**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CHUYÊN NGÀNH**

ĐỀ TÀI : XÂY DỰNG HỆ THỐNG NHẬN DIỆN BIỂN SỐ XE

**Giảng viên hướng dẫn: VƯƠNG XUÂN CHÍ**

**Sinh viên thực hiện: VÕ QUỐC DƯƠNG**

**MSSV: 2000001592**

**Khoá: 2020**

**Chuyên ngành:** TRÍ TUỆ NHÂN TẠO

Tp HCM, tháng 9 năm 2023

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CHUYÊN NGÀNH**

ĐỀ TÀI : XÂY DỰNG HỆ THỐNG NHẬN DIỆN BIỂN SỐ XE

**Giảng viên hướng dẫn: VƯƠNG XUÂN CHÍ**

**Sinh viên thực hiện: VÕ QUỐC DƯƠNG**

**MSSV: 2000001592**

**Khoá: 2020**

**Chuyên ngành:** TRÍ TUỆ NHÂN TẠO

Tp HCM, tháng 9 năm 2023

# LỜI MỞ ĐẦU

Sự phát triển mạnh mẽ của công nghệ thông tin trong những năm gần đây đã tạo nên những thay đổi to lớn đối với cuộc sống con người. Nó ngày càng khẳng định được vai trò của mình trong sự phát triển chung của xã hội.Trong thời đại ngày nay mọi việc con người có thể làm thì hầu như máy móc cũng có thể làm được. đó là nhờ những người lập trình viên đã biến những chiếc máy tính và những ngoại vi của nó trở thành những hệ thống thông minh có thể thực hiện bất cứ công việc gì.

Đơn giản như công việc gửi xe, chỉ vài năm trước đây khi đi đến bất kì bãi gửi xe nào chúng ta cũng thấy tất cả xe đều phải được nhân viên coi xe gi bằng giấy, việc này là vô cũng bất tiện và nhiều rủi ro như giấy rất dễ bị hư hao, rách hoặc dính nước…không những vậy việc ghi giấy không thể tránh khỏi sai sót khi người nhân viên coi xe một thoáng mất tập trung và ghi sai, vì vậy rất mất thời gian. Trong khuôn khổ chương trình đã học, cũng như yêu cầu của đồ án chuyên ngành, em đã viết một chương trình : “XÂY DỰNG HỆ THỐNG NHẬN DIỆN BIỂN SỐ XE”

Song vì thời gian có hạn, và còn thiếu nhiều kinh nghiệm và kỹ năng còn hạn chế. Nên chắc chắn không tránh khỏi những thiếu sót. Vì vậy em rất mong nhận được nhiều ý kiến đóng góp từ thầy cô và bạn bè để chúng em hoàn thành tốt đề tài này. Chân thành cảm ơn thầy cô và các bạn.

# LỜI CẢM ƠN

Em xin được bày tỏ lòng cảm ơn sâu sắc và chân thành đến Thầy Vương Xuân Chí, người đã dành thời gian và nỗ lực để truyền đạt kiến thức và kinh nghiệm quý báu về Trí tuệ nhân tạo cho em trong môn học này.

Thầy là một giảng viên tuyệt vời, luôn nhiệt tình và tận tâm trong việc giảng dạy, giúp cho chúng em hiểu sâu và áp dụng được những kiến thức chuyên sâu của lĩnh vực này. Bên cạnh đó, Thầy cũng đã trang bị cho chúng em những kỹ năng và phương pháp nghiên cứu cần thiết để tiếp tục phát triển trong lĩnh vực này.

Không chỉ giảng dạy, Thầy còn là một người hướng dẫn và cố vấn rất tuyệt vời. Thầy luôn sẵn sàng trả lời các câu hỏi của chúng em và đưa ra các lời khuyên hữu ích để giúp chúng em hoàn thiện tiểu luận của mình.

Cuối cùng, em xin chân thành cảm ơn Thầy Vương Xuân Chí và mong rằng sẽ có thêm nhiều sinh viên được học tập và được Thầy truyền đạt kiến thức về Trí tuệ nhân tạo trong tương lai.

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ III NĂM HỌC 2023 - 2024** |

**PHIẾU CHẤM THI ĐỒ ÁN**

BM-ChT-11

Môn thi: Đồ án chuyên ngành trí tuệ nhân tạo Lớp học phần: 20DTH1A

Nhóm sinh viên thực hiện : 04

1. Võ Quốc Dương Tham gia đóng góp: 100%

Ngày thi:19/9/2023 Phòng thi:L.510

Đề tài tiểu luận/báo cáo của sinh viên : Xây dựng hệ thống nhận diện biển số xe

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  |  |  |
| Nội dung |  |  |  |
| * Các nội dung thành phần |  |  |  |
| * Lập luận |  |  |  |
| * Kết luận |  |  |  |
| Trình bày |  |  |  |
| **TỔNG ĐIỂM** |  |  |  |

|  |  |
| --- | --- |
|  | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* |

**Mục Lục**

[LỜI MỞ ĐẦU 1](#_Toc145717335)

[CHƯƠNG 1 : GIỚI THIỆU TỔNG QUAN 7](#_Toc145717337)

[1.1. Tổng quan đề tài 7](#_Toc145717338)

[1.2. Mục tiêu nghiên cứu đề tài 7](#_Toc145717339)

[1.3. Phương pháp nghiên cứu 7](#_Toc145717340)

[1.4. Các nghiên cứu liên quan 8](#_Toc145717341)

[1.4.1. Bài báo trong nước 8](#_Toc145717342)

[1.4.2. Bài báo ngoài nước 9](#_Toc145717343)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 10](#_Toc145717344)

[2.1. PyQt5 và Qt Designer : 10](#_Toc145717345)

[2.1.1. Qt : 10](#_Toc145717346)

[2.1.2. PyQt : 10](#_Toc145717347)

[2.1.3. Qt Designer : 11](#_Toc145717348)

[2.2. Sơ lược về ngôn ngữ Python : 11](#_Toc145717349)

[2.2.1. Nguồn gốc cái tên “ Python” 12](#_Toc145717350)

[2.2.2. Tính năng của Python 12](#_Toc145717351)

[2.2.3. Công cụ lập trình Python – Visual Studio 12](#_Toc145717352)

[2.3. Trí tuệ nhân tạo : 12](#_Toc145717353)

[2.3.1. Khái niệm 12](#_Toc145717354)

[2.3.2. Ứng dụng AI trong thực tế 13](#_Toc145717355)

[2.4. Tổng quan bài toán nhận dạng biển số xe : 13](#_Toc145717356)

[2.4.1. Khái niệm biển số xe : 13](#_Toc145717357)

[2.4.2. Xử lý ảnh và Open CV : 14](#_Toc145717358)

[2.4.3. Nhận dạng đối tượng 14](#_Toc145717359)

[2.4.4. Thị giác máy tính 14](#_Toc145717360)

[2.4.5. Hướng giải quyết 15](#_Toc145717361)

[CHƯƠNG 3 : MÔ HÌNH THỰC NGHIỆM 16](#_Toc145717362)

[3.1. Phân tích yêu cầu 16](#_Toc145717363)

[3.1.1. Yêu cầu chức năng 16](#_Toc145717364)

[3.1.2. Yêu cầu hệ thống 16](#_Toc145717365)

[3.2. Biểu đồ sử dụng 16](#_Toc145717366)

[3.2.1. Chức năng đưa dữ liệu vào 16](#_Toc145717367)

[3.2.2. Chức năng đưa ra ảnh đã threshold 16](#_Toc145717368)

[3.2.3. Chức năng đưa ra các contour tìm được 16](#_Toc145717369)

[3.2.4. Chức năng nhận dạng và hiển thị biển số xe 17](#_Toc145717370)

