

**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



Nguyễn Hoàng Trung Vương – 1910156

Tôn Thất Nhật Minh – 1910136

Đào Xuân Hải – 1911147

**NGHIÊN CỨU PHƯƠNG PHÁP NHẬN DIỆN XE**

**KHÓA LUẬN TỐT NGHIỆP KỸ SƯ CÔNG NGHỆ THÔNG TIN**

**GIÁO VIÊN HƯỚNG DẪN**

Th.S Đoàn Minh Khuê

***Đà Lạt, 2023***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

Đà Lạt, ngày ….. tháng …… năm ……

Giáo viên hướng dẫn

[Ký tên và ghi rõ họ tên]

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 1**

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

Đà Lạt, ngày ….. tháng …… năm ……

Giáo viên hướng dẫn

[Ký tên và ghi rõ họ tên]

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 2**

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

Đà Lạt, ngày ….. tháng …… năm ……

Giáo viên hướng dẫn

[Ký tên và ghi rõ họ tên]

**Trường Đại học Đà Lạt**

**Khoa Công nghệ Thông tin**

**---****---**

ĐỀ CƯƠNG THỰC HIỆN ĐỒ ÁN

**Tên đề tài:** Phương pháp nghiên cứu nhận diện xe .

**Sinh viên thực hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Email liên hệ** |
| 1 | Nguyễn Hoàng Trung Vương | 1910156 | CTK43 | 1910156@dlu.edu.vn |
| 2 | Tôn Thất Nhật Minh | 1910136 | CTK43 | 1910136@dlu.edu.vn |
| 3 | Đào Xuân Hải | 1911147 | CTK43 | 1911147@dlu.edu.vn |

**Giáo viên hướng dẫn: ThS. Đoàn Minh Khuê**

1. **Mục tiêu đề tài:**

* Ghi lại và lưu trữ hình ảnh biển số xe
* Phát hiện và tách vùng biển số xe.
* Trích xuất vùng chứa ký tự của biển số xe.
* In được đúng ký tự chứa trong biển số xe được trích xuất.

1. **Nội dung đề tài:**

* Tìm hiểu lý thuyết về ngôn ngữ Python, nghiên cứu thư viện OpenCV và sử dụng Python trong xây dựng code nhận diện xe.
* Hiểu rõ được cách thức hoạt động và các tính năng, module phổ biến của OpenCV.
* Ứng dụng có các chức năng sau: quản trị, lưu trữ dữ liệu hình ảnh biển số xe, kết nối với dữ liệu biển số xe có sẵn, quản lý xe ra/ vào (Đếm số lượng xe nhập/ xuất), xuất file excel.
* Thiết kế cơ sở dữ liệu hoặc lấy dữ liệu có sẵn để training mô hình.
* Thiết kế giao diện ứng dụng.
* Viết báo cáo đề tài.

1. **Phần mềm và công cụ sử dụng:** PyCharm, Visual Code, GitHub, Camera.
2. **Dự kiến kết quả đạt được:**

* Tích lũy được nhiều kinh nghiệm và kiến thức về lĩnh vực AI, Tesseract OCR engine, mô hình CNN, OpenCV .
* Xây dựng được ứng dụng có khả năng ghi lại và lưu trữ hình ảnh biển số xe/ quản lý được số lượng và biển số xe nhập/ xuất bãi giữ nhờ tính năng nhận dạng biển số. Giám sát bãi giữ xe dễ dàng.
* Hoàn thành báo cáo đề tài.

1. **Kế hoạch thực hiện**

* **Giai đoạn 1 :** 03/03/2023 - 09/04/2023
* Tìm hiểu mô hình cần sử dụng để phục vụ đề tài.
* Thiết kế cơ sở dữ liệu hoặc sử dụng CSDL có sẵn.
* Áp dụng mô hình đã chọn để phục vụ đề tài.
* Training mô hình.
* **Giai đoạn 2 :** 10/04/2023 - 16/04/2023
* Thiết kế phần mềm.
* Viết báo cáo tiến độ lần 1.
* **Giai đoạn 3 :** 08/05/2022 – 14/05/2022
* Hoàn thiện phần mềm.
* Viết báo cáo tiến độ lần 2.
* **Giai đoạn 4 :** 14/05/2022 – 30/05/2022
* Tiếp tục nâng cấp phần mềm.
* Báo cáo cuối kỳ và nộp khoá luận tốt nghiệp.

1. **Kết luận hướng phát triển**

* **Kết luận :** Phương pháp nhận diện xe của ngành công nghệ thông tin tại nước ta đã có những tác động tích cực đến lĩnh vực giám sát tự động cụ thể hơn là việc giám sát đối với các phương tiện giao thông. Hệ thống giám sát đều lấy biển số xe làm mục tiêu giám sát để giám sát các phương tiện trong hệ thống giao thông. Tuy nhiên nó vẫn còn là lĩnh vực cần nhiều công sức đầu tư để cải thiện phương pháp cũng như hiểu quả nhận diện một cách chính xác nhất. Nhóm em hy vọng hệ thống nhận diện biển số xe nói riêng và nhận dạng nói chung sẽ được sử dụng rộng rãi với độ chính xác cao phục vụ cho nhiều lĩnh vực cuộc sống để giám sát, quản lý công việc thay con người.
* **Hướng phát triển :**
* Tiếp tục hoàn thiện Tesseract OCR engine.
* Training cho hệ thống.
* Nghiên cứu phương pháp mới.

*Đà Lạt, ngày 30 tháng 05 năm 2023*

**Giáo viên hướng dẫn SV thực hiện**

(Ký tên) (Ký tên)

**BCN Khoa Tổ trưởng Bộ môn**

(Ký tên) (Ký tên)

**LỜI CAM ĐOAN**

Đề tài **“Nghiên cứu phương pháp nhận diện xe”** là công trình nghiên cứu của riêng của nhóm em với sự giúp đỡ của nhiều cá nhân, tổ chức hoàn toàn không có sự sao chép giả mạo của các tác giả khác. Các nguồn tài liệu trích dẫn, số liệu nội dung chuyên đề trung thực. Đồng thời nhóm em xin cam kết rằng kết quả quá trình nghiên cứu của đồ án tốt nghiệp này chưa từng được công bố trong bất kì công trình nghiên cứu nào. Nhóm em xin cam đoan và chịu hoàn toàn trách nhiệm trước nhà trường về vấn đề này.

Đà Lạt, *ngày 30 tháng 5 năm 2023*

Sinh viên

**(**Ký tên)

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành tới Thạc sĩ Đoàn Minh Khuê và cô Trần Thị Linh đã hướng dẫn nhóm em trong quá trình thực hiện bài đồ án tốt nghiệp suốt 4 năm học tại trường đại học Đà Lạt. Sự trải nghiệm và kiến thức nhóm em thu được từ hai người đã giúp nhóm phát triển kỹ năng và cải thiện khả năng nghiên cứu của mình.

Nhóm em cũng xin tri ân đến tất cả các giáo viên của khoa Công Nghệ Thông Tin - trường đại học Đà Lạt, những người đã truyền đạt kiến thức, kinh nghiệm và giúp nhóm em phát triển kỹ năng của mình trong suốt hành trình học tập. Những kiến thức nền tảng, các kỹ năng thực tiễn và kinh nghiệm từ các giáo viên đã góp phần giúp nhóm em xây dựng được sự nghiệp và phát triển tốt hơn trong tương lai.

Nhóm em rất biết ơn và hoan nghênh việc các giáo viên của trường đã cống hiến và hướng dẫn tận tình trong suốt 4 năm học tập và nghiên cứu tại trường đại học Đà Lạt. Những kiến thức chuyên môn, sự hỗ trợ về mặt thực tiễn, cũng như sự khuyến khích và động viên của các giáo viên đã giúp nhóm em cảm thấy tự tin hơn khi bước vào cuộc sống và sẽ là động lực vững chắc cho em trên hành trình tương lai.

Một lần nữa, Nhóm em xin chân thành cảm ơn các giáo viên của khoa Công Nghệ Thông Tin - trường đại học Đà Lạt để nhóm em có thể hoàn thành bài đồ án tốt nghiệp của mình một cách tốt đẹp và chuẩn bị cho những thách thức trên hành trình tương lai.

**DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Thuật ngữ và từ viết tắt** | **Diễn giải** |
| Cascade | Phân tầng |
| Classifier | Bộ phân loại |
| Feature | Đặc trưng |
| OpenCV | OpenCV Source Computer Vision  (Thị giác máy tính) |
| OCR | Optical Character Recognition  (Nhận dạng chữ in) |
| Tesseract OCR | Công cụ nhận dạng chữ in |
| Weak classifier | Bộ phân loại yếu |

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc136209591)

[CHƯƠNG 1: GIỚI THIỆU 3](#_Toc136209592)

[***1.1.* *Giới thiệu về đề tài* 3**](#_Toc136209593)

[***1.2.* *Lý do chọn đề tài* 3**](#_Toc136209594)

[***1.3.* *Nội dung đề tài* 3**](#_Toc136209595)

[***1.4.* *Mục tiêu và phạm vi đề tài* 4**](#_Toc136209596)

[***1.5.* *Tầm quan trọng của đề tài* 5**](#_Toc136209597)

[CHƯƠNG 2: TÁCH BIỂN SỐ 6](#_Toc136209598)

[***2.1 Tổng quan về tách biển số 6***](#_Toc136209599)

[***2.2 Tìm vùng màu trắng 7***](#_Toc136209600)

[***2.3 Xác định vùng chứa biển số 9***](#_Toc136209601)

[***2.4 Cắt vùng chứa biển số 11***](#_Toc136209602)

[***2.5 Tìm góc nghiêng và góc xoay 13***](#_Toc136209603)

[***2.5.1. Biến đổi Radon 13***](#_Toc136209610)

[***2.5.1.1.* *Tổng quan về biến đổi Randon* 13**](#_Toc136209611)

[***2.5.1.2.* *Các bước thực hiện* 15**](#_Toc136209612)

[***a.* *Biến đổi về ảnh nhị phân* 16**](#_Toc136209613)

[***b.* *Thực hiện biến đổi randon trên ảnh biên với θ = 0:179* 16**](#_Toc136209614)

[***c.* *Tìm giá trị lớn nhất của R trong biến đổi Radon* 16**](#_Toc136209615)

[***2.5.2.* *Tìm góc nghiêng và xoay* 16**](#_Toc136209616)

[***2.6* *. Cắt biến số chính xác* 19**](#_Toc136209617)

[CHƯƠNG 3: PHÂN ĐOẠN KÝ TỰ 21](#_Toc136209618)

[***3.1. Tổng quan về phân đoạn ký tự* 21**](#_Toc136209619)

[***3.2.* *Nhị phân biển số xe* 21**](#_Toc136209620)

[***3.1.* *Chuẩn hóa biển số* 23**](#_Toc136209621)

[***3.4. Phân đoạn ký tự* 23**](#_Toc136209622)

[***3.4.1. Chương trình chia đôi biển số* 26**](#_Toc136209623)

[***3.4.2. Chương trình phân vùng kí tự* 26**](#_Toc136209624)

[CHƯƠNG 4: NHẬN DẠNG KÝ TỰ 30](#_Toc136209625)

[**4.1. *Tổng quát nhận dạng ký tự* 30**](#_Toc136209626)

[***4.2. Lựa chọn phương pháp* 31**](#_Toc136209627)

[***4.2.1. Phương pháp nhận dạng cổ điển* 31**](#_Toc136209628)

[***4.2.2. Phương pháp ứng dụng mạng neural* 31**](#_Toc136209629)

[***4.3. Giới thiệu về mạng neural (neural networks)* 31**](#_Toc136209630)

[***4.3.1. Khái niệm* 31**](#_Toc136209631)

[***4.3.2. Mô hình của một mạng neural nhân tạo* 32**](#_Toc136209632)

[***4.3.3. Thiết kế 1 mạng Neural* 33**](#_Toc136209633)

[***4.3.4. Cơ sở lý thuyết và giải thuật cho huấn luyện lan truyền ngược* 33**](#_Toc136209634)

[***4.4. Ứng dụng mạng lan truyền ngược vào nhận dạng ký tự* 37**](#_Toc136209635)

[***4.4.1. Thiết kế mạng lan truyền ngược* 37**](#_Toc136209636)

[***4.4.2. Quá trình nhận dạng* 38**](#_Toc136209637)

[***4.4.4. Phương pháp tăng khả năng tổng quát của mạng* 39**](#_Toc136209638)

[***4.4.5. Huấn luyện mạng lan truyền ngược* 41**](#_Toc136209639)

[CHƯƠNG 5: PHƯƠNG PHÁP TESSERACT OCR 42](#_Toc136209640)

[***5.1. Tổng quan về phương pháp* 42**](#_Toc136209641)

[***5.1.1. Tổng quan* 42**](#_Toc136209642)

[***5.1.2. Chức năng của Tesseract* 42**](#_Toc136209643)

[***5.1.3. Kiến trúc giải thuật nhận dạng chữ* 42**](#_Toc136209644)

[***5.2. Code* 45**](#_Toc136209645)

[***5.2.1. Nhận diện bằng file ảnh* 45**](#_Toc136209646)

[***5.2.2. Nhận diện bằng webcam – realtime* 52**](#_Toc136209647)

[TÀI LIỆU THAM KHẢO 53](#_Toc136209648)

**DANH MỤC HÌNH ẢNH**

[**Hình 1. 1:** Quy trình nhận diện biển số 4](#_Toc136181893)

[**Hình 2. 1:** Sơ đồ tổng quát của khối tách biển số 6](#_Toc136184158)

[**Hình 2. 2:** Sơ đồ chi tiết của khối tách biển số 7](#_Toc136184159)

[**Hình 2. 3:** Ảnh tối và ảnh sáng 8](#_Toc136184160)

