**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**

**Khoa Công Nghệ Thông Tin**



**ĐỒ ÁN CƠ SỞ 5**

**NHẬN DẠNG BIỂN SỐ XE**

**SỬ DỤNG THƯ VIỆN OPENCV**

Sinh viên thực hiện : **BÙI HỮU HẢO**

**NGUYỄN HỮU DŨNG**

Lớp : **18IT4**

Giảng viên hướng dẫn : **TS LÊ THỊ THU NGA**

***Đà nẵng, tháng 06 năm 2021***

**ĐẠI HỌC ĐÀ NẴNG**

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



**ĐỒ ÁN CƠ SỞ 5**

**NHẬN DẠNG BIỂN SỐ XE**

**SỬ DỤNG THƯ VIỆN OPENCV**

Sinh viên thực hiện : **BÙI HỮU HẢO**

**NGUYỄN HỮU DŨNG**

Lớp : **18IT4**

Giảng viên hướng dẫn : **TS LÊ THỊ THU NGA**

***Đà Nẵng, tháng 06 năm 2021***

**NHẬN XÉT**

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

*Đà Nẵng, ngày tháng 05 năm 2021*

**Giảng viên hướng dẫn**  
  
  
  
 *TS. Lê Thị Thu Nga*

**LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành đến tất cả các thầy cô giảng viên của Trường Đại học Công nghệ Thông tin và Truyền Thông Việt - Hàn nói chung và cô Lê Thị Thu Nga nói riêng đã cung cấp cho chúng em kiến thức, kỹ năng cần thiết, hướng dẫn cho chúng em hoàn thành tốt đề tài này.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, đồ án này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, góp ý kiến của các thầy cô để Chúng em có điều kiện bổ sung, nâng cao kiến thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[**DANH MỤC CÁC BẢNG** 5](#_Toc76492709)

[**LỜI MỞ ĐẦU** 6](#_Toc76492710)

[**CHƯƠNG I: TỔNG QUAN** 8](#_Toc76492711)

[**1.** **Tổng quan về bài toán nhận dạng:** 8](#_Toc76492712)

[**1.1** **Định nghĩa về bài toán nhận dạng:** 8](#_Toc76492713)

[**1.2** **Khó khăn trong việc nhận dạng** 8](#_Toc76492714)

[**2.** **Tổng quan về bài toán nhận dạng biển số xe:** 8](#_Toc76492715)

[**3.** **Ngôn ngữ lập trình Python** 9](#_Toc76492716)

[**3.1** **Lịch sử ngôn ngữ lập trình python :** 9](#_Toc76492717)

[**3.2** **Giới thiệu và những đặc điểm của Python :** 9](#_Toc76492718)

[**3.3** **Những ứng dụng của Python:** 11](#_Toc76492719)

[**4.** **Thư viện OpenCV (OpenSource Computer Vision)** 11](#_Toc76492720)

[**CHƯƠNG II: CƠ SỞ LÝ THUYẾT** 13](#_Toc76492721)

[**1.** **Xử lý ảnh và các vấn đề cơ bản trong xử lý ảnh:** 13](#_Toc76492722)

[**1.1** **Xử lý ảnh là gì ?** 13](#_Toc76492723)

[**1.2** **Định nghĩa về xử lý ảnh:** 13](#_Toc76492724)

[**1.3** **Các bước chính trong xử lý ảnh số:** 14](#_Toc76492725)

[**1.4 Ứng dụng của xử lý ảnh :** 15](#_Toc76492726)

[**2.** **Thuật toán KNN (K-Neareast Neighbors)** 16](#_Toc76492727)

[**3.** **Bài toán nhận diện biển số xe** 17](#_Toc76492728)

[**3.1 Mô hình đề xuất** 17](#_Toc76492729)

[**3.2 Các quá trình xử lý trong hệ thống** 18](#_Toc76492730)

[**CHƯƠNG III: XÂY DỰNG ỨNG DỤNG VÀ TRIỂN KHAI CÀI ĐẶT** 21](#_Toc76492731)

[**1.** **Cài đặt ngôn ngữ lập trình Python và Thư viện OpenCV** 21](#_Toc76492732)

[**2.** **Kết quả** 22](#_Toc76492733)

[**KẾT LUẬN** 27](#_Toc76492734)

[**1.** **Kết luận** 27](#_Toc76492735)

[**2.** **Hướng phát triển** 27](#_Toc76492736)

[**TÀI LIỆU THAM KHẢO** 28](#_Toc76492737)

**DANH MỤC CÁC BẢNG**

[*Hình 1: Logo của Python* 10](#_Toc76492226)

[*Hình 2: Mô tả về ứng dụng của thư viện OpenCV* 12](#_Toc76492227)

[*Hình 3: Quá trình xử lý ảnh* 13](#_Toc76492228)

[*Hình 4: Các bước cơ bản trong một hệ thống xử lý ảnh* 13](#_Toc76492229)

[*Hình 5: Các bước chính trong xử lý ảnh số* 15](#_Toc76492230)

[*Hình 6: Hoạt động của thuật toán KNN* 17](#_Toc76492231)

[*Hình 7: Mô hình ứng dụng* 18](#_Toc76492232)

[*Hình 8: Sơ đồ tổng quát của khối xác định vùng chứa biển số xe* 18](#_Toc76492233)

[*Hình 9: Sơ đồ khối tách biển số* 19](#_Toc76492234)

[*Hình 10: Các giai đoạn trong việc phân đoạn từng ký tự* 20](#_Toc76492235)

[Hình 11: *Giao diện Download Python* 21](file:///C:\Users\Administrator\Downloads\đồ%20án%20cơ%20sở%205.docx#_Toc76492236)

[*Hình 12: Giao diện tải về thư viện OpenCV* 21](#_Toc76492237)

[*Hình 13: Hình mẫu* 22](#_Toc76492238)

[*Hình 14: Ảnh imgGrayscaleScene* 23](#_Toc76492239)

[*Hình 15: Ảnh imgThreshScen* 23](#_Toc76492240)