[3.3. Thiết kế hệ thống 17](#_Toc145717371)

[3.4. Phát hiện vị trí và tách biển số xe 18](#_Toc145717372)

[3.4.1. Mô hình 18](#_Toc145717373)

[3.4.2. Chuyển sang ảnh xám 19](#_Toc145717374)

[3.4.3. Tăng độ tương phản 19](#_Toc145717375)

[3.4.4. Giảm nhiễu bằng bộ lọc Gausian 21](#_Toc145717376)

[3.4.5. Phát hiện cạch 21](#_Toc145717377)

[3.4.6. Vẽ contour 24](#_Toc145717378)

[3.5. Phân đoạn kí tự 25](#_Toc145717379)

[3.5.1. Xoay biển số 25](#_Toc145717380)

[3.5.2. Tìm vùng đối tượng 27](#_Toc145717381)

[3.5.3. Tìm và tách kí tự 27](#_Toc145717382)

[3.6. Nhận diện kí tự 27](#_Toc145717383)

[3.6.1. Thuật toán KNN 27](#_Toc145717384)

[3.6.2. Hướng giải quyết 29](#_Toc145717385)

[3.7 Kết quả chương trình 31](#_Toc145717386)

[CHƯƠNG 4 : KẾT LUẬN 33](#_Toc145717387)

[1.1 Hạn chế: 33](#_Toc145717388)

[1.2 Hướng phát triển: 33](#_Toc145717389)

**DANH MỤC HÌNH**

Hình 3.1 17

Hình 3.2 18

Hình 3.3 20

Hình 3.4 20

Hình 3.5 20

Hình 3.6 21

Hình 3.7 22

Hình 3.8 23

Hình 3.9 23

Hình 3.10 25

Hình 3.11 26

Hình 3.12 26

Hình 3.13 27

Hình 3.14 28

Hình 3.15 29

Hình 3.16 30

Hình 3.17 30

Hình 3.18 30

Hình 3.19 31

Hình 3.20 32

# CHƯƠNG 1 : GIỚI THIỆU TỔNG QUAN

* 1. **Tổng quan đề tài**

Sự phát triển mạnh mẽ của công nghệ thông tin trong những năm gần đây đã tạo nên những thay đổi to lớn đối với cuộc sống con người. Nó ngày càng khẳng định được vai trò của mình trong sự phát triển chung của xã hội.Trong thời đại ngày nay mọi việc con người có thể làm thì hầu như máy móc cũng có thể làm được. đó là nhờ những người lập trình viên đã biến những chiếc máy tính và những ngoại vi của nó trở thành những hệ thống thông minh có thể thực hiện bất cứ công việc gì.

Đơn giản như công việc gửi xe, chỉ vài năm trước đây khi đi đến bất kì bãi gửi xe nào chúng ta cũng thấy tất cả xe đều phải được nhân viên coi xe ghi bằng giấy, việc này là vô cũng bất tiện và nhiều rủi ro như giấy rất dễ bị hư hao, rách hoặc dính nước…không những vậy việc ghi giấy không thể tránh khỏi sai sót khi người nhân viên coi xe một thoáng mất tập trung và ghi sai, vì vậy rất mất thời gian. Để bắt kịp với xu thế hiện nay, em muốn nghiên cứu cũng như tiếp cận bài toán nhận dạng này. Vì thế em muốn thực hiện đề tài nhận dạng biển số xe bằng ngôn ngữ lập trình Python. .Mục tiêu dự kiến của em là hiểu thêm về bài toán nhận dạng, ứng dụng được nó vào thực tế.

* 1. **Mục tiêu nghiên cứu đề tài**
* Xây dựng được chương trình giúp người dùng nhận dạng được các biển số thông qua các hình ảnh hoặc camera đầu vào.
* Chương trình có thể nhận dạng được số xe một cách chính xác và in nó ra màn hình.
* Hiểu rõ hơn về các thuật toán nhận dạng và cách hoạt động nhằm nâng cao hiểu biết về công nghệ hiện nay
  1. **Phương pháp nghiên cứu**
* Tìm hiểu về ngôn ngữ lập trình Python
* Tìm hiểu về các thư viện hỗ trợ lập trình
* Tham khảo các chức năng, các điểm ưu nhược của các ứng dụng đã có, từ đó quyết định các ứng dụng và hướng phát triển
* Tham khảo và lập trình giao diện cho ứng dụng
* Phân tích các chức năng và yêu cầu cần có
* Nghiên cứu về cách thức hoạt động của bài toán nhận dạng
* Nghiên cứu và hiểu được thuật toán của bài toán nhận dạng biển số xe
* Kiểm thử phần mềm, sửa chữa và phát triển thêm các ý tưởng mới
* Xây dựng báo cáo hoàn chỉnh
  1. **Các nghiên cứu liên quan**
     1. **Bài báo trong nước**

***Bài báo 1:*** "Phát triển hệ thống nhận dạng biển số xe ô tô sử dụng deep learning."

**Tác giả:** Đào Văn Hùng và đồng nghiệp.

**Dữ liệu:** Dữ liệu được sử dụng trong nghiên cứu bao gồm các hình ảnh biển số xe ô tô được chụp trong các điều kiện khác nhau, bao gồm ánh sáng mạnh yếu, góc chụp khác nhau, v.v.

**Thuật toán:** Trong nghiên cứu này, tác giả sử dụng mạng neural tích chập (CNN) để xây dựng mô hình nhận dạng biển số xe ô tô. Các bước xử lý được thực hiện trên các hình ảnh gồm: tiền xử lý ảnh, phát hiện vùng chứa biển số xe, cắt và cân bằng ảnh biển số xe, và nhận dạng biển số xe bằng mô hình CNN.

**Kết quả:** Mô hình nhận dạng biển số xe ô tô được xây dựng trong nghiên cứu đạt được độ chính xác trên tập dữ liệu kiểm tra lên đến 97.8%, cho thấy khả năng ứng dụng cao trong thực tế.

***Bài báo 2:*** Sử dụng mạng neural tích chập (CNN) trong nhận dạng biển số xe ô tô.

**Tác giả:** Lê Trung Thành, đồng tác giả là Lê Viết Long và Trương Minh Tuấn.

**Dữ liệu:** dữ liệu này được tạo ra bằng cách thu thập hình ảnh từ các nguồn khác nhau, bao gồm cả hình ảnh được chụp từ các camera giám sát giao thông

**Thuật toán sử dụng:** là mạng neural tích chập (CNN), một thuật toán deep learning được sử dụng rộng rãi trong xử lý hình ảnh và nhận dạng đối tượng.

**Kết quả đạt được:** Kết quả của nghiên cứu cho thấy mạng neural tích chập (CNN) có khả năng xử lý hình ảnh và nhận dạng biển số xe ô tô với độ chính xác cao, đạt được độ chính xác trên tập dữ liệu kiểm tra là khoảng 95%. Mô hình CNN đã vượt qua các phương pháp truyền thống trước đó trong việc nhận dạng biển số xe.

* + 1. **Bài báo ngoài nước**

***Bài báo 1:*** A license plate recognition system based on multiple features extraction and SVM classification

**Tác Giả:** Xuewu Zhang, Shuihua Wang & Jingyu Yang.

**Dữ liệu:** Dữ liệu trong bài báo này bao gồm các hình ảnh về các phương tiện giao thông đường bộ được chụp từ nhiều góc độ khác nhau, với độ phân giải cao và đầy đủ thông tin về biển số xe.

