản trị viên, sẽ dễ xảy ra lỗi trong quá trình chạy các ứng dụng Python.

- Sử dụng Anaconda thì nó sẽ sử dụng một phiên bản Python tách biệt với Python có sẵn trong Mac, mọi thứ sẽ hoạt động trơn tru, ta làm việc mà chẳng lo lỗi.

**Anaconda giúp quản lý gói và môi trường ảo dễ dàng hơn**

- Một trong những ưu điểm lớn nhất của Anaconda là quản lý môi trường ảo (virtual environment) dễ dàng. Nếu bạn mới dùng Anaconda lần đầu tiên, có thể bạn chưa quan tâm lắm đến vấn đề này, tuy nhiên khi bắt đầu với một số project, bạn sẽ thấy virtual environment rất quan trọng, giúp ta tránh xung đột về phiên bản Python cũng như phiên bản package của project nếu có mang đi sang máy khác.[3]

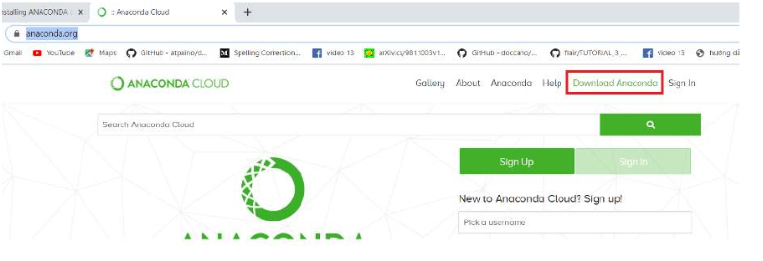
# **Chương 3: THIẾT KẾ DỰ ÁN**

## 3.1. Cài đặt các công cụ

### **3.1.1. Cài đặt Anaconda**

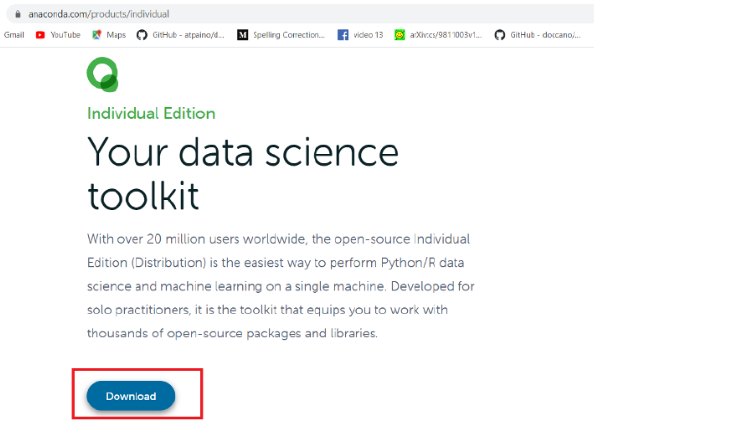
+ **Bước 1:**Vào trang web của Anaconda https://anaconda.org/

+ **Bước 2:** Chọn Download Anaconda



*Hình 3.1 Download Anaconda*

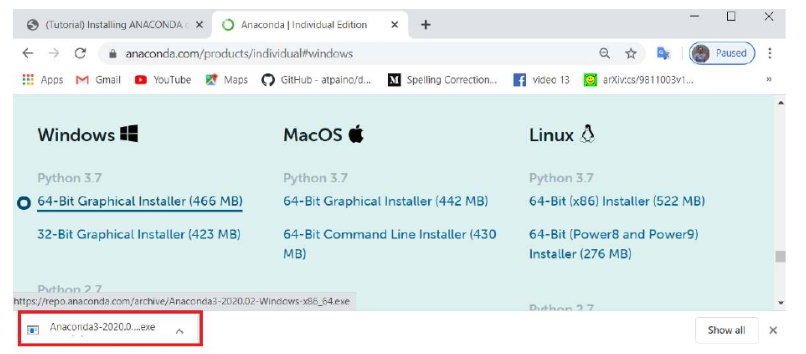
+ **Bước 3:** Chọn nút Download

****

*Hình 3.2 Bắt đầu tải về*

Chọn phiên bản Anaconda phù hợp với máy bạn.

- Nếu bạn dùng hệ điều hành window - 64 bit, bạn chọn 64-Bit Graphical Installer (466 MB) giống hình bên dưới

- Nếu bạn dùng hệ điều hành window - 32 bit, bạn chọn 32-Bit Graphical Installer (423 MB)  
 

