

MỤC LỤC

MỤC LỤC.....	1
DANH MỤC HÌNH ẢNH	4
LỜI CẢM ƠN.....	6
LỜI MỞ ĐẦU	8
CHƯƠNG I: TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ BÀI TOÁN NHẬN DẠNG BIÊN SỐ XE Ô TÔ Ở CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO	10
1.1 TỔNG QUAN VỀ XỬ LÝ ẢNH.....	10
1.1.1 Một số khái niệm cơ bản trong xử lý ảnh	10
1.1.1.1 Phần tử ảnh (Pixel (Picture Element))	10
1.1.1.2 Độ phân giải của ảnh (Resolution)	11
1.1.1.3 Mức xám (Gray Level)	11
1.1.1.4 Sử dụng các bộ lọc:.....	12
1.1.2 Quan hệ giữa các điểm ảnh	13
1.1.2.1 Các lân cận của điểm ảnh (Image Neighbors)	13
1.1.2.2 Các mối liên kết điểm ảnh	13
1.1.2.3 Đo khoảng cách giữa các điểm ảnh	14
1.1.3 Chu tuyến của một đối tượng ảnh	15
1.1.3.1 Định nghĩa Chu tuyến	15
1.1.3.2 Định nghĩa Chu tuyến đối ngẫu	15
1.1.3.3 Định nghĩa Chu tuyến ngoài	15
1.1.3.4 Định nghĩa Chu tuyến trong	16
1.1.3.5 Định nghĩa điểm trong và điểm ngoài chu tuyến.....	16
1.1.3.6 Bổ đề Chu tuyến đối ngẫu	16
1.1.3.7 Bổ đề Phần trong/ngoài của chu tuyến	16
1.1.3.8 Định lý về tính duy nhất của chu tuyến ngoài	17
1.1.4 Biên và các phương pháp phát hiện biên	17
1.1.4.1 Các Phương pháp phát hiện biên trực tiếp.....	17
1.1.4.2 Phát hiện vùng chứa biên số xe	25
1.1.4.3 Phương pháp phát hiện biên gián tiếp	25
1.1.4.4 Biểu diễn đường biên.....	28
1.1.4.5 Biểu diễn đường biên bằng hàm Radius – vector.....	30

1.2 NHẬN DẠNG BIÊN SỐ XE Ô TÔ Ở CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO	31
1.2.1 Khái quát về biển số xe ở Cộng hòa Dân chủ Nhân dân Lào.....	31
1.2.2 Khái niệm & ứng dụng về nhận dạng	32
1.2.3 Phân loại biển số xe	34
1.2.4 Điểm ảnh, biên ảnh	39
1.2.5 Vai trò của việc nhận dạng	42
CHƯƠNG II: MỘT SỐ VẤN ĐỀ TRONG NHẬN DẠNG ẢNH BIÊN SỐ XE Ô TÔ CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO	43
2.1 KỸ THUẬT BÓC TÁCH ẢNH	43
2.1.1 Thuật toán nhị phân hoá ảnh đầu vào.....	43
2.1.2 Thuật toán tách cạnh, dò biên	44
2.1.3 Tính toán các số đo đặc trưng của đối tượng ảnh nhị phân.....	46
2.1.4 Thuật toán phân vùng, phân cụm	47
2.2 KỸ THUẬT TÁCH CÁC KÝ TỰ	49
2.2.1 Các phương pháp tách ký tự và nhận dạng ký tự.....	50
2.2.1.1 Tách ký tự	50
2.2.2 Nhận dạng ký tự.....	54
2.3 KỸ THUẬT TRÍCH CHỌN ĐẶC TRƯNG ẢNH	58
2.3.1 Màu sắc	59
2.3.1.1 Độ đo tương đồng về màu sắc	60
2.3.1.2 Không gian màu.....	62
2.3.2 Kết cấu.....	62
2.3.3 Đặc trưng hình dạng.....	62
2.3.3.1 Độ đo tương đồng cho hình dạng	63
2.3.4 Đặc trưng cục bộ bất biến.....	63
2.4 NHẬN DẠNG ẢNH BIÊN SỐ XE	69
2.4.1 Tiền xử lý	70
2.4.1.1 Trích chọn đặc trưng ảnh.....	70
2.4.2 Nhận dạng.....	71
3.1. MÔ TẢ VÀ GIỚI THIỆU VỀ BIÊN SỐ XE Ô TÔ CỦA CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO	72
3.2 THƯ VIỆN OPENCV	73
3.2.1 OpenCV là gì?	73

3.2.2 Những ai sử dụng OpenCV	74
3.2.3 Nguồn gốc của OpenCV	74
3.3 BỘ THƯ VIỆN MÃ NGUỒN MỞ OPENCV	75
3.3.1 Một vài thư viện của openCV	75
3.3.2 Các thao tác ảnh cơ bản	76
3.4 HÀM OPENCV HỖ TRỢ XÂY DỰNG ỨNG DỤNG NHẬN DẠNG BIỂN SỐ XE.	77
3.4.1 Hàm cvLoadImage.....	77
3.4.2 Hàm cv ReleaseImage	78
3.4.3 Hàm cvCreateImage	78
3.4.4 Hàm cvCvtClolor	79
3.4.5 Hàm cvNamedWindow.....	80
3.4.6 Hàm cvShowImage.....	80
3.4.7 Hàm cvResizeWindow	80
3.4.8 Hàm cvThreshold.....	81
3.4.9 Hàm cvFindContours.....	82
3.4.10 Hàm cvBoundingRect	83
3.4.11 Hàm cvCircle.....	84
3.4.12 Hàm cvMemStorage.....	84
3.4.13 Hàm cvBoundingRect	85
3.4.14 Hàm cvDrawContours.....	85
3.5 VISUAL C/C++	86
3.6. GIAO DIỆN CỦA CHƯƠNG TRÌNH NHẬN DẠNG BIỂN SỐ XE.	86
KẾT LUẬN	93
TÀI LIỆU THAM KHẢO.....	94

DANH MỤC HÌNH ẢNH

Hình 1.1. Quá trình xử lý ảnh	10
Hình 1.2 Kỹ thuật lọc trung vị	12
Hình 1.3 Kỹ thuật lọc trung bình	12
Hình 1.4 Ví dụ về chu tuyến của đối tượng ảnh	15
Hình 1.5 Chu tuyến trong, chu tuyến ngoài	16
Hình 1.6 Sơ đồ giải quyết.....	25
Hình 1.8 a) Hàm radius – vector	28
b) Vấn đề của hàm radius – vector khi hình không phải là hình sao	30
Hình 1.9 a) Hình sao X	29
b) Hàm radius – vector của hình X	31
Hình 1.10 biển số xe cũ hiện này	40
Hình 1.11 biển số xe mới	42
Hình 2.1 Nhị phân hoá ảnh	44
Hình 2.2 Tách cạnh Sobel, Prewitt.....	46
Hình 2.3 Các hình ảnh trung gian xử lý bóc tách ảnh biển số xe	49
Hình 2.4 Lưu đồ tách ký tự	50
Hình 2.5 Các bước tách ký tự sử dụng kỹ thuật tách ngưỡng.....	52
Hình 2.6 Các bước tách ký tự sử dụng kỹ thuật sobel	53
Hình 2.7 Sơ đồ tổng quát một hệ nhận dạng	58
Hình 2.8 Định nghĩa bộ lọc DoG	65
Hình 2.9 Xác định điểm cực trị.....	66
Hình 2.10 Định vị điểm hấp dẫn	66
Hình 2.11 Trích ảnh Gaussian.....	67
Hình 2.12 Tính độ lớn và hướng của gradient	68
Hình 2.13 Sơ đồ khối của một hệ thống nhận dạng	68
Hình 3.1 Giao diện của chương trình.....	87
Hình 3.2 Chọn ảnh trên Menu(File).....	88
Hình 3.3 Chọn ảnh trong button (Load Image).....	88

Hình 3.4 Ảnh đã được chọn	89
Hình 3.5 Nhận dạng biển số xe trong button (Show Result)	89
Hình 3.5 Phát hiện biển số xe.....	86
Hình 3.6 Tách các ký tự	90
Hình 3.7 nhận dạng các ký tự.....	86
Hình 3.8 Các Dữ liệu sau nhận dạng.....	90
Hình 3.9 biểu đồ của thời gian phát hiện	91
Hình 3.10 biểu đồ của thời gian nhận dạng	91
Hình 3.11 biểu đồ của tổng thời gian.....	92

.....

LỜI CẢM ƠN

Em xin cảm ơn về nội dung luận văn tốt nghiệp với tên đề tài: **“Nghiên cứu nhận dạng biển số xe ô tô Cộng hòa dân chủ nhân dân Lào”** không sao chép nội dung cơ bản từ các luận văn khác hay các sản phẩm tương tự không phải do em làm ra. Sản phẩm của luận văn là chính bản thân em nghiên cứu và xây dựng.

Nếu có gì sai em xin chịu trách nhiệm trước Trường Đại học Công nghệ thông tin và Truyền thông – Đại học Thái Nguyên.

Thái Nguyên, ngày 24 tháng 8 năm 2016
HỌC VIÊN

KHAMPHOUMY Phonevilay

LỜI CẢM ƠN

Để hoàn thành chương trình cao học và viết luận văn tốt nghiệp, em đã nhận được sự hướng dẫn, giúp đỡ góp ý nhiệt tình của quý thầy cô trong trường Đại học Công nghệ Thông tin và Truyền thông – Đại học Thái Nguyên.

Trong quá trình học tập và rèn luyện tại khoa Công nghệ thông tin – đại học Thái Nguyên, đến nay em đã kết thúc khóa học 2 năm và hoàn thành luận văn tốt nghiệp. Để có được kết quả này em xin chân thành cảm ơn

Ban chủ nhiệm khoa Công nghệ thông tin cùng các thầy, cô giáo trong khoa đã giảng dạy, quan tâm và tạo điều kiện thuận lợi để chúng em học tập và rèn luyện trong suốt thời gian theo học tại trường Công nghệ Thông tin và Truyền thông.

Em xin chân thành cảm ơn **TS. Nguyễn Văn Huân** – trưởng khoa hệ thống thông tin kinh tế, và các thầy - cô trong phòng hệ thống thông tin kinh tế đã tận tình giúp đỡ và tạo điều kiện cho tôi hoàn thành báo cáo này.

Thái Nguyên, ngày 24 tháng 8 năm 2016

HỌC VIÊN

KHAMPHOUMY Phonevilay

LỜI MỞ ĐẦU

Ngày nay trên thế giới bên cạnh việc tăng trưởng kinh tế là sự phát triển của các ngành khoa học kỹ thuật nói chung, trong đó ngành công nghiệp sản xuất các phương tiện giao thông lại là một trong những ngành có tốc độ phát triển cực nhanh. Sự phát triển ấy, được thể hiện rõ ràng nhất thông qua hình ảnh các phương tiện giao thông trên thế giới ngày một tăng cao và đa dạng. Tuy nhiên, điều đó lại gây ra một áp lực đối với cơ quan và các cấp quản lý, làm cho công tác quản lý và giám sát sẽ khó khăn hơn. Đây cũng là một trong những vấn nạn ở Lào. Công tác quản lý phương tiện giao thông nói chung và quản lý ô tô, xe máy là vô cùng phức tạp... cũng như công tác phát hiện, xử phạt các hành vi vi phạm giao thông, chống trộm... sẽ tốn nhiều thời gian và công sức hơn.

Để làm giảm lượng nhân lực trong việc công tác quản lý, kiểm soát phương tiện giao thông, trên thế giới đã nhanh chóng xây dựng hệ thống giám sát tự động đối với các phương tiện giao thông. Các hệ thống giám sát đều lấy biển số xe là mục tiêu giám sát. Hệ thống này đã được sử dụng rộng rãi tuy nhiên ở Việt Nam đây vẫn là một lĩnh vực mới mẻ. Do đó em chọn làm đề tài **“nghiên cứu nhận dạng biển số xe ô tô Cộng hòa dân chủ nhân dân Lào”** để giải quyết vấn đề đặt ra.

Cụ thể, báo cáo sẽ tập trung giải quyết các vấn đề sau:

- Nghiên cứu lý thuyết về xử lý ảnh số
- Nghiên cứu một số kỹ thuật xử lý ảnh trong ứng dụng tự động nhận dạng biển số xe trong ảnh chụp từ camera.
- Tìm hiểu về thư viện OpenCV – thư viện hỗ trợ cho việc viết chương trình.
- Demo chương trình nhận dạng biển số xe.

Nội dung chính của báo cáo ngoài phần mở đầu, tài liệu tham khảo, mục lục được trình bày trong 3 phần chính:

- Chương I: Tổng quan về xử lý ảnh và bài toán nhận dạng biển số xe ở ô tô Cộng hòa dân chủ nhân dân Lào.

•Chương II: Một số vấn đề trong nhận dạng ảnh biển số xe ô tô Cộng hòa dân chủ nhân dân Lào.

•Chương III: Chương trình nhận dạng biển số xe ô tô Cộng hòa dân chủ nhân dân Lào.

Mặc dù có nhiều cố gắng nhưng do hạn chế về thời gian cũng như kinh nghiệm nên báo cáo này chắc chắn còn nhiều thiếu sót, rất mong nhận được sự góp ý của các thầy cô và các bạn để ứng dụng mà báo cáo đề cập được hoàn thiện hơn.

Thái Nguyên, ngày 24 tháng 8 năm 2016

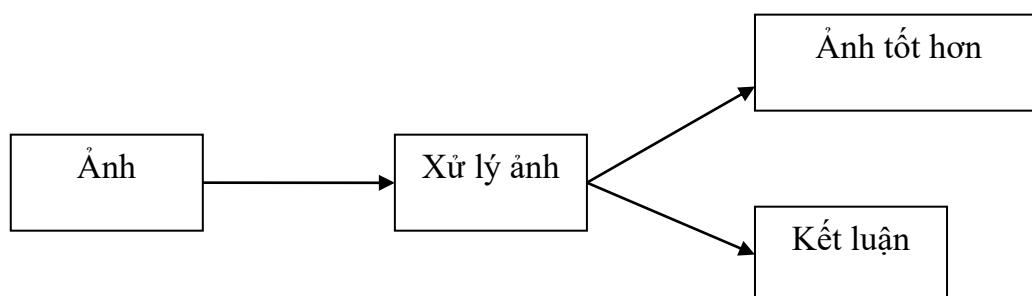
Học viên

Phonevilay KHAMPHOUMY

CHƯƠNG I: TỔNG QUAN VỀ XỬ LÝ ẢNH VÀ BÀI TOÁN NHẬN DẠNG BIỂN SỐ XE Ô TÔ Ở CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO

1.1 TỔNG QUAN VỀ XỬ LÝ ẢNH

Quá trình xử lý nhận dạng ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc là một kết luận.



Hình 1.1. Quá trình xử lý ảnh

1.1.1 Một số khái niệm cơ bản trong xử lý ảnh

Như đã đề cập trong phần giới thiệu, chúng ta đã thấy được một cách khái quát các vấn đề chính trong xử lý ảnh. Để hiểu chi tiết hơn, trước tiên ta xem xét các khái niệm (thuật ngữ) thường dùng trong xử lý ảnh đó là phần tử ảnh (pixel), độ phân giải của ảnh (resolution), ảnh số (digitize image) và mức xám (grey level)...

1.1.1.1 Phần tử ảnh (Pixel (Picture Element))

Ảnh trong thực tế là một ảnh liên tục về không gian và về giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính cần thiết phải tiến hành số hoá ảnh. Trong quá trình số hoá, người ta biến đổi tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hoá về không gian) và lượng hoá thành phần giá trị mà về nguyên tắc mắt thường không phân biệt được hai điểm kề nhau. Trong quá trình này, người ta sử dụng khái niệm Picture element mà ta quen gọi hay viết là Pixel – phần tử ảnh. Ở đây cũng cần phân biệt khái niệm pixel hay đề cập đến trong các hệ

thống đồ hoạ máy tính. Để tránh nhầm lẫn ta tạm gọi khái niệm pixel này là pixel thiết bị. Khái niệm pixel thiết bị có thể xem xét như sau: khi ta quan sát màn hình (trong chế độ đồ hoạ), màn hình không liên tục mà gồm nhiều điểm nhỏ, gọi là pixel. Mỗi pixel gồm một cặp tọa độ x, y và màu.