**Thuật toán sử dụng:** SVM(Surport Vector Machine)

**Kết quả đạt được:** thử nghiệm của hệ thống đã cho thấy độ chính xác trung bình là 95,6% trên tập dữ liệu kiểm tra.

***Bài Báo 2****:* Automatic license plate recognition system**.**

**Tác giả:** Tarek Gaber, Aboul Ella Hassanien, Sanaa Elsayed, và Nashwa El-Bendary.

**Dữ liệu:** Nghiên cứu sử dụng tập dữ liệu chứa hình ảnh các biển số xe được chụp từ nhiều góc độ và điều kiện ánh sáng khác nhau, bao gồm cả hình ảnh đặc biệt như biển số xe gắn trên xe máy.

**Thuật toán:** Nghiên cứu sử dụng một số kỹ thuật xử lý hình ảnh như lọc Gauss, cân bằng sáng, phân ngưỡng, lỗ lớn, và kỹ thuật phát hiện biên cạnh Sobel. Ngoài ra, nghiên cứu sử dụng SVM để phân loại và nhận dạng biển số xe.

**Kết quả:** Kết quả của nghiên cứu cho thấy phương pháp nhận dạng biển số xe dựa trên SVM đạt được độ chính xác cao, đặc biệt là trong điều kiện ánh sáng yếu và ảnh nhiễu. Kết quả nhận dạng trên tập dữ liệu thực tế đạt được độ chính xác trên 95%.

# CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

* 1. **PyQt5 và Qt Designer :**
     1. **Qt :**

Qt là một Application framework đa nền tảng viết trên ngôn ngữ C++ , được dùng để phát triển các ứng dụng trên desktop, hệ thống nhúng và mobile. Hỗ trợ cho các platform bao gồm : Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS và một số platform khác.

* + 1. **PyQt :**

PyQt là Python interface của Qt, kết hợp của ngôn ngữ lập trình Python và thư viện Qt, là một thư viện bao gồm các thành phần giao diện điều khiển (widgets , graphical control elements).

PyQt được phát triển bởi Riverbank Computing Limited.

Các class của PyQt5 được chia thành các module, bao gồm :

* QtCore : là module bao gồm phần lõi không thuộc chức năng GUI, ví dụ dùng để làm việc với thời gian, file và thư mục, các loại dữ liệu, streams, URLs, mime type, threads hoặc processes.
* QtGui : bao gồm các class dùng cho việc lập trình giao diện (windowing system integration), event handling, 2D graphics, basic imaging, fonts và text.
* QtWidgets : bao gồm các class cho widget, ví dụ : button, hộp thoại, … được sử dụng để tạo nên giao diện người dùng cơ bản nhất.
* QtMultimedia : thư viện cho việc sử dụng âm thanh, hình ảnh, camera,…
* QtBluetooth : bao gồm các class giúp tìm kiếm và kết nối với các thiết bị có giao tiếp với phần mềm
* QtNetwork : bao gồm các class dùng cho việc lập trình mạng, hỗ trợ lập trình TCP/IP và UDP client, server hỗ trợ việc lập trình mạng
* QtPositioning : bao gồm các class giúp việc hỗ trợ xác định vị.
* Enginio : module giúp các client truy cập các Cloud Services của Qt.
* QtWebSockets : cung cấp các công cụ cho WebSocket protocol.
* QtWebKit : cung cấp các class dùng cho làm việc với các trình duyệt Web, dựa trên thư viện WebKit2
* QtWebKitWidgets : các widget cho WebKit.
* QtXml : các class dùng cho làm việc với XML file.
* QtSvg : dùng cho hiển thị các thành phần của SVG file.
* QtSql : cung cấp các class dùng cho việc làm việc với dữ liệu.
* QtTest : cung cấp các công cụ cho phép test các đơn vị của ứng dụng với PyQt5.
  + 1. **Qt Designer :**

Qt Designer là một công cụ Qt để thiết kế và xây dựng giao diện người dùng đồ họa (GUI) với Qt Widget. Bạn có thể soạn và tùy chỉnh các cửa sổ hoặc hộp thoại của mình theo cách bạn sẽ thấy (What you see is what you get – Những gì bạn đang thấy là những gì bạn sẽ nhận được) và kiểm tra chúng bằng các kiểu và độ phân giải khác nhau.

Các tiện ích và biểu mẫu được tạo bằng Qt Designer tích hợp liền mạch với mã được lập trình, sử dụng cơ chế vị trí và tín hiệu của Qt để bạn có thể dễ dàng gán hành vi cho các phần tử đồ họa. Tất cả các thuộc tính được đặt trong Qt Designer có thể được thay đổi động trong mã. Hơn nữa, các tính năng như quảng cáo widget và plugin tùy chỉnh cho phép bạn sử dụng các thành phần của riêng mình với Qt Designer .

* 1. **Sơ lược về ngôn ngữ Python :**

Phiên bản sử dụng trong đồ án: Python 3.7.8

Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

* + 1. **Nguồn gốc cái tên “ Python”**

Rossum – Tác giả của Python, là fan của một sê-ri chương trình hài cuối những năm 1970, và cái tên “Python” được lấy từ tên một phần trong sê-ri đó “Monty Python’s Flying Circus”.

* + 1. **Tính năng của Python**
* Đơn giản, dể học
* Miễn phí với mã nguồn mở
* Có thể chạy python trên hầu hết các nền tảng khác nhau
* Ngôn ngữ thông dịch cấp cao
* Thư viện tiêu chuẩn lớn
* Hướng đối tượng
  + 1. **Công cụ lập trình Python – Visual Studio**
  1. **Trí tuệ nhân tạo :**
     1. **Khái niệm**

AI (Artificial intelligence) là trí thông minh nhân tạo. Nó là sự mô phỏng các quá trình trí tuệ của con người bằng máy móc, đặc biệt là các hệ thống hệ máy tính. Các ứng dụng cụ thể của AI bao gồm xử lý các ngôn ngữ tự nhiên, nhận dạng giọng nói và thị giác, quản lý hệ thống.

Bản chất của AI là mô hình, thuật toán được huấn luyện trên một tập dữ liệu để giải quyết một bài toán rất cụ thể. AI giống như một đứa bé sơ sinh vậy, nó bắt đầu thu nhận hình ảnh và học cách phân loại. Nó chỉ có thể nhận diện gương mặt ba, mẹ hoặc ông bà chứ không thể làm gì khác nữa.

Hiện nay, công nghệ AI có 3 hướng chính:

* Xử lý hình ảnh (Computer Vision)
* Xử lý ngôn ngữ tự nhiên (Natural Language Processing)
* Xử lý tín hiệu âm thanh (Audio Signal Processing)

Trong đồ án này, đề tài em thực hiện chủ yếu phát triển theo hướng xử lý hình ảnh.

* + 1. **Ứng dụng AI trong thực tế**
* Hệ thống nhận diện khuôn mặt thông qua các camera được gắn tại sân bay và các tòa nhà
* Các trợ lý ảo (như Siri, Google Assistant, Alexa) có khả năng nghe, hiểu, trả lời và làm việc cho mình
* Những ứng dụng trong y sinh, dùng AI để chẩn đoán bệnh dựa trên phim chụp X-quang, X-ray và MRI
* Quen thuộc hơn thì có những dòng smart TV, áp dụng công nghệ AI để cải tiến chất lượng hình ảnh hoặc nhận diện giọng nói…
  1. **Tổng quan bài toán nhận dạng biển số xe :**
     1. **Khái niệm biển số xe :**

