ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN-TIN HỌC

BÁO CÁO

MÔN: PHÂN TÍCH XỬ LÝ ẢNH

Thành phố Hồ Chí Minh – 2026

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN-TIN HỌC

BÁO CÁO

MÔN: PHÂN TÍCH XỬ LÝ ẢNH

ĐỀ TÀI: NHẬN DẠNG VÀ ĐỌC BIỂN SỐ XE

Giảng viên: Thầy Huỳnh Thanh Sơn

Nhóm sinh viên thực hiện: Đoàn Anh Quân - 23110111

Trần Tấn Hiệp - 23110082

Lớp: 23TTH1

Thành phố Hồ Chí Minh – 2026

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Huỳnh Thanh Sơn, người đã định hướng chủ đề và cung cấp cho chúng em nhiều kiến thức cần thiết trong quá trình thực hiện báo cáo với đề tài:

Trong suốt quá trình thực hiện, chúng em đã học hỏi thêm được nhiều kiến thức mới về lĩnh vực mạng máy tính, đặc biệt là cơ chế hoạt động và tối ưu của các giao thức truyền vận hiện đại.

Chúng em cũng xin chân thành cảm ơn quý Thầy Cô trong Bộ môn đã tạo điều kiện thuận lợi cho chúng em trong suốt quá trình học tập và nghiên cứu.

Cuối cùng, chúng em xin gửi lời cảm ơn đến các bạn trong nhóm đã cùng nhau nỗ lực, thảo luận và hoàn thiện bài báo cáo với tinh thần hợp tác và trách nhiệm cao.

MỤC LỤC

DANH MỤC HÌNH

1 GIỚI THIỆU

1.1 Tổng quan

Nội dung :

- Tìm hiểu về biển số xe và hệ thống nhận dạng biển số xe
- Phát biểu bài toán và hướng giải quyết
- Nghiên cứu một số thuật toán xử lý ảnh và nhận dạng kí tự ứng dụng trong việc nhận dạng biển số xe

1.2 Nhiệm vụ đề tài

Từ nội dung nêu trên, đề tài của em sẽ bao gồm các nhiệm vụ sau:

- Tìm hiểu khái quát về xử lý ảnh và bài toán nhận dạng biển số xe.
- Tìm hiểu thông tin về biển số xe và phân loại biển số xe của Việt Nam.
- Tìm hiểu các công đoạn chính của bài toán nhận dạng biển số xe gồm 2 khâu chính:
  - Phát hiện vị trí và tách biển số xe
  - Phân đoạn kí tự trong biển số xe

2 TỔNG QUAN BÀI TOÁN NHẬN DIỆN BIỂN SỐ XE

2.1 Khái niệm biển số xe.

Ở Việt 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, được cơ quan công an 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 sắt, có dạng hình chữ nhật hoặc hơi vuông, trên đó có in số và chữ (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 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 ô tô có 2 loài là biển số ngắn: kích thước (330 x 165) mm, bốn góc được bo tròn. Vị trí Công an hiệu được dập ở giữa 2 hàng chữ, số trên và dưới, cách mép trái 5 mm và biển số dài: kích thước (520 x 110) mm, bốn góc được bo tròn. Vị trí Công an hiệu dập phía trên của nét gạch ngang, mép trên Công an hiệu thẳng hàng với mép trên của dãy chữ và số. Biển số xe mô tô có kích thước biển số xe mô tô 190

x 140 mm, bốn góc được bo tròn. Ký hiệu bảo mật Công an hiệu được dập ở vị trí phía trên nét gạch ngang hàng trên của biển số xe mô tô, cách mép trên của biển số xe mô tô 5 mm.