[*Hình 16: Ảnh chứa những ký tự có thể là biển số* 24](#_Toc76492241)

[*Hình 17: Ảnh chưa những chuỗi ký tự phù hợp* 24](#_Toc76492242)

[*Hình 18: Hình ảnh sau khi trích xuất* 25](#_Toc76492243)

[*Hình 19: Hình ảnh sau khi xử lý khung* 25](#_Toc76492244)

[*Hình 20: Hình ảnh sau khi tìm chuỗi phù hợp* 25](#_Toc76492245)

[*Hình 21: Hình ảnh sau khi nhận diện ký tự trong khung* 26](#_Toc76492246)

[*Hình 22: Kết quả thu được* 26](#_Toc76492247)

**LỜI MỞ ĐẦU**

1. **Lý do chọn đề tài**

Trong những năm gần đây, xã hội phát triển thì khoa học ngày càng phát triển, từ cuộc cách mạng công nghiệp đầu tiên cho đến bây giờ là cách mạng công nghiệp 4.0. Là cuộc cách mạng mà cốt lõi là công nghệ thông tin với các thiết bị thông minh như robot, xe tự hành, những ứng dụng nhận diện để tăng năng suất lao động và phục vụ cho nhu cầu đời sống con người.

Song song với sự phát triển kinh tế thì đã kéo theo sự bùng nổ của cơ sở hạ tầng giao thông và số lượng phương tiện giao thông. Qua đó gây ra những khó khăn trong công tác quản lý mà nguồn nhân lực con người khó có thể đảm đương được.

Từ những vấn đề thực tế trên, ở bài đồ án cơ sở 5 thì tôi đã nghiên cứu và quyết định lựa chọn bài toán “Nhận dạng biển số xe sử dụng thư viện OpenCV” để có thể phục vụ trong công tác quản lý phương tiện giao thông cũng như áp dụng vào công tác bảo vệ quản lý xe ở các khu vực trông giữ xe.

1. **Mục tiêu của đề tài:**

Ứng dụng hoàn thành có thể áp dụng vào giao thông, giúp quá trình quản lí các phương tiện cũng như quản lí các bãi giữ xe được tốt hơn.

1. **Cấu trúc báo cáo**

Chương I: Tổng quan

1. Tổng quan về bài toán nhận dạng
2. Tổng quan về bài toán nhận diện biển số xe
3. Ngôn ngữ lập trình python
4. Thư viện opencv

Chương II: Cơ sở lí thuyết

1. Xử lí ảnh và các vấn đề cơ bản trong xử lí ảnh
2. Mô hình nhận dạng biển số
3. Đánh giá mô hình

Chương III: Xây dựng ứng dụng và triển khai cài đặt

1. Cài đặt ngôn ngữ python và thư viện openv
2. Quy trình thực hiện
3. Kết quả đạt được

Kết luận và hướng phát triển

**CHƯƠNG I: TỔNG QUAN**

1. **Tổng quan về bài toán nhận dạng:**
2. **Định nghĩa về bài toán nhận dạng:**

Nhận dạng là quá trình phân loại các đối lượng được biểu diễn theo mô hình nào đó và gán cho chúng vào một lớp dựa theo những quy luật và các mẫu chuẩn.

Quá trình nhận dạng dựa vào mẫu học biết trước gọi là nhận dạng có thầy ( supervised learning) và trường hợp còn lại là học không thầy ( non supervised learning)

Nhận dạng là một bài toán quan trọng trong ngành thị giác máy tính

1. **Khó khăn trong việc nhận dạng**

Đối với con người, việc nhân dạng trong ảnh không phải là việc khó khăn, nhưng đối với một hệ thống nhân tạo thì việc nhân dạng đòi hỏi phải giải quyết rất nhiều vấn đề như:

* Sự xuất hiện hoặc thiếu một số thành phần: Các thành phần miêu tả đối tượng có thể xuất hiện hoặc không trong hình ảnh làm cho bài toán nhận dạng trở nên khó khăn hơn nhiều.
* Tư thế, góc chụp: Ảnh chụp có thể thay đổi rất nhiều khi thay đổi góc chụp giữa camera với đối tượng. Chẳng hạn như chụp thẳng, chụp xéo...
* Sự biến dạng của đối tượng: Biến dạng của đối tượng cũng có thể làm ảnh hưởng đến các thông số của đối tượng đó.
* Sự che khuất: Đối tượng có thể bị che khuất bởi các đối tượng khác.
* Sự phức tạp của hình nền: Hình nền phức tạp cũng sẽ khiến cho việc nhận dạng trở nên khó khăn.
* Điều kiện của ảnh: Ảnh chụp trong các điều kiện khác nhau về ánh sáng, camera,...ảnh hưởng rất nhiều đến chất lượng của ảnh

1. **Tổng quan về bài toán nhận dạng biển số xe:**

Hệ thống nhận dạng biển số xe là hệ thống có khả năng phân tích hình ảnh và xác định biển số trên xe, thông qua video, thiết bị ghi hình và hình ảnh.

Có nhiều cách thức khác nhau để phân loại ứng dụng nhận diện biển số xe. Trong đó cách phân loại dựa trên mục đích sử dụng là thông dụng nhất và có thể chia ứng dụng nhận dạng biển số xe thành hai loại sau:

* Loại 1: Giới hạn vùng nhìn
* Loại 2: Không giới hạn vùng nhìn

Để giải quyết được bài toán nhận dạng biển số xe, yêu cầu đặt ra là phải giải quyết được 3 bài toán con:

* Bài toán thứ nhất: xác định vùng chứa biển số xe và tách biển số.
* Bài toán thứ hai: phân đoạn từng ký tự.
* Bài toán thứ ba: nhận dạng ký tự.