Ở [Việt Nam](https://vi.wikipedia.org/wiki/Vi%E1%BB%87t_Nam), biển kiểm soát xe cơ giới (hay còn gọi tắt là biển kiểm soát, biển số xe) là [tấm biển gắn trên mỗi xe cơ giới](https://vi.wikipedia.org/wiki/Bi%E1%BB%83n_%C4%91%C4%83ng_k%C3%BD_xe), được cơ quan [công an](https://vi.wikipedia.org/wiki/C%C3%B4ng_an_nh%C3%A2n_d%C3%A2n_Vi%E1%BB%87t_Nam) cấp (đối với xe quân sự do Bộ Quốc phòng cấp) khi mua xe mới hoặc chuyển nhượng xe. Biển số xe được làm bằng hợp kim [nhôm](https://vi.wikipedia.org/wiki/Nh%C3%B4m) [sắt](https://vi.wikipedia.org/wiki/S%E1%BA%AFt), có dạng [hình chữ nhật](https://vi.wikipedia.org/wiki/H%C3%ACnh_ch%E1%BB%AF_nh%E1%BA%ADt) hoặc hơi [vuông](https://vi.wikipedia.org/wiki/H%C3%ACnh_vu%C3%B4ng), trên đó có in [số](https://vi.wikipedia.org/wiki/S%E1%BB%91) và [chữ](https://vi.wikipedia.org/wiki/Ch%E1%BB%AF_vi%E1%BA%BFt) (biển xe dân sự không dùng các chữ cái I, J, O, Q, W. Chữ R chỉ dùng cho xe rơ-moóc, sơ-mi rơ-moóc) cho biết: Vùng và địa phương quản lý, các con số cụ thể khi tra trên máy tính còn cho biết danh tính người chủ hay đơn vị đã mua nó, thời gian mua nó phục vụ cho công tác an ninh, đặc biệt trên đó còn có hình [Quốc huy Việt Nam](https://vi.wikipedia.org/wiki/Qu%E1%BB%91c_huy_Vi%E1%BB%87t_Nam) dập nổi.

Tiêu chuẩn về kích thước: Ở mỗi nước thường có tiêu chuẩn về kích thước nhất định, còn riêng Việt Nam tỉ lệ kích thước giữa các biển số là gần như giống nhau. Biển số xe có 2 loại, kích thước như sau: Loại biển số dài có chiều cao 110 mm, chiều dài 470 mm; loại biển số ngắn có chiều cao 200 mm, chiều dài 280 mm nên ta sẽ giới hạn tỉ lệ cao/rộng là 3.5 ≤ cao/rộng ≤ 6.5 (biển một hàng) và 0.8 ≤ cao/rộng ≤ 1.5 (biển hai hàng).

Số lượng kí tự trong biển số xe nằm trong khoảng [7,9]. Chiều cao của chữ và số: 80mm, chiều rộng của chữ và số: 40mm.

Từ những đặc điểm trên ta có thể thiết lập nhưng thông số, điều khiển để lọc chọn những đối tương phù hợp mà ta cần.

* + 1. **Xử lý ảnh và Open CV :**

Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người. Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế,....

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. Opencv có rất nhiều ứng dụng

* + 1. **Nhận dạng đối tượng**

Nhận dạng đối tượng là quá trình xác định và phân loại các đối tượng trong hình ảnh hoặc dữ liệu đầu vào. Để làm điều này, chúng ta sử dụng các công nghệ như mạng nơ-ron tích chập (CNNs), phát hiện đối tượng (Object Detection), và học sâu (Deep Learning) để trích xuất đặc trưng từ hình ảnh và xác định các đối tượng cụ thể trong đó. Nhận dạng đối tượng có nhiều ứng dụng trong trí tuệ nhân tạo, bao gồm trong lĩnh vực xe tự hành, giám sát an ninh, xử lý hình ảnh y tế và nhiều lĩnh vực khác.

* + 1. **Thị giác máy tính**

Thị giác máy tính trong học máy là lĩnh vực nghiên cứu và ứng dụng trí tuệ nhân tạo để máy tính có khả năng hiểu và xử lý thông tin từ hình ảnh và video. Nó sử dụng mạng nơ-ron tích chập (CNNs) để tự động trích xuất đặc trưng từ hình ảnh và thực hiện các nhiệm vụ như phân loại, phát hiện đối tượng, và nhiều ứng dụng khác. Thị giác máy tính có nhiều ứng dụng quan trọng trong thế giới thực và luôn tiếp tục phát triển.

* + 1. **Hướng giải quyết**

Hiện nay trên thế giới đã có rất nhiều cách tiếp cận khác nhau với việc nhận dạng biển số xe, tuy nhiên trong phạm vi đồ án này em sẽ giải quyết vấn đề theo 3 bước chính:

* 1. Phát hiện vị trí và tách biển số xe từ một hình ảnh có sẵn từ đầu vào là camera
  2. Phân đoạn các kí tự có trong biển số xe
  3. Nhận diện các kí tự đó rồi đưa về mã ASCII

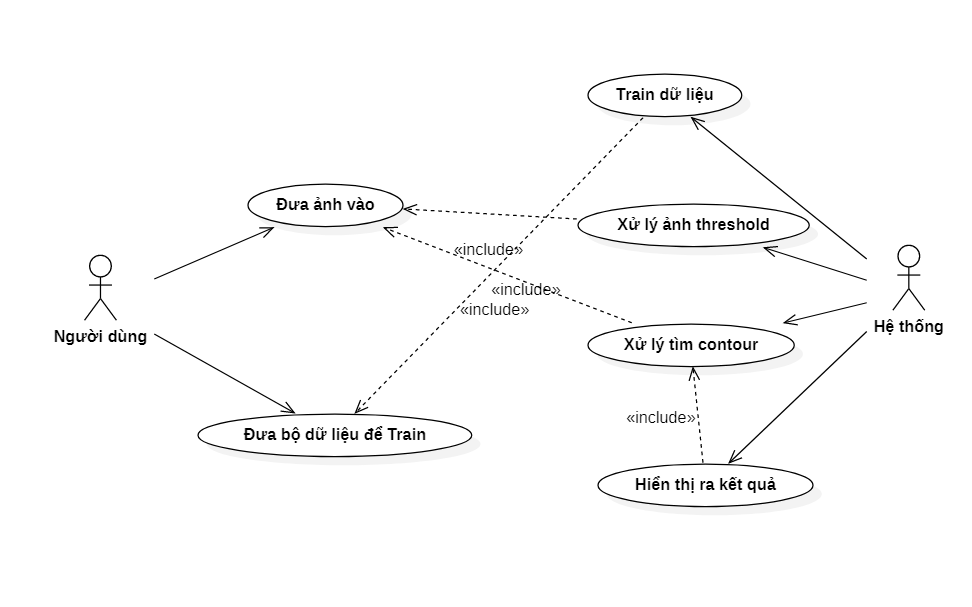
Phát hiện vị tí và tách biển số xe

Phân đoạn kí tự

Nhận diện kí tự

# CHƯƠNG 3 : MÔ HÌNH THỰC NGHIỆM

* 1. **Phân tích yêu cầu**
     1. **Yêu cầu chức năng**
* Có thể đọc dữ liệu ảnh đầu vào
* Có thể đọc dữ liệu từ video
* Tách được biển số từ video và ảnh đầu vào
* Đọc biển số đã tách
* Hiển thị biển số nhận dạng được ra màng hình
* Có giao diện dễ sử dụng
  + 1. **Yêu cầu hệ thống**
* Chương trình chạy trên hệ điều hành windows
* Chương trình được viết trên Visua Studio Code
* Phiên bản python 3.7.8
  1. **Biểu đồ sử dụng**
     1. **Chức năng đưa dữ liệu vào**
* Các tác nhân: Người dùng
* Điều kiện trước: Người dùng phải chọn ảnh/video đầu vào
* Điều kiện sau: Hệ thống nhận được ảnh/video
* Mô tả: Ca sử dụng này người dùng thực hiện nạp dữ liệu đầu vào và hệ thống đọc được dữ liệu ảnh/video
  + 1. **Chức năng đưa ra ảnh đã threshold**
* Các tác nhân: hệ thống
* Điều kiện trước: Có ảnh đầu vào
* Điều kiện sau: Hệ thống đưa ra bức ảnh đã threshold
* Mô tả: Ca sử dụng này nhằm đưa ra ảnh sau khi threshold để dễ dàng nhận dạng biển số xe
  + 1. **Chức năng đưa ra các contour tìm được**
* Các tác nhân: Hệ thống
* Điều kiện trước: Ảnh đã qua xử lý và được tách biển số ra
* Điều kiện sau: Hiển thị các contour tìm được
* Mô tả: Ca sử dụng này nhằm đưa ra các contour tìm được để nhận dạng dễ dàng
  + 1. **Chức năng nhận dạng và hiển thị biển số xe**
* Các tác nhân: Hệ thống
* Điều kiện trước: Ảnh đã qua xử lý
* Điều kiện sau: Hệ thống đưa ra kết quả sau khi nhận dạng và in lên ảnh gốc
* Mô tả: Đưa ra kết quả nhận dạng được