2.2 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 như:

- Nhận dạng ảnh
- Xử lý hình ảnh
- Phục hồi hình ảnh/video
- Thực tế ảo
- Các ứng dụng khác

2.3 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 2 bước chính:

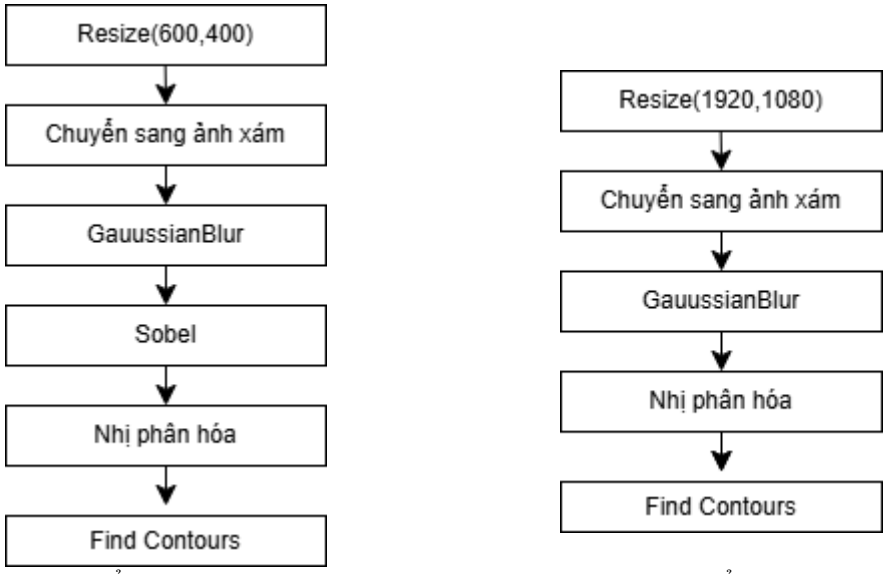
1. Tìm và tách biển số xe
2. Tách từng kí tự

Hình 2.3 - 1 Các bước chính trong nhận dạng biển số xe

3 TÌM VÀ TÁCH BIỂN SỐ XE

3.1 Hướng giải quyết

Sơ đồ sẽ tóm gọn các bước để xác định và tách biển số với biển 1 hàng và 2 hàng :



Hình 3.1 - 1 Hướng giải quyết của biển 1 và 2 hàng

Đầu tiên, ảnh đầu vào sẽ được **resize về kích thước** nhằm chuẩn hóa dữ liệu, giúp giảm khối lượng tính toán và đảm bảo tính ổn định cho các bước xử lý sau. Tiếp theo, ảnh được **chuyển sang ảnh xám** để loại bỏ thông tin màu sắc RGB không cần thiết, từ đó tập trung vào sự khác biệt về cường độ sáng giữa biển số và môi trường xung quanh. Sau đó, ta áp dụng **bộ lọc Gaussian Blur** nhằm làm mịn ảnh và giảm nhiễu, hạn chế các chi tiết nhỏ gây ảnh hưởng đến quá trình phát hiện đặc trưng. Ở bước tiếp theo , toán tử **Sobel(chỉ áp dụng với 1 hàng)theo một hướng** được sử dụng để làm nổi bật các biên có sự thay đổi cường độ sáng mạnh, đặc biệt là các cạnh ngang đặc trưng của biển số. Kết quả thu được sẽ được đưa qua bước **nhị phân hóa**, giúp tách rõ vùng đối tượng (biển số) khỏi nền. Cuối cùng, thuật toán **Find Contours** được sử dụng để phát hiện và trích xuất các đường bao, từ đó xác định vị trí và khoanh vùng biển số trong ảnh phục vụ cho các bước nhận diện tiếp theo.

3.2 Resize

Resize ảnh là bước tiền xử lý nhằm chuẩn hóa kích thước ảnh đầu vào trước khi đưa vào các thuật toán xử lý ảnh. Trong thực tế, các frame trích xuất từ video có thể có độ phân giải khác nhau, điều này gây khó khăn cho việc lựa chọn tham số (kernel, ngưỡng, kích thước bộ lọc…) và

làm giảm tính ổn định của hệ thống. Việc đưa tất cả ảnh về cùng kích thước giúp đảm bảo các thuật toán phía sau hoạt động nhất quán và cho kết quả đồng đều.