Cặp tọa độ x, y tạo nên độ phân giải (resolution). Màn hình máy tính có nhiều loại với độ phân giải khác nhau: màn hình CGA có độ phân giải là 320x200; màn hình VGA là 640x350,...

Như vậy, một ảnh là một tập hợp các điểm ảnh. Khi được số hoá, nó thường được biểu diễn bởi bảng hai chiều $I(n,p)$: n dòng và p cột. Ta nói ảnh gồm $n \times p$ pixels. Người ta thường ký hiệu $I(x,y)$ để chỉ một pixel. Một pixel có thể lưu trữ trên 1, 4, 8 hay 24 bit.

1.1.1.2 Độ phân giải của ảnh (Resolution)

Độ phân giải (Resolution) của ảnh là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị. Trong đó khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bố, đó chính là độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

Độ phân giải của ảnh trên màn hình CGA (Color Graphic Adaptor) là một lưới điểm theo chiều ngang màn hình: 300 điểm chiều dọc * 200 điểm ảnh (320*200). Rõ ràng, cùng màn hình CGA 12'' ta nhận thấy mịn hơn màn hình CGA 17'' độ phân giải 320*200. Điều này do cùng một mật độ (độ phân giải) nhưng diện tích màn hình rộng hơn thì độ mịn (liên tục của các điểm) kém hơn.

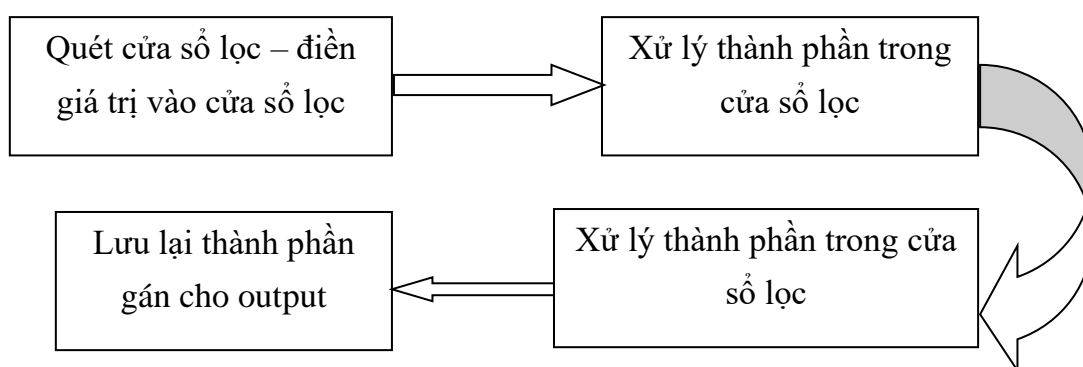
1.1.1.3 Mức xám (Gray Level)

Mức xám là kết quả sự mã hoá tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số - kết quả của quá trình lượng hoá. Cách mã hoá kinh điển thường dùng 16, 32 hay 64 mức. Mã hoá 256 mức là phổ dụng nhất do lý do kỹ thuật. Vì $2^8 = 256$ (0,1,...255), nên với 256 mức, mỗi pixel sẽ được mã hoá bởi 8 bit.

1.1.1.4 Sử dụng các bộ lọc

❖ Kỹ thuật lọc trung vị

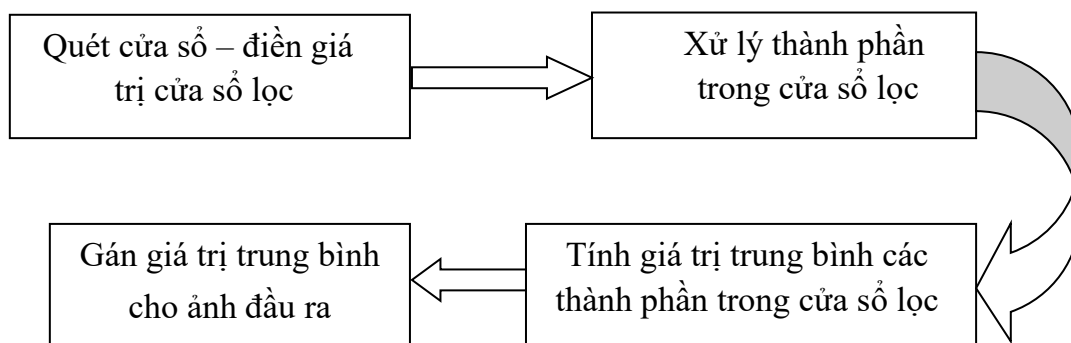
Ý tưởng: sử dụng một cửa sổ lọc (ma trận 3x3) quét qua lần lượt từng điểm ảnh của ảnh đầu vào input. Tại vị trí mỗi điểm ảnh lấy giá trị của các điểm ảnh tương ứng trong vùng 3x3 của ảnh gốc lắp vào ma trận lọc, sau đó sắp xếp các điểm ảnh trong cửa sổ này theo thứ tự tăng dần, gán điểm ảnh nằm chính giữa (trung vị) của dãy giá trị điểm ảnh đã được sắp xếp ở trên cho giá trị điểm ảnh đang xét của ảnh đầu ra output. Sơ lược ý tưởng trên:



Hình 1.2 Kỹ thuật lọc trung vị

❖ Kỹ thuật lọc trung bình

Ý tưởng: Tương tự như kỹ thuật lọc trung vị, sử dụng một cửa sổ lọc (ma trận 3x3) quét qua lần lượt từng điểm ảnh của ảnh đầu vào input. Tại vị trí mỗi điểm ảnh lấy giá trị của các điểm ảnh tương ứng trong vùng 3x3 của ảnh gốc lắp vào ma trận lọc. Với hai bước: tính tổng các thành phần trong cửa sổ lọc và sau đó lấy tổng chia cho các phần tử của cửa sổ lọc. Sơ lược ý tưởng trên:



Hình 1.3 Kỹ thuật lọc trung bình

1.1.2 Quan hệ giữa các điểm ảnh

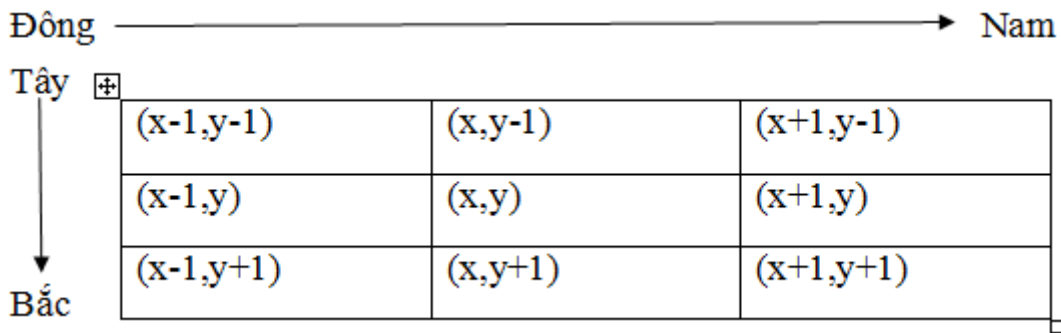
Một ảnh số giả sử được biểu diễn bằng hàm $f(x,y)$. Tập con các điểm ảnh là S ; cặp điểm ảnh có quan hệ với nhau ký hiệu là p,q . Chúng ta nêu một số các khái niệm sau.

1.1.2.1 Các lân cận của điểm ảnh (Image Neighbors)

Giả sử có điểm ảnh p tại tọa độ (x,y) . p có 4 điểm lân cận gần nhất theo chiều đứng và ngang (có thể coi như lân cận 4 hướng chính: Đông, Tây, Nam, Bắc).

$$\{(x-1,y),(x,y-1),(x,y+1),(x+1,y)\} = N_4(p)$$

Trong đó: Số 1 là giá trị logic; $N_4(p)$ tập 4 điểm lân cận của p .



Các lân cận chéo: Các điểm lân cận chéo $N_p(P)$ (Có thể coi lân cận chéo là 4 hướng: Đông – Nam, Đông – Bắc, Tây – Nam, Tây – Bắc)

$$N_p(P) = \{(x-1,y-1), (x+1,y-1), (x-1,y+1), (x+1,y+1)\}$$

Tập kết hợp: $N_8(p) = N_4(P) + N_p(P)$ là tập hợp 8 điểm lân cận của điểm ảnh p .

Chú ý: Nếu (x,y) nằm ở biên (mép) ảnh; một số điểm sẽ nằm ngoài ảnh.

1.1.2.2 Các mối liên kết điểm ảnh

Các mối liên kết được sử dụng để xác định giới hạn (Boundaries) của đối tượng vật thể hoặc xác định vùng trong một ảnh. Một liên kết được đặc trưng bởi tính liên hệ giữa các điểm và mức xám của chúng.

Giả sử V là tập các giá trị mức xám. Một ảnh có các giá trị cường độ sáng từ thang mức xám từ 32 đến 64 được mô tả như sau:

$$V = \{32, 33, \dots, 64\}.$$

Có 3 loại liên kết:

a) Liên kết 4: Hai điểm ảnh p và q được nói là liên kết 4 với các giá trị cường độ sáng V nếu q nằm trong một các lân cận của p , tức q thuộc $N_4(P)$.

b) Liên kết 8: Hai điểm ảnh p và q nằm trong một các lân cận 8 của p , tức q thuộc $N_8(P)$.

c) Liên kết m (liên kết hỗn hợp): Hai điểm ảnh p và q với các giá trị cường độ sáng V được nói là liên kết m nếu:

q thuộc $N_4(P)$ hoặc q thuộc $N_8(P)$.

1.1.2.3 Đo khoảng cách giữa các điểm ảnh

Khoảng cách $D(p,q)$ giữa hai điểm ảnh p tọa độ (x, y) , q tọa độ (s, t) là hàm khoảng cách (Distance) hoặc Metric nếu:

a) $D(p,q) \geq 0$ (với $D(p,q) = 0$ nếu và chỉ nếu $p = q$)

b) $D(p,q) = D(q,p)$

c) $D(p,z) \leq D(p,q) + D(q,z)$; z là một điểm ảnh khác.

Khoảng cách Euclide: Khoảng cách Euclide giữa hai điểm ảnh $p(x, y)$ và $q(s, t)$ được định nghĩa như sau:

$$D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$$

Khoảng cách khối: Khoảng cách $D_4(p,q)$ được gọi là khoảng cách khối đô thị (City – Block Distance) và được xác định như sau:

$$D_4(p,q) = |x - s| + |y - t|$$

Giá trị khoảng cách giữa các điểm ảnh r : giá trị bán kính r giữa điểm ảnh từ tâm điểm ảnh đến tâm điểm ảnh q khác. Ví dụ: Màn hình CGA 12” (12” * 2,54 cm = 30,48 cm = 304,8mm) độ phân giải 320 * 200; tỷ lệ 4/3 (chiều dài/chiều rộng). Theo định lý Pitago về tam giác vuông, đường chéo sẽ lấy tỷ lệ 5 phần (5/4/3: đường chéo/chiều dài/chiều rộng màn hình); Khi đó độ dài thật là (305/244/183) chiều rộng màn hình 183mm ứng với màn hình CGA 200 điểm ảnh theo chiều dọc.

Như vậy, khoảng cách điểm ảnh lân cận của CGA 12” là khoảng 1mm.

Khoảng cách $D_8(p,q)$ còn gọi là khoảng cách bàn cờ (Ches – Board Distance) giữa điểm ảnh p,q được xác định như sau:

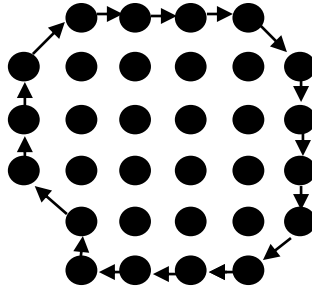
$$D_8(p,q) = \max(|x - s|, |y - t|)$$

1.1.3 Chu tuyến của một đối tượng ảnh

1.1.3.1 Định nghĩa Chu tuyến

Chu tuyến của một đối tượng ảnh là dãy các điểm của đối tượng ảnh P_1, \dots, P_n sao cho P_i và P_{i+1} là các 8-láng giềng của nhau ($i=1, \dots, n-1$) và P_1 là 8-láng giềng của P_n , $\forall i \exists Q$ không thuộc đối tượng ảnh và Q là 4-láng giềng của P_i (hay nói cách khác $\forall i$ thì P_i là biên 4). Kí hiệu $\langle P_1 P_2 \dots P_n \rangle$.

Tổng các khoảng cách giữa hai điểm kế tiếp của chu tuyến là độ dài của chu tuyến và kí hiệu $Len(C)$ và hướng $P_i P_{i+1}$ là hướng chẵn nếu P_i và P_{i+1} là các 4 – láng giềng (trường hợp còn lại thì $P_i P_{i+1}$ là hướng lẻ). Trong hình dưới đây biểu diễn chu tuyến của ảnh, trong đó, P là điểm khởi đầu chu tuyến.



Hình 1.4 Ví dụ về chu tuyến của đối tượng ảnh

1.1.3.2 Định nghĩa Chu tuyến đối ngẫu

Hai chu tuyến $C = \langle P_1 P_2 \dots P_n \rangle$ và $C^\perp = \langle Q_1 Q_2 \dots Q_m \rangle$ được gọi là đối ngẫu của nhau nếu và chỉ nếu $\forall i \exists j$ sao cho:

- P_i và Q_j là 4-láng giềng của nhau.
- Các điểm P_i là vùng thì Q_j là nền và ngược lại.

1.1.3.3 Định nghĩa Chu tuyến ngoài

Chu tuyến C được gọi là chu tuyến ngoài (Hình 1.3a) nếu và chỉ nếu

- Chu tuyến đối ngẫu C^\perp là chu tuyến của các điểm nền

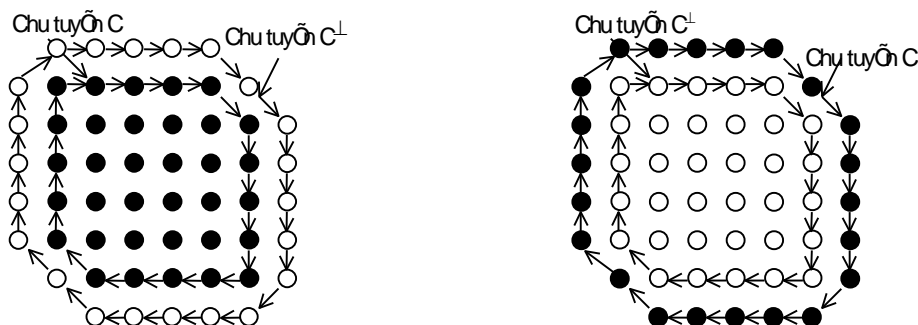
b) Độ dài của C nhỏ hơn độ dài C^\perp

1.1.3.4 Định nghĩa Chu tuyến trong

Chu tuyến C được gọi là chu tuyến trong (Hình 1.3b) nếu và chỉ nếu:

a) Chu tuyến đối ngẫu C^\perp là chu tuyến của các điểm nền

b) Độ dài của C lớn hơn độ dài C^\perp



a) Chu tuyến ngoài

b) Chu tuyến trong

Hình 1.5 Chu tuyến trong, chu tuyến ngoài

1.1.3.5 Định nghĩa điểm trong và điểm ngoài chu tuyến

Giả sử $C = \langle P_1 P_2 \dots P_n \rangle$ là chu tuyến của một đối tượng ảnh và P là một điểm ảnh. Khi đó:

a) Nếu nửa đường thẳng xuất phát từ P sẽ cắt chu tuyến C tại số lẻ lần, thì P được gọi là điểm trong chu tuyến C và kí hiệu $in(P, C)$

b) Nếu $P \notin C$ và P không phải là điểm trong của C , thì P được gọi là điểm ngoài chu tuyến C và kí hiệu $out(P, C)$.