Hình 3.1 Lưu đồ nhận dạng biển số xe

* 1. **Thiết kế hệ thống**

Hệ thống tự động nhận dạng biển số xe là hệ thống có khả năng phân tích, xử lý hình ảnh biển số xe trên các hình ảnh chụp được từ các thiết bị thu hình, cho kết quả đầu ra là chuỗi ký tự trên biển số. Nguồn hình ảnh cho ứng dụng có rất nhiều nhưng thường là hình ảnh được thu nhận trực tiếp từ camera. Đầu vào của hệ thống là ảnh phương tiện giao thông có biển số xe. Đầu ra là chuỗi ký tự trên biển số xe.

* 1. **Phát hiện vị trí và tách biển số xe**
     1. **Mô hình**

Đầu vào là file video/ảnh

Tách ra từng frame ảnh

Chuyển sang ảnh xám

Tăng độ tương phản

Kết quả

Giảm nhiễu bằng bộ lọc Gauss

Mô hình KNN

Threshold

Phát hiện cạnh Canny

Tìm và tách vùng kí tự

Tìm vị trí và lọc biển số bằng Contour

Xoay biển số

Hình 3.2 Lưu đồ nhận dạng biển số xe

Đầu tiên từ video đưa vào ta sẽ cắt ra được từng frame ảnh để xử lý tách biển số. Sau đó ta sẽ chuyển ảnh sang màu xám. Tiếp theo ta tăng độ tương phản với hai phép toán hình thái học Top Hat và Black Hat để làm nổi bật thêm biển số giữa phông nền, hỗ trợ cho việc xử lý nhị phân sau này. Sau đó, ta giảm nhiễu bằng bộ lọc Gauss để loại bỏ những chi tiết nhiễu có thể gây ảnh hưởng đến quá trình nhận diện, đồng thời làm tăng tốc độ xử lý.

Việc lấy ngưỡng sẽ giúp ta tách được thông tin biển số và thông tin nền, ở đây em chọn lấy ngưỡng động (Adaptive Threshold). Tiếp đó ta sử dụng thuật toán phát hiện cạnh Canny để trích xuất những chi tiết cạnh của biển số. Trong quá trình xử lý máy tính có thể nhầm lẫn biển số với những chi tiết nhiễu, việc lọc lần cuối bằng các tỉ lệ cao/rộng hay diện tích của biển số sẽ giúp xác định được đúng biển số. Cuối cùng, ta sẽ xác định vị trí của biển số trong ảnh bằng cách vẽ Contour bao quanh.

* + 1. **Chuyển sang ảnh xám**

Ảnh xám (Gray Scale) đơn giản là một hình ảnh trong đó các màu là các sắc thái của màu xám với 256 cấp độ xám biến thiên từ màu đen đến màu trắng, nằm trong giải giá trị từ 0 đến 255, nghĩa là cần 8 bits hay 1 byte để biểu diễn mỗi điểm ảnh này. Lý do cần phải phân biệt giữa ảnh xám và các ảnh khác nằm ở việc ảnh xám cung cấp ít thông tin hơn cho mỗi pixel. Với ảnh thông thường thì mỗi pixel thường được cung cấp 3 trường thông tin trong khi với ảnh xám chỉ có 1 trường thông tin, việc giảm khối lượng thông tin giúp tăng tốc độ xử lý, đơn giản hóa giải thuật nhưng vẫn đảm bảo các tác vụ cần thiết

Ở bài này em sẽ chuyển ảnh xám từ hệ màu HSV thay vì RGB vì với không gian màu HSV ta có ba giá trị chính là: Vùng màu (Hue), độ bão hòa (Saturation), cường độ sáng (Value). Vì lý do đó không gian màu HSV thích nghi tốt hơn đối với sự thay đổi ánh sáng từ môi trường ngoài. Khi chuyển đổi, ảnh xám ta cần là ma trận các giá trị cường độ sáng tách ra từ hệ màu HSV.

* + 1. **Tăng độ tương phản**
       1. **Black Hat**

Black Hat là kết quả của phép trừ ảnh đóng với ảnh gốc. Điều này được sử dụng để trích xuất các chi tiết nhỏ từ các cấu trúc trong ảnh gốc, đặc biệt tăng cường các đối tượng sáng trong nền tối.

Hình 3.3 Black hat

* + - 1. **Top Hat**

Top Hat là kế quả của phép trừ ảnh đã mở với ảnh gốc. Điều này được sử dụng để trích xuất các chi tiết nhỏ từ các cấu trúc trong ảnh gốc, đặc biệt tăng cường đối tượng tối trong nền sáng.

Hình 3.4 Top Hat

Để làm tăng độ tương phản của biển số, em sử dụng chủ yếu hai phép Top Hat và Black Hat. Ý tưởng chung là ảnh đầu ra sẽ là ảnh gốc cộng thêm ảnh qua phép Top Hat và trừ đi ảnh qua phép Black Hat. Những chi tiết đã sáng sẽ sáng hơn và những chi tiết tối lại càng tối hơn, từ đó sẽ làm tăng độ tương phản cho biển số.

Hình 3.5 Ảnh sau khi tăng tương phản

* + 1. **Giảm nhiễu bằng bộ lọc Gausian**

Noise được hiểu cơ bản là các dạng chấm hạt nhỏ phân bố trên hình ảnh. Noise có thể làm biến dạng các chi tiết trong ảnh khiến cho chất lượng ảnh thấp.

Bộ lọc Gaussian được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gaussian , sau đó cộng chúng lại để tạo thành ảnh đầu ra.

Hình 3.6 Ảnh sau khi lọc Gaussian

* + 1. **Phát hiện cạch**

Trong hình ảnh, thường tồn tại các thành phần như: vùng trơn, góc/cạnh và nhiễu. Cạnh trong ảnh mang đặc trưng quan trọng, thường là thuộc đối tượng trong ảnh. Do đó, để phát hiện cạnh trong ảnh, có nhiều giải thuật khác nhau như toán tử Sobel, toán tử Prewitt, Zero crossing .... nhưng ở đây em chọn giải thuật Canny vì hương pháp này hơn hẳn các phương pháp khác do ít bị tác động của nhiễu và cho khả năng phát hiện các biên yếu. Phương pháp này đi theo 4 bước chính:

* Bước 1 : Giảm nhiễu

Thông thường để giảm nhiễu dùng bộ lọc gaussian kích thước 5x5

* Bước 2 : Tính toán Gradient

Vì đường biên trong ảnh là nơi phân cách giữa các đối tượng khác nhau, nên tại đó gradien của nó sẽ có biến đổi mạnh mẽ nhất. Để tính toán gradien trong ảnh, ta có thể sử dụng bộ lọc Sobel, hoặc trực tiếp nhân chập ma trận ảnh với các mặt nạ theo hướng x và y, chẳng hạn:

|  |  |
| --- | --- |
|  |  |

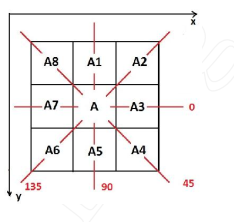
Tìm gradient và hướng được làm tròn về 4 hướng: hướng ngang (0 độ), hướng chéo bên phải (45 độ), hướng dọc (90 độ) và hướng chéo trái (135 độ).

|  |  |
| --- | --- |
|  |  |

* Bước 3 : Loại bỏ những điểm không phải là cực đại

Bước này sẽ tìm ra những điểm ảnh có khả năng là biên ảnh nhất bằng cách loại đi những giá trị không phải là cực đại trong bước tìm gradien ảnh ở trên. Ta thấy rằng, với giá trị của góc θ  ở trên thì biên của đối tượng có thể tuân theo bốn hướng, và ta có bốn khả năng sau:

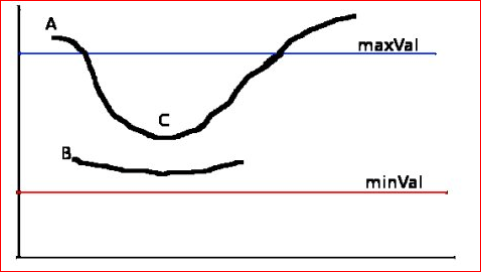
* Nếu θ = 0 , khi đó điểm A sẽ được xem xét là một điêm trên biên độ nếu độ lớn gradien tại A lớn hơn gradien của các điểm tại A3, A7.
* Nếu θ = 45, khi đó điểm A được xem là một điểm trên biên độ nếu độ lớn gradien tại A lớn hơn độ lớn gradien của các điểm tại A4, A8
* Nếu θ = 90, khi đó điểm A sẽ được coi là một điểm nằm trên biên độ nếu độ lớn gradien tại A lớn hơn độ lớn gradien của các điểm tại A1, A5.
* Nếu θ = 135, khi đó điểm A được xem là một điểm nằm trên biên độ nếu độ lớn gradien tại A lớn hơn độ lớn gradien của các điểm tại A2, A6



Hình 3.7 Loại bỏ những điểm không là cực đại

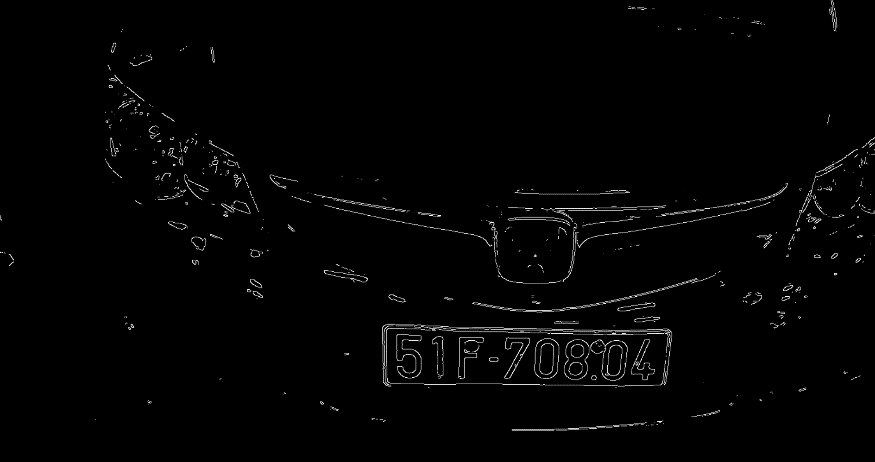
* Bước 4 : Lọc ngưỡng

Lọc ngưỡng: ta sẽ xét các pixel dương trên mặt nạ nhị phân kết quả của bước trước. Nếu giá trị gradient vượt ngưỡng max\_val thì pixel đó chắc chắn là cạnh. Các pixel có độ lớn gradient nhỏ hơn ngưỡng min\_val sẽ bị loại bỏ. Còn các pixel nằm trong khoảng 2 ngưỡng trên sẽ được xem xét rằng nó có nằm liền kề với những pixel được cho là "chắc chắn là cạnh" hay không. Nếu liền kề thì ta giữ, còn không liền kề bất cứ pixel cạnh nào thì ta loại. Sau bước này ta có thể áp dụng thêm bước hậu xử lý loại bỏ nhiễu (tức những pixel cạnh rời rạc hay cạnh ngắn) nếu muốn



Hình 3.8 Lọc ngưỡng

Sau khi sử dụng phát hiện biên canny, dù đã trích xuất được những chi tiết cạnh của biển số, tuy nhiên vẫn còn quá nhiều chi tiết thừa trong hình ảnh, từ đây chúng ta sẽ vẽ contour, áp dụng nhưng đặc điểm của biển số để lọc lấy ra biển số chính xác.



Hình 3.9 Ảnh sau khi phát hiện biên canny

* + 1. **Vẽ contour**

Có thể hiểu Contour là tập hợp các điểm tạo thành đường cong kín bao quanh một đối tượng nào đó. Thường dùng để xác định vị trí, đặc điểm của đối tượng.

Hàm trong OpenCV được biểu diễn như sau:

**findContours**(InputOutputArray image, OutputArratOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset = Point() )Cáctham số:   
**image** : hình ảnh cần tìm biên, là ảnh nhị phân.  
**contours** : lưu trữ các đường biên tìm được, mỗi đường biên được lưu trữ dưới dạng một vector của các điểm.  
**hierarchy** :  chứa thông tin về hình ảnh như số đường viền, xếp hạng các đường viền theo kích thước, trong ngoài, ..  
**mode** :CV\_RETR\_EXTERNAL : khi sử dựng cờ này nó chỉ lấy ra những đường biên bên ngoài, nhưng biên bên trong của vật thể bị loại bỏ.  
CV\_RETR\_LIST : Khi sử dụng cờ này nó lấy ra tất cả các đường viền tìm được.  
CV\_RETR\_CCOMP : khi sử dụng cờ này nó lấy tất cả những đường biên và chia nó làm 2 level, những đường biên bên ngoài đối tượng, và những đường biên bên trong đối tượng.  
CV\_RETR\_TREE : khi sử dụng cờ này nó lấy tất cả các đường biên và tạo ra một hệ thống phân cấp đầy đủ của những đường lồng nhau.  
**method** :CV\_CHAIN\_APPROX\_NONE : sử dụng cờ này sẽ lưu trữ tất cả các điểm của đường viền .  
CV\_CHAIN\_APPROX\_SIMPLE : Ví dụ : một hình chữ nhật sẽ được mã hoá bằng toạ độ của 4 đỉnh.  
CV\_CHAIN\_APPROX\_TC89\_L1 or CV\_CHAIN\_APPROX\_TC89\_KCOS :  Áp dụng thuật toán xấp xỉ Tech-Chin.



Hình 3.10 Vẽ Contour với OpenCV

Vì có quá nhiều đường bao quanh các vật không phải là biển số nên chúng ta sẽ áp dụng những đặc trưng riêng về tỉ lệ cao/ rộng, diện tích trong khung hình cố định để lọc ra đúng biển số.

Đầu tiên ta làm xấp xỉ contour thành một hình đa giác và chỉ lấy những đa giác nào chỉ có 4 cạnh. Nghĩa là lúc xấp xỉ contour bộ nhớ chỉ ghi nhớ vị trí các đỉnh của đa giác đó thành một mảng. Số cạnh của đa giác sẽ bằng số đỉnh và bằng chiều dài của mảng đó.