*Hình 3.3 Giao diện sau khi tải xong*

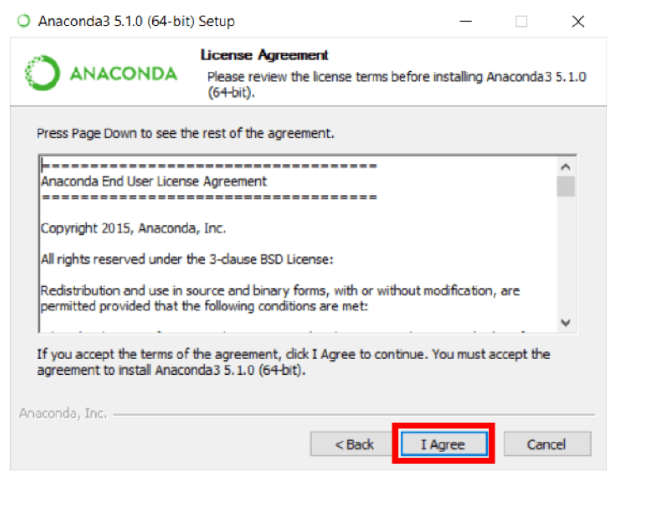
+ **Bước 4: Cài Đặt**

Sau khi Download xong bộ cài bạn tiến hành cài đặt

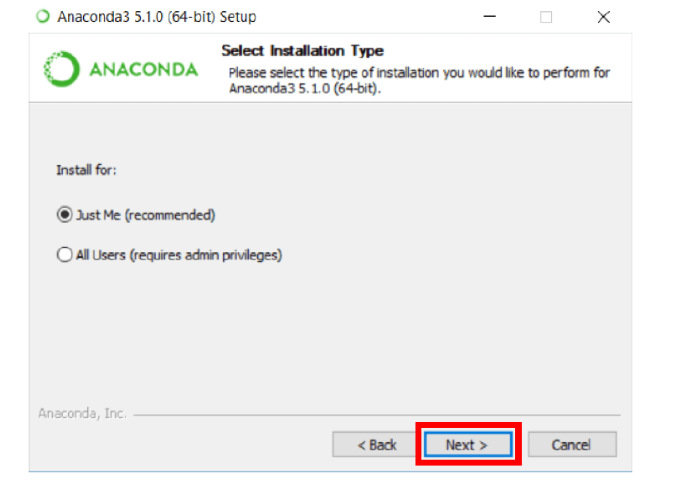
Khi màn hình sau xuất hiện bạn chọn next 

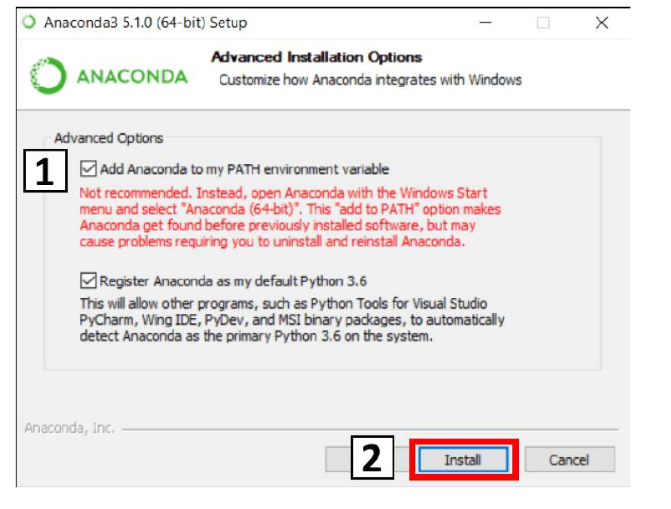
*Hình 3.4 Tiến hành cài đặt*

Đọc điều khoản sử dụng và click chọn I Agree

*Hình 3.5 Đọc điều khoản và tiếp tục cài đặt*

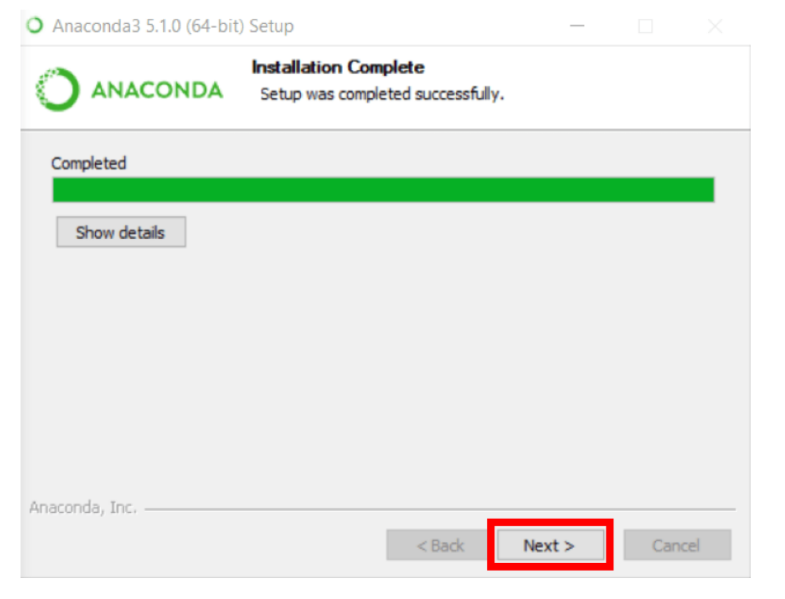
Chọn Next

*Hình 3.6 Chọn Next để tiếp tục*

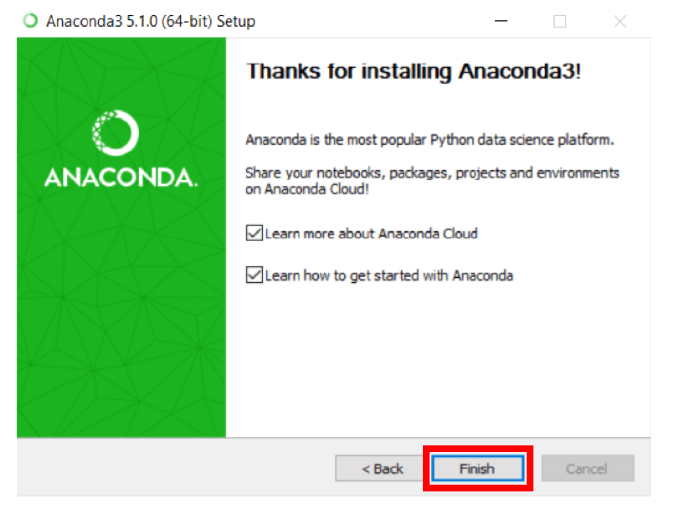
Nhấn chọn Add Anaconda to my PATH environment variable (mặc dù không được gợi ý), sau đó chọn Install 