[**Hình 2. 4:** Ảnh sau khi nhị phân 8](#_Toc136184161)

[**Hình 2. 5:** Sơ đồ thuật giải tìm và tách vùng màu trắng 9](#_Toc136184162)

[**Hình 2. 6 :** Hình dạng biển số khi bị nghiêng 10](#_Toc136184163)

[**Hình 2. 7:** Biển số sau khi nhị phân 10](#_Toc136184164)

[**Hình 2. 8:** Hình thể hiện vùng ảnh cần tách ra với biên an toàn 11](#_Toc136184165)

[**Hình 2. 9:** Ảnh chụp ban đầu 11](#_Toc136184166)

[**Hình 2. 10:** Ảnh sau khi cắt thô 12](#_Toc136184167)

[**Hình 2. 11:** Sơ đồ thuật giải của bước xác định vùng chứa biển số và cắt thô 13](#_Toc136184168)

[**Hình 2. 12:** Phương pháp biến đổi Radon 14](#_Toc136184169)

[**Hình 2. 13:** Hình chiếu đơn giản theo góc *θ* 15](#_Toc136184170)

[**Hình 2. 14:** Biến đổi Radon dưới dạng hình học 15](#_Toc136184171)

[**Hình 2. 15:** Ảnh được tách biên 17](#_Toc136184172)

[**Hình 2. 16:** Anh biển số sau khi xoay về phương thẳng đứng 18](#_Toc136184173)

[**Hình 2. 17:** Thuật giải tìm góc nghiêng biển số 19](#_Toc136184174)

[**Hình 2. 18:** Biển số sau khi cắt hoàn chỉnh 20](#_Toc136184175)

[**Hình 2. 19:** Thuật giải cắt biển số chính xác 21](#_Toc136184176)

[**Hình 3. 1:** Sơ đồ khối phân đoạn ký tự 21](#_Toc136184325)

[**Hình 3. 2:** Ảnh sau khi được nhị phân 21](#_Toc136184326)

[**Hình 3. 3**: Giải thuật nhị phân biển số 22](#_Toc136184327)

[**Hình 3. 4**: Ảnh biển số sau khi được chuẩn hóa 23](#_Toc136184328)

[**Hình 3. 5:** Thuật giải chuẩn hóa biển số 23](#_Toc136184329)

[**Hình 3. 6:** Tổng số các bit theo 1 hàng của biển số 24](#_Toc136184330)

[**Hình 3. 7**: Hình thể hiện thông số Min\_area 25](#_Toc136184331)

[**Hình 3. 8:** Hình thể hiện thông số Digit\_width 25](#_Toc136184332)

[**Hình 3. 9:**Các ký tự được cắt khỏi biển số 25](#_Toc136184333)

[**Hình 3. 10**: Giải thuật phân vùng ký tự 26](#_Toc136184334)

[**Hình 3. 11:** Thuật giải phân vùng từng ký tự 27](#_Toc136184335)

[**Hình 3. 12**: Giải thuật chương trình phân vùng 28](#_Toc136184336)

[**Hình 3. 13:** Thuật giải chương trình con vitricuctieu 29](#_Toc136184337)

[**Hình 4. 1**: Thuật giải nhận dạng ký tự 30](#_Toc136185561)

[**Hình 4. 2**: Mô tả toán học tổng quát của mạng Neural 32](#_Toc136185562)

[**Hình 4. 3**: Neural 1 ngõ vào 32](#_Toc136185563)

[**Hình 4. 4**: Neural nhiều ngõ vào 33](#_Toc136185564)

[**Hình 4. 5:** Cấu tạo 1 neural 34](#_Toc136185565)

[**Hình 4. 6:** Hàm truyền logsig 34](#_Toc136185566)

[**Hình 4. 7**: Hàm truyền tansig 34](#_Toc136185567)

[**Hình 4. 8**: Hàm truyền purelin ( tuyến tính ) 35](#_Toc136185568)

[**Hình 4. 9:** Cấu trúc mạng 1 lớp 35](#_Toc136185569)

[**Hình 4. 10:** Cấu trúc mạng 2 lớp 35](#_Toc136185570)

[**Hình 4. 11**: Cấu trúc mạng dùng để nhận dạng ký tự số 38](#_Toc136185571)

[**Hình 4. 12:** Hình minh họa hoạt động của mạng trong nhận dạng 39](#_Toc136185572)

[**Hình 4. 13**: Trưòng hợp quá khớp 39](#_Toc136185573)

[**Hình 4. 14:** Hàm xấp xỉ khi mạng ngừng học sớm 40](#_Toc136185574)

[**Hình 4. 15:** Hàm lỗi khi ngừng học sớm để huấn luyện mạng 41](#_Toc136185575)

[**Hình 5. 1:** Kiến trúc nhận dạng văn bản chữ in trong Tesseract 43](#_Toc136185581)

[**Hình 5. 2:** Đường cơ sở hình cong 43](#_Toc136185582)

[**Hình 5. 3:** Cắt các ký tự liền nhau 44](#_Toc136185583)

[**Hình 5. 4:** Sơ đồ nhận dạng từ 44](#_Toc136185584)

[**Hình 5. 5:** Các đặc trưng ký tự được nhận dạng 45](#_Toc136185585)

# **LỜI NÓI ĐẦU**

Xin được trình bày phần lời nói đầu của luận án về phương pháp nghiên cứu nhận diện xe.

Bước đầu tiên trong quá trình nghiên cứu và phát triển công nghệ nhận diện xe là thu thập dữ liệu và xử lý. Việc này đòi hỏi sự hợp tác chặt chẽ giữa nhiều bộ phận và tiến hành một cách chính xác để đảm bảo tính toàn vẹn của dữ liệu thu thập được.

Phần chính của luận án này sẽ trình bày về quá trình nghiên cứu phát triển phương pháp nhận diện xe dựa trên các công nghệ hiện đại, bao gồm học máy và thị giác máy tính.

* Trong y học, cải thiện ảnh X – quang và nhận dạng đường biên mạch máu từ ảnh chụp bằng tia X, ứng dụng vào các xét nghiệm lâm sàng như phát hiện và nhận dạng u não, nội soi cắt lớp,...
* Trong thiên văn học, hệ thống chụp hình gắn trên tàu vũ trụ hạn chế về kích thước và trọng lượng, do đó chất lượng hình ảnh nhận được bị giảm chất lượng như mờ, méo hình học, nhiễu nền. Các hình ảnh đó được xử lý bằng máy tính.
* Trong các lĩnh vực công nghiệp, người máy càng đóng vai trò quan trọng. Chúng thực hiện các công việc nguy hiểm, đòi hỏi có tốc độ và độ chính xác cao vượt quá khả năng của con người. Người máy sẽ trở nên tinh vi hơn và thị giác máy tính đóng vai trò quan trọng hơn. Người ta sẽ không chỉ đòi hỏi người máy phát hiện và nhận dạng các bộ phận công nghiệp mà còn phải “hiểu” được những gì chúng “thấy” và đưa ra hành động phù hợp. Xử lý ảnh sẽ tác động đến thị giác của máy tính.
* Ngoài ra, xử lý và nhận dạng còn được ứng dụng trong lĩnh vực ít được nói đến hơn. Công an giao thông thường hay chụp ảnh trong môi trường không thuận lợi, ảnh thường bị nhoè nên cần được xử lý và nhận dạng để có thể nhìn thấy biển số xe.

Bên cạnh đó, chúng em cũng sử dụng các phương pháp kinh điển, như phân tích đặc trưng và nhận dạng mẫu, để tạo ra một phiên bản nhận diện xe chính xác và đáng tin cậy hơn.

Kết quả của luận án này sẽ giúp cải thiện hiệu quả của các hệ thống nhận diện xe, đảm bảo an toàn giao thông và nâng cao đời sống của những người tham gia giao thông.

Xin chân thành cảm ơn Ban giám khảo và các đồng nghiệp đã ủng hộ và góp ý cho luận án của chúng em. Mong rằng luận án sẽ có giá trị thực tiễn và đóng góp tích cực đối với ngành công nghiệp ô tô và giao thông vận tải.

# **CHƯƠNG 1: GIỚI THIỆU**

* 1. ***Giới thiệu về đề tài***

Trong khuôn khổ của luận án này, nhóm em sẽ tập trung vào một chủ đề cụ thể là phương pháp nhận diện xe, đặc biệt là nhận diện biển số xe. Đây là một trong những lĩnh vực đang nhận được sự quan tâm đặc biệt của các nhà quản lý giao thông và những người đang nghiên cứu về các công nghệ và phương pháp để nâng cao hiệu quả quản lý và kiểm soát giao thông.Trong thời đại của cuộc cách mạng công nghiệp 4.0, công nghệ ngày càng được ứng dụng rộng rãi trong các lĩnh vực đời sống, đặc biệt là lĩnh vực giao thông. Trong dòng chảy phát triển của xã hội hiện đại, nhu cầu vận chuyển hàng hóa và người dân tăng cao, dẫn tới ngập tràn xe cộ trên đường phố, đặc biệt là ở các thành phố lớn. Vì vậy, nhu cầu quản lý và kiểm soát giao thông đang trở thành một vấn đề cấp bách.

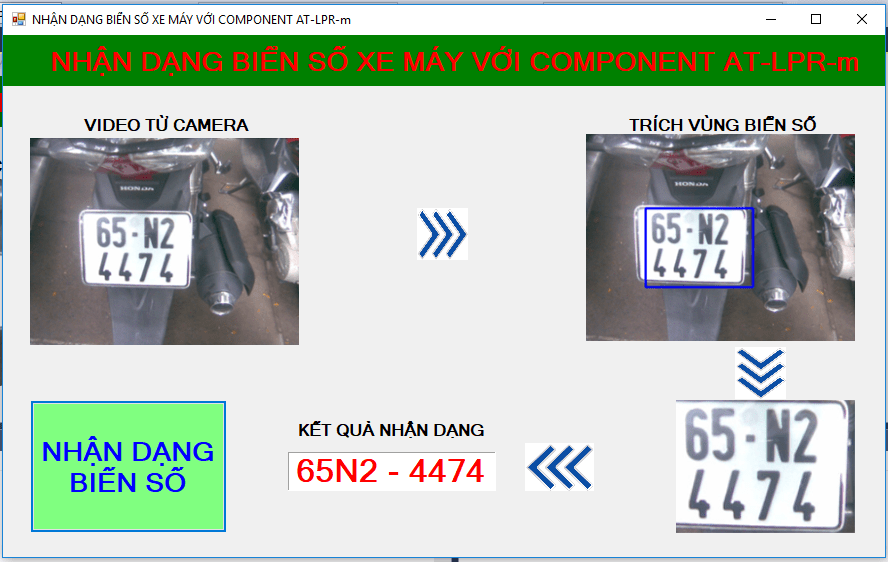
* 1. ***Lý do chọn đề tài***

Lý do nhóm em chọn đề tài này là vì bản thân chúng em cũng là một người sử dụng phương tiện giao thông trong đời sống hàng ngày. Nhóm em đã từng bị thiệt hại về tài sản và an ninh khi bị mất cắp xe, và chính vì thế tụi em hiểu rõ vấn đề và nguy cơ mất an toàn giao thông. Bên cạnh đó, vài năm gần đây, bọn em đã dành nhiều thời gian nghiên cứu các công nghệ mới như học sâu, các mô hình nhận diện và các thuật toán xử lý ảnh. Do đó, đề tài nghiên cứu phương pháp nhận diện xe (nhận diện biển số xe) đã được nhóm em lựa chọn vì nó phù hợp với năng lực và hứng thú của bọn em.

* 1. ***Nội dung đề tài***

Trong tài liệu này, chúng ta sẽ tìm hiểu về nguyên lý của nhận dạng biển số xe xem nó sẽ gồm các bước nào, chi tiết thực hiện ra sao từ trong lý thuyết và các áp dụng vào trong code.

Quá trình nhận dạng biển số sẽ sẽ diễn ra cơ bản như sau:



**Hình 1. 1:** Quy trình nhận diện biển số

* **Tách biển số:** khối này có chức năng tách biển số từ ảnh chụp bằng các phương pháp xử lý ảnh. Kết quả của khối là ảnh màu RGB (Red Green Blue) được cắt ra từ ảnh chụp. Đây là một công việc rất khó khăn vì ta không biết được vị trí chính xác của biển số. Ngoài ra, còn phụ thuộc vào độ sáng của ảnh chụp.
* **Phân đoạn ký tự:** sau khi tách biển số, chúng ta bắt đầu phân đoạn ký tự. Khối này thực hiện tách từng ký tự có trong biển số, tạo thành tập ảnh riêng biệt các ký tự phục vụ việc nhận dạng ký tự. Ảnh của mỗi ký tự là ảnh trắng đen.
* **Nhận dạng ký tự:** sau khi phân đoạn, tách được các ký tự trong biển số và tạo thành một chuỗi ký tự. Chuỗi này đưa vào khối nhận dạng để tiến hành nhận dạng từng ký tự trong chuỗi.
  1. ***Mục tiêu và phạm vi đề tài***

Mục tiêu của luận án này là đề xuất một phương pháp nhận diện xe (nhận diện biển số xe) hiệu quả, chính xác và đáng tin cậy hơn. Từ đó cải thiện hiệu quả quản lý và kiểm soát giao thông, đảm bảo an toàn cho người dân và hàng hóa.

Phạm vi của đề tài sẽ tập trung vào việc phân tích, thiết kế, xây dựng và thực hiện một hệ thống nhận diện xe chuyên sâu dựa trên các kỹ thuật và công nghệ mới nhất. Đồng thời, phương pháp đề xuất cũng sẽ được kiểm tra và đánh giá một cách cẩn thận về hiệu suất và độ chính xác để đảm bảo sự tin cậy của nó trong thực tế. Ngoài ra, luận án cũng sẽ tập trung vào việc nghiên cứu các bài toán trong nhận diện xe và nhận diện biển số, và đề xuất các giải pháp để cải thiện hiệu quả của các hệ thống nhận diện xe trong thực tế.