1. **Ngôn ngữ lập trình Python**
   1. **Lịch sử ngôn ngữ lập trình python :**

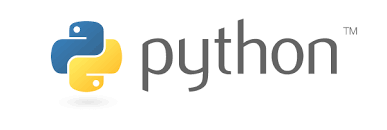
Python đã được hình thành vào cuối những năm 1980, và việc thực hiện nó vào tháng 12 năm 1989 bởi Guido van Rossum tại Centrum Wiskunde & Informatica (CWI) ở Hà Lan như là một kế thừa cho ngôn ngữ ABC (tự lấy cảm hứng từ SETL) có khả năng xử lý ngoại lệ và giao tiếp với Hệ điều hành Amoeba. Van Rossum là tác giả chính của Python, và vai trò trung tâm của ông trong việc quyết định hướng phát triển của Python.

Python 2.0 đã được phát hành vào ngày 16 tháng 10 năm 2000 và có nhiều tính năng mới, bao gồm bộ thu gom rác theo chu kỳ (cycle-detecting garbage) và hỗ trợ Unicode. Với việc phát hành này quá trình phát triển đã được thay đổi và trở nên minh bạch hơn và cộng đồng hậu thuẫn

Python 3.0 được phát hành năm 2008, sau một thời gian dài thử nghiệm.Cho tới năm 2017, Python đang có phiên bản 3.7

* 1. **Giới thiệu và những đặc điểm của 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.



*Hình 1: Logo của Python*

Những đặc điểm của Python:

* Ngôn ngữ lập trình đơn giản, dễ học – dễ học. Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#.
* Miễn phí, mã nguồn mở : Có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật
* Khả năng di động linh hoạt **:** Python có thể lấy một mã và chạy nó trên bất kỳ máy nào, không cần phải viết mã khác nhau cho các máy khác nhau. Điều này làm cho Python trở thành một ngôn ngữ di động.
* Khả năng mở rộng và có thể nhúng: Python có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của Python những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.
* Ngôn ngữ thông dịch cấp cao: Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Trong nội bộ, mã nguồn của nó được chuyển đổi thành một hình thức ngay lập tức được gọi là bytecode.
* Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến: Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code.
* Hướng đối tượng: Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.
  1. **Những ứng dụng của Python:**
* Các ứng dụng web
* Dùng ngôn ngữ Python để viết ngôn ngữ lập trình kịch bản (scripting language)
* Nghiên cứu mang tính học thuật với dữ liệu lớn
* Ngành khoa học dữ liệu (data science)
* Machine Learning và Trí thông minh nhân tạo (AI)
* Lĩnh vực IoT – Internet Vạn Vật
* Lập trình game

1. **Thư viện OpenCV (OpenSource Computer Vision)**

OpenCV (OpenSource Computer Vision) là một thư viện mã nguồn mở.

OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại.

Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android.

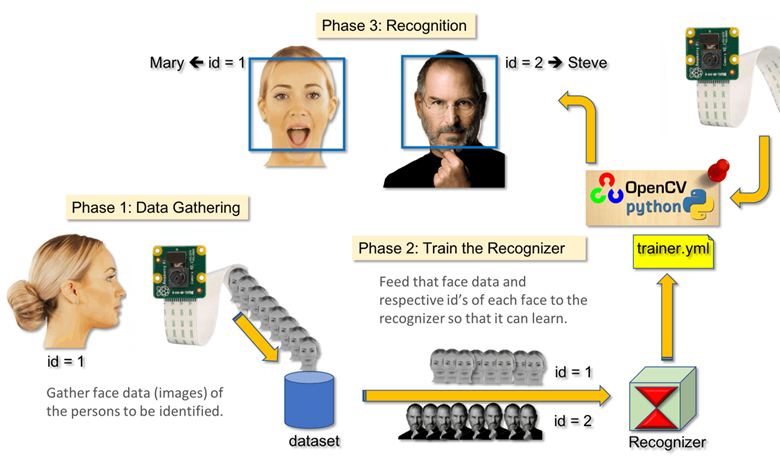
OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi.

Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot

OpenCV đang được sử dụng rộng rãi trong các ứng dụng bao gồm:

* Nhận diện hình ảnh
* Kiểm tra và giám sát tự động
* Robot và xe hơi tự lái
* Phân tích hình ảnh y tế
* Tìm kiếm và phục hồi hình ảnh/video
* Phim - cấu trúc 3D từ chuyển động
* Nghệ thuật sắp đặt tương tác

Đối với sinh viên ngành điện tử viễn thông việc ứng dụng thư viện mã nguồn mở OpenCV có thể thực hiện được rất nhiều các bài toán lý thú trên các bo mạch phát triển sẵn như Raspberry pi hay Adruino.



*Hình 2: Mô tả về ứng dụng của thư viện OpenCV*

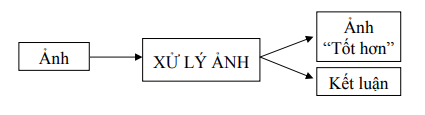
Chức năng OpenCV

* Image/video I/O, xử lý, hiển thị (core, imgproc, highgui)
* Phát hiện các vật thể (objdetect, features2d, nonfree)
* Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
* Computational photography (photo, video, superres)
* Machine learning & clustering (ml, flann)
* CUDA  acceleration (gpu)

**CHƯƠNG II: CƠ SỞ LÝ THUYẾT**

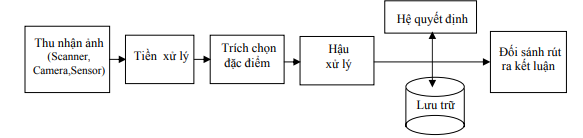
1. **Xử lý ảnh và các vấn đề cơ bản trong xử lý ảnh:**
2. **Xử lý ảnh là gì ?**

Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ hoạ đó phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ hoạ đóng một vai trò quan trọng trong tương tác người máy. Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc một kết luận.



*Hình 3: Quá trình xử lý ảnh*

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến P(c1, c2,..., cn). Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều. Sơ đồ tổng quát của một hệ thống xử lý ảnh:



*Hình 4: Các bước cơ bản trong một hệ thống xử lý ảnh*

1. **Định nghĩa về xử lý ảnh:**

Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của ngƣời sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh.