Tiếp theo ta tính toán tỉ lệ cao/rộng và diện tích của biển số phù hợp, sau đó ta lưu tất cả những biển số có trong hình dưới dạng tọa độ các đỉnh

Từ đây, ta cắt hình ảnh biển số từ các tọa độ vị trí đã biết để phục vụ cho mục đích tiếp theo “Tách các kí tự trong biển số”. Lưu ý ở đây ta cắt từ ảnh nhị phân luôn để máy tính xử lý nhanh hơn, tốn ít thời gian hơn.

* 1. **Phân đoạn kí tự**
     1. **Xoay biển số**

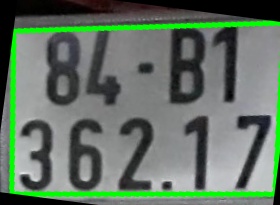
Khi chụp ảnh đầu vào, không phải lúc nào biển số cũng ở chính diện, có thể bị méo sang trái, sang phải, nghiêng góc dẫn đến nếu cứ sử dụng ảnh biển số đã cắt mà không điều chỉnh góc độ dẫn đến ảnh kí tự được cắt ra đưa vào bộ nhận diện rất dễ bị sai. Ví dụ giữa số 1 và số 7, số 2 và chữ Z, chữ B và số 8,...



Hình 3.11 Ảnh biển số chưa xoay

Phương pháp xoay ảnh em sử dụng ở đây là:

1. Lọc ra tọa độ 2 đỉnh A,B nằm dưới cùng của biển số
2. Từ 2 đỉnh có tọa độ lần lượt là A(x1, y1) và B(x2,y2) ta có thể tính được cạnh đối và cạnh kề của tam giác ABC
3. Ta tính được góc quay
4. Xoay ảnh theo góc quay đã tính. Nếu ngược lại điểm A nằm cao hơn điểm B ta cho góc quay âm



Hình 3.12 Ảnh biển số đã xoay

* + 1. **Tìm vùng đối tượng**

Từ ảnh nhị phân, ta lại tìm contour cho các kí tự (phần màu trắng). Sau đó vẽ những hình chữ nhật bao quanh các kí tự đó. Tuy nhiên việc tìm contour này cũng bị nhiễu dẫn đến việc máy xử lý sai mà tìm ra những hình ảnh không phải kí tự. Ta sẽ áp dụng các đặc điểm về tỉ lệ chiều cao/rộng của kí tự, diện tích của kí tự so với biển số, những đường màu vàng là đường contour và nếu so sánh với ảnh nhị phân thì có rất nhiều đường nhiễu như đường viền biển số, dấu gạch, dấu chấm... Sau khi đã áp dụng các điều kiện thì sẽ vẽ ra những hình chữ nhật màu xanh bao quanh các kí tự.



Hình 3.13 Tìm vùng đối tượng

* + 1. **Tìm và tách kí tự**

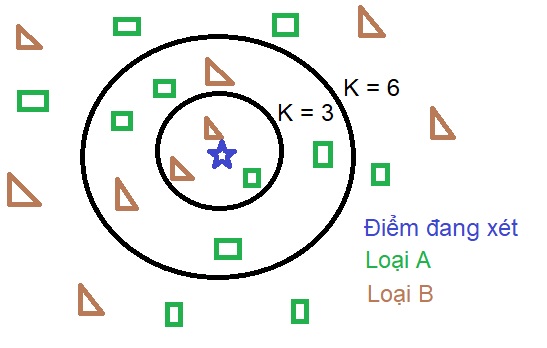
Sau khi đã nhận dạng từng kí tự bằng hình chữ nhật và cũng đã có tọa độ vị trí 4 đỉnh của hình đó, ta lúc này có thể cắt hình ảnh kí tự đó ra phục vụ cho giai đoạn sau “Nhận diện kí tự”. Lưu ý ở đây ta cắt ảnh nhị phân chứ không cắt từ ảnh gốc.

* 1. **Nhận diện kí tự**
     1. **Thuật toán KNN**

KNN là một trong những thuật toán học có giám sát đơn giản nhất trong Machine Learning, có thể sử dụng cho cả bài toán phân loại và hồi quy. Ví dụ để phân biệt các giống chó khác nhau, thì khi so sánh những đặc điểm mắt, lông, tai … với những giống chó đã thấy thì ta có thể xem được nó thuộc giống nào.

KNN hoạt động theo quy trình gồm 4 bước chính:

1. Xác định tham số K (số láng giềng gần nhất).
2. Tính khoảng cách từ điểm đang xét đến tất cả các điểm trong tập dữ liệu cho trước
3. Sắp xếp các khoảng cách đó theo thứ tự tăng dần
4. Xét trong tập K điểm gần nhất với điểm đang xét, nếu số lượng điểm của loại nào cao hơn thì coi như điểm đang xét thuộc loại đó



Hình 3.14 Ví dụ về KNN

Việc điểm đang xét thuộc loại nào còn phụ thuộc vào hệ số K hay trọng số khoảng cách... mà người dùng đặt sao cho phù hợp với bài toán đang xét. Chẳng hạn ở hình trên nếu ta xét K = 3 thì điểm đang xét sẽ thuộc loại B, ngược lại nếu K = 6 thì nó thuộc loại A. Ngoài ra người ta có thể để trọng số cao hơn cho những điểm gần hơn hay ít khi sử dụng K = 1 để đảm bảo kết quả đầu ra được tối ưu.

Thông thường việc tính khoảng cách đến các điểm sẽ theo công thức Euclid:

Khi thực hiện so sánh có thể bỏ qua dấu căn bậc 2. Ngoài ra nếu như khoảng cách giữa các biến quá lớn như biến x lớn hơn xấp xỉ 1000000 lần thì ta cũng cần chuẩn hóa lại dữ liệu theo công thức:

* + 1. **Hướng giải quyết**

Ở giai đoạn cuối này được thực hiện theo những bước sau:

1. Tạo tập dữ liệu để huấn luyện
2. Huấn luyện mô hình KNN
3. Đưa hình ảnh từ bước “Phân đoạn kí tự” vào mô hình KNN đã tạo để đưa ra kết quả
4. In ra kết quả biển số

Bước 1 và 2 ta sẽ tạo ra mô hình KNN riêng biệt với code chính. Để khi cần nhận diện kí tự ta không cần phải làm lại các bước từ đầu. Đầu tiên em tạo tập dữ liệu (tập hình ảnh của các chữ số và kí tự) để train từ phần mềm paint. Trong phần mềm Paint ta viết các chữ số và kí tự (trừ kí tự O, I, J) với phông chữ “Biển số xe Việt Nam”, có thể xoay các kí tự này lần lượt với các góc . Kết quả có dạng như sau:

Bước 3 và 4. Ta thực hiện đưa ảnh đang xét vào và tính khoảng cách đến tất cả các điểm trong mẫu, kết quả sẽ là mã ASCII đại điện cho hình ảnh đó. Cuối cùng ta in biển số xe ra hình. Tuy nhiên ở Việt Nam có hai loại biển số là biển một hàng và biển hai hàng. Về ý tưởng chung để phân biệt hai hàng này ta dựa vào vị trí của hình ảnh kí tự, nếu vị trí nằm thấp 1/3 chiều cao của biển số thì kí tự sẽ được xếp vào hàng một. Ngược lại sẽ được xếp vào hàng hai.