* 1. ***Tầm quan trọng của đề tài***

Đề tài nghiên cứu phương pháp nhận diện xe (nhận diện biển số xe) có tầm quan trọng rất lớn trong thực tiễn hiện nay. Việc áp dụng các phương pháp nhận diện xe giúp các cơ quan chức năng trong việc quản lý và kiểm soát giao thông một cách thông minh, chính xác và đáng tin cậy. Dưới đây là một số lý do cho tầm quan trọng của đề tài này.

1. Giảm thiểu tối đa những rủi ro về an toàn giao thông: Nếu việc nhận diện xe được thực hiện một cách nhanh chóng, đáng tin cậy và chính xác, các cơ quan chức năng có thể giảm thiểu tối đa các rủi ro về an toàn giao thông, từ đó cải thiện đời sống và tăng cường an ninh cho người dân.
2. Nâng cao hiệu quả quản lý và kiểm soát giao thông: Với sự hỗ trợ của các công nghệ như nhận diện xe, các cơ quan chức năng có thể thực hiện việc quản lý và kiểm soát giao thông một cách thông minh và hiệu quả hơn, từ đó giảm thiểu tắc nghẽn giao thông, tăng cường an ninh và đảm bảo lợi ích cho người dân.
3. Đáp ứng nhu cầu quản lý và kiểm soát giao thông: Trong bối cảnh Việt Nam đang phát triển nhanh chóng, nhu cầu vận chuyển hàng hóa và người dân tăng cao, đảm bảo an ninh và quản lý giao thông trở thành một nhu cầu cấp bách. Do đó, việc nghiên cứu và áp dụng phương pháp nhận diện xe là rất cần thiết để đáp ứng nhu cầu này.
4. Đóng góp tích cực vào sự phát triển của ngành công nghiệp ô tô: Việc áp dụng các công nghệ mới như nhận diện xe đóng góp tích cực vào sự phát triển của ngành công nghiệp ô tô. Các công nghệ này có thể được ứng dụng trong việc sản xuất và phát triển các sản phẩm liên quan đến ô tô và xe cộ, từ đó tạo ra cơ hội mới cho việc phát triển ngành công nghiệp nước nhà.
5. Đưa Việt Nam vào cuộc chơi công nghệ toàn cầu: Nghiên cứu và áp dụng phương pháp nhận diện xe giúp đưa Việt Nam vào cuộc chơi công nghệ toàn cầu. Việc ứng dụng các công nghệ mới như học sâu, các mô hình nhận diện và các thuật toán xử lý ảnh sẽ giúp các chuyên gia và nghiên cứu viên Việt Nam tiếp cận với các công nghệ tiên tiến nhất trong lĩnh vực này, từ đó phát triển thêm năng lực và cơ hội kinh doanh toàn cầu.

Tóm lại, việc nghiên cứu và áp dụng phương pháp nhận diện xe (nhận diện biển số xe) là rất cần thiết trong việc tăng cường an ninh giao thông và quản lý giao thông trong các thành phố hiện đại. Chúng đóng góp tích cực cho sự phát triển của ngành công nghiệp ô tô và gi

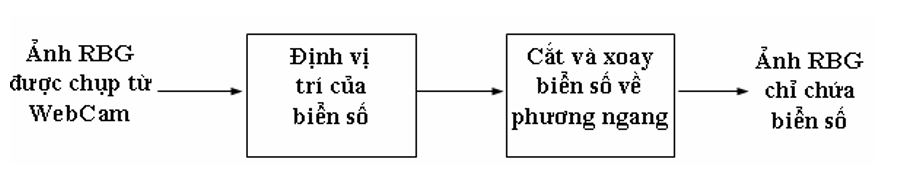
# **CHƯƠNG 2: TÁCH BIỂN SỐ**

## **Tổng quan về tách biển số**

Tách biển số là một bước rất quan trọng trong quá trình nhận dạng biển số xe.

Khối tách biển số xe được chia làm 2 giai đoạn chính:

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

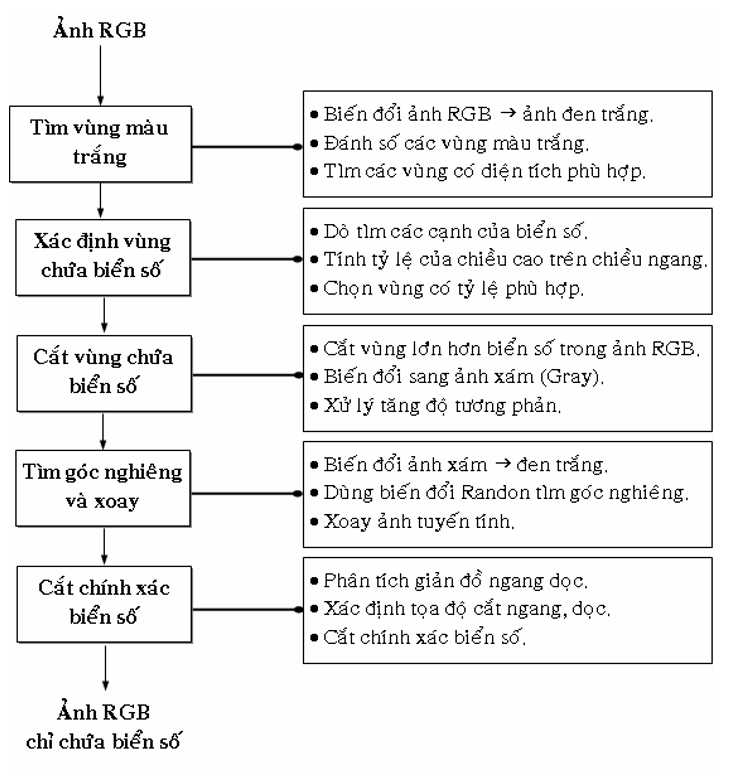


**Hình 2. 1:** Sơ đồ tổng quát của khối tách biển số

Việc định vị biển số xe dựa vào các tính chất của biển số như: hình dạng, màu sắc của biển số so với nền. Biển số xe có hình chữ nhật với kích thước chiều dài và rộng không thay đổi. Từ đó chúng ta có thể tìm các vùng có hình tứ giác trong hình, tính tỉ lệ giữa hai chiều của hình so với khoảng giá trị cho trước để xem thử vùng đó, có chứa biển số hay không. Ngoài ra, định vị biển số còn dựa trên màu sắc của biển số. Hầu hết các biển số xe Việt Nam đều nền trắng, chữ đen. Kết hợp hai tính chất trên, chúng ta xác định được vùng chứa biển số.

Sau khi định vị biển số xe, chúng ta tiến hành cắt biển số xe. Biển số xe được cắt theo 2 bước. Bước đầu tiên là cắt vùng rộng hơn vùng chứa biển số. Sau đó tìm góc nghiêng của biển số và thực hiện xoay biển số về phương thẳng đứng. Bước hai là cắt biển số ra khỏi vùng trên. Thực hiện việc cắt biển số qua hai bước như trên làm tăng độ chính xác, biển số được cắt nguyên vẹn, không cắt phạm chữ, trừ trường hợp ảnh bị chói, độ tương phản không đều hoặc bị che khuất thì kết quả của việc tách biển số mới không chính xác.

Phần tách biển số sử dụng rất nhiều giải thuật và phương pháp xử lý đối với ảnh số như sơ đồ sau đây:



**Hình 2. 2:** Sơ đồ chi tiết của khối tách biển số

* 1. **Tìm vùng màu trắng**

Vì biển số màu trắng nên người thực hiện sẽ nhị phân hóa ảnh bằng cách gắn giá trị cho các pixel trắng là 1, còn ngược lại là 0. Đầu tiên, chúng ta sẽ biến đổi nh gốc thành ảnh xám (có mức sáng từ 0 đến 255), sau đó nhị phân hóa với một ngưỡng thích hợp. Nếu ảnh được chụp vào ban đêm hay ban ngày nhưng ít ánh sáng thì mức ngưỡng sẽ là 120. Còn ban ngày, nhiều ánh sáng là 190.

Vấn đề ở đây là làm sao chúng ta nhận biết được đó là ảnh sáng hay ảnh tối? Người thực hiện đã dựa vào lược đồ mức xám (histogram – tần số xuất hiện của mức xám) của ảnh [8]. Nếu tần số xuất hiện các pixel có giá trị <64 thì đó là ảnh tối, ngược lại là ảnh sáng.

Ví dụ cho hai ảnh gốc có biển số như sau:

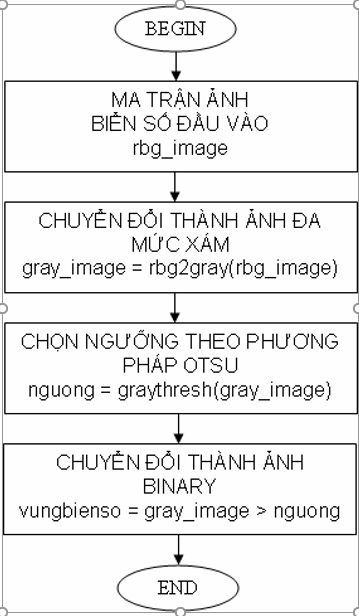


**Hình 2. 3:** Ảnh tối và ảnh sáng



**Hình 2. 4:** Ảnh sau khi nhị phân

Đây là sơ đồ thuật giải của bước tìm và tách vùng màu trắng:



**Hình 2. 5:** Sơ đồ thuật giải tìm và tách vùng màu trắng

## **Xác định vùng chứa biển số**

Ảnh nhận được sau khi nhị phân hóa sẽ có nhiều vùng màu trắng, và biển số sẽ nằm trong vùng màu trắng thỏa điều kiện:

* 0.75 < chiều cao / chiều ngang < 0.91
* 16000 ≤ S\_pixel\_trang ≤ 61000
* ≤ S\_pixel\_trang / S\_biển số ≤ 0.7

Với:

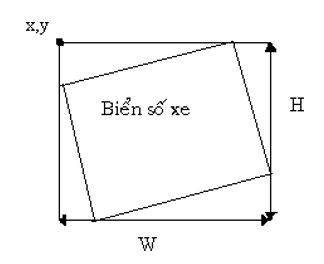
Chiều cao là H {H = start(i).Boundingbox(4)}

Chiều ngang là W {W = start(i).Boundingbox(3)}

Diện tích vùng trắng là S\_pixel\_trang { S\_pixel\_trang = start(i).Area }

Diện tích biển số là S\_biển số { S\_biển số = W × H }

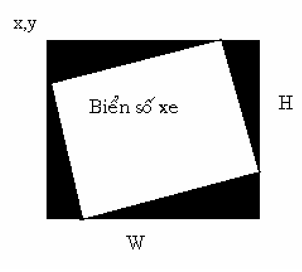
Tỉ số 2 kích thước của biển số là 0.75 nhưng khi bị nghiêng thì là 0.91.



**Hình 2. 6 :** Hình dạng biển số khi bị nghiêng

Sau điều kiện thứ nhất, sẽ có những vùng trắng không chứa biển số nhưng có tỉ số giữa chiều cao và chiều ngang phù hợp thì vẫn được chọn. Ta sẽ loại bỏ những vùng này bằng điều kiện thứ hai – diện tích vùng trắng. Vì khoảng cách từ webcam tới xe cố định (ta lấy trước khoảng cách chụp) nên số lượng pixel trắng chứa trong biển số cố định.

Đối với điều kiện thứ ba, nếu biển số nằm ngay ngắn thì tỉ số là 0.7 còn nếu biển số bị nghiêng, sẽ xuất hiện pixel đen nên tỉ số này giảm còn 0.38.

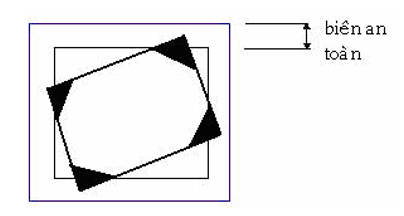


**Hình 2. 7:** Biển số sau khi nhị phân

## **Cắt vùng chứa biển số**

Sau khi xác định được vùng nào chứa biển số, ta tiến hành cắt biển số trên ảnh màu RBG. Do sự tương đồng của ảnh nhị phân và ảnh màu RGB nên toạ độ một pixel trên ảnh nhị phân tương ứng tọa độ trên ảnh màu RGB. Do đó khi xác định được tọa độ trên ảnh nhị phân, ta dùng tọa độ này để cắt trên ảnh màu RGB.

Khi ta cắt biển số ra khỏi ảnh mà biển số bị nghiêng thì ta tiến hành cắt theo hình chữ nhật lớn hơn hình chữ nhật lớn thực sự một lượng biên an toàn để tránh việc mất thông tin biển số.