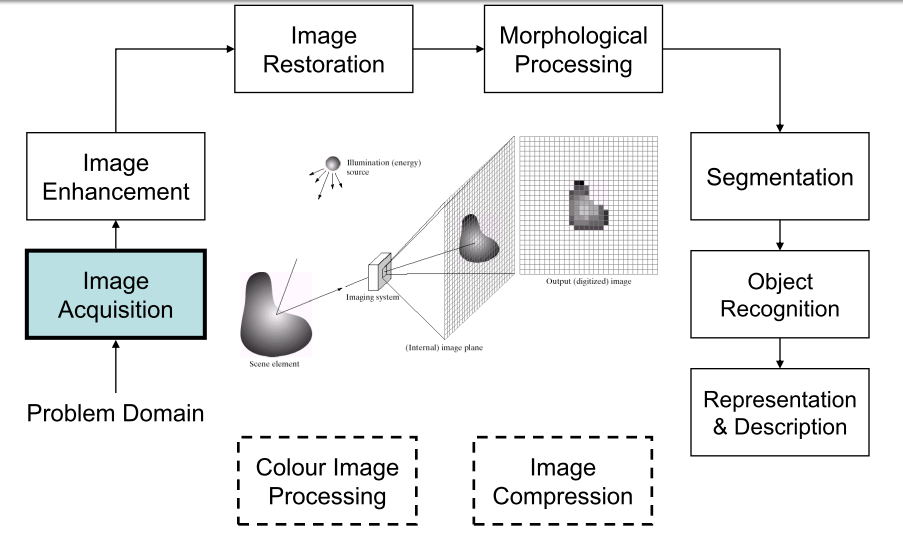
Cũng như xử lý dữ liệu bằng đồ hoạ, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ả n h số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên. Mục đích của xử lý ảnh gồm:

* Biến đổi ảnh làm tăng chất lượng ảnh
* Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có t h ể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh,… Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

1. **Các bước chính trong xử lý ảnh số:**

Các bước chính trong xử lý ảnh số :

* Thu nhận ảnh
* Biến đổi thông tin hình ảnh về các cấu trúc được lưu trữ trong máy tính, có thể hiển thị qua các thiết bị ngoại vi như máy in, màn hình,…
* Ảnh được thu từ nhiều người khác nhau : Camera chụp ảnh, video, máy scan,…
* Tiền xử lý ảnh
* Là quá trình sử dụng các kỹ thuật xử lý ảnh để làm ảnh tốt lên theo mục đích sử dụng
* Mục đích : Lọc nhiễu, nâng cao độ tương phản
* Phân đoạn ảnh
* Là quá trình phân chia nội dung các đối tượng cần khảo sát ra khỏi ảnh.
* Phân chia các đối tượng tiếp giáp nhau.
* Phân tách các đối tượng riêng biệt thành các đối tượng con.
* Biểu diễn và mô tả
* Tìm các vùng đặc trưng của điểm ảnh và biểu diễn lại thông qua các điểm đặc trưng
* Nhận dạng và nội suy
* Nhận dạng là quá trình xác định ảnh bằng cách so sánh với mẫu chuẩn đã được lưu trước.
* Nội suy là phán đoán dựa trên cơ sở nhận dạng
* Theo lý thuyết về nhận dạng,các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản:
* Nhận dạng theo tham số
* Nhận dạng theo cấu trúc

*Hình 5: Các bước chính trong xử lý ảnh số*

**1.4 Ứng dụng của xử lý ảnh :**

* Khôi phục hình ảnh, chỉnh sửa, điều chỉnh độ phân giải: Ứng dụng này tương tự như photoshop: từ một hình ảnh được chụp từ máy ảnh, ta có thể chỉnh sửa, xử lý để làm ảnh đẹp hơn hoặc phù hợp nhu cầu người dung như: làm mờ, lấy biên, chỉnh độ nét, chỉnh độ phân giải, phục hồi và nhận dạng ảnh….
* Trong y học: Các thuật toán xử lý ảnh cho phép biến đổi hình ảnh được tạo ra từ nguồn bức xạ X -ray hay nguồn bức xạ siêu âm thành hình ảnh quang học trên bề mặt film X-quang hoặc trực tiếp trên bề mặt màn hình hiển thị.
* UV imaging: Lĩnh vực này liên quan nhiều đến thám hiển, do thám. Cách hoạt động như sau: để phân tích thiệt hại của một trận động đất mà con người không thể tới được. Mặt đất nơi đó sẽ được quét bởi vệ tinh hoắc một máy bay sau đó truyền dữ liệu, hình ảnh về máy chủ để phân tích. Sẽ rất nhanh chóng so với việc chờ đợi con người tới đó. Một trận động đất có thể diện tích rất rộng mà con người không thể nào phân tích hết được.
* Truyền và mã hóa: Lĩnh vực này không cần giải thích nhiều chắc các bạn cũng hiểu được. Mình sẽ cho một ví dụ. Ngày nay con người sử dụng internet để truyền nhận các ảnh, video một cách nhanh chóng. Hình ảnh khi ta chụp sẽ được mã hóa và truyền theo internet. Rất nhanh sau vài giây là người bạn có thể nhận được một bức ảnh.
* Thị giác máy tính và robot: Hiện tại công nghệ robot đang phát triển nhanh chóng, và càng ngày càng giống con người hơn. Thị giác của máy tính cũng là một phần quan trọng. Làm thế nào để robot có thể nhìn mọi thứ, tránh vật cảng, nhận dạng các vật..? Đó chính là nhờ một hệ thống quá trình xử lý ảnh phức tạp
* Phát hiện vật cản: Phát hiện vật cản cũng là một lĩnh vực mới và được thực hiện bởi xử lý ảnh: tính toán khoảng cách từ robot tới vật cản bằng cách xác định được các đối tượng khác nhau trong hình ảnh sau đó xử lý và tính toán chúng.
* Công nghệ nhận dạng: Xử lý ảnh dung để xác định, nhận dạng các đối tượng, các mối nguy hiểm, nhận dạng vân tay, khuôn mặt, hoặc các loại bệnh trong lĩnh vự y tế.