1.1.3.6 Bổ đề Chu tuyến đối ngẫu

Giả sử $E \subseteq \mathfrak{S}$ là một đối tượng ảnh và $C = \langle P_1 P_2 \dots P_n \rangle$ là chu tuyến của E , $C^\perp = \langle Q_1 Q_2 \dots Q_m \rangle$ là chu tuyến đối ngẫu tương ứng. Khi đó:

a) Nếu C là chu tuyến trong thì $in(Q_i, C) \forall i (i=1, \dots, m)$

b) Nếu C là chu tuyến ngoài thì $in(P_i, C^\perp) \forall i (i=1, \dots, n)$

1.1.3.7 Bổ đề Phần trong/ngoài của chu tuyến

Giả sử $E \subseteq \mathfrak{S}$ là một đối tượng ảnh và C là chu tuyến của E . Khi đó:

a) Nếu C là chu tuyến ngoài thì $\forall x \in E$ sao cho $x \notin C$, ta có $in(x, C)$

b) Nếu C là chu tuyến trong thì $\forall x \in E$ sao cho $x \notin C$, ta có $\text{out}(x, C)$

1.1.3.8 Định lý về tính duy nhất của chu tuyến ngoài

Giả sử $E \subseteq \mathfrak{S}$ là một đối tượng ảnh và C_E là chu tuyến ngoài của E . Khi đó C_E là duy nhất.

1.1.4 Biên và các phương pháp phát hiện biên

Biên là vấn đề quan trọng trong trích chọn đặc điểm nhằm tiến tới hiểu ảnh. Cho đến nay chưa có định nghĩa chính xác về biên, trong mỗi ứng dụng người ta đưa ra các độ đo khác nhau về biên, một trong các độ đo đó là độ đo về sự thay đổi đột ngột về cấp xám.

Ví dụ: Đối với ảnh đen trắng, một điểm được gọi là điểm biên nếu nó là điểm đen có ít nhất một điểm trắng bên cạnh.

Tập hợp các điểm biên tạo nên biên hay đường biên của đối tượng.

Xuất phát từ cơ sở này người ta sử dụng hai phương pháp phát hiện biên cơ bản:

Phát hiện biên trực tiếp: Phương pháp này làm nổi biên dựa vào sự biến thiên mức xám của ảnh. Kỹ thuật chủ yếu dùng để phát hiện biên ở đây là kỹ thuật lấy đạo hàm. Nếu lấy đạo hàm bậc nhất của ảnh ta có các kỹ thuật Gradient, nếu lấy đạo hàm bậc hai của ảnh ta có các kỹ thuật Laplace. Ngoài ra còn có một số các tiếp cận khác.

Phát hiện biên gián tiếp: Nếu bằng cách nào đó ta phân được ảnh thành các vùng thì ranh giới giữa các vùng đó gọi là biên. Kỹ thuật dò biên và phân vùng ảnh là hai bài toán đối ngẫu nhau vì dò biên để thực hiện phân lớp đối tượng. Khi đã phân lớp xong nghĩa là đã phân vùng được ảnh. Và ngược lại, khi đã phân vùng ảnh xong nghĩa là đã phân lớp thành các đối tượng, do đó có thể phát hiện được biên.

1.1.4.1 Các Phương pháp phát hiện biên trực tiếp

Kỹ thuật phát hiện biên Gradient

Theo định nghĩa, gradient là một vectơ có các thành phần biểu thị tốc độ thay đổi giá trị của điểm ảnh, ta có:

$$\begin{cases} \frac{\partial f(x, y)}{\partial x} = f_x \approx \frac{f(x+dx, y) - f(x, y)}{dx} \\ \frac{\partial f(x, y)}{\partial y} = f_y \approx \frac{f(x, y+dy) - f(x, y)}{dy} \end{cases}$$

Trong đó, dx, dy là khoảng cách (tính bằng số điểm) theo hướng x và y.

*** Nhận xét:**

Tuy ta nói là lấy đạo hàm nhưng thực chất chỉ là mô phỏng và xấp xỉ đạo hàm bằng các kỹ thuật nhân chập vì ảnh số là tín hiệu rời rạc nên đạo hàm không tồn tại.

Ví dụ: Với $dx = dy = 1$, ta có:

$$\begin{cases} \frac{\partial f}{\partial x} \approx f(x+1, y) - f(x, y) \\ \frac{\partial f}{\partial y} \approx f(x, y+1) - f(x, y) \end{cases}$$

Do đó, mặt nạ nhân chập theo hướng x là: $A = \begin{pmatrix} -1 & 1 \end{pmatrix}$ và hướng y là $B = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

Chẳng hạn:

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 \end{pmatrix}$$

Ta có:

$$I \otimes A = \begin{pmatrix} 0 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ * & * & * & * \end{pmatrix}; I \otimes B = \begin{pmatrix} 0 & 3 & 3 & * \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & * \\ * & * & * & * \end{pmatrix}$$

$$I \otimes A + I \otimes B = \begin{pmatrix} 0 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ 3 & 0 & 0 & * \\ * & * & * & * \end{pmatrix}$$

Kỹ thuật tìm tách ngưỡng tự động

Ngưỡng θ trong kỹ thuật tách ngưỡng thường được cho bởi người sử dụng. Kỹ thuật tìm tách ngưỡng tự động nhằm tìm ra ngưỡng θ một cách tự động dựa vào histogram theo nguyên lý trong vật lý là vật thể tách làm 2 phần nếu tổng độ lệch trong từng phần là tối thiểu.

Giả sử, ta có ảnh $I \sim$ kích thước $m \times n$

$G \sim$ là số mức xám của ảnh kể cả khuyết thiếu

$t(g) \sim$ số điểm ảnh có mức xám $\leq g$

$$m(g) = \frac{1}{t(g)} \sum_{i=0}^g i \cdot h(i) \sim \text{mômen quán tính TB có mức xám } \leq g$$

Hàm $f: g \mapsto f(g)$

$$f(g) = \frac{t(g)}{mxn - t(g)} [m(g) - m(G-1)]^2$$

Tìm θ sao cho:

$$f(\theta) = \max_{0 \leq g < G-1} \{f(g)\}$$

Ví dụ: Tìm ngưỡng tự động của ảnh sau

$$I = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Lập bảng

gh(g)	t(g)	g.h(g)	$\sum_{i=0}^g ih(i)$	m(g)	f(g)
01515	0	0	0	1.35	
15 20	5	5	0,25	1.66	
24 24	8	13	0,54	1.54	
33 27	9	22	0,81	1.10	
42 29	8	30	1,03	0.49	
51 30	5	35	1,16	∞	

Ngưỡng cần tách $\theta=1$ ứng với $f(\theta)=1.66$

Kỹ thuật Prewitt

Kỹ thuật sử dụng 2 mặt nạ nhập chấp xấp xỉ đạo hàm theo 2 hướng x và y là:

$$H_x = \begin{matrix} -1 & \begin{pmatrix} 0 & 1 \\ -1 & 0 & 1 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

$$H_y = \begin{matrix} -1 & \begin{pmatrix} -1 & -1 \\ 0 & 0 & 0 \\ 1 \end{pmatrix} \end{matrix}$$

Các bước tính toán của kỹ thuật Prewitt

+ **Bước 1:** Tính $I \otimes H_x$ và $I \otimes H_y$

+ **Bước 2:** Tính $I \otimes H_x + I \otimes H_y$

Ví dụ:

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$I \otimes H_x = \begin{pmatrix} 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ 0 & 0 & -10 & -10 & * & * \\ 0 & 0 & -5 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

$$I \otimes H_y = \begin{pmatrix} 15 & 15 & 10 & 5 & * & * \\ 0 & 0 & 0 & 0 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ -15 & -15 & -10 & -5 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

$$I \otimes H_x + I \otimes H_y = \begin{pmatrix} 15 & 15 & 0 & -5 & * & * \\ 0 & 0 & -15 & -15 & * & * \\ -15 & -15 & -20 & -15 & * & * \\ -15 & -15 & -15 & -10 & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix}$$

Kỹ thuật Sobel

Tương tự như kỹ thuật Prewitt kỹ thuật Sobel sử dụng 2 mặt nạ nhân chập theo 2 hướng x, y là:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$H_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Các bước tính toán tương tự Prewitt

+ **Bước 1:** Tính $I \otimes H_x$ và $I \otimes H_y$

+ **Bước 2:** Tính $I \otimes H_x + I \otimes H_y$

Kỹ thuật Canny

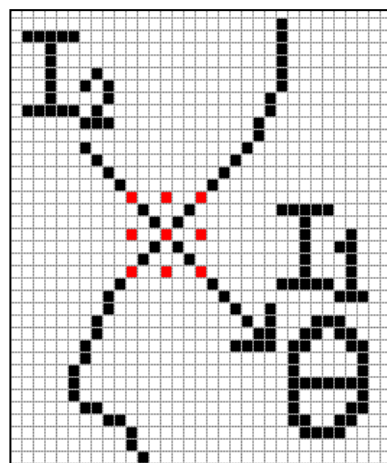
Đây là một thuật toán tương đối tốt, có khả năng đưa ra đường biên mảnh, và phát hiện chính xác điểm biên với điểm nhiễu.

Thuật toán

+ **Bước 1:** Làm trơn ảnh

Tính $I \otimes H$, với:

$$H = \frac{1}{115} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$



Gọi G là kết quả lọc nhiễu: $G = I \otimes H$

+ **Bước 2:** Tính gradient của ảnh bằng mặt nạ Prewitt, kết quả đặt vào G_x, G_y .

$$G_x = G \otimes H_x, G_y = G \otimes H_y$$

+ **Bước 3:** Tính gradient hướng tại mỗi điểm (i,j) của ảnh. Hướng này sẽ được nguyên hóa để nằm trong 8 hướng $[0..7]$, tương đương với 8 lân cận của một điểm ảnh.

+ **Bước 4:** Dùng ràng buộc “loại bỏ những điểm không phải là cực đại” để xóa bỏ những điểm không phải là biên. Xét (i,j) , θ là gradient hướng tại (i,j) . I_1, I_2 là hai điểm lân cận của (i,j) theo hướng θ . Theo định nghĩa điểm biên cực bộ thì (i,j) là biên nếu $I(i,j)$ cực đại địa phương theo hướng gradient \rightarrow Nếu $I(i,j) > I_1$ và $I(i,j) > I_2$ thì mới giữ lại $I(i,j)$, ngược lại xóa $I(i,j)$ về điểm ảnh nền.

+ **Bước 5:** Phân ngưỡng. Với các điểm được giữ lại, thực hiện lấy ngưỡng gradient biên độ lần cuối để xác định các điểm biên thực sự.

Kỹ thuật la bàn

Kỹ thuật sử dụng 8 mặt nạ nhân chập theo 8 hướng $0^0, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, 315^0$

$$\begin{aligned} H_1 &= \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} & H_2 &= \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \\ H_3 &= \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} & H_4 &= \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \\ H_5 &= \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix} & H_6 &= \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \\ H_7 &= \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} & H_8 &= \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \end{aligned}$$

Các bước tính toán thuật toán La bàn

+ **Bước 1:** Tính $I \otimes H_i$; $i = 1, 8$

+ **Bước 2:** $\sum_{i=1}^8 I \otimes H_i$

Kỹ thuật phát hiện biên Laplace

Các phương pháp đánh giá gradient ở trên làm việc khá tốt khi mà độ sáng thay đổi rõ nét. Khi mức xám thay đổi chậm, miền chuyển tiếp trải rộng, phương pháp cho hiệu quả hơn đó là phương pháp sử dụng đạo hàm bậc hai Laplace.

Toán tử Laplace được định nghĩa như sau:

Ta có:

$$\begin{aligned} \nabla^2 f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ \frac{\partial^2 f}{\partial x^2} &= \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) \approx \frac{\partial}{\partial x} (f(x+1, y) - f(x, y)) \end{aligned}$$

$$\begin{aligned} &\approx [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)] \\ &\approx f(x+1, y) - 2f(x, y) + f(x-1, y) \end{aligned}$$

Tương tự,

$$\begin{aligned} \frac{\partial^2 f}{\partial y^2} &= \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right) \approx \frac{\partial}{\partial y} (f(x, y+1) - f(x, y)) \\ &\approx [f(x, y+1) - f(x, y)] - [f(x, y) - f(x, y-1)] \\ &\approx f(x, y+1) - 2f(x, y) + f(x, y-1) \end{aligned}$$

$$\text{Vậy: } \nabla^2 f = f(x+1, y) + f(x, y+1) - 4f(x, y) + f(x-1, y) + f(x, y-1)$$

Dẫn tới:

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Trong thực tế, người ta thường dùng nhiều kiểu mặt nạ khác nhau để xấp xỉ rời rạc đạo hàm bậc hai Laplace. Dưới đây là ba kiểu mặt nạ thường dùng:

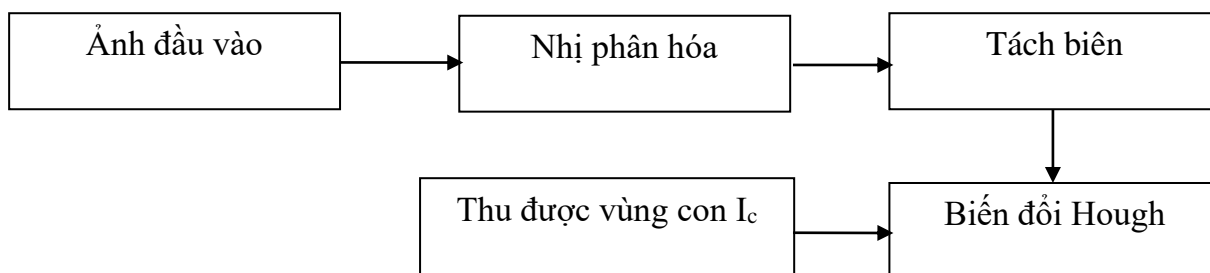
$$\mathbf{H}_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \mathbf{H}_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \mathbf{H}_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

$$\text{VD: } \mathbf{I} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 5 & 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

1.1.4.2 Phát hiện vùng chứa biển số xe

Sơ đồ các bước được mô tả trong hình dưới:

Ảnh đầu vào: là một ảnh có 256 mức xám, được nhị phân hóa thành ảnh nhị phân. Mục đích của giai đoạn nhị phân hóa ảnh là nhằm làm nổi bật vùng biển số xe. Khi ta tách biên, vùng bao của biển số xe sẽ hiện lên rõ ràng. Sau đó dùng phương pháp phát hiện biên để có được biên dọc vào ngang của ảnh. Kết quả của công đoạn này, ảnh thu được là ảnh nhị phân chỉ chứa các cạnh dọc và ngang. Thực hiện biến đổi Hough cho các đoạn biên vừa lấy được và xác định các đoạn thẳng đi qua tập các điểm biên của mỗi biên, kết quả sẽ là các đoạn thẳng ngang và dọc. Giao của những đoạn thẳng này sẽ cho ra vùng con I_c



Hình 1.6 Sơ đồ giải quyết

1.1.4.3 Phương pháp phát hiện biên gián tiếp

+ Thuật toán dò biên tổng quát

Biểu diễn đối tượng ảnh theo chu tuyến thường dựa trên các kỹ thuật dò biên. Có hai kỹ thuật dò biên cơ bản. Kỹ thuật thứ nhất xét ảnh biên thu được từ ảnh vùng sau một lần duyệt như một đồ thị, sau đó áp dụng các thuật toán duyệt cạnh đồ thị. Kỹ thuật thứ hai dựa trên ảnh vùng, kết hợp đồng thời quá trình dò biên và tách biên. Ở đây ta quan tâm cách tiếp cận thứ hai.

Trước hết, giả sử ảnh được xét chỉ bao gồm một vùng ảnh 8-liên thông \mathfrak{I} , được bao bọc bởi một vành đai các điểm nền. Dễ thấy \mathfrak{I} là một vùng 4-liên thông chỉ là một trường riêng của trường hợp trên.