Ngoài ra, resize ảnh còn giúp **giảm độ phức tạp tính toán**, từ đó tăng tốc độ xử lý, đặc biệt quan trọng trong các bài toán xử lý video hoặc thời gian thực. Kích thước được lựa chọn vì vẫn giữ được các đặc trưng quan trọng của biển số (hình dạng, cạnh, tỉ lệ) trong khi số điểm ảnh cần xử lý đã được giảm đáng kể. Nhờ đó, bước resize vừa đảm bảo hiệu quả xử lý, vừa hạn chế mất mát thông tin cần thiết cho các bước nhận diện biển số tiếp theo.

3.3 Chuyển ảnh xám và tăng độ tương phản

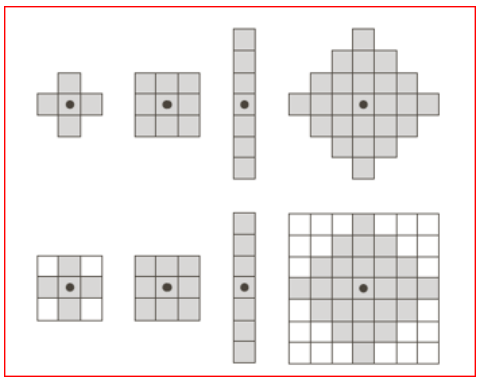
3.3.1 Ảnh xám

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.

3.3.2 Phép toán hình thái học

Hình thái học toán học là một lý thuyết và kỹ thuật để phân tích và xử lý cấu trúc hình học, hình ảnh đầu ra được xác định chủ yếu dựa vào phần từ cấu trúc (structuring elements/kernel)



Hình 3.3.2 - 1 Ví dụ về phần tử cấu trúc

Hình thái học toán học đã được phát triển cho hình ảnh nhị phân, và sau đó được mở rộng cho ảnh xám,... Đây là một trong những kỹ thuật được áp dụng trong giai đoạn tiền xử lý. Hai phép toán thường dùng là phép giãn nở (Dilation) và phép co (Erosion) . Từ hai phép toán cơ bản này người ta phát triển thành một số phép toán như phép đóng (Closing) và phép mở (Opening) và phép Top Hat, Black Hat.

- **Phép mở**

Là thực hiện phép co trước sau đó mới thực hiện phép giãn nở. Phép toán mở được ứng dụng trong việc loại bỏ các phần lồi lõm và làm cho đường bao các đối tượng trong ảnh trở nên mượt mà hơn.

Ảnh gốc

Ảnh sau khi dùng phép mở

Hình 3.3.2 - 2 Phép mở

- **Phép đóng**

Thực hiện phép giãn nở trước sau đó mới thực hiện phép co. Phép toán đóng được dùng trong ứng dụng làm trơn đường bao các đối tượng, lấp đầy các khoảng trống biên và loại bỏ những hố nhỏ.

Ảnh gốc

Ảnh sau khi dùng phép đóng

Hình 3.3.2 - 3 Phép đóng

- **Phép Top Hat**

Phép Top Hat là kết quả của phép trừ ảnh của ảnh ban đầu với ảnh sau khi thực hiện phép mở, dùng để làm nổi bật nhưng chi tiết trắng trong nền tối

Ảnh gốc

Ảnh sau khi dùng phép Top Hat

Hình 3.3.2 - 4 Phép Top Hat

- **Phép Black Hat**

Phép Black Hat là kết quả của phép trừ ảnh của ảnh sau khi thực hiện phép đóng với ảnh ban đầu. Dùng làm nổi bật chi tiết tối trong nền trắng.

Ảnh gốc

Ảnh sau khi dùng phép Black Hat

Hình 3.3.2 – 5 Phép Black Hat

3.3.3 Tăng độ tương phản

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.