**Hình 2. 8:** Hình thể hiện vùng ảnh cần tách ra với biên an toàn

**Trong chương trình, người thực hiện chọn vùng biên an toàn là 30 pixel. Trong bước này, ta không cắt biển số ra khỏi ảnh ngay, mà chỉ cắt vùng chứa biển số. Việc cắt này có thể xem như là cắt thô.**



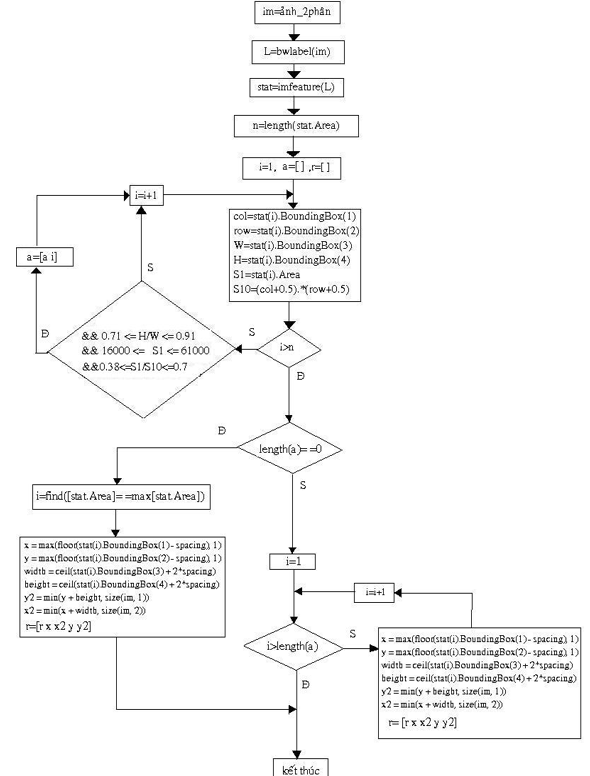
**Hình 2. 9:** Ảnh chụp ban đầu



**Hình 2. 10:** Ảnh sau khi cắt thô

Ở đây, chúng ta có thể thấy các điều kiện trên chưa chặt chẽ, vì vậy ta lưu tất cả các thông số “x, y, W, H” của vùng chứa biển số vào biến “r”.

Và sau đây là thuật giải của bước xác định vùng chứa biển số và cắt thô:



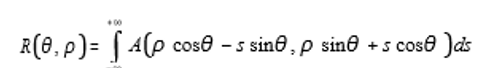
**Hình 2. 11:** Sơ đồ thuật giải của bước xác định vùng chứa biển số và cắt thô

* 1. **Tìm góc nghiêng và góc xoay**

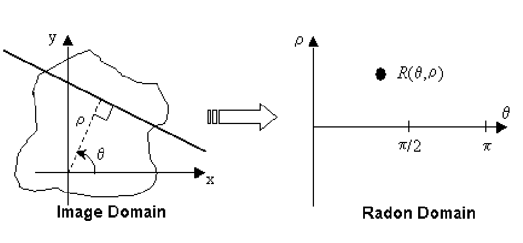
1. 5. 1. **Biến đổi Radon**
         1. ***Tổng quan về biến đổi Randon***

Dùng để biến đổi các ảnh trong không gian 2 chiều với các đường thẳng thành miền Radon, trong đó mỗi đường thẳng trong ảnh sẽ cho 1 điểm trong miền Radon.

Công thức toán học của biến đổi Radon:

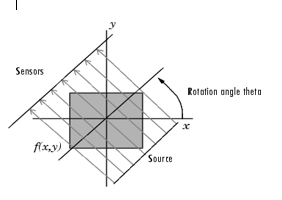


Phương trình trên biểu diễn việc lấy tích phân dọc theo đường thẳng s trên ảnh, trong đó ρ là khoảng cách của đường thẳng so với gốc tọa độ O, và θ là góc lệch so với phương ngang.



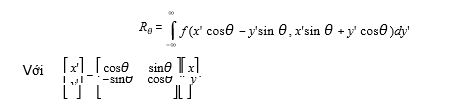
**Hình 2. 12:** Phương pháp biến đổi Radon

Trong xử lý ảnh số, biến đổi Radon tính toán hình chiếu của ma trận ảnh dọc theo 1 hướng xác định. Hình chiếu của 1 hàm số 2 chiều là f (x, y) là tập hợp các tích phân đường. Hàm Radon tính toán tích phân đường dọc theo các tia song song theo các phương khác nhau (bằng cách xoay hệ trục tọa độ xung quanh O theo các giá trị θ khác nhau), chiều rộng của các tia là 1 pixel. Hình dưới đây biểu diễn 1 hình chiếu đơn giản theo 1 giá trị của góc θ.

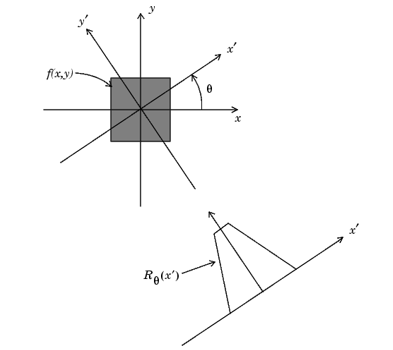


**Hình 2. 13:** Hình chiếu đơn giản theo góc *θ*

Công thức tổng quát trên có thể viết lại như sau:

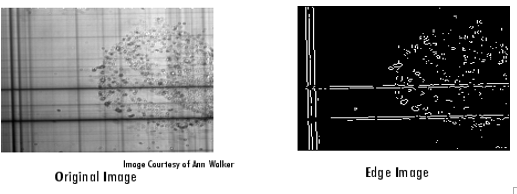


Hình sau sẽ biểu diễn phương pháp biến đổi Radon dưới dạng hình học:

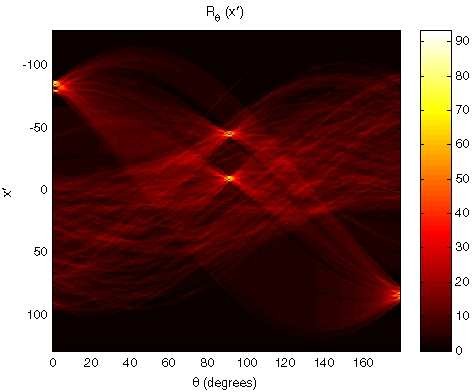


**Hình 2. 14:** Biến đổi Radon dưới dạng hình học

* + - 1. ***Các bước thực hiện***
  1. ***Biến đổi về ảnh nhị phân***

****

* 1. ***Thực hiện biến đổi randon trên ảnh biên với*** [***θ = 0:179***](https://docs.google.com/document/d/1FPQ2qc0iM268i3RW0eEI27iVx9VP4IUo/edit#heading=h.3j2qqm3)



* 1. ***Tìm giá trị lớn nhất của R trong biến đổi Radon***

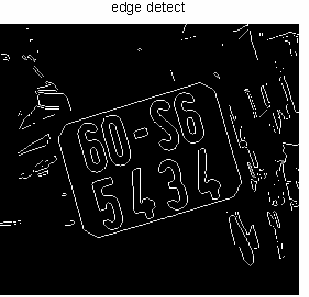
Vị trí các giá trị lớn nhất này tương ứng với các giá trị của đường thẳng trong

ảnh ban đầu.

* + 1. ***Tìm góc nghiêng và xoay***

Biển số được chụp với nhiều góc nghiêng khác nhau, do đó ta phải tìm góc nghiêng và xoay về phương thẳng. Đây là việc rất quan trọng, vì nếu không quay về phương thẳng thì khi cắt biển số sẽ bị phạm vào biển số. Chúng ta xác định góc nghiêng bằng phương pháp biến đổi Radon.

Trước khi biến đổi Radon, ảnh chứa biển số được biến đổi thành ảnh được tách biên nhị phân [1],[2].



**Hình 2. 15:** Ảnh được tách biên

Sau đó, ta tiến hành biến đổi Radon để tìm góc xoay. Thực hiện biến đổi Radon với góc θ chạy trong khoảng (0: 180), ta sẽ được một ma trận với các điểm R(θ) với từng góc θ và tọa độ pixel tương ứng.

Sau khi biến đổi Radon, chúng ta xác định được góc Rmax, ứng với Rmax thì ta có được θ max và góc lệch là (90o - θ max). Sau đó ta sử dụng hàm Rotate trong MATLAB để xoay ảnh với góc lệch tìm được.



**Hình 2. 16:** Anh biển số sau khi xoay về phương thẳng đứng

Và đây là thuật giải của bước này:



**Hình 2. 17:** Thuật giải tìm góc nghiêng biển số

* 1. ***. Cắt biến số chính xác***

Sau khi xoay biển số về phương thẳng đứng ta thực hiện việc cắt biển số. Đây là một việc rất quan trọng, kết quả của nó quyết định tới kết quả của hệ thống nhận dạng biển số.

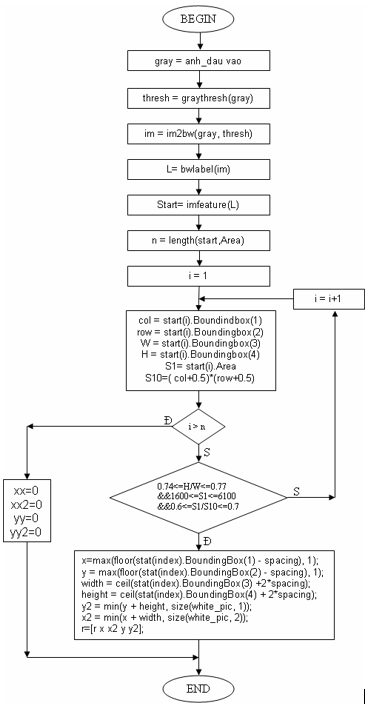
Ta thực hiện lại việc chọn vùng biển số nhưng với biên là 0 pixel và 3 điều kiện chính xác hơn:

* + - 1600 < diện tích vùng trang < 6100
    - 0.73 < width/height < 0.77
    - 0.6 < số pixel trang / số pixel đen < 0.7

****

**Hình 2. 18:** Biển số sau khi cắt hoàn chỉnh

Dưới đây là thuật giải :



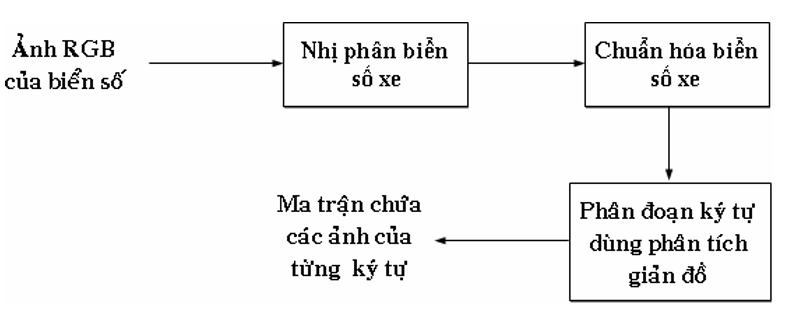
**Hình 2. 19:** Thuật giải cắt biển số chính xác

# **CHƯƠNG 3: PHÂN ĐOẠN KÝ TỰ**

## ***3.1. Tổng quan về phân đoạn ký tự***

Kết quả của khối tách biển số là một ảnh màu RGB có chứa biển số xe. Để nhận dạng các ký tự trong biển số, ta tiến hành phân đoạn ký tự trong biển số. Phân đoạn ký tự là việc cắt các ký tự trong biển số xe.

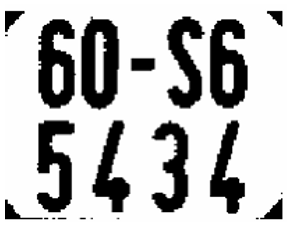
Sau khi nhận kết quả của khối tách biển số, khối phân đoạn ký tự bắt đầu tiến hành tách từng ký tự trong biển số. Trước khi phân đoạn ký tự, ảnh của biển số được chuyển thành ảnh nhị phân. Ảnh nhị phân được chuẩn hóa về kích chuẩn, sau đó tiến hành cắt các ký tự. Kết quả của quá trình phân đoạn là một ma trận chứa các ảnh đen trắng của ký tự.



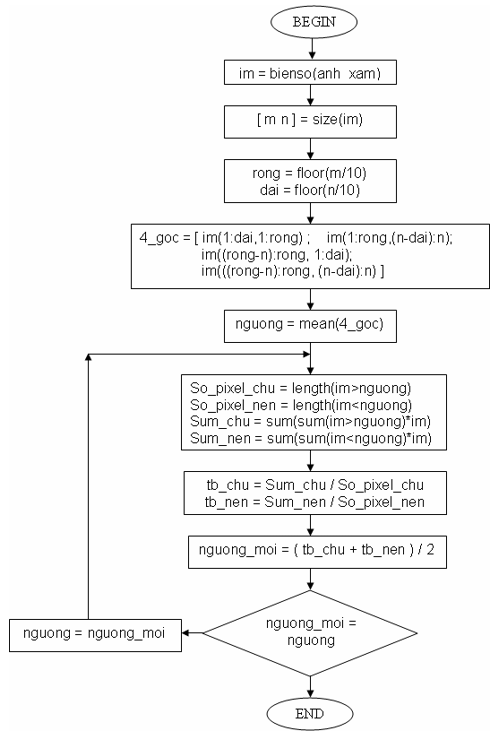
**Hình 3. 1:** Sơ đồ khối phân đoạn ký tự

* 1. ***Nhị phân biển số xe***

Đây là bước quan trọng để nhận dạng biển số xe. Bước này sẽ tìm mức ngưỡng tối ưu, sau đó tiến hành nhị phân hóa ảnh với ngưỡng vừa tìm được (nhằm làm tăng độ tương phản của ký tự với nền biển số).



**Hình 3. 2:** Ảnh sau khi được nhị phân

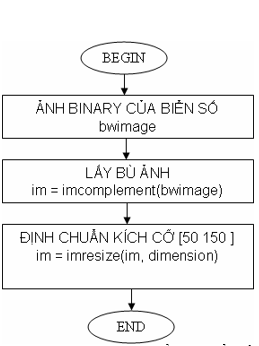


**Hình 3. 3**: Giải thuật nhị phân biển số

* 1. ***Chuẩn hóa biển số***

Biển số được chuẩn hóa về kích thước [50 150], sau đó được lấy bù. 