Về cơ bản, các thuật toán dò biên trên một vùng đều bao gồm các bước sau:

- Xác định điểm biên xuất phát

- Dự báo và xác định điểm biên tiếp theo
- Lặp bước 2 cho đến khi gặp điểm xuất phát

Do xuất phát từ những tiêu chuẩn và định nghĩa khác nhau về điểm biên, và quan hệ liên thông, các thuật toán dò biên cho ta các đường biên mang các sắc thái rất khác nhau.

Kết quả tác động của toán tử dò biên lên một điểm biên r_i là điểm biên r_{i+1} (8-láng giềng của r_i). Thông thường các toán tử này được xây dựng như một hàm đại số Boolean trên các 8-láng giềng của r_i . Mỗi cách xây dựng các toán tử đều phụ thuộc vào định nghĩa quan hệ liên thông và điểm biên. Do đó sẽ gây khó khăn cho việc khảo sát các tính chất của đường biên. Ngoài ra, vì mỗi bước dò biên đều phải kiểm tra tất cả các 8-láng giềng của mỗi điểm nên thuật toán thường kém hiệu quả. Để khắc phục các hạn chế trên, thay vì sử dụng một điểm biên ta sử dụng cặp điểm biên (một thuộc \mathfrak{S} , một thuộc $\overline{\mathfrak{S}}$), các cặp điểm này tạo nên tập nền vùng, kí hiệu là NV và phân tích toán tử dò biên thành 2 bước:

- Xác định cặp điểm nền vùng tiếp theo.
- Lựa chọn điểm biên

Trong đó bước thứ nhất thực hiện chức năng của một ánh xạ trên tập NV lên NV và bước thứ hai thực hiện chức năng chọn điểm biên.

Thuật toán dò biên tổng quát

Bước 1: Xác định cặp nền-vùng xuất phát

Bước 2: Xác định cặp nền-vùng tiếp theo

Bước 3: Lựa chọn điểm biên vùng

Bước 4: Nếu gặp lại cặp xuất phát thì dừng, nếu không quay lại bước 2.

1. Xác định cặp nền- vùng xuất phát

Việc xác định cặp nền vùng xuất phát được thực hiện bằng cách duyệt ảnh lần lượt từ trên xuống dưới, từ trái sang phải rồi kiểm tra điều kiện lựa chọn cặp ảnh nền vùng.

2. Xác định cặp nền – vùng tiếp theo

Ta xác định cặp nền – vùng tiếp theo bằng toán tử dò biên. Toán tử dò biên được định nghĩa như sau :

T là một ánh xạ .

$$T : NV \rightarrow NV$$

$$(b,r) \rightarrow (b',r')$$

Ta gọi T là toán tử dò biên cơ sở nếu nó thỏa mãn điều kiện b', r' là các 8-láng giềng của r .

3. Lựa chọn điểm biên :

Giả sử $(b,r) \in NV$; gọi $K(b,r)$ là hàm chọn điểm biên. Biên của một dạng \mathfrak{S} có thể định nghĩa theo một trong ba cách

- Tập những điểm thuộc \mathfrak{S} có mặt trên NV, tức là $K(b,r) = r$
- Tập những điểm thuộc phủ \mathfrak{S} có trên NV, tức là $K(b,r) = b$
- Tập những điểm ảo nằm giữa cặp nền-vùng, tức là $K(b,r)$ là những điểm nằm giữa hai điểm b và r .

Cách định nghĩa thứ ba tương ứng với mỗi cặp nền - vùng với một điểm biên. Còn với cách thứ nhất và thứ hai, một số cặp nền – vùng có thể có chung một điểm biên.

Bởi vậy, quá trình chọn điểm biên được thực hiện như sau:

```
i=1;(bi,ri)=(b0,r0);
While (K(bi,ri) != K(bn,rn) & i<=8 )
{
    (bi+1,ri+1) = T(bi,ri);
    i++;
}
```

Điều kiện dừng :

Cặp nền-vùng thứ n trùng với cặp nền vùng xuất phát $(b_n, r_n) = (b_0, r_0)$

1.1.4.4 Biểu diễn đường biên

Với các kỹ thuật dò biên ở trên, ta thu được đường biên của đối tượng. Đường biên là một chu tuyến (tuyến đóng) $P = [g_0, g_1, \dots, g_n]$. Có nhiều cách để mô tả chu tuyến theo các ứng dụng khác nhau, có thể kể đến như phương pháp biến đổi Fourier, sử dụng phép biến đổi wavelet, sử dụng phép biến đổi khoảng cách, khoảng cách Hausdorff, dùng mô men biên...

a). Thể hiện đường biên

Giả sử $P = [g_0, g_1, \dots, g_n]$ là đường biên của đối tượng. Để mô tả P, trước hết ta cần thể hiện P bằng một dãy số thực hoặc số phức $u_k, k = 0..n$.

- Mô tả dạng tọa độ phức: Sử dụng các tọa độ của các điểm trên đường biên.

$$u_k = x_k + iy_k; k = \overline{0, n}, i \text{ là đơn vị phức } i^2 = -1.$$

- Mô tả dạng khoảng cách đến trọng tâm: sử dụng hàm khoảng cách từ các điểm trên đường biên đến trọng tâm.

$$u_k = \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2}, k = \overline{0, n}$$

Trong đó:

$$x_c = \frac{1}{N} \sum_{k=0}^{N-1} x_k, y_c = \frac{1}{N} \sum_{k=0}^{N-1} y_k$$

- Mô tả diện tích: Trong dạng mô tả này, ta dùng diện tích của tam giác có một đỉnh là trọng tâm, còn 2 đỉnh kia là 2 điểm nằm kề nhau trên đường biên.

$$u_k = S_{\Delta C g_k g_{k+1}} = \frac{1}{2} |x_k y_{k+1} - x_{k+1} y_k|, k = \overline{0, n-1}$$

Ngoài ra còn một số dạng mô tả khác như hàm về góc tích lũy, hàm tín hiệu độ cong

b). Mô tả đường biên sử dụng phép biến đổi Fourier

Giả sử P được thể hiện bằng một dãy số phức $u_k, u_k = x_k + iy_k; k = \overline{0, n}$. Khi đó phép biến đổi Fourier rời rạc lên dãy u_k được cho bởi:

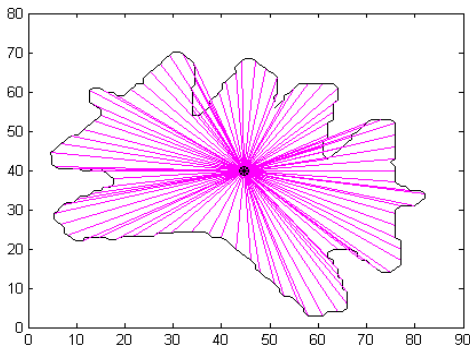
$$F_m = \sum_{k=0}^N u_k e^{-i2\pi km/(N+1)}, m = 0, 1, \dots, N$$

Khi đó, P hoàn toàn có thể được mô tả bởi bộ các hệ số Fourier: $F = (F_0, F_1, F_2, \dots, F_N)$, trong đó F_0 thể hiện trọng tâm của P , F_1 và F_{n-1} thể hiện elip chính, từ đó cho ta kích thước và góc nghiêng của P . Ngoài ra, các hệ số có ý nghĩa nhất được tập trung ở hai đầu của dãy F . Các tính chất này cho phép ta dễ dàng chuẩn hóa theo nghĩa bất biến đồng dạng và trích chọn ra bộ hệ số Fourier làm đặc trưng cho đường biên.

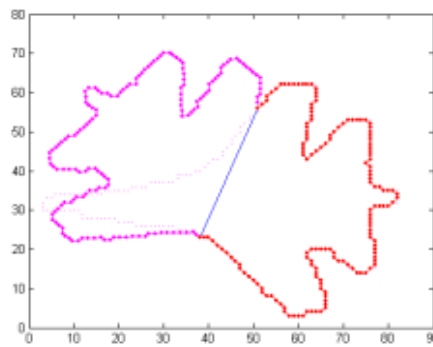
Ta có thể thu được các tọa độ phức $(u_k, k = \overline{0, N})$ từ phép biến đổi Fourier ngược:

$$u_k = \frac{1}{N+1} \sum_{m=0}^N F_m e^{i2\pi km/(N+1)}, k = 0, 1, \dots, N$$

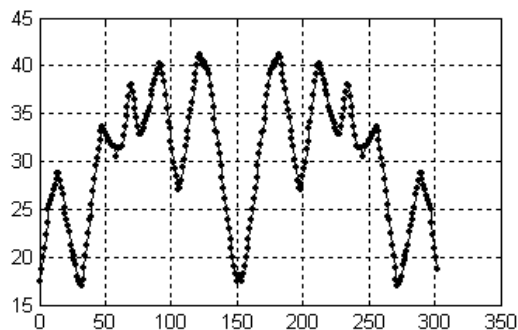
Các thử nghiệm thực tế đã chỉ ra rằng, các thể hiện tốt nhất sử dụng cho phép biến đổi Fourier trong mô tả chu tuyến là các thể hiện khoảng cách đến trọng tâm, tọa độ phức.



Hình 1.7 Biểu diễn đường biên



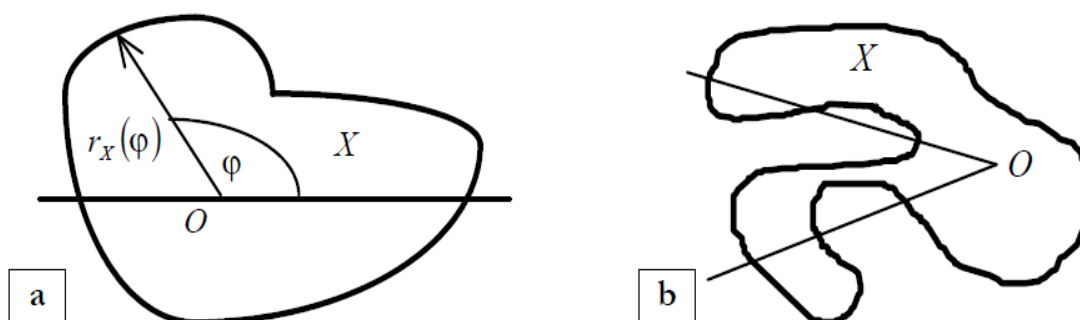
Hình 1.8 Khoảng cách đến trọng tâm



Hình 1.9 Thể hiện khoảng cách

1.1.4.5 Biểu diễn đường biên bằng hàm Radius – vector

Thường thì các đường biên của hình được mô tả bởi các hàm radius – vector được định nghĩa theo cách sau: một điểm tham chiếu O nằm trong hình X được lựa chọn, nó thường là điểm trung tâm của hình X và thường là trung tâm của lực hấp dẫn, hoặc là trung tâm của đĩa nhỏ nhất mà chứa hoàn toàn hình hoặc một điểm vật lý quan trọng. Tiếp theo, các tham chiếu thích hợp dòng l qua điểm O – điểm tham chiếu được chọn, thường là song song với trục x hoặc trục y . Hàm radius – vector là $r_x(\varphi)$, là hàm khoảng cách từ điểm O tới các đường biên theo hướng - ray. Trong đó (hình 1.10a)

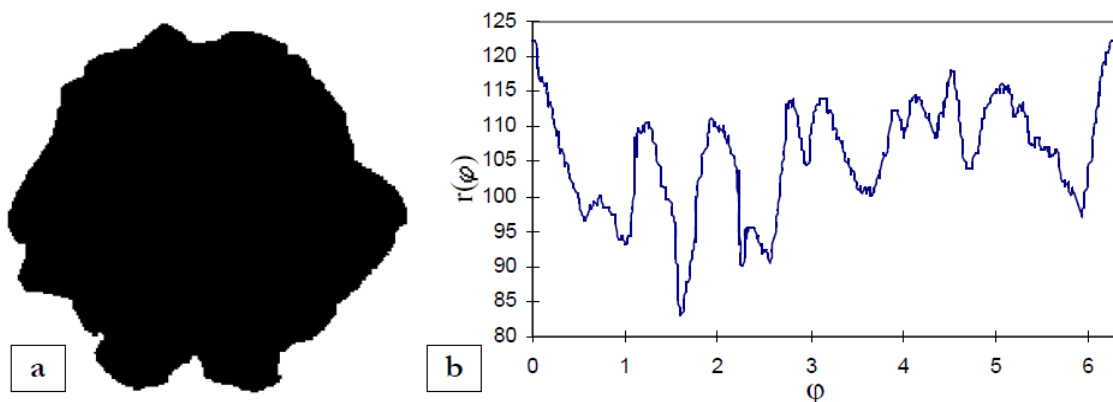


Hình 1.10 a) Hàm radius – vector

b) Vấn đề của hàm radius – vector khi hình không phải là hình sao

Nó là cần thiết, tuy nhiên, đối với bất kỳ điểm biên p thì các đoạn thẳng từ O sẽ làm cho p nằm trong hình vẽ. trong trường hợp này hàm radius – vector hoàn thành việc nối hình: nếu $r_x(\varphi)$ được cho, so với hình vẽ có thể được tái tạo lại hoàn toàn. Nếu hình sao nhỏ bị vi phạm chỉ bởi bất thường nhỏ ở đường biên của nó thì có thể phục hồi nó bằng cách làm mịn.

Tuy nhiên, trong trường hợp tổng quát, mô tả đường biên bởi các hàm radius – vector không thích hợp với hình vẽ không có dạng hình ngôi sao. Một ví dụ về một hình sao và hàm radius – vector của nó được đưa ra trong hình 1.11.



Hình 1.11 a) Hình sao X b) Hàm radius – vector của hình X

Một chỉ số để định lượng sự khác biệt giữa các hàm radius – vector của các hình khác nhau có thể thu được bằng cách xác định độ lệch bình phương trung bình của các hàm radius – vector từ một đường tròn có diện tích bằng nhau. Điều này ‘roughness coefficient’ có thể được định nghĩa là:

$$\hat{R}_X = \frac{1}{2\pi} \int_0^{2\pi} r_X^2(\varphi) d\varphi - \left(\frac{1}{2\pi} \int_0^{2\pi} r_X(\varphi) d\varphi \right)^2$$

1.2 NHẬN DẠNG BIỂN SỐ XE Ô TÔ Ở CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO

1.2.1 Khái quát về biển số xe ở Cộng hòa Dân chủ Nhân dân Lào

Đất nước Lào là một quốc gia không giáp biển duy nhất tại vùng Đông Nam Á. Lào giáp giới nước Myanmar và Trung Quốc phía tây bắc, Việt Nam ở phía đông, Campuchia ở phía nam, và Thái Lan ở phía tây. Trong xu hướng hội nhập kinh tế quốc tế, WTO hay khối ASEAN nền kinh tế của Lào ngày càng phát triển, kéo theo các công nghệ hiện đại như công nghệ thông tin ngày càng được ứng dụng nhiều trong các lĩnh vực như y tế, giao thông vận tải, giáo dục,....

Biển số xe của nước Cộng hòa Dân chủ Nhân dân Lào hiện nay có bao gồm cả chữ và số. Tuy nhiên theo dự định trong thời gian tới sẽ có sự thay đổi về cách đánh mã số, ký hiệu trên biển số xe của nước Cộng hòa Dân chủ Nhân dân Lào. Kiểu biển mới sẽ không bao gồm chữ Lào mà chỉ gồm các số và ký tự chữ cái A, B, C,...tên các tỉnh trong đất nước Lào và có tất cả 4 màu như là: đỏ, trắng, vàng, xanh dương.

Màu đỏ (xe quân đội, xe công an).

Màu trắng chữ đen (xe tạm thời, doanh nghiệp)

Màu trắng chữ xanh (xe công ty tư nhân, đại sứ quán)

Màu vàng (xe tư nhân)

Màu xanh dương (xe cơ quan)

1.2.2 Khái niệm & ứng dụng về nhận dạng

a) Khái niệm:

Hệ thống nhận dạng biển số xe là hệ thống có khả năng phân tích hình ảnh và xác định biển số trên xe, thông qua video, thiết bị ghi hình và hình ảnh. Sau cùng là xác định các thông tin như: chủ sở hữu xe, theo dõi xe với tốc độ chậm.