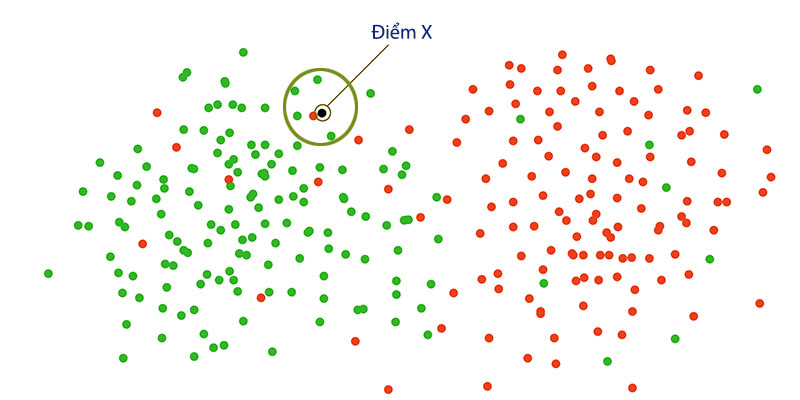
1. **Thuật toán KNN (K-Neareast Neighbors)**

K-nearest neighbor là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

Với KNN, trong bài toán Classification, label của một điểm dữ liệu mới (hay kết quả của câu hỏi trong bài thi) được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong training set. Label của một test data có thể được quyết định bằng major voting (bầu chọn theo số phiếu) giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra label. Chi tiết sẽ được nêu trong phần tiếp theo.

Trong bài toán Regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.

Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu. Hình dưới đây là một ví dụ về KNN trong classification với K = 1.



*Hình 6: Hoạt động của thuật toán KNN*

1. **Bài toán nhận diện biển số xe**

**3.1 Mô hình đề xuất**

Trích chọn đặc trung

Nhận dạng biển số

Phân đoạn từng ký tự

Tách biển số

Ảnh RGB xe ô tô

*Hình 7: Mô hình ứng dụng*

**3.2 Các quá trình xử lý trong hệ thống**

3.2.1 Xác định vùng chứa biển số xe và tách biển số xe ô tô

a) Xác định vùng chứa biển số xe ô tô

Vì biển số xe có những đặc trưng cơ bản được quy định bởi các cơ quan chức năng nên ta có thể dựa vào đặc trưng này để phân biệt với các đối tượng khác. Theo quy định của Bộ Công an, biển số xe đằng trước của các loại xe ô tô là một hình chữ nhật, có kích thước 520 x 110 (mm), phông nền màu trắng và các ký tự in hoa màu đen. Các ký tự chữ số bao gồm từ 0 đến 9 và các ký tự chữ cái bao gồm A, B, C, D, E, F, G, H, K, L, M, N, P, S, T, U, V, X, Y, Z (20 ký tự).

Xác định vùng chứa biển số xe được chia làm 2 giai đoạn chính:

* Giai đoạn 1: xác định ví trí của biển số trong ảnh
* Giai đoạn 2: dùng các giải thuật để tách biển số xe ra khỏi ảnh chụp và xoay biển số xe về phương ngang.

Ảnh chỉ chứa biển số

Tách biển số

Định vị trí của biển số

Ảnh chụp từ Camera

*Hình 8: Sơ đồ tổng quát của khối xác định vùng chứa biển số xe*

b) Tách biển số

Sau khi định vị biển số xe, tiến hành tách biển số xe. Biển số xe được tách theo 2 bước:

* Bước 1: Xác định vùng chứa ký tự, sau đó loại bỏ các vùng không có ký tự vì đó không phải là vùng chứa biển số.
* Bước 2: Tách biển số ra khỏi vùng trên.

Ảnh

Tìm vùng màu trắng

Biển đổi ảnh RGB -> Ảnh xám

Nhị phân ảnh xám để tìm vùng trắng

Tìm các vùng có diện tích phù hợp

Tìm đường biên của các vùng trắng

Dò và loại bỏ các khung ảnh có mật độ pixel không phù hợp

Chọn vùng có tỉ lệ phù hợp

Xác định các vùng có chứa biển số

Cắt các vùng có tỉ lệ phù hợp với biển số

Biến đổi sang ảnh xám (Gray)

Nhị phân ảnh xám và tìm vùng bao quanh

Cắt vùng có nghi vấn chứa biển số

Phân tích ảnh bao quanh các hình chữ nhật

Xác định các hình chữ nhật đủ với số lượng

Tìm vùng chứa ký tự

Cắt vùng ảnh khi mà xác định đủ só hình chữ nhật

Cắt biển số chính xác

Cắt chính xác biển số

Ảnh chỉ chứa biển số

*Hình 9: Sơ đồ khối tách biển số*

3.2.2 Phân đoạn từng ký tự

Các giai đoạn trong việc phân đoạn ký tự như sau:

* Gia đoạn 1: tiến hành phân ngưỡng

Sau khi có hình ảnh biển số xe nhị phân thì tìm vùng trắng của ký tự để xác định được các đường viền bao quanh ký tự.

* Giai đoạn 2: tìm vùng đối tượng.

Khi tìm được hình Contours hay là đường viền bao quanh các ký tự thì ta dùng 1 hình chữ nhật có kích thước tương đương với các ký tự để nhận dạng các ký tự riêng biệt.

* Giai đoạn 3: tìm và tách vùng ký tự.

Sau khi đã nhận dạng từng ký tự bằng hình chữ nhật thì ta tiến hành cắt các vùng chữ nhật đó để tạo ra các ký tự riêng biệt.

Sau khi thực hiện phân vùng ta sẽ được ma các ma trận tương ứng với các ký tự trên biển số. Lần lượt từng ma trận ký tự sẽ được đưa vào chương trình nhận dạng. Kết quả cuối cùng sẽ là ký tự và chương trình sẽ hiển thị ký tự này dưới dạng text.