Ảnh gốc

Ảnh sau khi tăng độ tương phản

Hình 3.3.3 - 1 Ảnh sau khi tăng độ tương phản

3.4 GaussianBlur

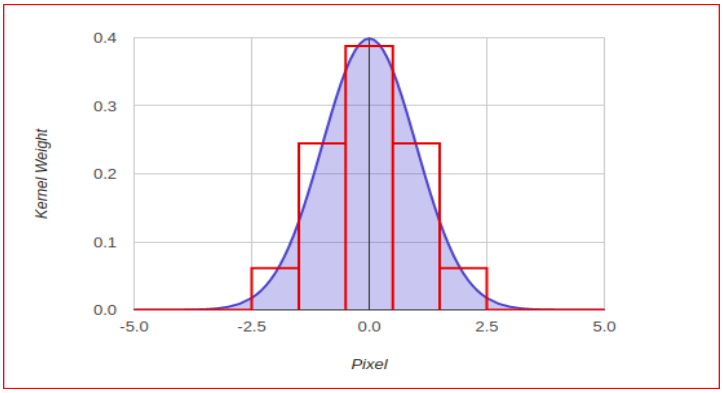
3.4.1 Nhiễu

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. Trên thực tế có nhiều loại nhiễu, nhưng người ta thường chia làm ba loại: nhiễu cộng, nhiễu nhân và nhiễu xung. Bản chất của nhiễu thường tương ứng với tần số cao và cơ sở lý thuyết của bộ lọc là chỉ cho những tín hiệu có tần số nhất định đi qua, nên người ta thường sử dụng bộ lọc thông thấp hay trung bình.

Hình 3.4.1 - 1 Nhiễu

3.4.2 Bộ lọc Gauss (Gauss filter)

Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gauss sau đó cộng chúng lại để tạo thành ảnh đầu ra. Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa. Trọng số của sự phụ thuộc được lấy theo hàm Gauss (cũng được sử dụng trong quy luật phân phối chuẩn).



Hình 3.4.2 – 1 Ma trận lọc Gauss

Giả sử ảnh là một chiều. Riem ảnh ở trung tâm sẽ có trọng số lớn nhất. Các điểm ảnh ở càng xa trung tâm sẽ có trọng số giảm dần khi khoảng cách từ chúng tới điểm trung tâm tăng lên. Như vậy điểm càng gần trung tâm sẽ càng đóng góp nhiều hơn vào giá trị điểm trung tâm.

Ảnh gốc

Ảnh sau khi làm mờ, giảm nhiễu

Hình 3.4.2 - 2 Kết quả sử dụng bộ lọc Gaussa

3.5 Sobel

3.5.1 Khái niệm

Sobel là một **toán tử phát hiện biên (edge detector)** trong xử lý ảnh số, dựa trên việc **xấp xỉ đạo hàm bậc nhất** của ảnh theo các hướng không gian. Mục tiêu của Sobel là xác định **vị trí và mức độ mạnh của biên**, tức những vùng mà **cường độ mức xám thay đổi đột ngột**.

- Gradient **lớn** → biên rõ

- Gradient **nhỏ** → vùng phẳng

Độ lớn gradient:

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

Sobel **xấp xỉ các đạo hàm riêng này bằng tích chập (convolution)** với các kernel rời rạc.

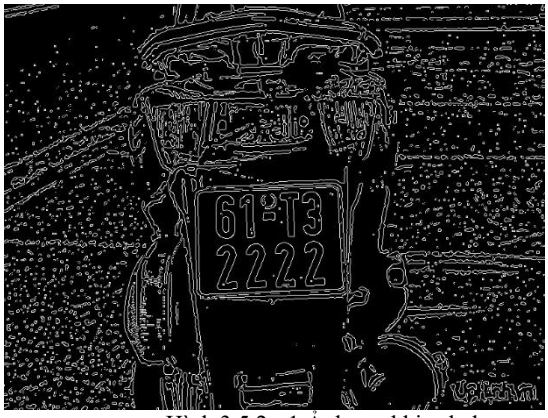
3.5.2 Sobel kernels

Kernel theo trục x (phát hiện biên dọc): $G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

- Nhạy với **sự thay đổi cường độ theo chiều ngang**
- Phát hiện **biên dọc**

Kernel theo trục y (phát hiện biên ngang): $G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