***) Phân loại ứng dụng nhận dạng biển số xe:**

Ứng dụng nhận dạng biển số xe là ứng dụng có khả năng phân tích hình ảnh và xác định biển số xe từ các hình ảnh chụp được từ các thiết bị thu hình. Nguồn hình ảnh cho ứng dụng có rất nhiều. Hình ảnh được trực tiếp thu nhận từ camera. Trong báo cáo tốt nghiệp chỉ dừng lại ở mức xác định được biển số xe (xác định các chữ) từ các bức ảnh.

Có nhiều cách thức khác nhau để phân loại các ứng dụng nhận dạng biển số xe. Một trong những cách đơn giản là phân loại ứng dụng nhận dạng biển số xe thông qua mục đích sử dụng. Có thể chia ứng dụng nhận dạng biển số xe thành hai loại sau:

Loại 1: Giới hạn vùng nhìn

- Đầu vào: Ảnh thu trực tiếp từ các thiết bị ghi nhận ảnh kỹ thuật số. Ảnh được ghi nhận thường chỉ giới hạn trong vùng có biển số xe.

- Nguyên lý hoạt động: Các phương tiện giao thông phải chạy với một tốc độ đủ chậm để máy ghi nhận hình ảnh có thể thu được ảnh vùng biển số xe.

- Ứng dụng: Những ứng dụng nhận dạng biển số xe loại này thường được dùng tại các trạm kiểm soát, các trạm thu phí, các bãi gửi xe tự động, các trạm gác cổng.

Loại 2: Không giới hạn vùng nhìn

•Đầu vào: Ảnh đầu vào thu được từ các thiết bị ghi hình tự động, không phụ thuộc vào góc độ, các đối tượng xung quanh, ảnh không cần bắt buộc chỉ chụp vùng chứa biển số xe, mà có thể ảnh tổng hợp như chứa thêm các đối tượng như người, cây, đường phố... miễn là vùng biển số xe phải đủ rõ để có thể thực hiện nhận dạng được các ký tự trong vùng đó.

•Nguyên lý hoạt động: do đặc tính không giới hạn vùng nhìn mà ảnh đầu vào có thể thu được từ một thiết bị ghi hình (camara, máy ảnh...). Do đó, công việc đầu tiên là dò tìm trong ảnh, để xác định đúng vùng nào là biển số xe. Sau đó, thực hiện tách vùng và nhận dạng. Cuối cùng tùy thuộc vào mục đích sử dụng mà kết quả nhận dạng được truyền đi hay lưu trữ để phục vụ nhu cầu của người dùng cuối.

Ứng dụng: Vì không phụ thuộc vào hình ảnh thu được nên có thể dùng ứng dụng tại nhiều nơi như tại những nơi điều tiết giao thông, tại các vị trí nhạy cảm của giao thông như ngã ba, ngã tư đường giao nhau. Kiểm soát, phát hiện những hành vi vi phạm an toàn giao thông.

b) Ứng dụng của hệ thống nhận dạng biển số xe:

Hệ thống nhận dạng biển số xe được xây dựng nhằm mục đích giám sát, kiểm soát các phương tiện. Dưới đây chúng ta đề cập đến một số ứng dụng phổ biến đối với hệ thống nhận dạng biển số xe:

+) Thu phí giao thông: Lắp đặt hệ thống “Nhận dạng biển số xe” tại các trạm thu phí nhằm hỗ trợ hoặc tự động hóa công tác thu phí.

+) Kiểm soát xe tại các đường biên giới: Mỗi quốc gia đều có những quy định riêng về biển số xe, để phục vụ cho công tác quản lý và phát hiện những phương tiện giao thông (xe) vượt biên giới bất hợp pháp. Việc lắp hệ thống “Nhận dạng biển số xe” tại các trạm kiểm soát sẽ góp phần hỗ trợ công tác kiểm tra và an ninh quốc gia.

+) Các trạm gác cổng: Việc lắp đặt hệ thống “Nhận dạng biển số xe” sẽ hỗ trợ hoặc tự động hóa công tác mở cổng cho xe vào.

Ngoài ra, hệ thống còn được ứng dụng vào công tác chống trộm xe, các bãi giữ xe tự động, điều tiết giao thông (chẳng hạn như Thành phố Dublin đã ứng dụng

công nghệ “Nhận dạng biển số xe tự động” trong việc điều tiết giao thông theo dạng biển số chẵn/lẻ).

1.2.3 Phân loại biển số xe

Biển số xe của nước Cộng hòa Dân chủ Nhân dân Lào hiện nay có bao gồm cả chữ và số, tuy nhiên theo dự định trong thời gian tới sẽ có sự thay đổi về cách đánh mã số, ký hiệu trên biển số xe của nước Cộng hòa Dân chủ Nhân dân Lào. Kiểu biển mới sẽ không bao gồm chữ Lào mà chỉ gồm các số và ký tự chữ cái A, B, C...

Sau đây là quy định biển số của 18 tỉnh thành, tất cả các huyện trong tỉnh, ký hiệu và số

Tỉnh/Thành phố	Huyện/Quận	Ký hiệu	Số xe
1- Attapue	1- Xay Set Tha 2- Sa Mak Khi Xay 3- Sa Nam Xay 4- Xien Xay 5- Phu Vong	PL	6777
2- Bokeo	1- Hauty Xai 2- Ton Pherng 3- Merng 4- Pha Au Dom 5- Pak Tha	SD	8882
3- Bolikhamxay	1- Tha Pha Bard 2- Pak Sun 3- Pak Ka Ding 4- Kham Kert 5- Vieng Thog 6- Bo Li Khun 7- Xay Cham Phone	GM	1112
4- Champasak	1- Pak Se		

	2- Sa Na Som Boun 3- Bar Jieng Cha Lern Souk 4- Pak Song 5- Pa Thoum Phone 6- Phon Thong 7- Cham Pa Sak 8- Su Khu Ma 9- Moun La Pa Mok 10- Khaung	PV	3334
5- Hoaphun	1- Xam Nua 2- Xieng Khor 3- Hiem 4- Vieng Xay 5- Hoa Meung 6- Xam Tay 7- Xop Bao 8- Aed 9- Kuan 10- Xon	EP	7772
6- Khammuan	1- Tha Khek 2- Ma Ha Xay 3- Nong Bok 4- Hin Boun 5- Yom Ma Lath 6- Boa La Pha 7- Na Kai 8- Xe Bung Fay 9- Xay Boa Thong 10- Khoun Kham	ES	8889

7-	Laungnamtha	1- Laung Nam Tha 2- Xing 3- Long 4- Vieng Thong 5- Na Lae	PV	1122
8-	Laungphabang	1- Laung Pha Bang 2- Xieng Ngan 3- Nan 4- Pak Au 5- Nam Bak 6- Ngoi 7- Pak Seng 8- Phon Thong 9- Jom Phet 10- Vieng Kham 11- Phu Khoun 12- Phon Thong	ES	8882
9-	Oudonxay	1- Xay 2- Lar 3- Na Mor 4- Nga 5- Baeng 6- Houn 7- Pak Baeng	PC	1234
10-	Phongsaly	1- Boun Tay 2- Khoa 3- May 4- Yot Au 5- Phong Sa Ly	SD	9998

		6- Sam Phun 7- Boun Nua		
11-	Xayyabury	1- Bo Tan 2- Hong sa 3- Kan Thao 4- Khoi 5- Ngan 6- Pak Lai 7- Phieng 8- Thong Mi Xay 9- Xay Ya Bu Ry 10- Xieng Hon 11- Xay Sa Than	PL	2333
12-	Salavun	1- Sa La Vun 2- La Khon Pheng 3- Vah Py 4- Rao Ngam 5- Toum Lan 6- Ta Aoi 7- Khong Se Don 8- Sa Moai	EP	6688
13-	Savannakhet	1- Kay Son 2- Au Thoum Ma 3- At Sa Phung Thong 4- Phin 5- Se Pon 6- Nong 7- Tha Pang Thong 8- Song Kone	PV	2122

		9- Cham Phone 10- Son Bu Ry 11- Xay Bu Ry 12- Vi La Bu Ry 13- At Sa Phone 14- Xay Phu Thong 15- Pha Larn Xay		
14-	Sekong	1- Tha Taeng 2- La Mam 3- Ka Leum 4- Dac Jing	GM	6878
15-	Nakhonlaung	1- Chan Tha Bu Ry 2- Xay Set Tha 3- Si Khod Ta Bong 4- Si Sat Ta Nak 5- Hard Sai Thong 6- Na Sai Thong 7- Xay Tha Ny 8- Xung Thong 9- Pak Ngeum	PC	6777
16-	Viengchan	1- Phuong 2- Hin Herb 3- Ka Sy 4- Keo Au Dom 5- Med 6- Phon Hong 7- Thu La Khom 8- Vung Vieng 9- Vieng Kham	PV	5556

	10- Sa Na Kham 11- Muen		
17- Xiengkhaung	1- Pack 2- Kham 3- Nong Hed 4- Khoun 5- Mork 6- Phu Kut 7- Pha Xay	GM	3373
18- Xaysomboun	1- Long Jaeng 2- Tha Thom 3- Long San 4- A Nu Vong 5- Hom	PV	4747

1.2.4 Điểm ảnh, biên ảnh

a) Biển số xe cũ hiện này



Ảnh biển số xe cũ của Quân đội



Ảnh biển số xe cũ của Công an



Ảnh biển số xe cũ của Tạm thời



Ảnh biển số xe cũ của Doanh nghiệp



Ảnh biển số xe cũ của Tư nhân



Ảnh biển số xe cũ của Đại sứ quán



Ảnh biển số xe cũ của Công ty tư nhân



Ảnh biển số xe cũ của Cơ quan

Hình 1.12 biển số xe cũ hiện này

b) Biển số xe mới



Ảnh biển số xe mới của Quân đội



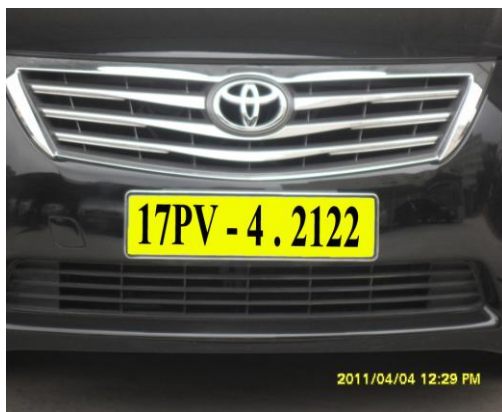
Ảnh biển số xe mới của Công an



Ảnh biển số xe mới Tạm thời



Ảnh biển số xe mới của Doanh nghiệp



Ảnh biển số xe mới của Tư nhân



Ảnh biển số xe mới của Đại sứ quán



Ảnh biển số xe mới của Công ty tư nhân

Ảnh biển số xe mới của Cơ quan

Hình 1.13 biển số xe mới

1.2.5 Vai trò của việc nhận dạng

Vai trò của việc nhận dạng đối tượng phụ thuộc nhiều vào các đặc trưng trích chọn mà các đặc trưng này chủ yếu được trích chọn của các đối tượng thu được từ ảnh.

Cùng với sự phát triển của khoa học kỹ thuật và nhu cầu đi lại ngày càng tăng số lượng phương tiện giao thông trên đường ngày càng xuất hiện nhiều. Với số lượng phương tiện giao thông lớn và còn không ngừng tăng lên như vậy đã làm nảy sinh nhiều vấn đề trong việc kiểm soát cũng như quản lý các phương tiện. Để giải quyết vấn đề này nhu cầu đặt ra là áp dụng các hệ thống tự động. Một trong những hệ thống như vậy là hệ thống tự nhận dạng biển số xe là hệ thống có khả năng thu nhận hình ảnh cũng như là ‘đọc’ và ‘hiểu’ các biển số xe một cách tự động.

Sẽ được sử dụng các kỹ thuật và thuật toán vào trong chương trình như là:

- Kỹ thuật bóc tách ảnh biển số xe
- Kỹ thuật tách các ký tự trên biển số xe
- Kỹ thuật trích chọn đặc trưng ảnh biển số xe
- Thuật toán nhị phân hoá ảnh đầu vào
- Thuật toán tách cạnh, dò biên
- Thuật toán phân vùng, phân cụm

CHƯƠNG II: MỘT SỐ VẤN ĐỀ TRONG NHẬN DẠNG ẢNH BIẾN SỐ XE Ô TÔ CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO

2.1 KỸ THUẬT BÓC TÁCH ẢNH

2.1.1 Thuật toán nhị phân hoá ảnh đầu vào

Nhị phân hoá ảnh đầu vào là chuyển ảnh đầu vào thành một ảnh đen trắng. Bằng cách này, ta hoàn toàn có thể trích rút ra được các thông tin cần thiết về các đối tượng có mặt trong ảnh như: diện tích, chu vi, hướng của đối tượng, tọa độ trọng tâm, tỉ lệ det,... Phép nhị phân hoá ảnh đầu vào được thực hiện bằng phép phân ngưỡng:

$$BW(r,c) = \begin{cases} 1 & \text{khi } I(r,c) \geq \theta \\ 0 & \text{khi } I(r,c) < \theta \end{cases}$$

Trong đó, I là ảnh đầu vào, BW là ảnh đen trắng đầu ra, θ là ngưỡng đã được chọn trước. Ngưỡng θ có thể chọn tự động sử dụng thuật toán Otsu. Trên hình 2.1 là kết quả phép tách cạnh ảnh đầu vào tự động, sử dụng thuật toán Otsu.

Ảnh ban đầu được sử dụng là ảnh 256 mức xám. Việc sử dụng ảnh 256 mức xám không làm giảm đi tính đa năng của ứng dụng. Trên thực tế, ảnh 256 mức xám vẫn được sử dụng nhiều, và nhiều thiết bị ghi hình cũng có khả năng tự chuyển ảnh màu thành ảnh 256 mức xám. Tuy nhiên, nếu để ảnh 256 mức xám thì việc phát hiện biên không hiệu quả, vì sự thay đổi liên tục của các mức xám làm cho việc xác định biên không phải dễ dàng, và việc tìm ra các vùng liên tục của biên khá hạn chế. Vì vậy, chúng ta thực hiện chuyển ảnh về dạng nhị phân để thực hiện việc lấy biên nhanh hơn.

```
void Binarize// Nhị phân hóa ảnh
{
// Ảnh đầu vào: ảnh 256 mức xám
// Đầu ra là ảnh nhị phân
P: là tổng số điểm ảnh được xét (m,n)
g(j,j) tương ứng là mức xám của điểm ảnh I(i,j)
 $\theta$ : là ngưỡng của ảnh được xác định theo phương pháp ở trên.
```

```

for(int i= 0; i< m; i++)
  for(int j= 0; j< n; j++)
    if(g(i, j)<=  $\theta$ )
      {
         $I_c(I, j)= 0$ 
      }else  $I_c(I, j)= 1$ 
    }

```

Vậy ta thu được ảnh nhị phân I_c , ảnh nhị phân thu được vẫn đảm bảo tách biệt giữa vùng chứa biển số xe với vùng xung quanh. Đồng thời loại bỏ những vùng đồng nhất và ít biến thiên.