Thực chất, quá trình nhận dạng là quá trình đổi ma trận điểm ảnh của các ký tự thành mã ASCII tương ứng với ký tự đó. Để làm được điều này người ta đem so sánh ma trận của ký tự với tất cả các ma trận trong tập mẫu, ma trận mẫu nào có khả năng giống nhiều nhất thì nó chính là ký tự cần tìm.

Tìm và tách vùng ký tự

Tìm vùng đối tượng

Phân ngưỡng

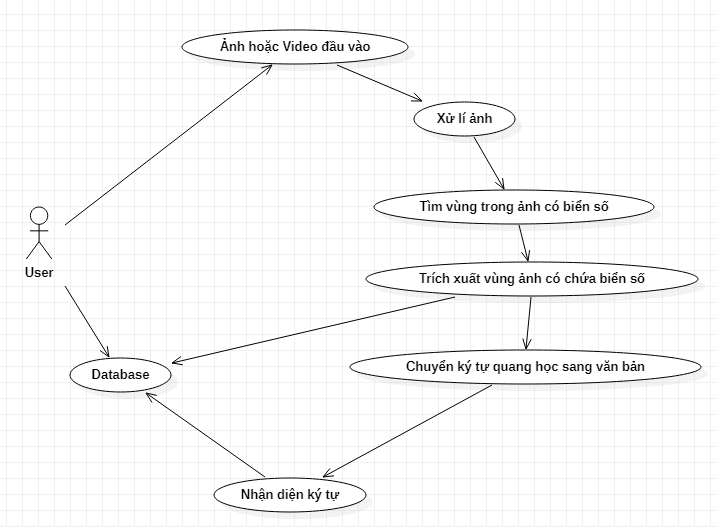
Ảnh đầu vào

*Hình 10: Các giai đoạn trong việc phân đoạn từng ký tự*

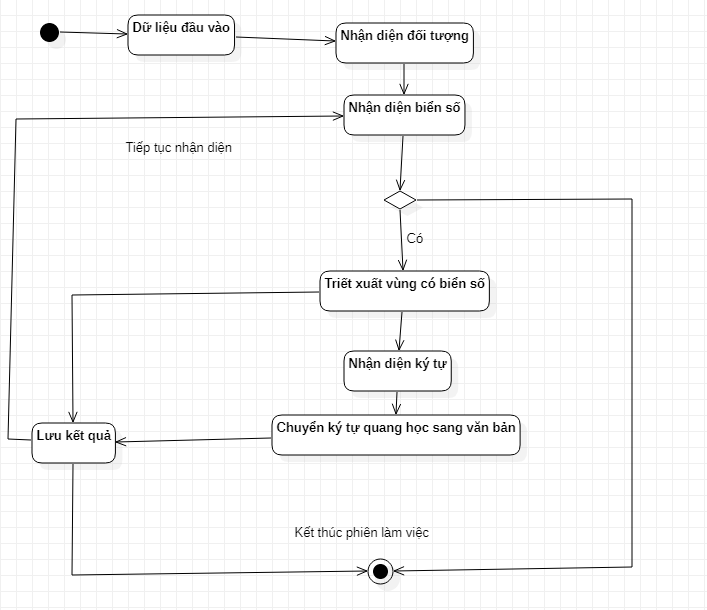
Nhận dạng ký tự ở đây sử dụng phương pháp KNN (K-Nearest Neighbors). Thuật toán KNN này hiểu đơn giản là tìm k phẩn tử giống phần tử được test nhất, kết quả là class nào xuất hiện nhiều nhất thì đó là kết quả cần tìm.