- Nhạy với **sự thay đổi cường độ theo chiều dọc**
- Phát hiện **biên ngang**



Hình 3.5.2 - 1 Ảnh sau khi sobel

3.6 Lọc biển số với contour

3.6.1 Một số phương pháp tìm contour

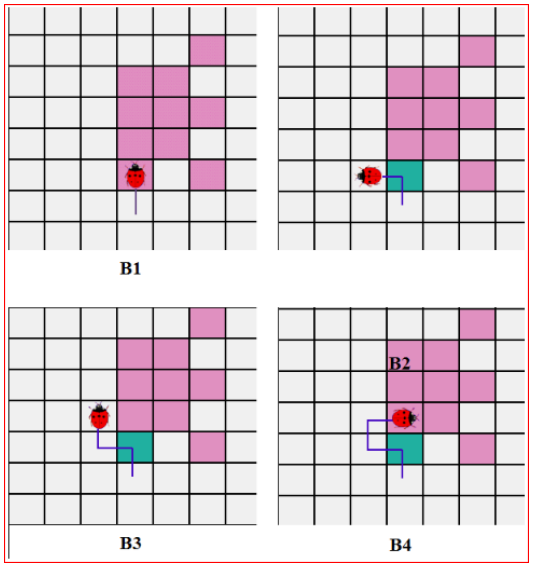
Có thể hiểu Contour là tập hợp các điểm tạo thành đường cong kín bao quanh một đối tượng nào đó. Thường dùng để xác định vị trí, đặc điểm của đối tượng. Có 4 thuật toán Contour Tracing chung nhất. Hai trong số đó có tên là: Square Tracing algorithm và Moore – Neighbor Tracing là dễ để thực hiện và thường xuyên được dùng để dò tìm contour của một mẫu. Dưới đây em sẽ trình bày kĩ hơn về 2 phương pháp trên:

3.6.2 Thuật toán Square Tracing

Duyệt từ pixel ngoài cùng bên trái phía dưới, đi lên cho tới khi gặp pixel có giá trị bằng 255 (pixel này sẽ được gọi là pixel start) thì bắt đầu di chuyển theo quy tắc sau:

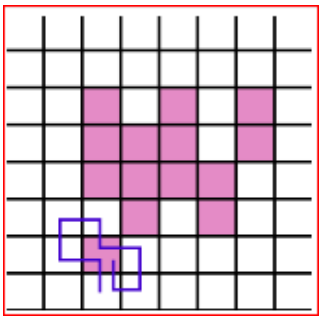
- Nếu gặp Pixel có giá trị bằng 255 thì rẽ trái.
- Nếu gặp Pixel có giá trị bằng 0 thì rẽ phải.
- Di chuyển cho tới khi quay lại pixel start thì dừng lại.

Hình sau mô tả cách hoạt động của thuật toán:

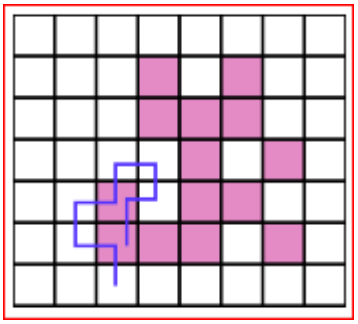


Hình 3.6.2 - 1 Thuật toán Square Tracing

Thuật toán sẽ kết thúc đúng khi di chuyển vào pixel start lần thứ 2 sau khi đi qua n pixel khác và theo đúng hướng đi vào pixel start lần đầu tiên. Và sai khi di chuyển vào pixel start mà không đúng hướng ban đầu. Vậy thuật toán này chỉ chạy đúng trên đối tượng 4 - connected.