*Hình 3.7 Install*

Chọn Next

*Hình 3.8 Tiếp tục*

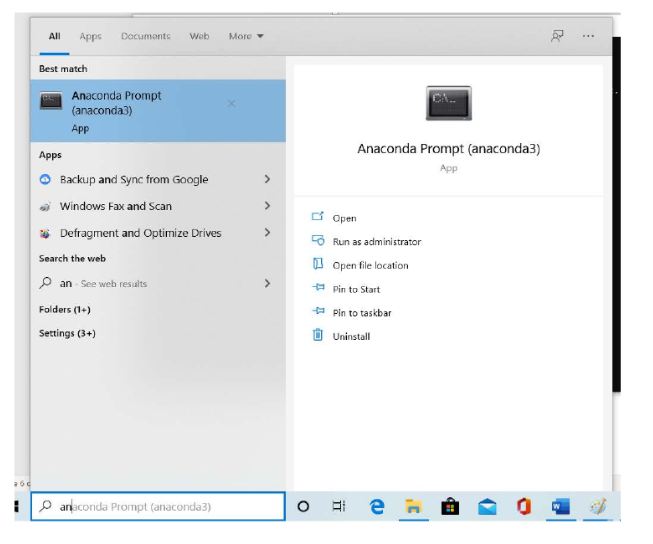
Chọn Finish



*Hình 3.9 Kết thúc*

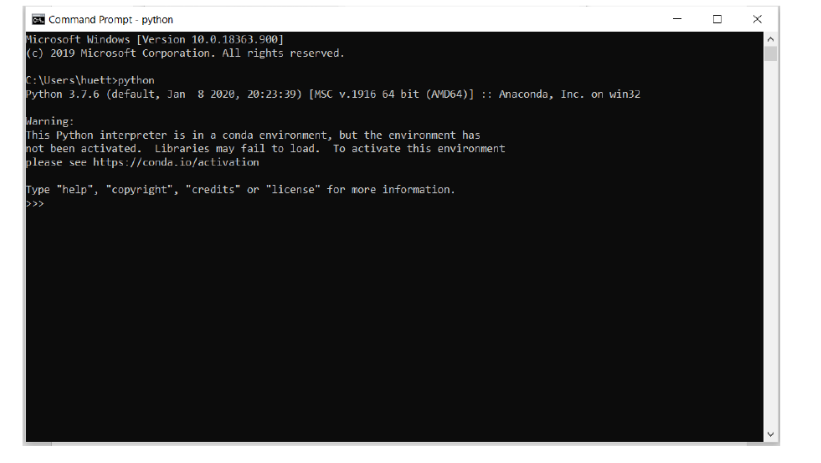
Kiểm tra cài đặt thành công

Vào cửa sổ hộp run bên cạnh window button ◊ gõ anaconda ◊ chọn Anaconda Prompt



*Hình 3.10 Kiểm tra*

Khi cửa sổ màn hình đen xuất hiện bạn gõ python vào vị trí dấu nhắc lệnh, nếu thấy màn hình làm việc với python xuất hiện là bạn đã cài đặt thành công

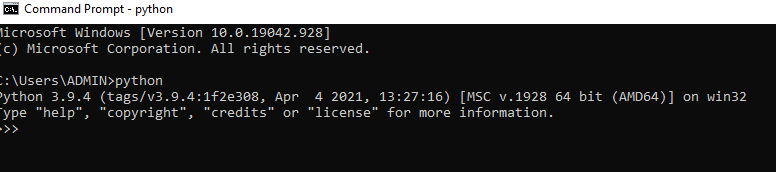


*Hình 3.11 Cài đặt thành công*

**3.1.2. Cài đặt OpenCV**

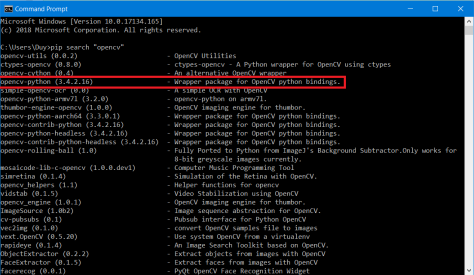
Tiến hành cài đặt OpenCV với ngôn ngữ lập trình Python

Trước khi cài đặt thì mình kiểm cần phải kiểm tra version của **Python**hiện tại của PC của mình bằng câu lệnh: python –version

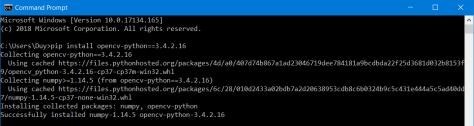


*Hình 3.12 Kiểm tra version Python*

Pip là hệ thống quản lý các package của Python. Mình có thể tìm kiếm các package mà pip hỗ trợ thông qua lệnh pip search.pip search ”opencv”

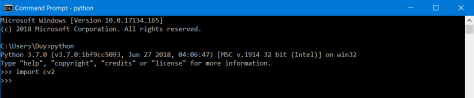


Hình 3.13 pip search “opencv”

Như hình trên mình thấy, pip hiện tại hỗ trợ package **opencv-python** (phiên bản 3.4.2.16), nên mình có thể cài đặt OpenCV cho Python thông qua pip. Sử dụng lệnh **pip install** để cài đặt OpenCV.pip install opencvpython==3.4.2.16

*Hình 3.14 Install opencv-python*

Kiểm tra lại, nếu có thể import được OpenCV mà không báo lỗi thì việc cài đặt thành công. Import cv2