Hình 2.1 Nhị phân hoá ảnh

2.1.2 Thuật toán tách cạnh, dò biên

Các điểm nằm trên biên (cạnh) của đối tượng trong ảnh có tính chất là tại đó biến thiên độ sáng lớn. Việc tìm ra biên của các đối tượng cho phép ta định hình được vùng đối tượng. Mặt khác, biên đối tượng có những đặc tính bất biến cho phép ta định hình được vùng biển số đối với các ảnh xe được chụp trong các điều kiện sáng khác nhau. Để tách cạnh, ta sử dụng các toán tử tách cạnh dựa trên xấp xỉ các toán tử vi phân. Sau đây là một số toán tử tách cạnh được nghiên cứu xây dựng phục vụ cho mục đích của bài toán:

+ Toán tử Sobel: Thực hiện phép cuộn ảnh với mặt nạ tách cạnh Sobel cho phép tìm kiếm các cạnh theo cả các hướng ngang và dọc và sau đó tổ hợp thông tin này vào trong cùng một metric. Các mặt nạ Sobel như sau:

Mặt nạ hàng

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Mặt nạ cuộn

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Các mặt nạ này được cuộn với ảnh. Tại mỗi vị trí pixel bây giờ ta có hai số: s_1 tương ứng với kết quả từ mặt nạ hàng, và s_2 tương ứng với kết quả từ mặt nạ cột. Ta sử dụng các số này để tính toán hai metrics, là cường độ cạnh và hướng cạnh được định nghĩa như sau:

$$\text{Cường độ cạnh: } \sqrt{s_1^2 + s_2^2}$$

$$\text{Hướng cạnh: } \tan^{-1} \left[\frac{s_1}{s_2} \right]$$

Hướng cạnh vuông góc với cạnh bởi vì hướng cạnh đã xác định là hướng của gradient, mà dọc theo đó các mức xám đang thay đổi.

+ Toán tử Prewitt: Thực hiện tương tự toán tử Sobel nhưng sử dụng mặt nạ Prewitt. Các mặt nạ được định nghĩa như sau:

Mặt nạ hàng

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Mặt nạ cuộn

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

+ Toán tử Canny: sử dụng 2 ngưỡng khác nhau (để dò tìm các cạnh mạnh và yếu), và gồm cả các cạnh yếu trong ảnh đầu ra chỉ khi chúng được nối liền với các cạnh mạnh. Toán tử này kém thích hợp hơn các toán tử trên khi ảnh bị “fooled” bởi nhiễu, và thích hợp hơn khi cần tìm ra cả các cạnh yếu.



1) Ảnh đầu vào



2) Kết quả tách cạnh Sobel



3) Kết quả tách cạnh Prewitt



4) Kết quả tách cạnh Canny

Hình 2.2 Tách cạnh Sobel, Prewitt

2.1.3 Tính toán các số đo đặc trưng của đối tượng ảnh nhị phân

Các đặc trưng nhị phân cho phép ta chọn ra được các đối tượng có đặc trưng nằm trong vùng liên quan đến biển số xe. Để trích chọn đặc trưng nhị phân, trước hết ta gán nhãn cho các đối tượng trong ảnh nhị phân. Trong ảnh nhị phân, mỗi vùng liên thông nhau được coi là một đối tượng. Kết quả của phép gán nhãn được minh họa như trong ví dụ sau đây:

$$\text{Ảnh đầu vào: } I = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \rightarrow \text{Ảnh đầu ra: } L = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 3 \\ 2 & 0 & 2 & 0 & 0 & 3 \\ 2 & 2 & 2 & 0 & 0 & 3 \end{bmatrix}$$

Tiếp theo, ta định nghĩa một hàm $I_i(r,c)$ như sau:

$$I_i(r,c) = 1, \text{ nếu } I(r,c) \in \text{đối tượng thứ } i \text{ (tức } L(r,c) = i),$$

$$I_i(r,c) = 0, \text{ nếu ngược lại.}$$

Bây giờ ta có thể định nghĩa các đặc trưng của các đối tượng có mặt trong ảnh nhị phân.

C1. Diện tích của đối tượng thứ i là:

$$A_i = \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} I_i(r,c)$$

Diện tích A_i được đo bằng đơn vị pixel và biểu thị được kích thước tương

quan của đối tượng.

C2. Toạ độ trọng tâm đối tượng là một cặp (\bar{r}_i, \bar{c}_i) được xác định như sau:

$$\bar{r}_i = \frac{1}{A_i} \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} r I_i(r, c), \quad \bar{c}_i = \frac{1}{A_i} \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} c I_i(r, c)$$

Đặc trưng này giúp xác định vị trí của đối tượng trong mặt phẳng ảnh hai chiều.

C3. Chu vi của đối tượng: Được định nghĩa bằng tổng số pixel (có giá trị 1) nằm trên biên của đối tượng đó. Để tìm chu vi của đối tượng, ta có thể áp dụng một bộ tách cạnh lên đối tượng, sau đó là đếm số pixel '1'.

Áp dụng các đặc trưng này lên đối tượng là ảnh biển số ô tô, xe máy, ta sẽ lọc bỏ được rất nhiều đối tượng không phải là ảnh biển số có mặt trong ảnh, chẳng hạn như: Đối tượng có diện tích hoặc chu vi quá lớn, đối tượng quá dẹt, đối tượng có tâm nằm ngoài vùng có khả năng chứa biển số, đối tượng có hướng không hợp lý...

2.1.4 Thuật toán phân vùng, phân cụm

Việc phân vùng, phân cụm ảnh cũng cho phép ta bóc tách được vùng ảnh biển số xe vì các pixel nằm trên vùng biển số thường có cấp xám gần nhau.

❖ *Phân đoạn ảnh* gồm 2 kỹ thuật chính là phân chia và hợp nhất. Quá trình diễn biến như sau:

1. Định nghĩa một mẫu thử tính đồng nhất. Ta phải định nghĩa một số đo tính đồng nhất, nó có thể tích hợp cả các thông tin độ sáng, màu, kết cấu, xác định một tiêu chuẩn về vùng và phải chuyển nó cho mẫu thử tính đồng nhất.

2. Phân chia ảnh thành các vùng có kích thước đều nhau.

3. Tính toán số đo đồng nhất cho mỗi vùng.

4. Nếu mẫu thử tính đồng nhất được chuyển vào cho một vùng, thì một phép hợp nhất sẽ được thử với các láng giềng lân cận của nó. Nếu tiêu chuẩn không thỏa mãn, thì vùng sẽ được phân chia.

5. Tiếp tục quá trình này cho đến khi tất cả các vùng đều được thử với mẫu thử.

❖ Quá trình phân cụm được thực hiện như sau:

1. Xét toàn bộ ảnh như là một vùng và tính toán các histogram cho ảnh đó.
2. Tìm kiếm đỉnh của histogram. Chọn ra đỉnh tốt nhất và đặt các ngưỡng ở ngay một trong hai bên của đỉnh đó. Phân đoạn ảnh thành 2 vùng dựa vào đỉnh này.
3. Phân ngưỡng nhị phân sao cho chỉ còn một vùng con kết nối (liên thông) được để lại.
4. Lặp lại các bước 1-3 cho mỗi vùng cho đến khi không còn vùng con mới có thể thể tạo ra được, tức là, khi đó không còn histogram nào có các đỉnh có ý nghĩa.

Trong hình 2.3 là các hình ảnh kết quả xử lý qua các bước trung gian có sử dụng các kỹ thuật trên để bóc tách vùng biển số của một ảnh ô tô:



1) Ảnh đầu vào



2) Phân ngưỡng tự động



3) Lọc bỏ đối tượng bé



4) Lọc bỏ theo hướng, tỉ lệ dẹt, ...



5) Kết quả bóc tách

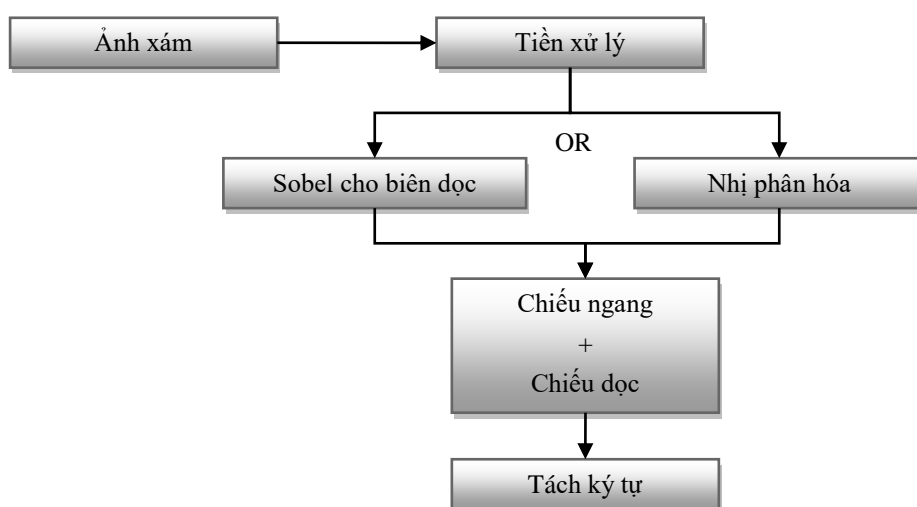
Hình 2.3 Các hình ảnh trung gian xử lý bóc tách ảnh biển số xe

2.2 KỸ THUẬT TÁCH CÁC KÝ TỰ

Sau khi bóc tách được biển số xe ra khỏi ảnh, ta tiếp tục bóc tách các ảnh ký tự có mặt trên biển số xe. Cơ chế bóc tách các ký tự cũng thực hiện được thông qua các kỹ thuật đã nêu trên, chỉ khác ở các tham số đặc trưng cần thiết đặt lại sao cho hợp lý.

Chúng ta sẽ đi lần lượt từng bước. Bắt đầu từ ảnh đầu vào cho đến khi tách được từng ký tự. Để dễ dàng hơn cho việc xử lý, ảnh đầu vào được chuyển thành ảnh 256 mức xám. Việc chuyển đổi này không làm mất đi tính chất và các đặc trưng của ảnh. Đặc điểm của các đối tượng ảnh vẫn được giữ nguyên sau khi chuyển đổi.

Việc tách ký tự này là một phần của bài toán phân đoạn ảnh trong hệ thống tự động nhận dạng biển số xe. Có 3 bước chính quan trọng trong việc tách ký tự là: tiền xử lý, nhị phân hóa (hoặc phát hiện biên sobel) và cuối cùng là tách ký tự. Tiếp theo nhị phân hóa để làm nổi bật phần ký tự. Sau đó tách rời các ký tự riêng biệt.



Hình 2.4 Lưu đồ tách ký tự

2.2.1 Các phương pháp tách ký tự và nhận dạng ký tự

2.2.1.1 Tách ký tự

Bước tiếp theo sau khi tách được biên số xe là tách ký tự trên biên số. Đây là một bước quan trọng trong quy trình nhận dạng biên số xe tự động bởi vì tất cả các bước sau đó đều phụ thuộc vào bước này. Nếu tách ký tự không chính xác thì mọi cố gắng tiếp theo đều vô nghĩa.

Việc tách ký tự có thể gặp phải một số khó khăn như dính ký tự, dính viền biên số với ký tự, không thể tách được ký tự do ký tự quá mờ hoặc biên số quá cũ. Đôi khi ký tự tách được lại có một số nhiễu đi kèm. Đặc biệt đối với Việt Nam, rất nhiều ký tự trên biên số bị che lấp bởi các đinh ốc gắn vào biển. Đây là một thách thức lớn đối với quá trình nhận dạng.

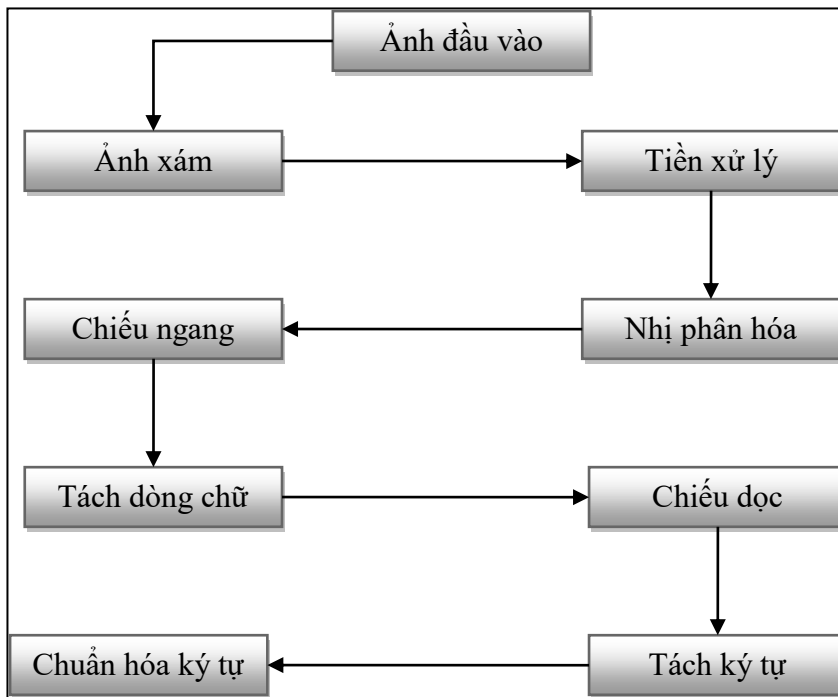
Có rất nhiều phương pháp để thực hiện tách ký tự, một số phương pháp có thể kể đến như phương pháp chiếu, phương pháp hình thái học, phương pháp phát hiện contour, phương pháp sử dụng trí tuệ nhân tạo... Tuy nhiên phương pháp chiếu có vẻ như được lựa chọn nhiều hơn. Trong bài báo cáo này em cũng tập

trung vào phương pháp chiếu. Với phương pháp chiếu này chúng ta cũng có nhiều cách thực hiện, có thể nhị phân ảnh rồi chiếu, có thể phát hiện biên rồi chiếu.

a. Phương pháp đã phát triển - Sử dụng ngưỡng xám và chiếu

Phương pháp này sử dụng một ngưỡng tự động để làm nổi ký tự rồi chiếu ngang, chiếu dọc để phân tách ký tự

Đầu tiên, ta nhập ảnh đầu vào thành ảnh xám mức xám, sau đó thông qua bước tiền xử lí: lọc nhiễu, chuẩn hóa histogram... Tiếp đó ta dùng thuật toán tách ngưỡng để nhị phân hóa ảnh xám đó. Thông thường người ta dùng thuật toán Otsu hoặc thuật toán tách ngưỡng thích ứng (adaptive thresholding). Sau đó ta dùng bước chiếu ngang để phát hiện ra dòng chữ trên biển số xe rồi tách các dòng kí tự ấy. Khi tách được dòng kí tự thì ta lại lấy phương pháp chiếu dọc để phát hiện ra các ký tự. Tiếp tục tách các ký tự vừa phát hiện được rồi chuẩn hóa các ký tự về một kích thước cố định.



Hình 2.5 Các bước tách ký tự sử dụng kỹ thuật tách ngưỡng

Các bước xử lý:

Bước 1 Chuyển ảnh đầu vào thành ảnh xám 256 mức xám

Bước 2 Tiền xử lý: lọc nhiễu, chuẩn hóa histogram...