1. **Phân tích và thiết kế hệ thống**

**4.1 Biểu đồ use case**

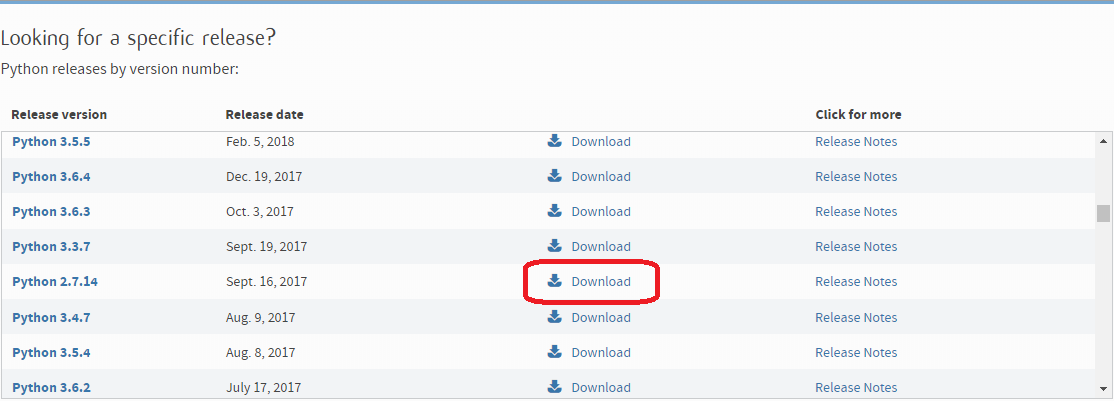


* 1. **Biểu đồ hoạt động**

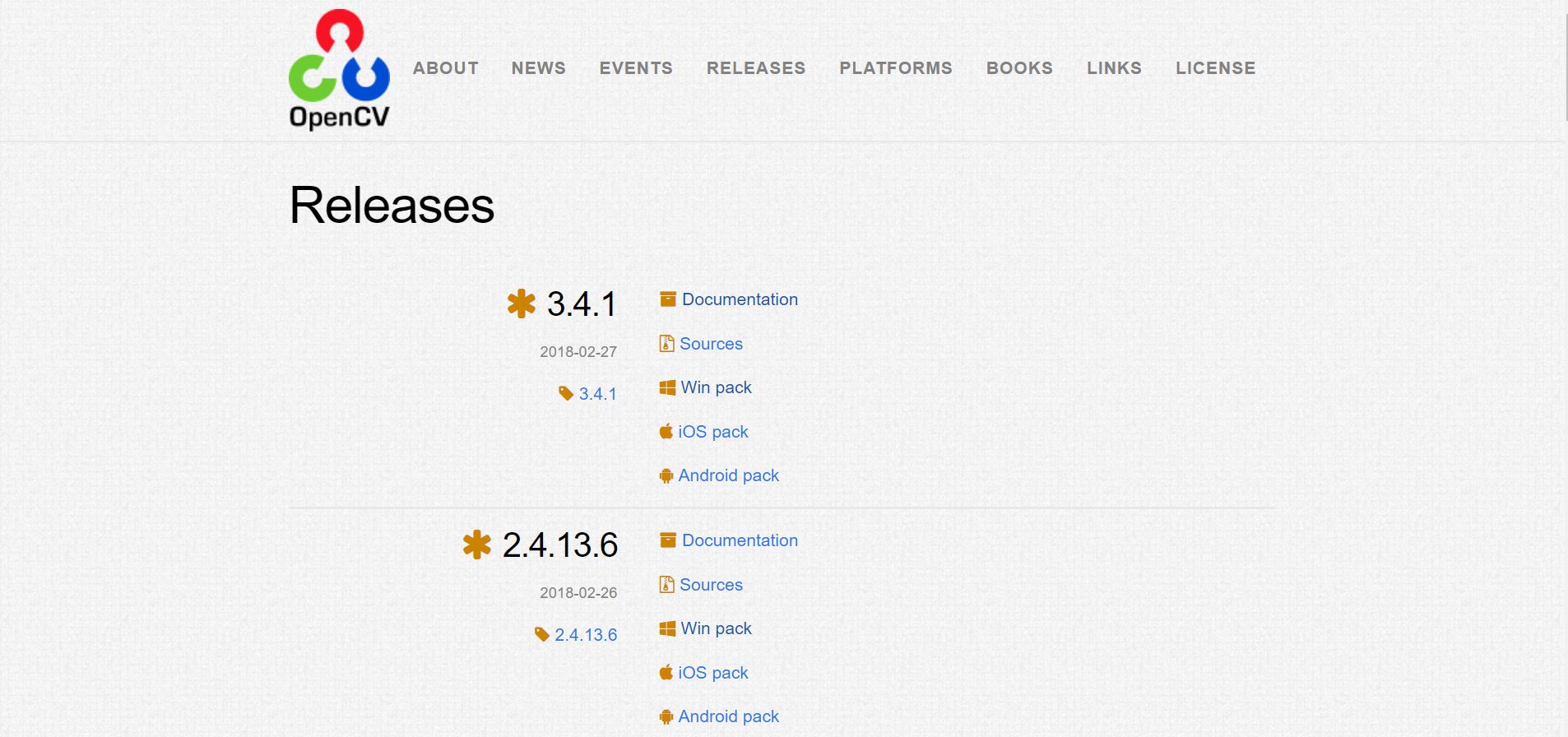


**CHƯƠNG III: XÂY DỰNG ỨNG DỤNG VÀ TRIỂN KHAI CÀI ĐẶT**

* + 1. **Cài đặt ngôn ngữ lập trình Python và Thư viện OpenCV**

Ở đây chúng em sẽ sử dụng Python phiên bản 2.7 và OpenCV

Hình 11: Giao diện Download Python



*Hình 12: Giao diện tải về thư viện OpenCV*

Sau khi cài đặt Python 2.7 và OpenCV ta copy file cv2.pyd từ thư mục *\face detection\opencv\build\python\2.7*

Copy vào thư mục *\Python27\Lib\site-packages*

* + 1. **Kết quả**
  1. Ảnh thu từ camera:

Để chụp ảnh thì ta có thể sử dụng camera hoặc Webcam thì khi lựa chọn thiết bị thu nhận ảnh cần phải quan tâm đến các thông số có thể ảnh hưởng tới chất lượng ảnh như là độ phân giải, số điểm ảnh, điều kiện hoạt động của thiết bị,vv

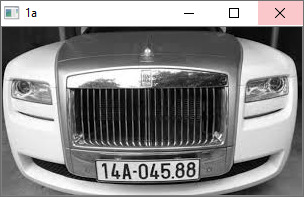
Tuy nhiên, ở báo cáo này thì chỉ nhằm mục đích nghiên cứu nên mình sẽ sử dụng một hình ảnh có sẵn để nhận diện.

****

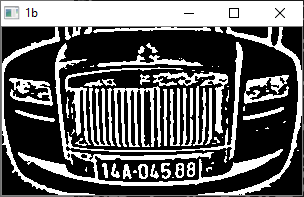
*Hình 13: Hình mẫu*

* 1. Định vị biển số xe:

Ở giai đoạn định vị biển số xe thì ta phải tiến hành xử lý hình ảnh đầu vào sang ảnh xám đồng thời đổi từ hệ màu RGB sang hệ màu HSV. Rồi lấy ngưỡng ảnh và tối ưu hóa tương phản. Thì kết quả thu được là ảnh imgGrayscaleScene và imgThreshScene*.*

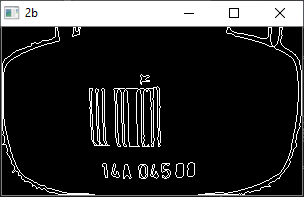


*Hình 14: Ảnh imgGrayscaleScene*



*Hình 15: Ảnh imgThreshScen*

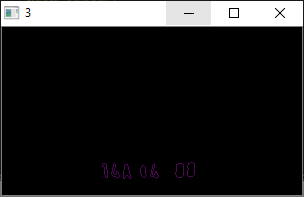
Tiếp theo, ta sẽ tìm những ký tự mà có thể là biển số .Ở bước này ta tiếp tục loại bỏ những nét không phải là ký tự, chỉ giữ lại những nét có thể là ký tự để có thể nhận biết được vùng có thể là bản số xe và tiếp tục xử lý ở các bước tiếp theo.