Hình 3.6.2 - 2 Thuật toán Square Tracing chạy đúng

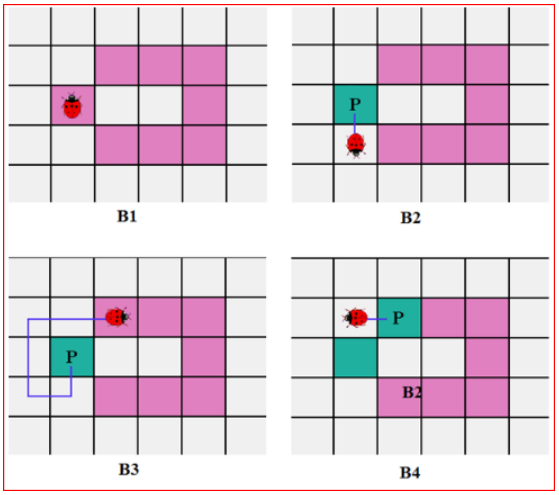


Hình 3.6.2 - 3 Thuật toán Square Tracing chạy sai

3.6.3 Thuật toán Moore – Neighbor

Thuật toán này có chút khác biệt so với thuật toán Square Tracking, cụ thể: khi gặp pixel có giá trị bằng 255 đầu tiên (pixel start) thì ta sẽ quay lại pixel trước đó, sau đó đi vòng qua các

pixel thuộc 8-connected theo chiều kim đồng hồ cho tới khi gặp pixel khác có giá trị bằng 255. Và điệu kiện kết thúc cũng giống như thuật toán Square Tracking

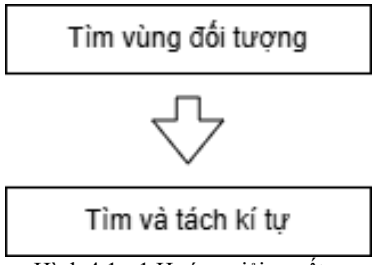


Hình 3.6.3 - 1 Thuật toán Moore - Neighbor.

4 TÁCH TỪNG KÍ TỰ

4.1 Hướng giải quyết

Ở giai đoạn này có những bước chính sau: Tìm tất cả các vùng kín cho là kí tự và lọc ra những kí tự đúng. Tách hình ảnh nhưng kí tự đó ra.



Hình 4.1 - 1 Hướng giải quyết

4.2 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

Hình 4.2 - 1 Ảnh sau khi tìm được vùng đối tượng

Trong ảnh 4.2 – 1 Sau khi đã áp dụng các điều kiện thì sẽ vẽ ra những hình chữ nhật màu đỏ bao quanh các kí tự.

4.3 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.

Hình 4.3 - 1 Ảnh sau khi nhận diện và tách kí tự

5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Ưu điểm:

- Pipeline nhận diện theo hướng xử lý ảnh + tách biển số + tách ký tự có lợi thế dễ cài đặt, dễ triển khai, phù hợp cho mục tiêu học tập/tiếp cận căn bản.
- Khá nhẹ nên máy tính với cấu hình yếu cũng có thể xử lý mượt mà so với các thuật toán khác như CNN, SVM.
- Phù hợp cho đối tượng sinh viên muốn tìm hiểu căn bản về xử lý ảnh hay trí tuệ nhân tạo.

Khuyết điểm:

- Hệ thống vẫn **nhạy với điều kiện môi trường**: phản chiếu, di ảnh/mờ do chuyển động, chói sáng, ký tự không rõ… dẫn tới nhận sai.

- Để chạy ổn định thường cần camera cố định + ánh sáng/phông nền được kiểm soát, và chưa tự động hoàn toàn, còn cần giám sát con người.

Vì vậy tốt nhất cần đặt camera cố định, với môi trường ánh sáng xung quanh được cài đặt trước. Phông nền cần hạn chế tối da nhưng chi tiết lóe sáng gây nhiễu. Phương pháp này vẫn còn cần sự giám sát của con người nhiều chứ chưa thể hoàn toàn tự động.

5.2 Hướng phát triển

Thay thuật toán nhận dạng ký tự sang mô hình tốt hơn (ví dụ **SVM/CNN**) hoặc dùng thư viện phát hiện hiện đại như **YOLO/YOLOv3** để tăng độ chính xác & tính tổng quát.Sử dụng camera chuyên dụng cho việc nhận diện biển số xe vì có khả năng chống chịu với sương mù, đêm tối, chói sáng,...