*Hình 3.15 Kiểm tra cài đặt opencv*

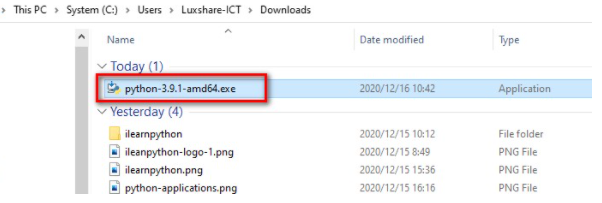
### **3.1.3. Cài đặt python**

**Bước 1 :** Tải file cài đặt tại trang web : <https://www.python.org/downloads/>



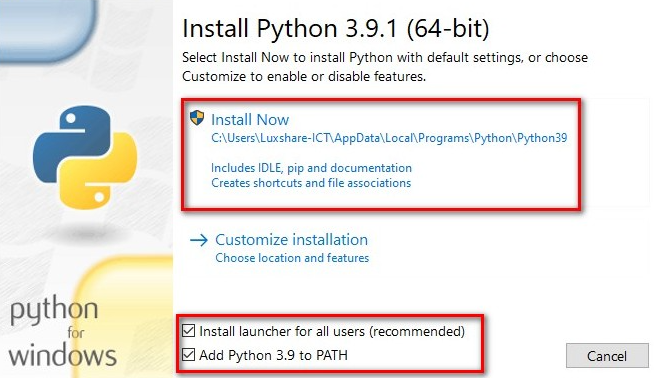
*Hình 3.16 Cài đặt python*

**Bước 2:** Sau khi Download file cài đặt thành công, bạn click vào file : python-3.9.1- amd64.exe để cài đặt .



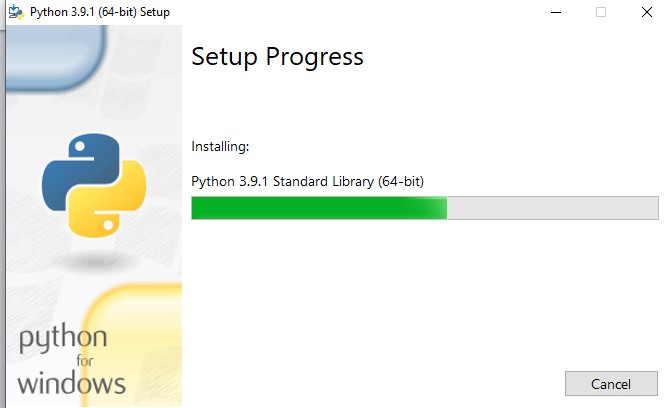
*Hình 3.17 Bắt đầu cài đặt*

**Bước 3 :** Tích chọn vào 2 ô checkbox và nhấn Install Now



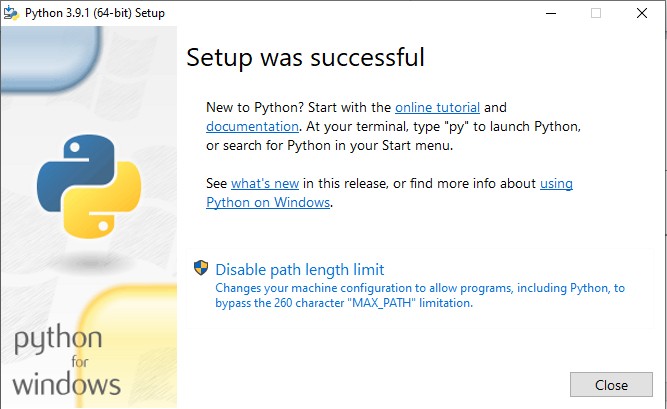
*Hình 3.18* ***Chọn checkbox và tiếp tục install***

**Bước 4:** Chờ quá trình cài đặt



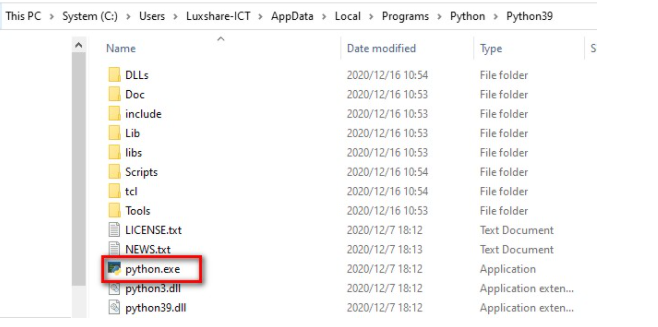
*Hình 3.19 Đang cài đặt*

**Bước 5 :** Quá trình cài đặt thành công như hình bên dưới



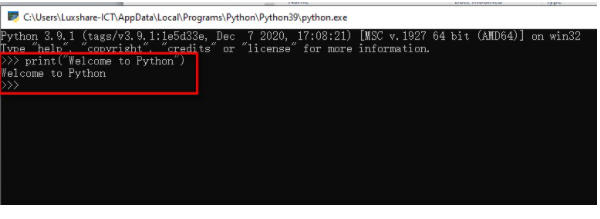
*Hình 3.20 Cài đặt thành công*

**Bước 6:** Kiểm tra sau khi cài đặt: Vào thư mục C:\Users\Luxshare-ICT\AppData\Local\Programs\Python\Python39  sau đó chọn file python.exe



*Hình 3.21 Kiểm tra thư mục*

**Bước 7:** Kiểm tra chạy dòng lệnh :  print (“Welcome to Python”)  nếu in kết là: Welcome to Ptyhon . Nghĩa là chúng ta đã cài đặt thành công

*Hình 3.22 Kiểm tra dòng lệnh Print*

### **3.1.4. Cài đặt Tesseract**

- Tải xuống Tesseract từ trang web chính thức: <https://github.com/tesseract-ocr/tesseract>. Cài đặt Tesseract trên máy tính của bạn bằng cách chạy trình cài đặt.

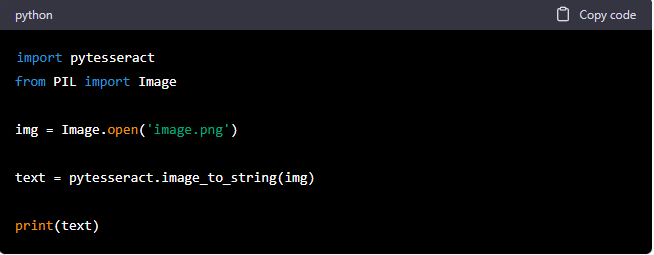
- Để sử dụng Tesseract trong các ứng dụng lập trình, bạn cần cài đặt các gói phụ thuộc như: Leptonica (một thư viện xử lý hình ảnh), và các gói ngôn ngữ OCR cho các ngôn ngữ bạn muốn hỗ trợ.