*Hình 16: Ảnh chứa những ký tự có thể là biển số*

* 1. Định vị ký tự:

Bước này chúng sẽ là xử lý những ký tự phù hợp nhất để tạo thành một dãy biển số

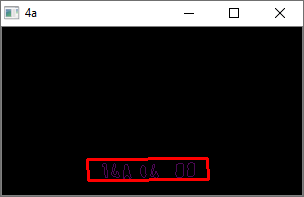


*Hình 17: Ảnh chưa những chuỗi ký tự phù hợp*

Sau khi có được chuỗi ký tự phù hợp nhất thì ta sẽ tiến hành vẽ khung đỏ ở xung quanh nó để đánh dấu. Rồi ta sẽ tách những phần được đánh dấu khung đỏ từ ảnh gốc để thành những ảnh độc lập chỉ chứa những ký tự mà ta vừa xác định rồi tiếp tục xử lý.

Trong quá trình xử lý thì sẽ có nhiều khung chứa ký tự phù hợp vì thế ta sẽ xử lý những khung này để tìm ra khung nào có chứa ký tự để tạo thành một biển số và tiếp tục xử lý ở bước tiếp theo

Ở trong hình minh họa thì chỉ có một khung chứa ký tự nên nó chỉ có thể nhận dạng ra một khung chứa ký tự

2020-08-05_104555

*Hình 18: Hình ảnh sau khi trích xuất*

Sau khi có được khung có chứa mã biển số phù hợp. Ta sẽ tiếp tục định vị từ ký tự trong khung để có thể tách từng ký tự trong khung đó. Thì cũng giống như bước Định vị biển số xe ở trên ta cũng sẽ chuyển về ảnh xám và ảnh phân ngưỡng.



*Hình 19: Hình ảnh sau khi xử lý khung*

Sau đó, ta tiếp tục loại bỏ những phần không cần thiết và chỉ tách lấy phần chứa chuỗi kí tự phù hợp. Đồng thời tách lấy biên





*Hình 20: Hình ảnh sau khi tìm chuỗi phù hợp*

* 1. Nhận dạng ký tự:

Đến bước này thì ta đã có thể xác được được vị trí của biển số xe và định vị được chuỗi ký tự ở trong biển số xe.

Ở bước này ta sẽ tách riêng từng kỳ tự thành từng khung và nhận diện ký tự đó là ký tự gì.

Ta sẽ sử dụng phương pháp KNN training với những text có sẵn thay vì sử dụng phần mềm Tesseract

Những khung chứa những ký tự không phù hợp thì không thể nhận diện. Ngoài ra còn do chất lượng ảnh hoặc là training chưa đủ thì cũng có thể xảy ra trường hợp nhận diện sai lệch hoặc là không nhận diện được.

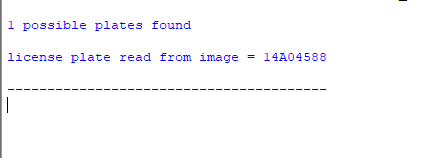


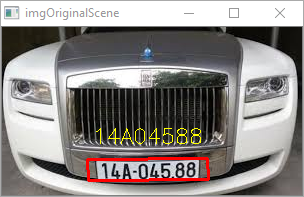
*Hình 21: Hình ảnh sau khi nhận diện ký tự trong khung*

* 1. Kết quả thu được

Ta thu được chuỗi ký tự trên khung là là biển số xe và in kết qủa trên cửa sổ “python shell”

Ngoài ra chuỗi ký tự còn hiện ở ảnh gốc để ta có thể so sánh

****



*Hình 22: Kết quả thu được*

# **KẾT LUẬN**

1. **Kết luận**

Sau thời gian tìm hiểu các thuật toán trong xử lý ảnh , ngôn ngữ lập trình python và thư viện OpenCV vào bài toán nhận dạng ký tự thì nhóm đã đạt mục tiêu đề ra là xây dựng một ứng dụng Nhận dạng biển số xe sử dụng thư viện OpenCV

Tuy nhiên trong quá trình làm đồ án thì do còn thiếu kiến thức cho nên vẫn còn một số hạn chế như:

* Code vẫn còn chưa được tối ưu, gây khó khăn trong quá trình đọc hiểu
* Tỉ lệ nhận diện đúng vẫn còn sai sót chỉ tương đối ổn định, chịu ảnh hưởng khi ảnh bị mờ hoặc mà góc chụp không chính diện.
* Độ chính xác của chương trình nhận dạng biển số xe phụ thuộc vào ánh sáng của môi trường.
* Việc nhận dạng ký tự còn có sai sót do không có được một tập mẫu hoàn ch

1. **Hướng phát triển**

Từ đề tài “Nhận dạng biển số xe sử dụng thư viện OpenCV” ta có thể ứng dụng vào trong thực tế như là ở các bãi giữ xe, hệ thống quản lý giao thông, xử lý vi phạm giao thông.

Tiếp tục nghiên cứu các bài toán xử lý ảnh và khắc phục những vấn đề còn thiếu xót

Xây dựng một ứng dụng hoàn chính, thuận tiện để có thể được sử dụng rộng rãi trong thực tế.

# **TÀI LIỆU THAM KHẢO**

**[1]** Xây dựng phần mềm nhận dạng biển số xe : <https://thigiacmaytinh.com/phan-mem-nhan-dien-bien-so-co-kem-source/>

**[2]** Giới thiệu về thư viện OpenCV : <https://fet.epu.edu.vn/chi-tiet-tin/gioi-thieu-ve-opencv-va-ung-dung-17-12249.html>

**[3]** Giới thiệu về python: <https://o7planning.org/vi/11385/gioi-thieu-ve-python>

**[4]** <https://en.wikipedia.org/wiki/OpenCV>

**[5]** <https://en.wikipedia.org/wiki/Python_(programming_language)>

**[6]** <https://github.com/>

**[7]** Nhận diện biển số xe với Opencv step by step :

<https://thorpham.github.io/blog/2018/04/11/regconite-plate-car/>