**Hình 3. 4**: Ảnh biển số sau khi được chuẩn hóa

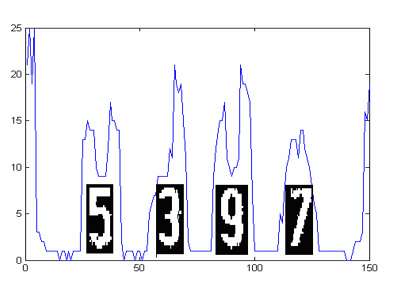


**Hình 3. 5:** Thuật giải chuẩn hóa biển số

## ***3.4. Phân đoạn ký tự***

Ma trận binary của biển số chính là ngõ vào của chương trình phân vùng ký tự. Trước khi phân vùng ký tự, ta chia ma trận ảnh biển số thành từng hàng và lần lượt đưa từng hàng vào chương trình phân vùng. Tuy nhiên, ở đây, người thực hiện chỉ tiến hành nhận dạng ký tự của hàng 2 nên ta chỉ ngõ vào của chương trình phân vùng ký tự là ma trận của hàng 2.

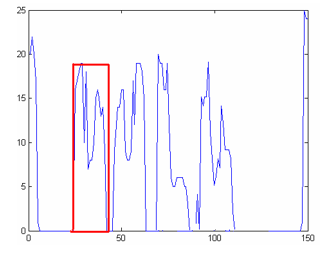
Để phân chia thành nhiều ma trận ký tự từ ma trận biển số, ta dựa vào tổng số pixel mức 1 ( mức 1 là màu trắng- màu của ký tự, mức 0 là màu đen – màu của nền).Với ma trận của hàng 2 sau khi đã chia đôi, giữa 2 ký tự có rất ít pixel có mức 1 (trong trường hợp lý tương, thì sẽ là 0 ). Như vậy khi cộng giá trị các pixel theo từng cột, như hình sau, ta thấy giá trị tại các vùng giữa 2 ký tự rất thấp ( đây cũng là tổng số pixel mức 1). Từ đó, giải thuật phân vùng sẽ những vùng này dựa vào giá trị của nó nhỏ hơn những vùng lân cận và sẽ phân chia thành từng vùng. Ở đây, ta sẽ tìm 4 phân vùng tương ứng với 4 ký tự.



**Hình 3. 6:** Tổng số các bit theo 1 hàng của biển số

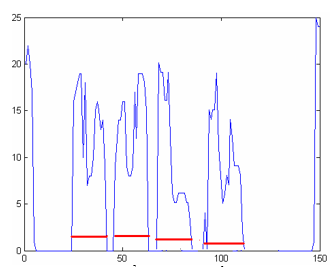
Chương trình có lựa chọn 2 thông số: Min\_area và digit\_width.

Min Area là diện tích cho phép nhỏ nhất của 1 ký tự, là tích của giá trị cột lớn nhất với độ rộng của phân vùng đó.



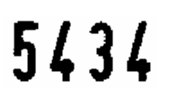
**Hình 3. 7**: Hình thể hiện thông số Min\_area

Digit\_width là độ rộng tối đa cho phép của 1 phân vùng ký tự.



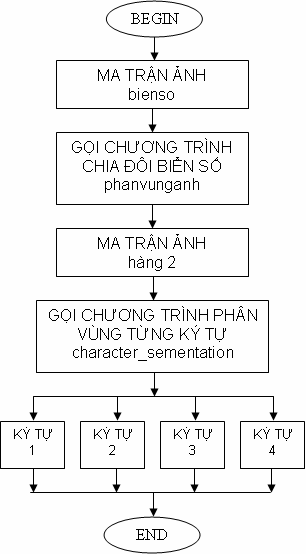
**Hình 3. 8:** Hình thể hiện thông số Digit\_width

Sau khi phân vùng được các ký tự ta tiến hành cắt các ký tự ra khỏi biển số.



**Hình 3. 9:**Các ký tự được cắt khỏi biển số

Dưới đây là sơ đồ thuật giải:



**Hình 3. 10**: Giải thuật phân vùng ký tự

### *3.4.1. Chương trình chia đôi biển số*

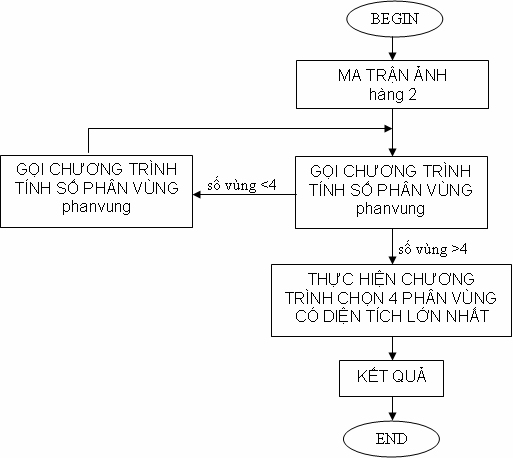
Đây là chương trình đơn giản, biển số sau khi được chuẩn hóa về kích thước [50 150] thì kích thước từng hàng sau khi chia đôi là [25 150].

### *3.4.2. Chương trình phân vùng kí tự*

Chương trình con này thực hiện việc tìm vùng ranh giới giữa 2 ký tự, tương ứng giữa 2 vùng ranh giới liên tiếp sẽ là vùng ký tự, chương trình sẽ trả về vị trí của vùng ký tự. Tuy nhiên, chương trình cũng có thể trả về số vùng ký tự lớn hơn 4 do ảnh hưởng của môi trường. Trường hợp này ta vẫn có thể tối ưu chương trình để tăng độ chính xác.

Với kết quả trả về nhỏ hơn 4, ta sẽ xóa các pixel tai các vùng ranh giới (gắn các giá trị đó bằng 0). Điều này làm cho việc thực hiện chương trình chính xác hơn.

Với kết quả trả về lớn hơn 4 (nghĩa là có 1 hay nhiều vùng không chứa ký tự), ta sẽ lấy 4 vùng có diện tích lớn nhất (vì thông thường, các vùng không có ký tự nhỏ hơn các vùng có ký tự).



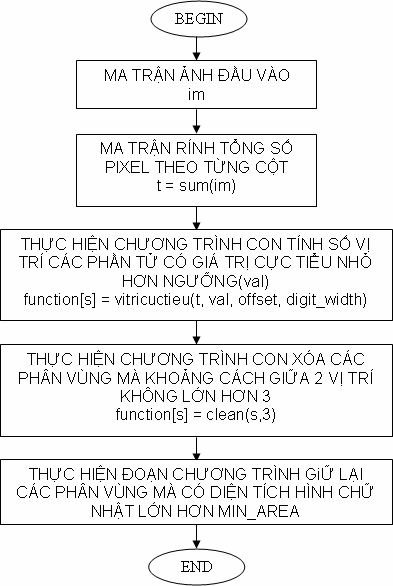
**Hình 3. 11:** Thuật giải phân vùng từng ký tự

#### **3.4.2.1. Chương trình chọn 4 phân vùng lớn nhất**

Trước tiên, Chương trình sẽ sắp xếp các phân vùng theo thứ tự diện tích từ nhỏ đến lớn, sau đó sẽ giữ lại 4 phân vùng có diện tích lớn nhất nhưng vẫn giữ nguyên vị trí của chúng trước khi sắp xếp.

#### **3.4.2.2. Chương trình phân vùng**

Chương trình này sẽ tìm ra vị trí ranh giới giữa các ký tự và trả về số phân vùng tương ứng với số ký tự.



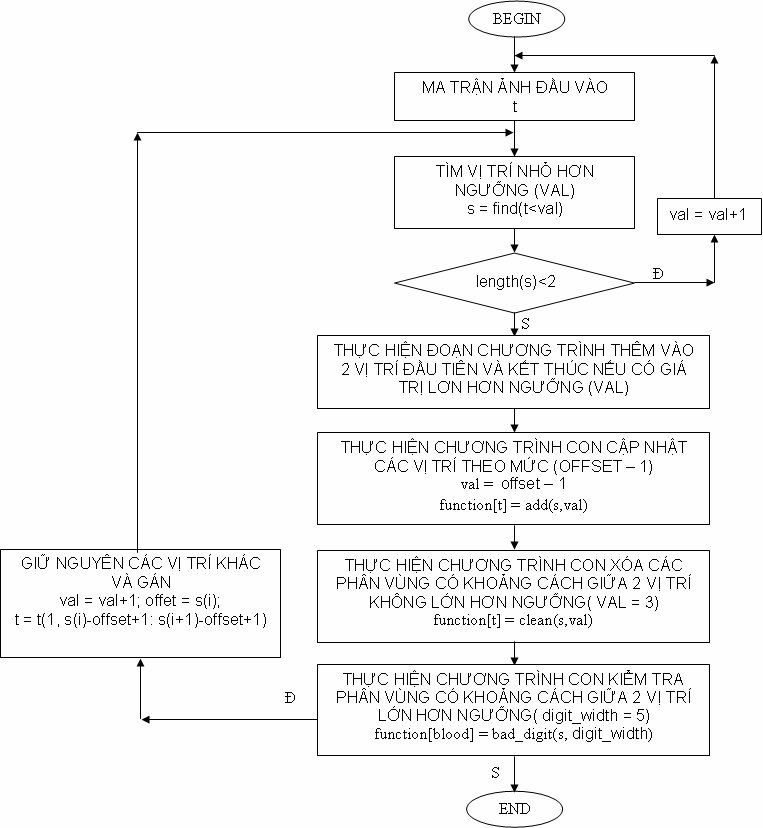
**Hình 3. 12**: Giải thuật chương trình phân vùng

Chương trình con vitricuctieu: tìm vị trí các cột trong ma trận đầu vào mà có giá trị ngưỡng nhỏ hơn giá trị ta đặt.

Chương trình con clean: xóa những vị trí ( đã tìm được trong Chương trình con vitricuctieu ) mà khoảng cách tới vị trí kế tiếp quá nhỏ.

#### **3.4.2.3. Chương trình con vitricuctieu**

Đây là chương trình con quan trọng nhất trong chương trình phân vùng. Chương trình quét toàn bộ ma trận ảnh đầu vào, tìm vị trí “khả nghi”, xóa các vị trí mà khoảng cách với vị trí kế tiếp khá lớn ( lớn hơn khoảng cách cho phép ) thì sẽ thực hiện lại chương trình này trong khoảng 2 vị trí đó để tiếp tục phân thành nhiều vùng nhỏ hơn.



**Hình 3. 13:** Thuật giải chương trình con vitricuctieu

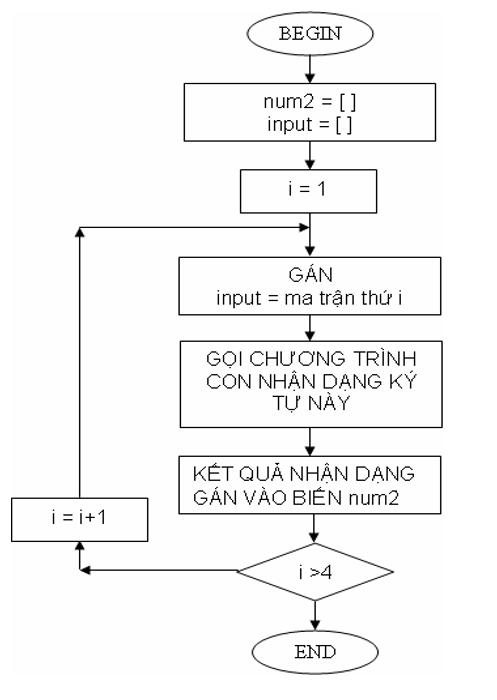
* *Tìm* *vị trí* *nhỏ hơn* *ngưỡng*: tìm các vị trí trong ma trận đầu vào có giá trị nhỏ hơn ngưỡng. Nếu không tìm được vị trí nào hoặc chỉ tìm được 1 vị trí thì ta sẽ tăng giá trị ngưỡng lên 1.
* *Đoạn* *chương* *trình* *thêm* *vào* *2* *vị trí* *đầu* *tiên* *và* *kết* *thúc* *nếu* *có* *giá* *trị lớn hơn ngưỡng*: những vị trí tìm được có thể có vị trí hai biên của ma trận đầu vào hoặc là không. Tuy nhiên, nếu biên vẫn có giá trị lớn hơn ngưỡng thì ta phải thêm vị trí này.

# **CHƯƠNG 4: NHẬN DẠNG KÝ TỰ**

## **4.1. *Tổng quát nhận dạng ký tự***

Sau khi thực hiện phân vùng ta sẽ được 4 ma trận tương ứng với 4 ký tự trên 1 hàng biển số. Lần lượt từng ma trận ký tự sẽ được đưa vào chương trình nhận dạng. Kết quả cuối cùng sẽ là 4 ký tự số 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 có chính là ký tự cần tìm.



**Hình 4. 1**: Thuật giải nhận dạng ký tự

Trong lĩnh vực nhân dạng, có 2 phương pháp để nhận dạng là phương pháp cổ điển và phương pháp sử dụng mang neural.

## ***4.2. Lựa chọn phương pháp***

### *4.2.1. Phương pháp nhận dạng cổ điển*

**Giới thiệu phương pháp**:

Phương pháp này sẽ có 1 tập ma trận ký tự mẫu. Phương pháp này khá đơn giản: ma trận ký tự cần nhận dạng khá giống với ma trận ký tự đó trong tập mẫu. Ví dụ, ta nhận dạng các số từ 0 đến 9 thì trong tập mẫu, ta sẽ tạo ra các ma trận ký tự từ 0 đến 9. Giả sử ma trận cần nhận dạng là số 1 thì ma trận này nhìn bằng mặt thấy cũng khá giống sao với ma trận số 1 trong tập mẫu.

**Ưu và nhược điểm:**

Phương pháp này tuy đơn giản nhưng hiệu quả trong trường hợp tập ảnh nhận dạng rõ nét, ít bị nhiễu. Nếu ảnh bị nhiễu thì ma trận đầu vào thay đổi, lúc này tập mẫu sẽ không còn chính xác. Trong 1 vài trường hợp biển số ban đầu bị nghiêng hoặc lệch, khi ta xoay và chuẩn hóa kích thước thì các ký tự bị nhiễu và khi so sánh với tập mẫu cũng không còn chính xác.