- Nếu bạn đang sử dụng Python, bạn có thể cài đặt Tesseract bằng pip



*Hình 3.23 Cài đặt Tesseract bằng pip*

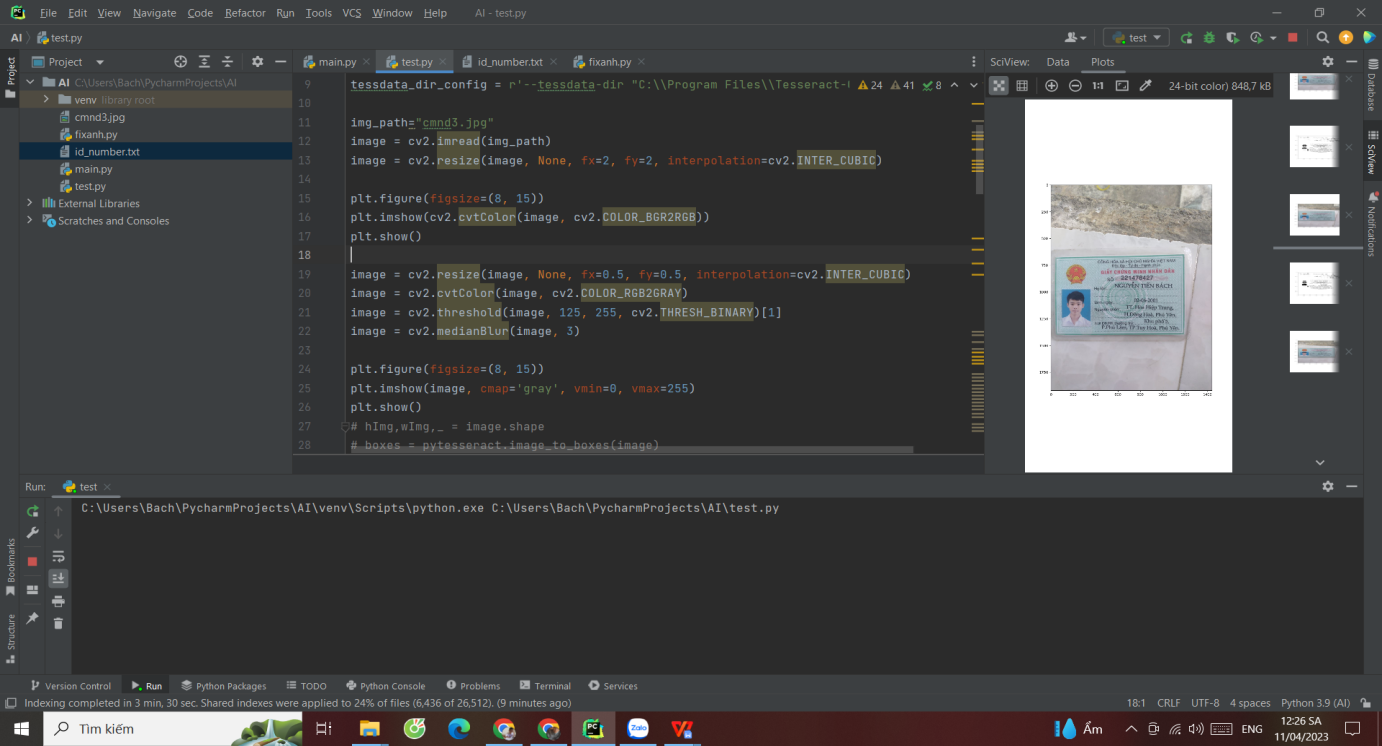
Sau khi cài đặt, bạn có thể sử dụng Tesseract trong ứng dụng của mình bằng cách gọi API của Tesseract. Ví dụ:



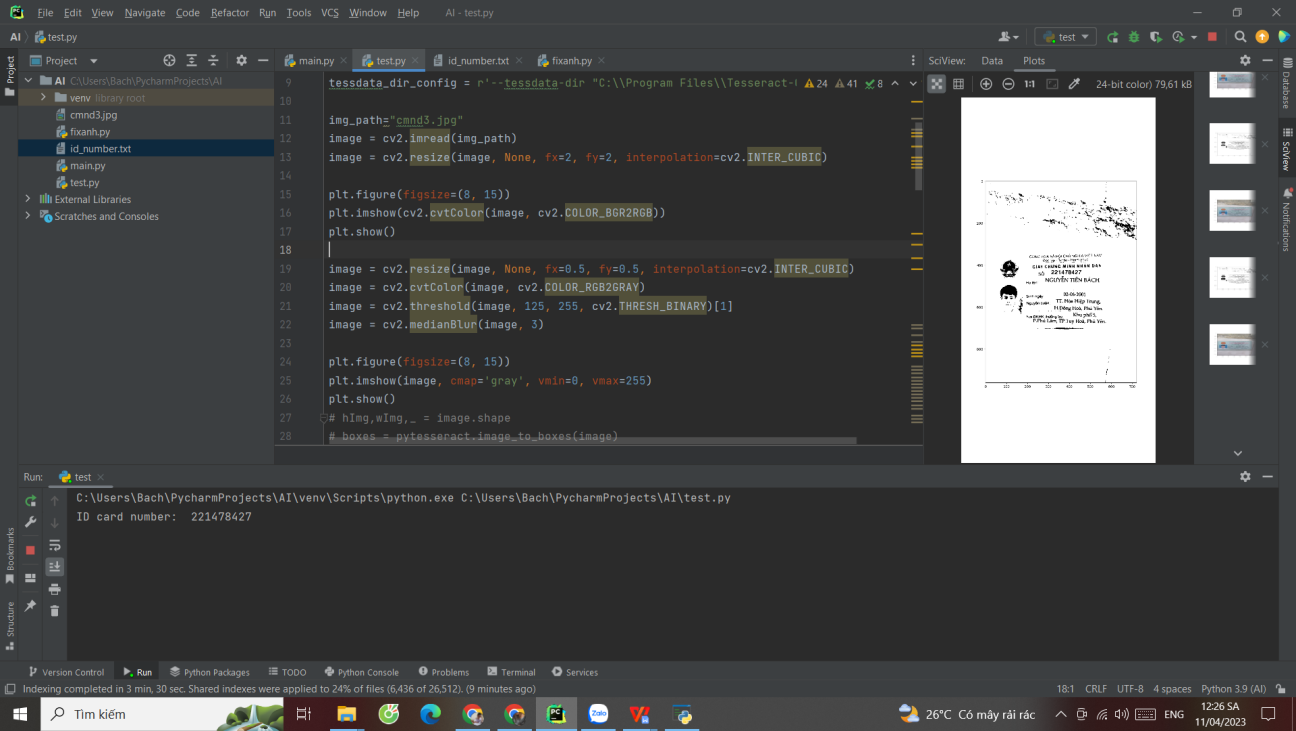
*Hình 3.24 Gọi API của Tesseract*

## **3.2. Demo**

Đầu tiên ta truyển dữ liệu hình ảnh căn cước công dân để nhận diện và đọc dữ liệu