Bước 3 Nhị phân hóa ảnh xám bằng thuật toán tách ngưỡng. Thông thường người ta thường dùng thuật toán tách ngưỡng Otsu hoặc thuật toán tách ngưỡng thích ứng (adaptive thresholding)

Bước 4 Chiếu ngang để phát hiện dòng chữ trên biển số xe

Bước 5 Tách các dòng ký tự

Bước 6 Chiếu dọc để phát hiện ra các ký tự

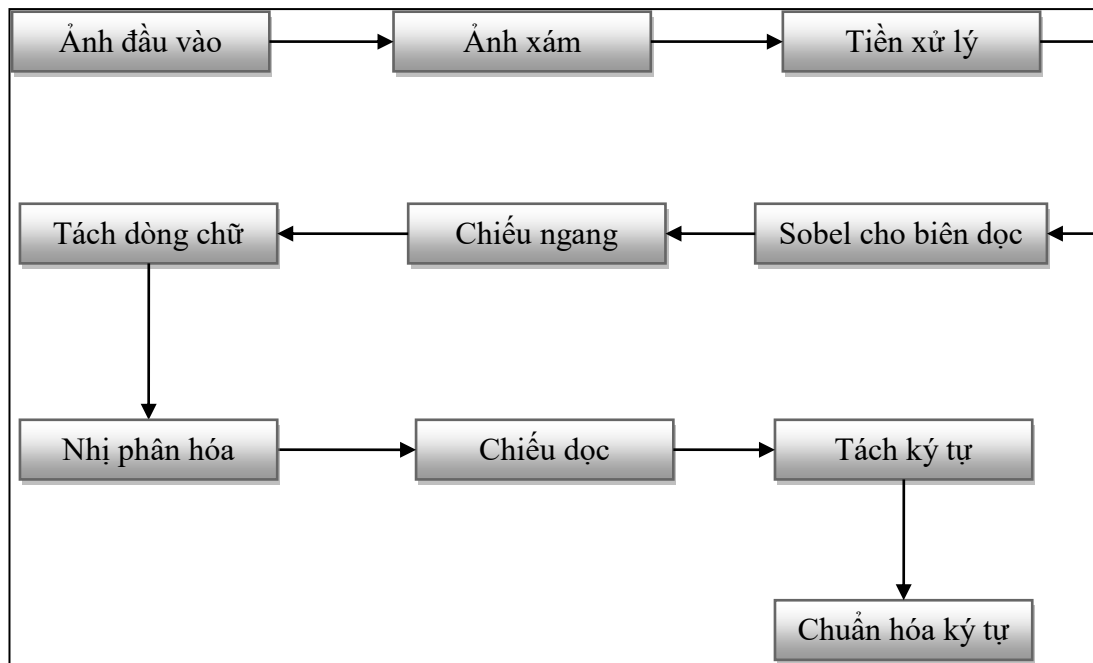
Bước 7 Tách các ký tự

Bước 8 Chuẩn hóa ký tự về một kích thước cố định

b. Phương pháp đề xuất - Sử dụng phát hiện biên sobel và chiếu

Số hóa bởi Trung tâm Học liệu – ĐHTN <http://www.lrc.tnu.edu.vn>

Đầu tiên, nhập ảnh đầu vào thành ảnh xám mức xám, sau đó tiền xử lý bao gồm lọc nhiễu và chuẩn hóa histogram. Khác với thuật tách ngưỡng, ta phát hiện biên dọc bằng thuật toán Sobel. Sau đó lại tiếp tục chiếu ngang để phát hiện dòng chữ trên biển số xe, rồi tách dòng ký tự. Tiếp tục dùng thuật toán tách ngưỡng để nhị phân hóa ảnh xám, rồi lại chiếu dọc để phát hiện ra các ký tự, tách các ký tự và bước cuối cùng là chuẩn hóa ký tự về một kích thước cố định tương tự như thuật toán tách ngưỡng.



Hình 2.6 Các bước tách ký tự sử dụng kỹ thuật sobel

Các bước xử lý:

Bước 1 Chuyển ảnh đầu vào thành ảnh xám 256 mức xám

Bước 2 Tiền xử lý: lọc nhiễu, chuẩn hóa histogram...

Bước 3 Phát hiện biên dọc bằng thuật toán sobel

Bước 4 Chiếu ngang để phát hiện dòng chữ trên biển số xe

Bước 5 Tách các dòng ký tự

Bước 6 Nhi phân hóa ảnh xám bằng thuật toán tách ngưỡng

Bước 7 Chiếu dọc để phát hiện ra các ký tự

Bước 8 Tách các ký tự

Bước 9 Chuẩn hóa ký tự về một kích thước cố định

2.2.2 Nhận dạng ký tự

Sau khi tách rời được các ký tự, việc tiếp theo là nhận dạng ký tự. Bài toán nhận dạng ký tự là bài toán đã được phát triển và sử dụng trong rất nhiều ứng dụng như nhận dạng chữ viết tay, nhận dạng chữ in. Với bài toán nhận dạng biển số xe, số lượng ký tự ít, các kiểu font chữ ít cho nên độ phức tạp cũng ít hơn, đó là một thuận lợi.

2.2.2.1 Tổng quan về nhận dạng

Nhận dạng là quá trình phân loại các đối tượng được biểu diễn theo một mô hình nào đó và gán cho chúng vào một lớp (gán cho đối tượng một tên gọi) dựa theo những quy luật và các mẫu chuẩn. Quá trình nhận dạng dựa vào những mẫu học biết trước gọi là nhận dạng có thầy hay *học có thầy* (supervised learning); trong trường hợp ngược lại gọi là *học không có thầy* (non supervised learning). Chúng ta sẽ lần lượt giới thiệu các khái niệm này.

❖ Không gian biểu diễn đối tượng và không gian diễn dịch

a) Không gian biểu diễn đối tượng

Các đối tượng khi quan sát hay thu thập được, thường được biểu diễn bởi tập các đặc trưng hay đặc tính. Người ta thường phân các đặc trưng theo các loại như: đặc trưng tô pô, đặc trưng hình học và đặc trưng chức năng. Việc biểu diễn ảnh theo đặc trưng nào là phụ thuộc vào ứng dụng tiếp theo.

Ở đây ta đưa ra một cách hình thức việc biểu diễn các đối tượng. Giả sử đối tượng X (ảnh, chữ viết, dấu vân tay, v...,v) được biểu diễn bởi n thành phần (n đặc trưng): $X = \{x_1, x_2, \dots, x_n\}$; mỗi x_i biểu diễn một đặc tính. Không gian biểu diễn đối tượng thường gọi tắt là không gian đối tượng X được định nghĩa:

$$X = \{X_1, X_2, \dots, X_m\}$$

Trong đó mỗi X_i biểu diễn một đối tượng. Không gian này có thể là vô hạn. Để tiện xem xét chúng ta chỉ xét tập X là hữu hạn.

b) Không gian diễn dịch là tập các tên gọi của đối tượng. Kết thúc quá trình nhận dạng ta xác định được tên gọi cho các đối tượng trong tập không gian đối tượng hay nói là đã nhận dạng được đối tượng. Một cách hình thức gọi Ω là tập tên đối tượng:

$$\Omega = \{w_1, w_2, \dots, w_k\} \text{ với } w_i, i = 1, 2, \dots, k \text{ là tên các đối tượng}$$

Quá trình nhận dạng đối tượng f là một ánh xạ $f: X \rightarrow \Omega$ với f là tập các quy luật để xác định một phần tử trong X ứng với một phần tử trong Ω . Nếu tập các quy luật và tập tên các đối tượng là biết trước như trong nhận dạng chữ viết (có 26 lớp từ A đến Z), người ta gọi là nhận dạng có thầy. Trường hợp thứ hai là nhận dạng không có thầy. Đương nhiên trong trường hợp này việc nhận dạng có khó khăn hơn.

❖ Mô hình và bản chất của quá trình nhận dạng

1) Mô hình

Việc chọn lựa một quá trình nhận dạng có liên quan mật thiết đến kiểu mô tả mà người ta sử dụng để đặc tả đối tượng. Trong nhận dạng, người ta phân chia làm 2 họ lớn:

- Họ mô tả theo tham số
- Họ mô tả theo cấu trúc.

Cách mô tả được lựa chọn sẽ xác định *mô hình* của đối tượng. Như vậy, chúng ta sẽ có 2 loại mô hình: *mô hình theo tham số* và *mô hình cấu trúc*.

• **Mô hình tham số** sử dụng một véctơ để đặc tả đối tượng. Mỗi phần tử của véctơ mô tả một đặc tính của đối tượng. Thí dụ như trong các đặc trưng chức năng,

người ta sử dụng các hàm cơ sở trực giao để biểu diễn. Và như vậy ảnh sẽ được biểu diễn bởi một chuỗi các hàm trực giao. Giả sử C là đường bao của ảnh và $C(i, j)$ là điểm thứ i trên đường bao, $i = 1, 2, \dots, N$ (đường bao gồm N điểm).

Giả sử tiếp :

$$x_0 = \frac{1}{N} \sum_{i=1}^N x_i$$

$$y_0 = \frac{1}{N} \sum_{i=1}^N y_i$$

là tọa độ tâm điểm. Như vậy, moment trung tâm bậc p, q của đường bao là:

$$\mu_{pq} = \frac{1}{N} \sum_{i=1}^N (x_i - x_0)^p (y_i - y_0)^q$$

Véc tơ tham số trong trường hợp này chính là các moment μ_{ij} với $i=1, 2, \dots, p$ và $j=1, 2, \dots, q$. Còn trong số các đặc trưng hình học, người ta hay sử dụng chu tuyến, đường bao, diện tích và tỉ lệ $T = 4\pi S/p^2$, với S là diện tích, p là chu tuyến.

Việc lựa chọn phương pháp biểu diễn sẽ làm đơn giản cách xây dựng. Tuy nhiên, việc lựa chọn đặc trưng nào là hoàn toàn phụ thuộc vào ứng dụng. Thí dụ, trong nhận dạng chữ (sẽ trình bày sau), các tham số là các dấu hiệu:

- số điểm chạc ba, chạc tư,

- số điểm chu trình,

- số điểm ngoặt,

- số điểm kết thúc,

chẳng hạn với chữ t có 4 điểm kết thúc, 1 điểm chạc tư,...

•**Mô hình cấu trúc:** Cách tiếp cận của mô hình này dựa vào việc mô tả đối tượng nhờ một số khái niệm biểu thị các đối tượng cơ sở trong ngôn ngữ tự nhiên. Để mô tả đối tượng, người ta dùng một số dạng nguyên thủy như đoạn thẳng, cung, v, \dots, v . Chẳng hạn một hình chữ nhật được định nghĩa gồm 4 đoạn thẳng vuông góc với nhau từng đôi một. Trong mô hình này người ta sử dụng một bộ kí hiệu kết thúc V_t , một bộ kí hiệu không kết thúc gọi là V_n . Ngoài ra có dùng một tập các luật sản xuất để mô tả cách xây dựng các đối tượng phù hợp dựa trên các đối tượng đơn

giản hơn hoặc đối tượng nguyên thủy (tập Vt). Trong cách tiếp cận này, ta chấp nhận một khẳng định là: cấu trúc một dạng là kết quả của việc áp dụng luật sản xuất theo những nguyên tắc xác định bắt đầu từ một dạng gốc ban đầu. Một cách hình thức, ta có thể coi mô hình này tương đương một văn phạm $G = (Vt, Vn, P, S)$ với:

- Vt là bộ ký hiệu kết thúc,
- Vn là bộ ký hiệu không kết thúc,
- P là luật sản xuất,
- S là dạng (ký hiệu bắt đầu).

2) Bản chất của quá trình nhận dạng

Quá trình nhận dạng gồm 3 giai đoạn chính:

- Lựa chọn mô hình biểu diễn đối tượng.
- Lựa chọn luật ra quyết định (phương pháp nhận dạng) và suy diễn quá trình học.

- Học nhận dạng.

Khi mô hình biểu diễn đối tượng đã được xác định, có thể là định lượng (mô hình tham số) hay định tính (mô hình cấu trúc), quá trình nhận dạng chuyển sang giai đoạn học. Học là giai đoạn rất quan trọng. Thao tác học nhằm cải thiện, điều chỉnh việc phân hoạch tập đối tượng thành các lớp.

Việc nhận dạng chính là tìm ra quy luật và các thuật toán để có thể gán đối tượng vào một lớp hay nói một cách khác gán cho đối tượng một tên.

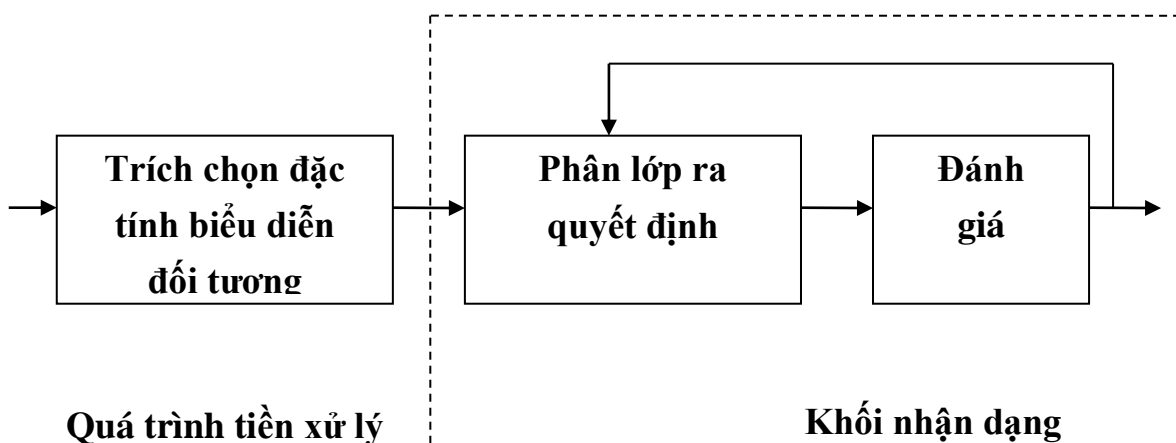
**Học có thầy (supervised learning)*

Kỹ thuật phân loại nhờ kiến thức biết trước gọi là học có thầy. Đặc điểm cơ bản của kỹ thuật này là người ta có một thư viện các mẫu chuẩn. Mẫu cần nhận dạng sẽ được đem sánh với mẫu chuẩn để xem nó thuộc loại nào. Thí dụ như trong một ảnh viễn thám, người ta muốn phân biệt một cánh đồng lúa, một cánh rừng hay một vùng đất hoang mà đã có các miêu tả về các đối tượng đó. Vấn đề chủ yếu là thiết kế một hệ thống để có thể đối sánh đối tượng trong ảnh với mẫu chuẩn và quyết định gán cho chúng vào một lớp. Việc đối sánh nhờ vào các thủ tục ra quyết

định dựa trên một công cụ gọi là *hàm phân lớp* hay *hàm ra quyết định*. Hàm này sẽ được đề cập trong phần sau.

***Học không có thầy (non supervised learning)**

Kỹ thuật học này phải tự định ra các lớp khác nhau và xác định các tham số đặc trưng cho từng lớp. Học không có thầy đương nhiên là khó khăn hơn. Một mặt, do số lớp không được biết trước, mặt khác những đặc trưng của các lớp cũng không biết trước. Kỹ thuật này nhằm tiến hành mọi cách gộp nhóm có thể và chọn lựa cách tốt nhất. Bắt đầu từ tập dữ liệu, nhiều thủ tục xử lý khác nhau nhằm phân lớp và nâng cấp dần để đạt được một phương án phân loại. Nhìn chung, dù là mô hình nào và kỹ thuật nhận dạng ra sao, một hệ thống nhận dạng có thể tóm tắt theo sơ đồ sau:



Hình 2.7 Sơ đồ tổng quát một hệ nhận dạng.

2.3 KỸ THUẬT TRÍCH CHỌN ĐẶC TRƯNG ẢNH

Đối tượng ảnh ký tự sau khi bóc tách sẽ được biểu diễn đường biên, từ đó trích rút các đặc trưng để đưa vào nhận dạng ảnh ký tự trên biển số xe. Các kỹ thuật này sẽ được báo cáo tập trung nghiên cứu và xây dựng các thử nghiệm tiên để xây dựng thành ứng dụng trong thời gian tới.

Trích chọn đặc trưng là cơ sở của tra cứu ảnh dựa vào nội dung. Theo nghĩa rộng, các đặc trưng có thể bao gồm cả các đặc trưng dựa vào văn bản và các đặc trưng trực quan như màu, kết cấu, hình dạng. Trong phạm vi đặc trưng trực quan, các đặc trưng có thể được phân loại tiếp thành các đặc trưng chung và các đặc trưng lĩnh vực cụ thể. Các đặc trưng trực quan chung gồm màu, kết cấu, và hình dạng trong khi các đặc trưng lĩnh vực cụ thể là phụ thuộc ứng dụng. Các đặc trưng lĩnh vực cụ thể bao gồm nhiều tri thức lĩnh vực.

Nhìn chung, không tồn tại một biểu diễn đơn tốt nhất cho một đặc trưng đã cho. Với mọi đặc trưng được cho tồn tại nhiều biểu diễn mô tả đặc trưng từ các cảnh hướng khác nhau.

2.3.1 Màu sắc

Màu là đặc trưng trực quan quan trọng đầu tiên và đơn giản nhất cho việc đánh chỉ số và tra cứu các ảnh. Nó cũng là đặc trưng được sử dụng phổ biến nhất trong tra cứu ảnh dựa vào nội dung.