### *4.2.2. Phương pháp ứng dụng mạng neural*

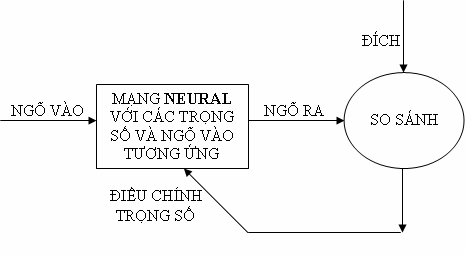
Cũng như phương pháp cổ điển, ta nhận dạng các ký tự nhờ vào vị trí và hàng của ký tự đó trên biển số, nhưng ở đây ta sử dụng nhiều tập mẫu để so sánh qua đó làm tăng độ chính xác khi nhận dạng. Hơn nữa trong MATLAB, có TOOLBOX hỗ trợ về phần neural làm cho việc nhận dạng trở nên dễ dàng hơn.

Vì lý do trên mà khi tiến hành nhận dạng ký tự, người thực hiện đã chọn phương pháp nhận dạng dùng mạng neural [3], [4].

## ***4.3. Giới thiệu về mạng neural (neural networks)***

### *4.3.1. Khái niệm*

Neural networks phát triển từ các nghiên cứu về trí tuệ nhân tạo, dựa trên việc mô phỏng cấp thấp hệ thống Neural sinh học – cố gắng bắt chước khả năng học và chấp nhận sai của bộ não có cấu trúc thấp.

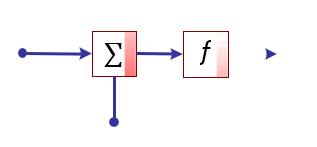


**Hình 4. 2**: Mô tả toán học tổng quát của mạng Neural

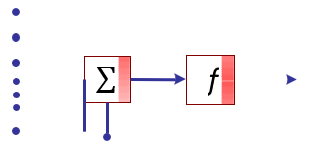
Tín hiệu ngõ vào sau khi qua mạng neural sẽ được tính toán và ngõ ra của mạng sẽ được so sánh với tín hiệu đích mong muốn. Mạng sẽ tiếp tục cập nhật và điều chỉnh trọng số và ngưỡng cho đến khi thỏa mãn ngõ ra yêu cầu.

### *4.3.2. Mô hình của một mạng neural nhân tạo*

Neural nhân tạo có 1 số ngõ vào ( từ các dữ liệu gốc hay từ ngõ ra của các neural khác). Mỗi kết nối đến ngõ vào đều có 1 cường độ ( hay trọng số ). Ngõ vào của neural có thể vô hướng hay hữu hướng. Mỗi neural có 1 giá trị ngưỡng. Tín hiệu được truyền qua hàm kích hoạt ( hay là hàm truyền) tạo giá trị ngõ ra của neural.



**Hình 4. 3**: Neural 1 ngõ vào



**Hình 4. 4**: Neural nhiều ngõ vào

### *4.3.3. Thiết kế 1 mạng Neural*

#### **4.3.3.1. Thu thập dữ liệu**

Trước hết ta cần xác định được tập dữ liệu – là tập bao gồm 1 số trường hợp, mỗi tập hợp chứa giá trị của ngõ vào và ngõ ra khác nhau.Sau đó xác định những biến nào sẽ được sử dụng, bao nhiêu trường hợp cần thu thập.

Việc lựa chọn các biến được sử dụng thường do trực giác quyết định và thường phụ thuộc vào công việc chuyên môn và lĩnh vực mà nó được ứng dụng.

#### **4.3.3.2. Các bước thực hiện**

Chọn cấu hình ban đầu (thường là 1 lớp ẩn có số neural ẩn bằng nửa tổng số neural ngõ vào và ngõ ra).

Thực hiện lặp đi lặp lại số thí nghiệm của mỗi cấu hình, giữ lại mạng tốt nhất (thường dựa vào sai số).

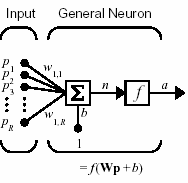
Trong mỗi lần thí nghiệm nếu xảy ra việc học chưa đủ ( kết quả thí nghiệm không đạt được xác suất như yêu cầu ) thì thử tăng số neural trong lớp ẩn. Nếu xảy ra việc học quá mức ( sai số ban đầu tăng lên ) thì hãy bỏ bớt 1 vài neural ẩn ( có thể bỏ lớp ẩn)

### *4.3.4. Cơ sở lý thuyết và giải thuật cho huấn luyện lan truyền ngược*

#### **4.3.4.1. Cấu trúc mạng lan truyền ngược**

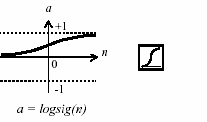
Các loại neural: (tansig, logsig, purelin).

Một loại neural có R ngõ vào thể hiện như sau:



**Hình 4. 5:** Cấu tạo 1 neural

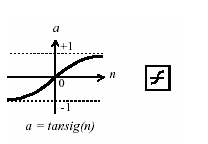
Mỗi ngõ vào tương ứng với 1 trọng số w. Tổng của giá trị ngưỡng b với tích của tích của các ngõ vào và các trọng số sẽ là ngõ vào của hàm truyền f ( f(Wp+b) ). Neural dùng nhiều hàm truyền khác nhau để tính toán ngõ ra. Mang nhiều lớp thường dùng hàm truyền log-sigmoid (logsig).



**Hình 4. 6:** Hàm truyền logsig

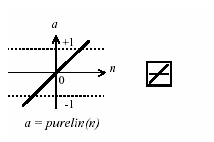
Hàm logsig tính toán giá trị ngõ ra nằm trong khoảng 0 và 1 trong khi giá trị ngõ vào từ -∞ đến +∞.

Tiếp đến là hàm truyền tan-sigmoid ( tansig)



**Hình 4. 7**: Hàm truyền tansig

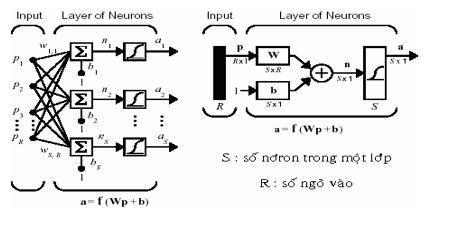
Hiếm khi hàm truyền tuyến tính ( purelin ) được sử dụng trong mạng lan truyền ngược ( thường chỉ sử dụng phổ biến trong mạng tuyến tính ).



**Hình 4. 8**: Hàm truyền purelin ( tuyến tính )

Cấu trúc các lớp trong mạng lan truyền ngược:

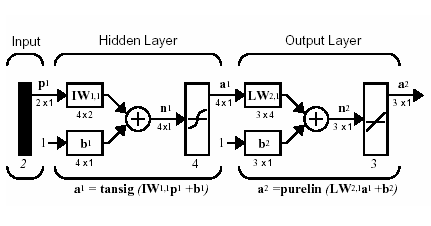
Mạng 1 lớp với S neural logsig có R ngõ vào được thể hiện như hình bên trái, còn bên phải là sơ đồ khối các lớp.

****

**Hình 4. 9:** Cấu trúc mạng 1 lớp

Mạng lan truyền ngược với các neural có hàm truyền đạt phi tuyến cho phép neural học được những mối liên quan tuyến tính và phi tuyến giữa ngõ vào và ngõ ra. Nếu giới hạn ngõ ra là 0 và 1 thì nên dùng hàm truyền logsig.

Mô hình mạng 2 lớp như sau:



**Hình 4. 10:** Cấu trúc mạng 2 lớp

#### **4.3.4.2. Huấn luyện mạng lan truyền ngược**

Mạng lan truyền ngược được huấn luyện để xấp xỉ 1 hàm phi tuyến , 1 mô hình liên kết hay phân lớp. Trong suốt quá trình huấn luyện, các trọng số và giá trị ngưỡng sẽ được điều chỉnh 1 cách hợp lý để làm tối thiểu giá trị hàm lỗi của mạng. Hàm lỗi mặc định trong mạng lan truyền ngược là trung bình phương lỗi (mean square error hay mse ) – lỗi trung bình phương giữa ngõ ra thực tế và ngõ ra được tính toán. Sau đây là 1 vài thuật toán huấn luyện mạng lan truyền ngược. Tất cả các thuật toán đều dùng độ dốc (gradient) của hàm lỗi để điều chỉnh các trọng số sao cho tối thiểu được giá trị hàm lỗi. Độ dốc được tính toán dựa vào kỹ thuật lan truyền ngược.

##### ***STEEPEST DESCENT GRADIENT***

Trong thuật toán lan truyền ngược, vector gradient của mặt phẳng sai số sẽ được tính toán. Vector này chỉ ra đường dốc nhất và vị trí hiện tai, vì thế nếu ta di chuyển theo nó 1 khoảng ngắn ta có thể đạt giá trị nhỏ nhất.

Tuy nhiên khó khăn ở đây là quyết định độ lớp của từng bước di chuyển. Bước lớn có thể hội tụ nhanh hơn nhưng có thể vượt quá điểm cần đến hay hay ra khỏi vùng có cực tiểu ( nếu mặt phẳng sai số bị lệch tâm). Ngược lại bước nhỏ có thể đi đến đúng hướng nhưng phải thực hiện lặp lại nhiều lần.

##### ***QUY TẮC HỌC THÍCH NGHI***

Phương pháp thực ra rất đơn giản. Mỗi phương pháp có 1 hệ số học e khác nhau.Khi cập nhật trọng số, nếu hướng lỗi hiện hành cùng bước với hướng trước, cho e lớn, còn ngược hướng, cho e nhỏ.

Hướng lỗi được xác định là dấu của dm, là đạo hàm riêng của hàm lỗi theo trọng số o bước m. Nếu dm dương, lỗi giảm khi trọng số giảm, nếu dm âm, lỗi giảm khi trong số tăng.

fm+1= *θ* fm + (1-*θ* ) dm

Nếu ta cho f là trung bình trọng số của các đạo hàm hiện tại và quá khứ, *θ* là trọng số của các hàm quá khứ , (1-*θ* ) là trọng số của các hàm hiện tai. Nếu f dương thì đây là lỗi “ giảm khi trọng số giảm ” và ngược lại cũng như đối với đạo hàm.

Dựa vào f và ta có thể đo chính xác cả hướng của lỗi đang giảm lẫn hướng của lỗi vừa mới giảm. Nếu chúng cùng dấu, việc giảm lỗi xảy ra theo hướng cũ, chúng khác dấu, ngược hướng với hướng cũ.

##### ***CONJUGATE GRADIENT DESCENT***

Đây là thuật toán định hướng đường, bằng cách lấy ra 1 hướng nhảy để di chuyển theo chiều ngang đa chiều, rồi chiếu đường thẳng theo hướng đó để xác định điểm nhỏ nhất và lặp lại. Hướng nhảy là hướng có độ dốc lớn nhất . Ý tưởng ở đây là khi thuật toán đã được cực tiểu hóa dọc theo 1 hướng cụ thể hóa nào đó, thỉ đạo hàm bậc 2 dọc theo hướng đó phải giữ ở zero. Các hướng liên hợp được giữ đạo hàm bậc 2 này với giả thiết mặt phẳng này là parabol. Nếu điều kiện này được giữ, N epoch sẽ đủ đạt được giá trị cực tiểu.

***Levenberg – Marquardt***

Là kỹ thuật mô phỏng vùng tin cậy: thay vì theo 1 định hướng cố định, ta giả sử mặt phẳng có hình dạng đơn giản sao cho cực tiểu có thể định vị trực tiếp, nếu giả thiết đúng. Thử mô hình và xem xét mức độ tốt của điểm được chọn. Mô hình được giả sử rằng mặt phẳng có hình dạng tốt sẽ đúng nếu gần đạt được cực tiểu. Ngoài điểm đó giả thiết có thể bị vi phạm, và mô hình có thể chọn những điểm sai để di chuyển. Mô hình chỉ có thể được tin cậy trong 1 vùng của điểm hiện tại và kích thước của vùng này thì chưa biết. Do đó, chọn các điểm mới để kiểm tra với điểm được chọn. Nếu điểm mới tốt, di chuyển đến điểm đó và tăng cường vai trò của mô hình trong việc lựa chọn điểm mới, còn nếu xấu, không di chuyển và tăng cường vai trò của bước dốc gradient trong việc lựa chọn điểm mới.

## ***4.4. Ứng dụng mạng lan truyền ngược vào nhận dạng ký tự***

Mạng lan truyền ngược được ứng dụng rộng rãi trong các vấn đề thực tế liên quan đến mạng neural. Việc thiết kế đơn giản cũng nhưng giải thuật huấn luyện hội tụ nhanh đã thể hiện mạng lan truyền ngược là 1 công cụ rất mạnh và phổ biến, đặc biệt trong vấn đề nhận dạng ký tự có cùng font, do đó việc ứng dụng nó cho 1 kết quả khả quan.

### *4.4.1. Thiết kế mạng lan truyền ngược*

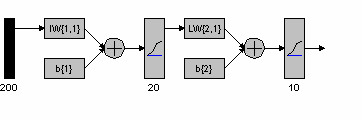
Các ký tự số sau khi được phân vùng sẽ được định chuẩn với kích thước 20×10. Sau đó ma trận ảnh của ký tự sẽ chuyển đổi thành ma trận một cột và trở thành ngõ vào của mạng neural. Như vậy, ta dùng vị trí của giá trị lớn nhất của neural ngõ ra, từ đó có thể biết được giá trị của ký tự số đó. Tập mẫu huấn luyện càng nhiều thì độ chính xác càng lớn.

Mạng nuôi tuyến lan truyền ngược gồm 2 lớp neural trong đó:

Lớp vào : 200 ngõ vào ( tương ứng với ma trận ký tự 20×10 sau khi chuyển thành ma trận cột ).