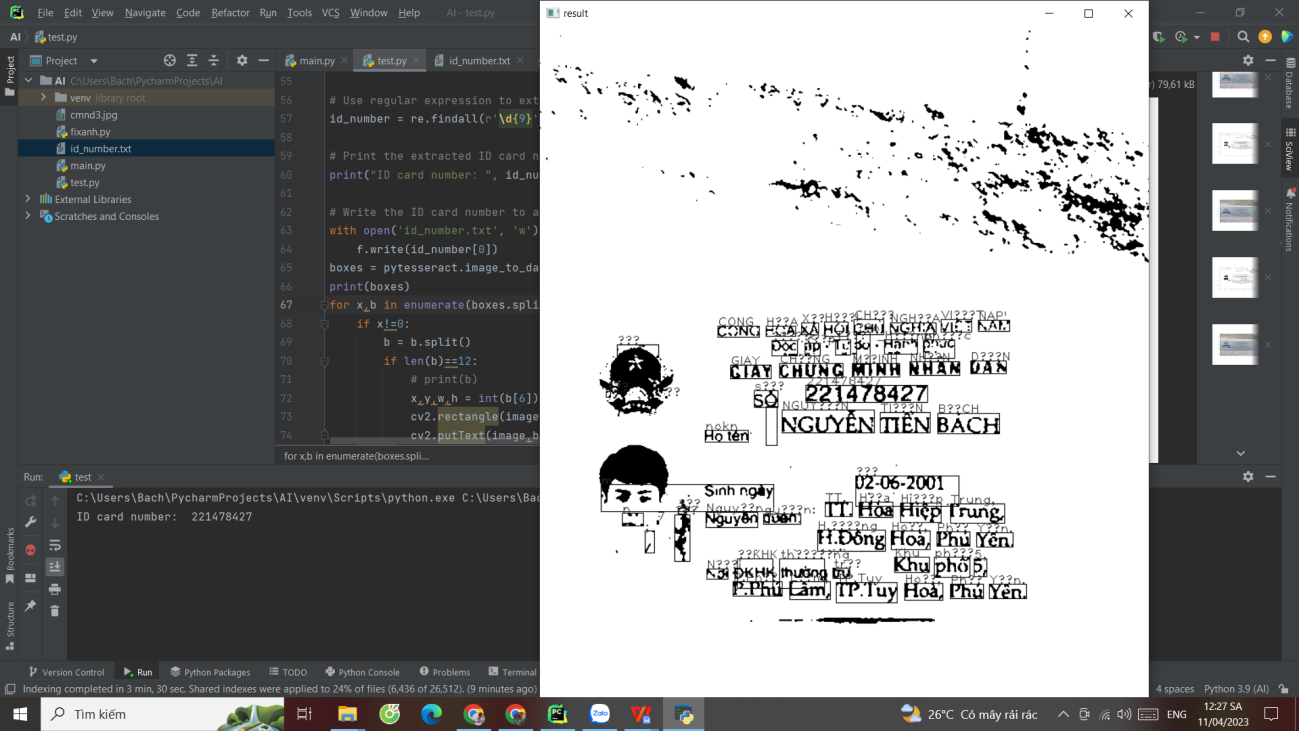
*Hình 3.25 Truyền dữ liệu ảnh đầu vào*

Tiếp theo ta chuyển đổi màu ảnh sang trắng đen để phân biệt các bệt ảnh bị lỗi từ đó ta có thể fix độ sáng nhằm để cho máy đọc dữ liệu text



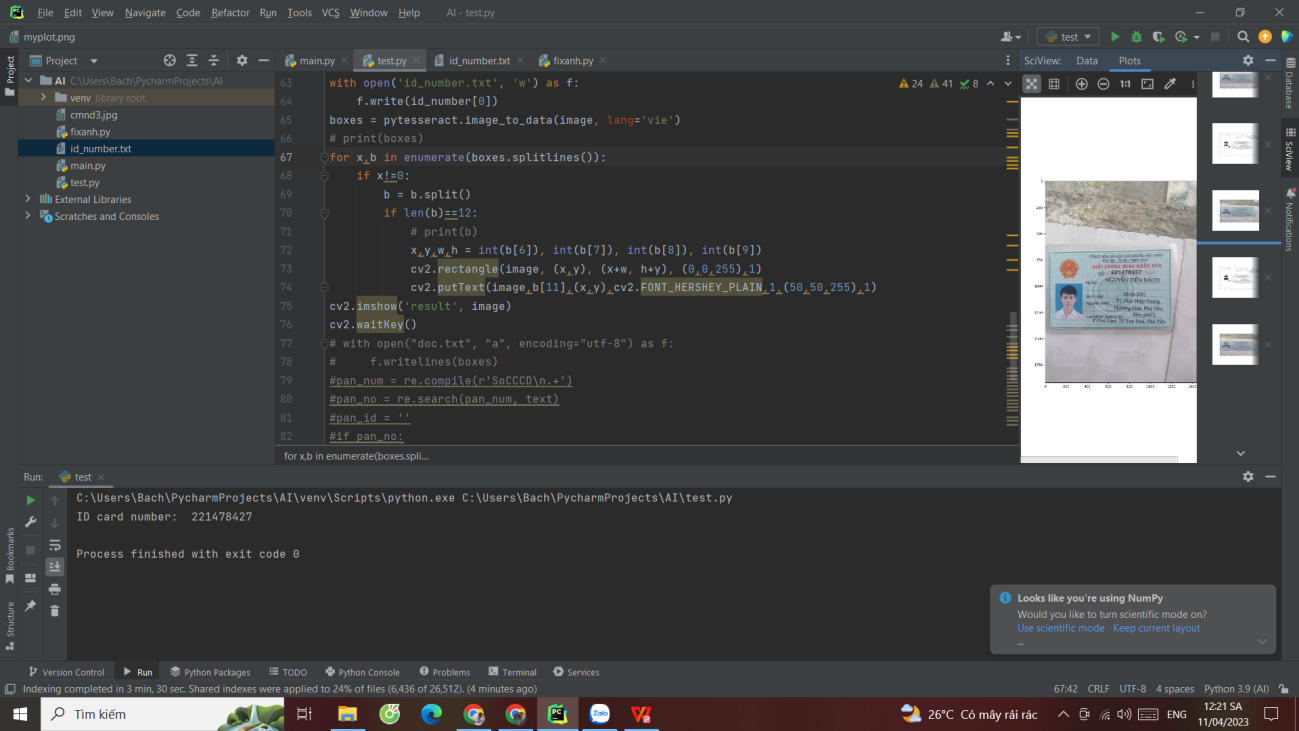
*Hình 3.26 Chuyển đổi màu ảnh sang trắng đen*