Dùng **camera chuyên dụng** để chịu được sương mù/ban đêm/chói sáng.

Cải thiện khâu xác định vị trí biển số bằng **Hough**, nhận diện theo **màu sắc**, và các kỹ thuật **giảm di ảnh khi xe chạy**.

Tích hợp vào hệ thống quản lý thực tế: **kho/bãi, quản lý phương tiện, tìm xe thất lạc, theo dấu…**

6 TÀI LIỆU THAM KHẢO

[1] M. J. Ahmed, M. Sarfaz, A. Zidouri, and K. G. AI-Khatib, “License plate recognition system,” Proc. IEEE Int. Conf. Electron. Circuits, Syst., vol. 2, no. January, pp. 898–901, 2003, doi: 10.1109/ICECS.2003.1301932.

[2] C. N. E. Anagnostopoulos, “License plate recognition: A brief tutorial,” IEEE Intell. Transp. Syst. Mag., vol. 6, no. 1, pp. 59–67, 2014, doi: 10.1109/MITS.2013.2292652.

[3] A. Badr, M. M. Abdel, A. M. Thabet, and A. M. Abdelsadek, “Automatic number plate recognition system,” Ann. Univ. Craiova, Math. Comput. Sci. Ser., vol. 38, no. 1, pp. 62–71, 2011, doi: 10.5120/ijca2018917277.

[4] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, “Automatic License Plate Recognition,” IEEE Trans. Intell. Transp. Syst., vol. 5, no. 1, pp. 42–53, 2004, doi: 10.1109/TITS.2004.825086.

[5] N. D. Linh, N. Van Nhan, and D. Van Dat, “12.pdf,” Tap chi thong tin khoa hoc va cong nghe Quang Binh, 2018.

[6] D. V. R. Mohan, M. T. Communication, S. Srkr, and E. College, “Number Plate Recognition by using open CV- Python,” pp. 4987–4992, 2019.

[7] Nguyễn Vĩnh An, “So sánh một số phương pháp phát hiện biên,” Tạp chí khoa học Trường Đại học Quốc gia Hà Nội, vol. 31, no. 2, pp. 1–7, 2015.

[8] F. Patel, J. Solanki, V. Rajguru, and A. Saxena, “Recognition of Vehicle Number Plate Using Image Processing Technique,” Adv. Emerg. Med., vol. 7, no. 1, pp. 2–8, 2018, doi: 10.18686/aem.v7i1.

[9] K. Sarbjit, “An Efficient Approach for Automatic Number Plate Recognition System under Image Processing,” Int. J. Adv. Res. Comput. Sci., vol. 5, no. (6), pp. 43–50, 2014.

[10] G. D. Yeshwant, S. Maiti, and P. B. Borole, “Automatic Number Plate Recognition System (ANPR System),” Int. J. Eng. Res., vol. 3, no. 7, p. 5, 2014, [Online]. Available: https://www.ijert.org/research/automatic-number-plate-recognition-system-anpr-system-IJERTV3IS071132.pdf.

[11] A. Zelinsky, Learning OpenCV---Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008)[On the Shelf], vol. 16, no. 3. 2009.

[12] Chris Dahms (2016), OpenCV 3 License Plate Recognition Python. https://www.youtube.com/watch?v=fJcl6Gw1D8k

[13] OpenCV. Morphological Transformations. https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html

[14] Nhận dạng biển số xe với opencv step by step. https://thorpham.github.io/blog/2018/04/11/regconite-plate-car/

[15] Find and draw Contours - OpenCV 3.4 with python 3. https://www.youtube.com/watch?v=_aTC-Rc4Io0

[16] Contour Features. https://docs.opencv.org/trunk/dd/d49/tutorial_py_contour_features.html

[17] Tìm hiểu về Contour, moments trong xử lý ảnh. https://congdongopencv.blogspot.com/2017/11/tim-hieu-ve-contour-moments-trong-xu-ly.html