Lớp ẩn: 20 neural

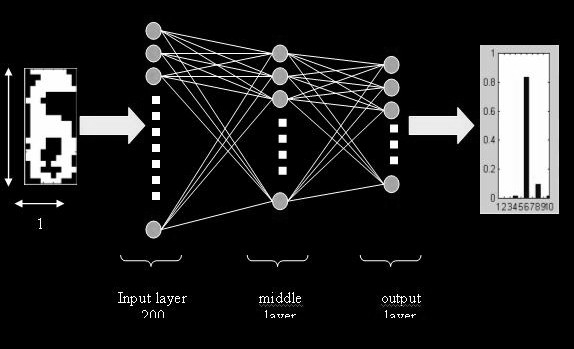
Lớp ngõ ra : 10 neural ( tương ứng để nhận dạng 10 ký tự số từ 0 đến 9). Dùng hàm truyền logsig có neural lớp ẩn và lớp ngõ ra.



**Hình 4. 11**: Cấu trúc mạng dùng để nhận dạng ký tự số

### *4.4.2. Quá trình nhận dạng*

Mạng sau khi được huấn luyện đưa vào sử dụng sẽ thực hiện nhận dạng đối với các ma trận ký tự số. Ví dụ, ma trận đầu vào lúc này là ma trận số 6. Như vậy, nếu mạng huấn luyện được kết quả tốt thì kết quả lan truyền ma trận này trong mạng là neural tương ứng với vị trí số 6 (trong hình này là neural thứ 6) sẽ có giá trị lớn nhất.

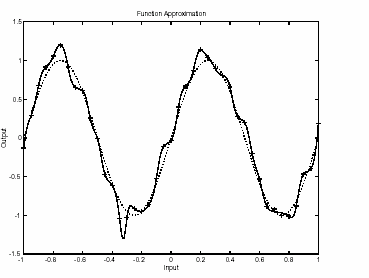


**Hình 4. 12:** Hình minh họa hoạt động của mạng trong nhận dạng

### *4.4.4. Phương pháp tăng khả năng tổng quát của mạng*

Một trong những vấn đề xuất hiện trong việc huấn luyện mạng neural được gọi là quá khớp. Lỗi trong tập huấn luyện có giá trị rất nhỏ nhưng khi đưa dữ liệu mới vào mạng thì lỗi lại quá lớn. Nghĩa là mạng có khả năng nhớ những tập huấn luyện rất tốt nhưng chưa có khả năng tổng quát đối với dữ liệu mới.

Hình sau thể hiện đáp ứng của mạng neural 1-20-1 được huấn luyện để xấp xỉ 1 hàm sin. Mạng như vậy sẽ quá khớp với dữ liệu và không có khả năng tổng quát.

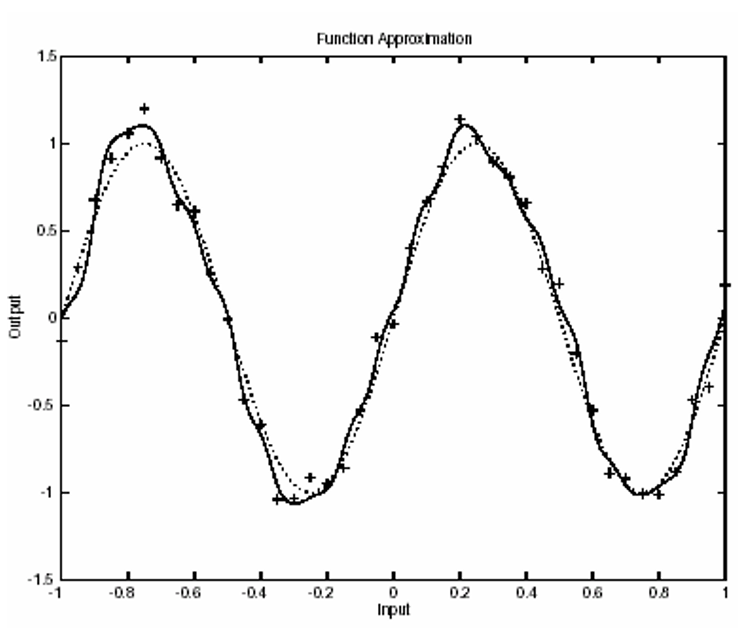


**Hình 4. 13**: Trường hợp quá khớp

Có rất nhiều phương pháp để giải quyết vấn đề này, trong đó phương pháp ngừng học sớm được xem là phương pháp hiệu quả nhất.

#### **Phương pháp ngừng học sớm**

Phương pháp này dùng 3 tập mẫu, một tập dùng để huấn luyện, một tập dùng để kiểm tra và một tập dùng để thử lại. Trong quá trình học tập mẫu huấn luyện thì lỗi của tập mẫu kiểm tra cũng được giám sát. Trong quá trình học tập thì lỗi của tập huấn luyện và kiểm tra đều giảm nhưng đến một lúc nào đó thì lỗi của tập huấn luyện bắt đầu tăng lên, lúc này xảy ra việc mạng neural bắt đầu quá khớp với tập huấn luyện. Chính vì vậy, ta cho dừng quá trình huấn luyện, thì tại thời điểm này, ta có mạng neural có khả năng tổng quát nhất, lúc này lỗi của tập thử lại ( không có trong tập huấn luyện và tập kiểm tra) là lỗi của một dữ liệu mới.



**Hình 4. 14:** Hàm xấp xỉ khi mạng ngừng học sớm

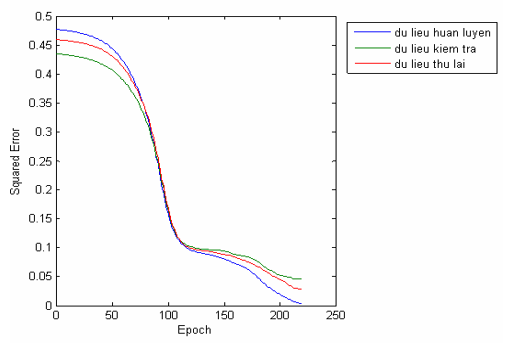
### *4.4.5. Huấn luyện mạng lan truyền ngược*

Việc huấn luyện có ý nghĩa quyết định đến độ chính xác cũng như sự thành công của chương trình. Kích thước của tập mẫu, số neural của lớp ẩn, khả năng tổng quát hóa để tránh trường hợp quá khớp của mạng cũng cần được cân nhắc và tính toán kỹ lưỡng để tăng độ chính xác của quá trình nhận dạng ký tự.

Để tăng khả năng tổng quát của neural, ta sẽ có 3 tập mẫu: một tập để huấn luyện, một tập để kiểm tra, một tập để thử lại. Trong quá trình học, khi hàm lỗi của tập kiểm tra bắt đầu tăng thì cho dừng học, thời điểm này, neural có khả năng tổng quát và lỗi của tập mẫu thử lại cũng chính là lỗi của 1 đối tượng bất kỳ nào đưa vào mạng.

Số lượng mẫu của các tập mẫu như sau:

* Tập mẫu huấn luyện gồm 30 mẫu cho mỗi ký tự số từ 0 đến 9.
* Tập mẫu dùng để kiểm tra gồm 5 mẫu cho mỗi ký tự.
* Tập mẫu dùng để thử lại gồm 5 mẫu cho mỗi ký tự.



**Hình 4. 15:** Hàm lỗi khi ngừng học sớm để huấn luyện mạng

# **CHƯƠNG 5: PHƯƠNG PHÁP TESSERACT OCR**

## ***5.1. Tổng quan về phương pháp***

### *5.1.1. Tổng quan*

Khác với những phần mềm vì mục tiêu thương mại, Tesseract là một thư viện – không phải là chương trình – nhận dạng ký tự quang học. Nó có mã nguồn mở, được công khai dưới giấy phép Apache, phiên bản 2.0, và được phát triển dưới sự tài trợ của Google từ năm 2006. Tesseract được đánh giá là một trong số ít những thư viện nhận dạng ký tự quang học mã nguồn mở tốt nhất hiện nay.

### *5.1.2. Chức năng của Tesseract*

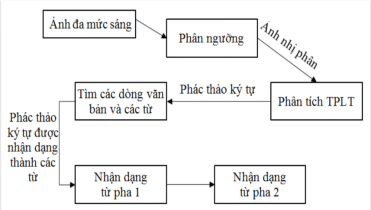
Như trên đã giới thiệu, Tesseract là một thư viện mã nguồn mở để hỗ trợ xây dựng ứng dụng nhận dạng ký tự quang học được phát triển từ năm 1995. Nó có thể chạy trên Linux, Windows và Mac, tuy nhiên do nguồn lực hạn chế nên chỉ có các phiên bản chạy trên Windows được kiểm tra chặt chẽ bởi các nhà phát triển.

Tesseract phiên bản 2.0 chỉ có thể đọc và hiển thị hình ảnh TIFF đơn giản chứa một cột văn bản. Những phiên bản đầu tiên này cũng không cho phép phân tích bố cục đối với văn bản nhiều cột hay chứa hình ảnh hoặc các công thức toán học. Kể từ phiên bản 3.0, Tesseract đã hỗ trợ định dạng văn bản đầu ra, xác định thông tin vị trí và phân tích bố cục trang. Mặt khác, một số định dạng mới như JPG, PNG, hoặc PDF được thêm vào bằng cách sử dụng thư viện Leptonica.

Nếu như phiên bản 1.0 của Tesseract chỉ có thể nhận dạng văn bản tiếng Anh thì sau này Tesseract được xây dựng để có thể được đào tạo để làm việc trong nhiều ngôn ngữ mới. Do đó từ phiên bản 2.0, Tesseract đã cho phép nhận dạng 6 ngôn ngữ bằng cách bổ sung tiếng Pháp, Tây Ban Nha... Tới phiên bản 3.0, nó đào tạo các ngôn ngữ tượng hình như ngôn ngữ Ả Rập, Trung Quốc dạng giản thể và ngôn ngữ Việt. Ngày nay, những phiên bản được cập nhật mới không chỉ mở rộng thêm nhiều ngôn ngữ mà còn nâng cao chất lượng nhận dạng.

### *5.1.3. Kiến trúc giải thuật nhận dạng chữ*

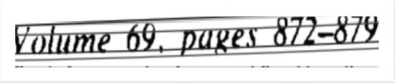
Cũng giống như hầu hết các chương trình nhận dạng ký tự quang học, Tesseract có một kiến trúc điển hình từ trên xuống. Bước đầu bằng chức năng tiền xử lý, một ngưỡng sẽ được chọn bởi bộ phân ngưỡng thông qua một quá trình phân tích các điểm trong ảnh với giải thuật Hình 2.2 Kiến trúc nhận dạng văn bản chữ in trong Tesseract như là Otsu, sau đó ngưỡng này sẽ được sử dụng để chuyển đổi ảnh màu hoặc ảnh xám đầu vào thành một ảnh nhị phân, giả thiết chứa các vùng văn bản hình dạng đa giác.



**Hình 5. 1:** Kiến trúc nhận dạng văn bản chữ in trong Tesseract

Giai đoạn tiếp theo là ảnh nhị phân được đưa vào bộ Phân tích thành phần liên thông (Connected component analysis) để tìm ra hình dạng phác thảo của những thành phần liên thông. Đây là một tiến trình phức tạp mất nhiều thời gian nhưng cần có để tách ra các ký tự có trong hình.

Khối Tìm các dòng văn bản và từ (Find text lines and words) thực hiện các chức năng như xác định dòng chặn dưới và chặn trên, đối với mỗi dòng thì cắt gọn từ trước khi xác định vùng của mỗi ký tự, ngoài ra cần nhận dạng khoảng cách giữa chữ và số. Lọc dãy dòng không chỉ tìm dãy ký tự trong từng dòng mà còn phát hiện các ký tự có độ cao chênh lệch trong dòng như ký tự drop-cap, ký tự chấm câu, ký tự dấu và nhiễu... Tuy nhiên, nếu ảnh số chứa các dòng có độ nghiêng hoặc cong thì giải thuật trở nên phức tạp. Để giúp giảm bớt mất thông tin khi nhận dạng ảnh nghiêng thì áp dụng giải thuật biến đổi Hough tìm góc nghiêng để đưa ảnh số trở lại vị trí thông thường. Trong trường hợp dòng có độ cong nào đó thì phải thiết lập các dòng cơ sở (Baseline) bằng cách sử dụng phương trình spline thích hợp cho nhiều phân đoạn.



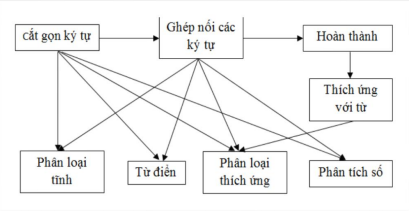
**Hình 5. 2:** Đường cơ sở hình cong

Cắt gọn từ sẽ xác định xem có các ký tự liền nhau trong một từ hay không. Nếu có nó sẽ cắt nhỏ các ký tự ra thành các ký tự riêng lẻ.



**Hình 5. 3:** Cắt các ký tự liền nhau

Nhận dạng khoảng cách giữa chữ và số là một bài toán rắc rối. Trong văn bản có nhiều phông chữ khác nhau dẫn tới khoảng cách giữa các từ và số khác nhau. Tesseract khắc phục khó khăn trên bằng cách đo khoảng cách được chọn gần ngưỡng nào đó như là giá trị mờ với sai số. Sơ đồ nhận dạng một từ là quy trình phân tích một từ được chia ra thành các ký tự:



**Hình 5. 4:** Sơ đồ nhận dạng từ

Mỗi ký tự cần nhận dạng có những đặc trưng riêng, có khoảng 50 tới 100 đặc trưng điển hình trong mỗi ký tự. Mỗi đặc trưng chứa 3 tham số là hoành độ, tung độ, và góc xoay. Trong khi đó mỗi ký tự mẫu có từ 10 tới 20 đặc trưng, mỗi đặc trưng có 4 tham số là hoành độ, tung độ, góc xoay, độ dài.