Đây là kết quả khi máy đọc xong dữ liệu ảnh đầu vào ta được một kết quả là thông tin của thẻ CMND



*Hình 3.27 Kết quả*

Cuối cùng ta lưu lại dữ liệu đã được đọc trên vào tệp id\_number.txt

*Hình 3.28 Lưu thông tin dữ liệu*

- Import các thư viện cần thiết: cv2: Thư viện OpenCV để xử lý hình ảnh và video. pytesseract: Thư viện để nhận dạng ký tự trong hình ảnh. PIL: Thư viện Pillow để làm việc với hình ảnh. matplotlib.pyplot: Thư viện để hiển thị hình ảnh và đồ thị. numpy: Thư viện để làm việc với mảng và ma trận

- Thiết lập đường dẫn đến tesseract OCR và tessdata: Dòng pytesseract.pytesseract.tesseract\_cmd="C:\\Program Files\\Tesseract-OCR\\tesseract" thiết lập đường dẫn tới tesseract OCR. Dòng tessdata\_dir\_config = r'--tessdata-dir "C:\\Program Files\\Tesseract-OCR\\tessdata" -l vie' thiết lập đường dẫn đến thư mục tessdata chứa dữ liệu ngôn ngữ (trong trường hợp này là tiếng Việt)

- Đọc và hiển thị hình ảnh: img\_path = "cmnd3.jpg" là đường dẫn đến file ảnh. image = cv2.imread(img\_path) đọc ảnh từ đường dẫn. image = cv2.resize(image, None, fx=2, fy=2, interpolation=cv2.INTER\_CUBIC) thay đổi kích thước ảnh. Sử dụng matplotlib.pyplot để hiển thị ảnh gốc

- Xử lý ảnh để tách lấy số CMND: image = cv2.resize(image, None, fx=0.5, fy=0.5, interpolation=cv2.INTER\_CUBIC) thay đổi kích thước ảnh. image = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY) chuyển ảnh sang ảnh xám. image = cv2.threshold(image, 125, 255, cv2.THRESH\_BINARY)[1] áp dụng ngưỡng nhị phân để làm sáng các ký tự trên nền đen. image = cv2.medianBlur(image, 3) làm mờ ảnh để làm giảm nhiễu

- Tìm contours và lấy số CMND: contours, hierarchy = cv2.findContours (image, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE) tìm contours trong ảnh. largest\_contour = max(contours, key=cv2.contourArea) tìm contour có diện tích lớn nhất. mask = np.zeros(image.shape[:2], dtype=np.uint8) tạo mặt nạ để tách số CMND. cv2.drawContours(mask, [largest\_contour], 0, 255, -1) vẽ contour lên mặt nạ. `id\_card = cv2.bitwise\_and(image, image, mask=mask)` áp dụng mặt nạ lên ảnh để tách lấy số CMND

- Nhận dạng số CMND từ ảnh: text = pytesseract.image\_to\_string(id\_card, lang='vie', config='--psm 6') sử dụng pytesseract để nhận dạng ký tự trong ảnh. id\_number = re.findall(r'\d{9}', text) sử dụng regular expression để trích xuất số CMND từ kết quả nhận dạng. In ra số CMND đã trích xuất

- Ghi số CMND vào file văn bản: with open('id\_number.txt', 'w') as f: mở file văn bản để ghi. f.write(id\_number[0]) ghi số CMND vào file. Hiển thị vùng chứa số CMND trên ảnh: boxes = pytesseract.image\_to\_data(image, lang='vie') sử dụng pytesseract để nhận dạng và trả về thông tin về vị trí của các ký tự

- Dùng vòng lặp để duyệt qua thông tin từng ký tự: b = b.split() tách thông tin của ký tự thành các phần.x,y,w,h = int(b[6]), int(b[7]), int(b[8]), int(b[9]) lấy vị trí và kích thước của ký tự.cv2.rectangle(image, (x,y), (x+w, h+y), (0,0,255),1) vẽ hình chữ nhật xung quanh ký tự.

cv2.putText(image,b[11],(x,y),cv2.FONT\_HERSHEY\_PLAIN,1,(50,50,255),1) đặt số hiệu của ký tự lên ảnh.

Hiển thị ảnh kết quả:

cv2.imshow('result', image) hiển thị ảnh với các ký tự được đánh số và vùng chứa số CMND được đánh dấu.

cv2.waitKey() để chờ người dùng nhấn một phím bất kỳ để thoát.

Mã này sử dụng thư viện OpenCV, pytesseract và các thư viện hỗ trợ khác để xử lý ảnh CMND, tách lấy số CMND và hiển thị kết quả lên ảnh

- Trong lĩnh vực xử lý ảnh và nhận dạng ký tự, việc trích xuất thông tin từ hình ảnh là một bài toán quan trọng. Trong bài viết này, chúng ta sẽ tìm hiểu về một đoạn mã Python dùng để trích xuất số CMND từ một ảnh chứa thông tin CMND.