Hình 3.15 Tập dữ liệu kí tự

|  |  |
| --- | --- |
|  |  |
| Hình 3.16 Biển số trước khi nhận diện | Hình 3.17 Biển số sau khi nhận diện |



Hình 3.18 Biển số xe được in lên ảnh gốc

* 1. **Kết quả chương trình**

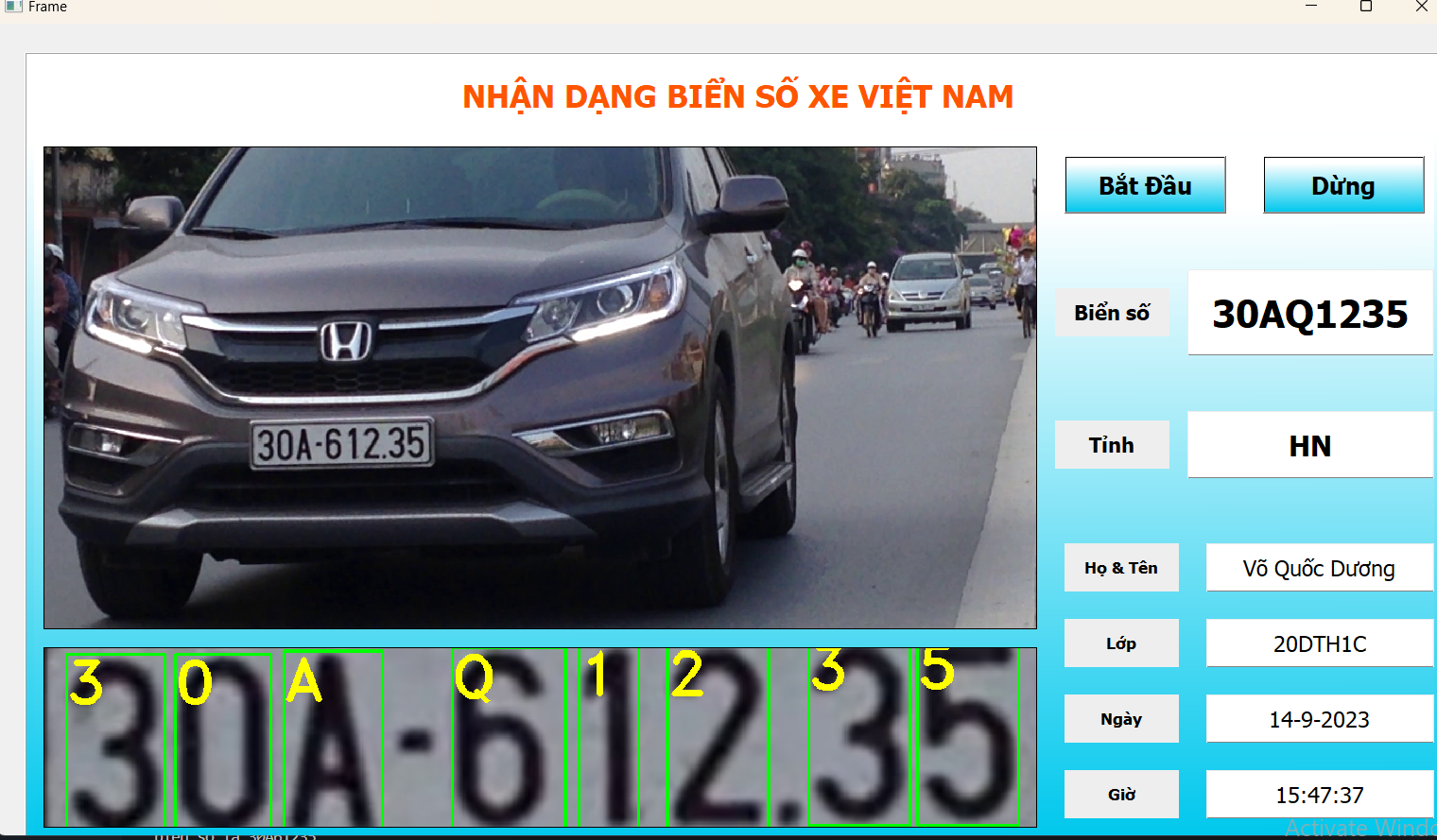
Sau khi nhấn nút chọn ảnh từ thư mục tùy ý, ảnh sẽ hiện lên ở khung label góc trên bên phải. Tiếp theo ta nhấn nút nhận dạng để tiến hành nhận dạng biển số và in lên label chính giữa. Nút thông tin dùng để hiển thị các thông tin của biển số xe như : tên, tỉnh,…



Hình 3.19 Ảnh demo nhận dạng biển số xe từ ảnh

Ta nhấn nút bắt đầu để tiến hành chạy video và nhận dạng ảnh, các thông tin trích xuất từ ảnh sẽ hiện ra ở các label bên phải như : biển số, tỉnh, tên,….Ta có thể nhấn nút dừng để dừng video và quá trình nhận dạng.

Ở phía dưới em cho hiển thị thêm ảnh vẽ contour để chúng ta hiểu rõ hơn bản chất.



Hình 3.20 Ảnh demo nhận dạng video

# CHƯƠNG 4 : KẾT LUẬN

Trong bài báo cáo này, chúng em đề xuất một hệ thống mới cho Nhận dạng Biển số Xe dựa trên việc lựa chọn đặc tính quan trọng. Hệ thống đề xuất gồm ba bước chính: (i) tiền xử lý và phân đoạn vùng ký tự, (ii) trích xuất các đặc tính quan trọng từ Vùng quan tâm (ROI), và (iii) kết hợp các đặc tính bằng một kỹ thuật mới.

Chúng em đã tập trung vào giải quyết các thách thức như biến đổi ánh sáng và tắc nghẽn giao thông. Kết quả mô phỏng cho thấy rằng phương pháp đề xuất đã thành công trong việc xử lý những khó khăn này. Hơn nữa, việc sử dụng một thiết kế xếp tầng có thể hiệu quả trong việc quản lý các vấn đề ở giai đoạn đầu, và lựa chọn đặc tính quan trọng có thể cải thiện độ chính xác của quá trình phân loại.

Tương lai, chúng em dự định bổ sung thêm bộ tính năng và triển khai các kỹ thuật lựa chọn đặc tính tối ưu hóa để giảm thiểu lỗi và tăng khả năng nhận dạng trong ứng dụng thực tế.Top of Form

## Hạn chế:

* Còn một số chức năng chưa hoàn thành
* Tốc độ khởi chạy chương trình còn chậm, chưa tối ưu
* Nhận dạng có thể bị sai, tỉ lệ chưa được cao
* Ảnh/video đầu vào phải rõ nếu không sẽ không nhận dạng được

## Hướng phát triển:

* Khắc phục các hạn chế, lập trình và kiểm thử các chức năng còn thiếu.
* Tối ưu mã nguồn tốc độ chạy của chương trình
* Thiết kế giao diện dễ nhìn, đẹp mắt hơn

**TÀI LIỆU THAM KHẢO**

*[1] Tài liệu nhận dạng biển số*

<https://viblo.asia/p/nhan-dien-bien-so-xe-viet-nam-Do754P9L5M6>

*[2] Nghiên cứu nhận dạng biển số bằng OpenCv*

<https://skhcn.quangbinh.gov.vn/3cms/upload/khcn/File/TapChiKHCN/2018/so1/12.pdf>

*[3] Thuật toán KNN và ví dụ đơn giản trong ngành ngân hàng.*

<https://bigdatauni.com/vi/tin-tuc/thuat-toan-knn-va-vi-du-don-gian-trong-nganh-ngan-hang.html>

*[4] K-nearest neighbor trong opencv2*

<https://viblo.asia/p/k-nearest-neighbour-trong-opencv2-V3m5W2wWlO7>

*[5] Chris Dahms (2016), KNN character recognition python.*

<https://www.youtube.com/watch?v=c96w1JS28AY&t=6s>

*[6] Nghiên cứu về Contour*

<https://congdongopencv.blogspot.com/2017/11/tim-hieu-ve-contour-moments-trong-xu-ly.html>