Văn bản luôn tồn tại độ dư thừa ký tự và từ vựng, vậy chức năng phân loại ký tự tạo ra danh sách rút gọn chứa các ký tự mà ký tự đối sánh có thể trùng khớp. Các lớp ký tự mẫu sinh ra các lớp véc tơ bít tương ứng với những đặc trưng của từng ký tự. Những đặc trưng của ký tự nhận dạng (Features of character) được đối sánh với lớp véc tơ bít của ký tự mẫu, và tính toán sự khác nhau giữa các đặc trưng của chúng. Bên cạnh đó có tham số thứ hai là độ dài của ký tự nhận dạng.



**Hình 5. 5:** Các đặc trưng ký tự được nhận dạng

Hệ số đánh giá đối sánh là tích hai tham số trên, cặp đối sánh nào có hệ số nhỏ nhất thì xem như chúng là tương tự nhau.

Chức năng phân loại tĩnh (static classifier) phù hợp với các ký tự có phông chữ bất kỳ, nhưng nó chủ yếu được dùng để nhận dạng các ký tự riêng như ký tự chú giải, dấu ngăn cách hay kết thúc câu... trong khi chức năng phân loại thích ứng (Adaptive classifier) dùng để nhận dạng các ký tự theo phông chữ chuẩn.

Bộ từ điển dùng để lưu trữ dữ liệu cho quá trình phân loại và nhận dạng. Mỗi ngôn ngữ có một bộ từ điển chứa các ký tự theo các phông chữ khác nhau với thuộc tính như chuẩn – normal, đậm – bold, nghiêng – italic và thuộc tính kết hợp. Từ điển cũng lưu trữ các từ hay sử dụng, từ chữ cái, từ số, từ chữ hoa, từ chữ thường.

## ***5.2. Code***

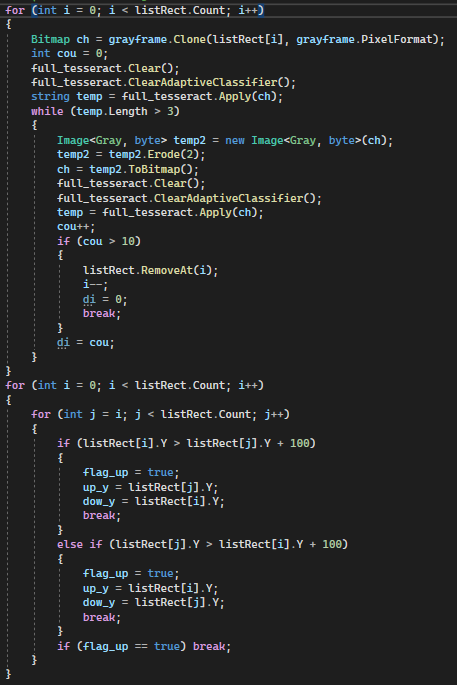
### *5.2.1. Nhận diện bằng file ảnh*

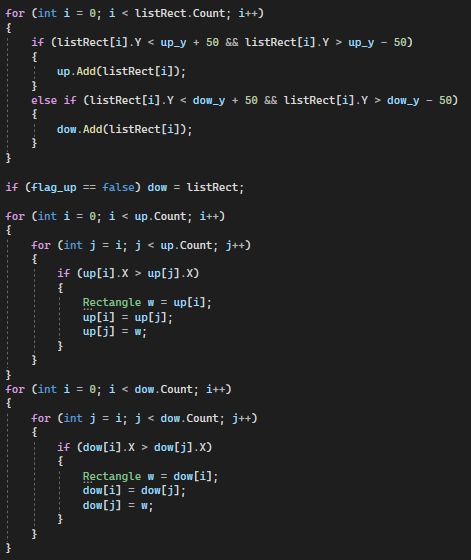
* Hàm nhận diện:
  + Đây là đoạn mã được sử dụng để thực hiện nhận diện các ký tự trong ảnh biển số xe. Nó tạo ra một danh sách các ảnh ký tự đã được xác định và các kích thước và vị trí của chúng trên ảnh biển số.

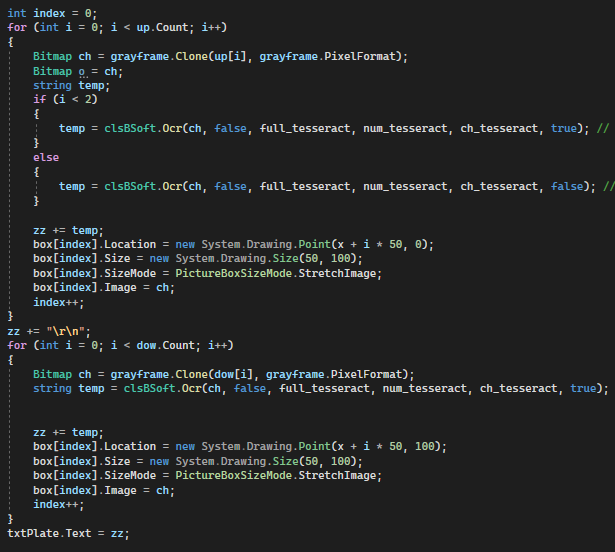


* + Đoạn mã sau đây được sử dụng để xử lý ảnh để nhận dạng được biển số xe và trích xuất được giá trị của nó.

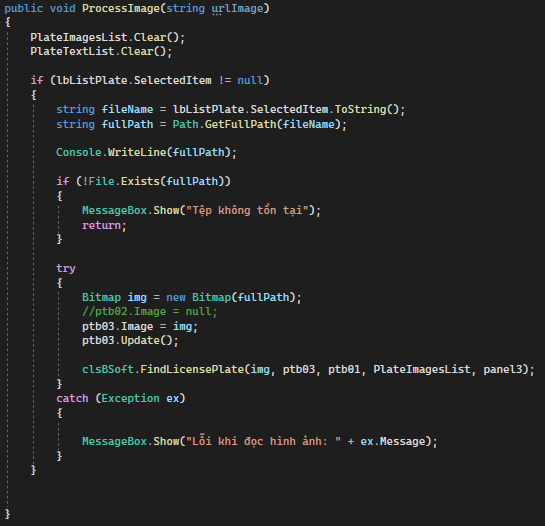
Đầu tiên, nó sử dụng một số kỹ thuật xử lý ảnh để tăng độ chính xác trong việc nhận dạng biển số xe. Sau đó, sẽ xác định vị trí các ký tự trên hình ảnh biển số xe và tách chúng ra để nhận dạng từng ký tự. Đoạn mã này sử dụng đối tượng `full\_tesseract` để nhận dạng ký tự bằng Tesseract OCR. Cuối cùng, nó trả về giá trị của biển số xe dưới dạng một chuỗi ký tự và hiển thị trên TextBox `txtPlate`.







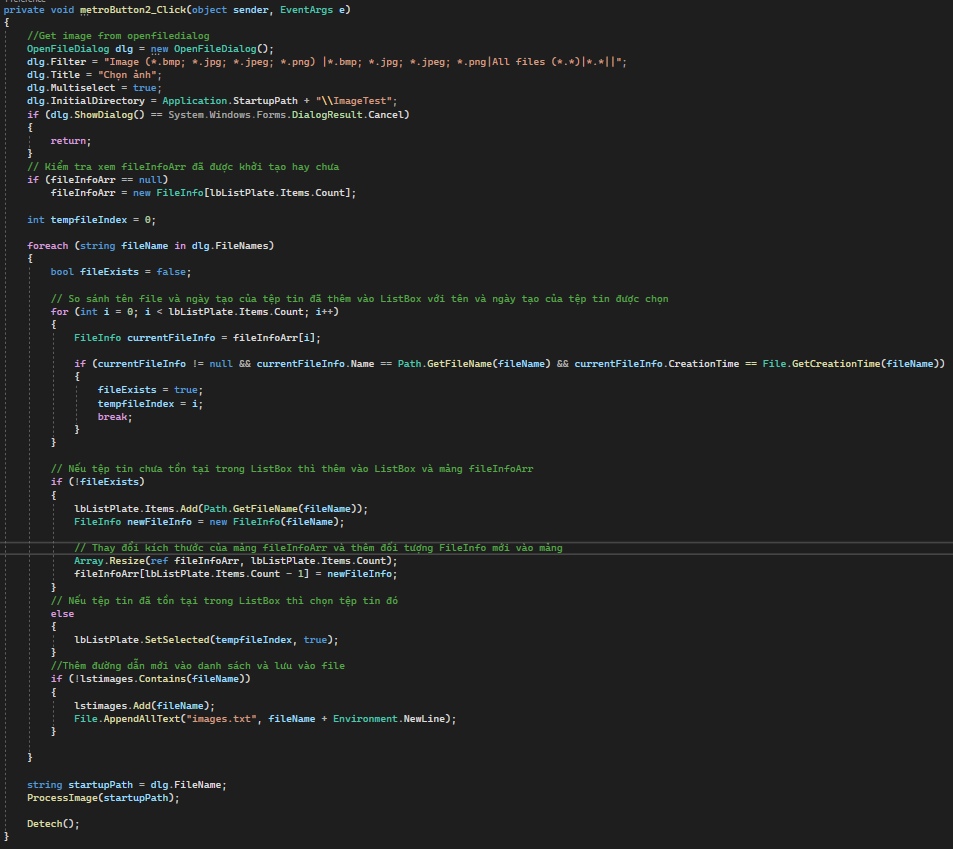
* Hàm xử lý hình ảnh biển số xe: Nó nhận đầu vào là đường dẫn của hình ảnh và sử dụng đối tượng `clsBSoft` để tìm kiếm và trích xuất biển số từ hình ảnh.



* Sự kiện chọn ảnh cần nhận dạng: Đoạn mã dưới mở hộp thoại lựa chọn tệp và cho phép người dùng chọn một hoặc nhiều tệp tin.

Sau khi người dùng chọn tệp, mã tiếp tục kiểm tra xem tệp đã được thêm vào ListBox hay chưa. Nếu như tệp đã được thêm vào trước đó, thì tệp đó sẽ được chọn trong ListBox để người dùng biết rằng tệp đã được thêm vào. Nếu tệp chưa thành viên, thì tệp sẽ được thêm vào ListBox và danh sách đường dẫn của hình ảnh sẽ được thêm vào một tệp tin địa phương có tên "images.txt".

Để hiển thị hình ảnh trên giao diện người dùng, mã gọi hàm `ProcessImage` để đọc hình ảnh và trích xuất biển số xe. Cuối cùng, đoạn mã gọi hàm `Detech` để hiển thị hình ảnh tìm thấy biển số.

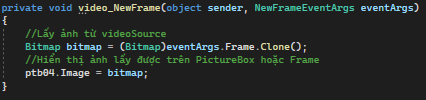


* Kết quả:

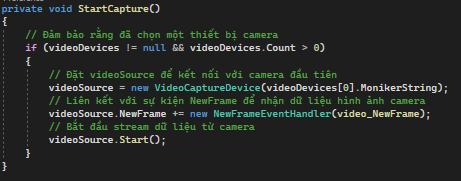


### *5.2.2. Nhận diện bằng webcam – realtime*

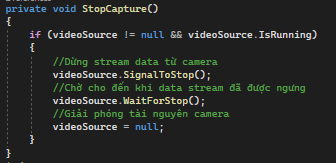
* Hiển thị video:

**

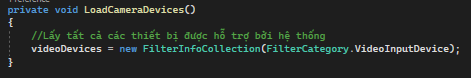
* Bắt đầu kết nối với camera:

**

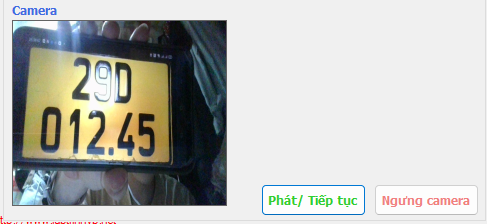
* Ngừng kết nối với camera:



* Load camera của hệ thống:

**

* Kết quả:

**

# **TÀI LIỆU THAM KHẢO**

1. PGS.TS Nguyễn Quang Hoan (2006), Giáo trình xử lý ảnh, Học viện bưu chính viễn thông, Xuất Bản Hà Nội.
2. Lương Mạnh Bá & Nguyễn Thanh Thủy (2000), Nhập môn xử lý ảnh số, Đại học Bách Khoa Hà Nội, Giáo dục.
3. Ollver TheoBald, (2015), [Machine Learning](https://www.thegioimaychu.vn/blog/thuat-ngu/machine-learning/) For Absolute Beginners: A Plain English Introduction.
4. Gary Bradski & Adrian Kaehler, (2008), Learning OpenCV, O’Reilly Media.
5. "License Plate Recognition Based on Deep Learning and Image Processing: A Survey," Neyshabouri, M. J., and Rezaie, S. (2019). https://www.sciencedirect.com/science/article/pii/S2405452618338037
6. "Vehicle Detection and License Plate Recognition Using Deep Learning," Pan, H., Zhang, Q., and Zhang, B. (2017).

https://www.mdpi.com/2076-3417/7/1/9

1. "Vehicle Recognition System Based on Deep Convolutional Neural Network," Pu, Q., and Zhou, Z. (2018). https://ieeexplore.ieee.org/document/8563822/
2. < <https://docs.opencv.org/>>.
3. "Vehicle Detection and Recognition using Machine Learning: A Review," Sharma, A., and Kaushik, S. (2019). https://www.sciencedirect.com/science/article/pii/S240545261834648X
4. "Vehicle Detection and Recognition: a Review of the Literature and Directions for Future Research," Zhang, L., Lin, L., and Liang, X. (2019). https://link.springer.com/article/10.1007/s11042-019-7234-8