- Mã nguồn bắt đầu bằng việc import các thư viện cần thiết như OpenCV, pytesseract, PIL (thuộc thư viện Pillow), matplotlib.pyplot và numpy. Đầu tiên, chúng ta cần cài đặt và thiết lập pytesseract để sử dụng các chức năng nhận dạng ký tự.

- Tiếp theo, chúng ta cung cấp đường dẫn đến file ảnh cần xử lý, sau đó đọc và hiển thị ảnh gốc. Mục đích của bước này là để người dùng có thể xem trước ảnh trước khi tiến hành xử lý.

- Tiếp theo, chúng ta tiến hành xử lý ảnh để tách lấy số CMND. Đầu tiên, ảnh được thu nhỏ để làm giảm kích thước và tốc độ xử lý. Sau đó, ảnh được chuyển đổi sang ảnh xám, áp dụng ngưỡng nhị phân để làm sáng các ký tự trên nền đen, và được làm mờ để giảm nhiễu.

- Tiếp theo, chúng ta sử dụng OpenCV để tìm các đường viền trong ảnh và xác định đường viền lớn nhất, chính là đường viền của CMND. Sau đó, chúng ta tạo một mặt nạ để tách lấy khu vực chứa số CMND bằng cách vẽ đường viền lên mặt nạ.

- Sau khi có mặt nạ, chúng ta sử dụng pytesseract để nhận dạng số CMND từ khu vực đã tách lấy. Kết quả nhận dạng được lưu vào biến text. Tiếp theo, chúng ta sử dụng regular expression để trích xuất số CMND từ kết quả nhận dạng.

- Sau khi có số CMND, chúng ta in số CMND đã trích xuất và ghi vào một file văn bản để lưu trữ thông tin.

- Cuối cùng, chúng ta hiển thị kết quả lên ảnh. Một khung chứa số CMND được vẽ xung quanh và số hiệu của từng ký tự cũng được đánh số trên ảnh. Điều này giúp người dùng có thể dễ dàng nhận biết và kiểm tra kết quả trích xuất.

Tóm lại, đoạn mã trên giúp chúng ta xử lý ảnh CMND và trích xuất số CMND từ ảnh đó. Việc này được thực hiện bằng cách sử dụng các thư viện như OpenCV và pytesseract để xử lý ảnh, tìm kiếm và trích xuất thông tin từ hình ảnh.

- Việc trích xuất số CMND từ ảnh có thể được ứng dụng trong nhiều lĩnh vực như nhận dạng thông tin cá nhân, tự động đăng ký thông tin, kiểm tra chính sách bảo hiểm và nhiều ứng dụng khác. Đoạn mã trên cung cấp một cơ sở để tiếp tục phát triển và ứng dụng trong các dự án thực tế.

- Chúng ta có thể tùy chỉnh và mở rộng đoạn mã này để phù hợp với các yêu cầu cụ thể của dự án, ví dụ như sử dụng các phương pháp xử lý ảnh khác, tăng cường độ chính xác của việc nhận dạng, xử lý các trường hợp ngoại lệ, và thực hiện các thao tác khác để trích xuất thông tin từ ảnh.

- Hy vọng rằng thông qua bài viết này, bạn đã hiểu rõ hơn về cách thức hoạt động và ứng dụng của đoạn mã xử lý ảnh và nhận dạng ký tự trong trường hợp trích xuất số CMND từ hình ảnh.

# **Chương 4: KẾT LUẬN**

## 4.1. Tổng kết

**4.1.1. Những điểm đã làm được**

- Quá trình tìm hiểu và thực hiện đề tài trải qua rất nhiều giai đoạn khác nhau để đi sâu vào nghiên cứu đề tài, nhờ sự giúp đỡ, định hướng của thầy cô và các bạn, em đã hoàn thành đồ án của mình. Qua đồ án, em đã học được những kiến thức từ thực tiễn và kiến thức chuyên ngành

+ Hoàn thiện các chức năng chính như nhận diện được chứng minh nhân dân, xác nhận tuổi, xác nhận giới tính, lưu vào cơ sở dữ liệu

+ Đã tìm hiểu được thư viện OpenCV, Tesseract và ngôn ngữ Python

+ Tìm hiểu được cách hoạt động nhận diện chứng minh nhân dân hoặc căn cước công dân.

### **4.1.2. Những điểm chưa làm được**

**-** Vì kiến thức còn hạn chế nên việc tổ chức bài toán chưa tối ưu, gặp nhiều khó khăn trong quá trình làm đồ án. Sản phẩm phát triển được chỉ theo mô phỏng mô hình tham khảo, chưa đầy đủ các chức năng và tối ưu thuật toán.

+ Tỉ lệ nhận diện đúng chưa quá cao

+ Kết quả đạt được chưa như mong muốn kỳ vọng của nhóm

## 4.2. Hướng phát triển:

Với đề tài “Nghiên cứu phương pháp nhận diện ký tự hình ảnh và xây dựng ứng dụng nhận diện chứng minh nhân dân”, em mong muốn học sâu hơn về kiến thức xử lí ảnh, phát triển đề tài hoàn thiện hơn đáp ứng được nhu cầu cho cuộc sống hiện nay. Có thể phát triển thêm nhiều chức năng hơn, chỉnh sửa ảnh tốt hơn, bổ sung nhiều chức năng hơn. Nghiên cứu các kĩ thuật xử lí ảnh, cải thiện hiệu suất cũng như tỉ lệ chính xác của việc nhận diện chứng minh nhân dân.

# **TÀI LIỆU THAM KHẢO**

[1] Tran Van Huy, Nhận diện văn bản với Tesseract OCR

[2] Nhận dạng chữ viết từ ảnh - Character Recognition Python P1 - tesseract OCR python\_https://www.youtube.com/watch?v=p2wdxQzSks8

[3] https://freetuts.net/cai-dat-anaconda-3416.html

[4] https://github.com/huytranvan2010/Tesseract-Guide

[5] https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/#.ZG2Jj3ZBy3A