Một ảnh màu tiêu biểu được thu từ một camera số, hoặc được tải xuống từ Internet thường có ba kênh màu (các ảnh xám chỉ có một kênh, các ảnh đa phổ có thể có nhiều hơn ba kênh). Tuy nhiên, các giá trị của dữ liệu ba chiều (3 kênh màu) từ ảnh màu không cho chúng ta một mô tả chính xác của màu trong ảnh, nhưng cho vị trí của các điểm ảnh này trong không gian màu. Các điểm ảnh có các giá trị $(1,1,1)$ sẽ xuất hiện khác nhau về màu trong các không gian màu khác nhau.

Tìm kiếm ảnh theo lược đồ màu là phương pháp phổ biến và được sử dụng nhiều nhất trong các hệ thống tìm kiếm ảnh theo nội dung. Đây là phương pháp đơn giản, tốc độ tìm kiếm tương đối nhanh tuy nhiên kết quả tìm kiếm có độ chính xác không cao. Đây có thể xem là bước lọc đầu tiên cho những bước tìm kiếm sau. Một số lược đồ màu được sử dụng như: lược đồ màu RGB, lược đồ màu HSI, lược đồ HSI cải tiến. Trong đó, lược đồ màu RGB được sử dụng phổ biến nhất.

Lược đồ màu RGB:

Đối với ảnh 256 màu, lược đồ màu của ảnh tương đương với lược đồ màu của ảnh xám. Đối với ảnh 24 bit màu, lược đồ miêu tả khả năng kết nối về cường độ của ba kênh màu R, G, B. Lược đồ màu này được định nghĩa như sau:

$$h_{R, G, B} = r, g, b \quad N \text{ Prob } R, G, B$$

Trong đó N là số lượng điểm có trong ảnh.

Lược đồ màu này được tính bằng cách rời rạc hóa từng màu trong ảnh, sau đó đếm số điểm ảnh của mỗi màu. Khi mà số lượng màu là có hạn, để thuận tiện hơn, người ta thường chuyển đổi ba kênh màu thành một biến giá trị duy nhất. Một cách khác để tính lược đồ màu của ảnh RGB là ta phân ra làm 3 lược đồ riêng biệt h_R [], h_G [], h_B []. Khi đó, mỗi lược đồ được tính bằng cách đếm kênh màu tương ứng trong mỗi điểm ảnh.

2.3.1.1 Độ đo tương đồng về màu sắc

Một số độ đo tương đồng được sử dụng như: Độ đo khoảng cách Oclit, độ đo Jensen-Shannon divergence (JSD).

Gọi $h(I)$ và $h(M)$ tương ứng là 2 lược đồ màu của hai ảnh I và ảnh M. Khi đó các loại độ đo màu được định nghĩa là một số nguyên (hoặc số thực) theo các loại độ đo tương ứng như sau:

Khoảng cách Oclit:

Đây là khoảng cách Oclit thông thường giữa các K bin:

$$\text{Intersection}(h(I), h(M)) = \sum_{j=1}^K \sqrt{(h(I) - h(M))^2}$$

Hoặc:

$$\text{Intersection}(h(I), h(M)) = \sum_{j=1}^K |h(I) - h(M)|$$

Độ đo Jensen-Shannon divergence (JSD) :

Độ đo Jensen-Shannon divergence sử dụng lược đồ màu RGB để tính toán độ tương đồng về màu sắc giữa 2 ảnh :

$$d_{JD}(H, H') = \sum_{m=1}^M H_m \log \frac{2H_m}{H_m + H'_m} + H'_m \log \frac{2H'_m}{H'_m + H_m}$$

Trong đó: H và H' là 2 biểu đồ màu được so sánh, H_m là bin thứ m của biểu đồ H

2.3.1.2 Không gian màu

Không gian màu là sự biểu diễn tập hợp các màu, một số không gian màu được sử dụng rộng rãi trong đồ họa máy tính. Những mô hình không gian màu có thể được phân biệt thành hướng phần cứng và hướng người sử dụng. Mô hình không gian màu hướng phần cứng bao gồm: RGB, IYQ, CMY dựa trên học thuyết 3 màu. Mô hình không gian màu hướng người sử dụng gồm: HLS, HCV, HSV,... dựa trên 3 tỷ lệ phần trăm của màu đó là: sắc màu, độ bão hoà, cường độ sáng.

Nếu chúng ta coi thông tin màu của một ảnh như tín hiệu một, hai, hoặc ba chiều, phân tích tín hiệu bằng việc sử dụng ước lượng mật độ xác suất là cách đơn giản nhất để mô tả thông tin màu của ảnh. Lược đồ là công cụ đơn giản nhất.

2.3.2 Kết cấu

Kết cấu là một đặc tính quan trọng khác của ảnh. Các biểu diễn kết cấu đa dạng đã được nghiên cứu trong nhận dạng mẫu và thị giác máy tính. Về cơ bản, các phương pháp biểu diễn kết cấu có thể được phân ra thành hai loại: cấu trúc và thống kê. Các phương pháp cấu trúc, gồm toán tử hình thái và đồ thị kề, mô tả kết cấu bởi nhận dạng cấu trúc gốc và các luật sắp đặt của chúng. Chúng có khuynh hướng hiệu quả nhất khi được áp dụng với các kết cấu đều. Các phương pháp thống kê, gồm các kỹ thuật phổ năng lượng Fourier, các ma trận đồng khả năng, phân tích thành phần chính bất biến - trượt, đặc trưng Tamura, phân rã Wold, trường ngẫu nhiên Markov mô hình fractal, và lọc đa phân giải như biến đổi Gabor và sóng, mô tả kết cấu bằng phân bố thống kê của cường độ ảnh. Một số biểu diễn kết cấu được sử dụng thường xuyên và đã được chứng minh là hiệu quả trong tra cứu ảnh dựa vào nội dung.

2.3.3 Đặc trưng hình dạng

Màu sắc và kết cấu là những thuộc tính có khái niệm toàn cục trong một ảnh. Trong khi đó, hình dạng không phải là một thuộc tính của ảnh. Nói tới hình dạng không phải là nhắc đến hình dạng của một ảnh. Thay vì vậy, hình dạng có khuynh hướng chỉ đến một khu vực đặc biệt trong ảnh, hay hình dạng chỉ là biên của một đối tượng nào đó trong ảnh.

Trong tìm kiếm ảnh theo nội dung, hình dạng là một cấp cao hơn so với màu sắc và kết cấu. Nó đòi hỏi sự phân biệt giữa các vùng để tiến hành xử lý về độ đo của hình dạng. Các hệ thống tìm kiếm ảnh theo nội dung thường khai thác hai nhóm biểu diễn hình dạng sau :

Biểu diễn hình dạng theo đường biên (contour-based descriptor) : Biểu diễn các đường biên bao bên ngoài và biểu diễn theo vùng (region-based descriptor): Biểu diễn một vùng toàn vẹn.

2.3.3.1 Độ đo tương đồng cho hình dạng

Độ đo về hình dạng rất nhiều trong phạm vi lý thuyết của bộ môn xử lý ảnh.

Chúng trải rộng từ những độ đo toàn cục dạng thô với sự trợ giúp của việc nhận dạng đối tượng, cho tới những độ đo chi tiết tự động tìm kiếm những hình dạng đặc biệt. Lược đồ hình dạng là một ví dụ của độ đo đơn giản. Kỹ thuật dùng đường biên hiệu quả hơn phương pháp trước, chúng tìm kiếm những hình dạng đối tượng gần giống với đường biên nhất. Phương pháp vẽ phác họa là phương pháp có nhiều đặc trưng rõ ràng hơn, không chỉ tìm kiếm những đường biên đối tượng đơn, mà còn đối với tập những đối tượng đã được phân đoạn trong một ảnh mà người dùng vẽ hay cung cấp.

2.3.4 Đặc trưng cục bộ bất biến

Người ta thường chia đặc trưng cục bộ thành 2 loại là những điểm trích xuất được từ điểm "nhô ra" (salient points) của ảnh và đặc trưng SIFT được trích chọn từ các điểm hấp dẫn Harris (interest points). Trong phần này, chúng tôi sẽ trình bày chi tiết về việc trích chọn các đặc trưng cục bộ bất biến (Scale Invariant Feature Transform SIFT) của ảnh.

Phần này trình bày phương pháp trích rút các đặc trưng cục bộ bất biến SIFT của ảnh. Các đặc trưng này bất biến với việc thay đổi tỉ lệ ảnh, quay ảnh, đôi khi là thay đổi điểm nhìn và thêm nhiễu ảnh hay thay đổi cường độ chiếu sáng của ảnh. Phương pháp được lựa chọn có tên là Scale-Invariant Feature Transform (SIFT) và đặc trưng trích rút được gọi là đặc trưng SIFT (SIFT

Feature). Các đặc trưng SIFT này được trích rút ra từ các điểm hấp dẫn cục bộ (Local Interest Point)

Điểm hấp dẫn (Interest Point (Keypoint)): Là vị trí (điểm ảnh) "hấp dẫn" trên ảnh. "Hấp dẫn" ở đây có nghĩa là điểm đó có thể có các đặc trưng bất biến với việc quay ảnh, co giãn ảnh hay thay đổi cường độ chiếu sáng của ảnh. Phương pháp trích rút các đặc trưng bất biến SIFT được tiếp cận theo phương pháp thác lọc, theo đó phương pháp được thực hiện lần lượt theo các bước sau:

Phát hiện các điểm cực trị Scale-Space (Scale-Space extrema detection):

Bước đầu tiên này tiến hành tìm kiếm các điểm hấp dẫn trên tất cả các tỉ lệ và vị trí của ảnh. Nó sử dụng hàm different-of-Gaussian để xác định tất cả các điểm hấp dẫn tiềm năng mà bất biến với quy mô và hướng của ảnh.

Định vị các điểm hấp dẫn (keypoint localization): Một hàm kiểm tra sẽ được đưa ra để quyết định xem các điểm hấp dẫn tiềm năng có được lựa chọn hay không?

Xác định hướng cho các điểm hấp dẫn (Orientation assignment): Xác định hướng cho các điểm hấp dẫn được chọn

Mô tả các điểm hấp dẫn (Keypoint descriptor): Các điểm hấp dẫn sau khi được xác định hướng sẽ được mô tả dưới dạng các vector đặc trưng nhiều chiều.

❖ Phương pháp trích chọn

- Phát hiện các điểm cực trị
- Định vị các điểm hấp dẫn
- Xác định hướng cho các điểm hấp dẫn
- Mô tả các điểm hấp dẫn

➤ **Phát hiện điểm cực trị**

❖ Định vị các vùng ảnh: tìm những khu vực chứa những tính chất giống nhau khi nhìn ảnh dưới các góc nhìn khác nhau

- Hàm không gian- tỷ lệ:

Tìm khu vực có khả năng bất biến với sự thay đổi tỷ lệ ảnh

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Số hóa bởi Trung tâm Học liệu – ĐHTN <http://www.lrc.tnu.edu.vn>

$I(x, y)$ Ảnh truy vấn

$G(x, y, \sigma)$ Biến tỷ lệ Gaussian

$L(x, y, \sigma)$ Hàm không gian tỷ lệ của ảnh I

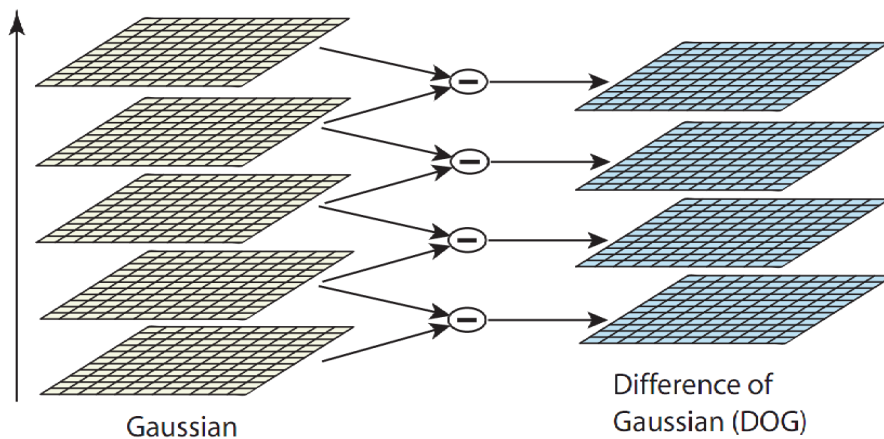
• Với

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

❖ Xem xét sự khác nhau của hàm không gian tỷ lệ giữa 2 tỷ lệ sai lệch nhau k lần

• Bộ lọc DoG:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$



Hình 2.8 Định nghĩa bộ lọc DoG

• Xấp xỉ hàm DoG

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G$$

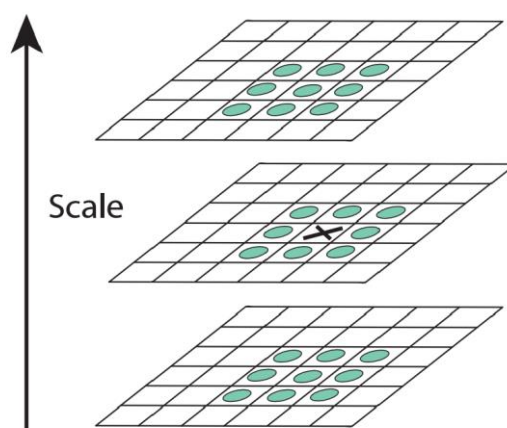
$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G$$

• **Xác định điểm hấp dẫn tiềm năng:**

So sánh mỗi điểm ảnh được xác định bởi hàm DoG với:

- 8 điểm ảnh xung quanh ở cùng tỷ lệ
- 9 điểm ảnh xung quanh ở các tỷ lệ liền trước
- 9 điểm ảnh xung quanh ở tỷ lệ liền sau.



Hình 2.9 Xác định điểm cực trị

→ Chọn là điểm hấp dẫn tiềm năng nếu nó đạt giá trị lớn nhất, hoặc nhỏ nhất.

➤ **Định vị điểm hấp dẫn**

❖ Phân tích điểm hấp dẫn tiềm năng: lấy các thông tin về:

- Vị trí ,
- Tỷ lệ,
- Tỷ lệ độ cong cơ sở (pricipal curvature).
- ❖ Loại bỏ điểm cực trị không phù hợp:
 - Điểm có độ tương phản thấp(không ổn định khi ảnh bị nhiễu)
 - Điểm ở những vị trí không thuận lợi dọc theo các cạnh.

❖ Phương pháp:

• Lấy mẫu: sử dụng các phương pháp lấy mẫu thích hợp để quyết định những điểm làm mẫu cho việc phân tích

- Khai triển Taylor cho bộ lọc DoG tại các điểm mẫu:

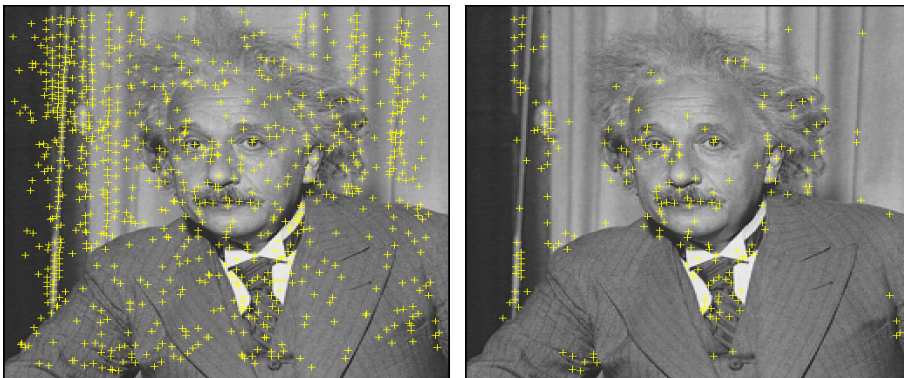
$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

$\mathbf{x} = (x, y, \sigma)^T$ Độ dịch so với các điểm lân cận của điểm lấy mẫu

- Vùng chứa điểm hấp dẫn được xác định $\hat{\mathbf{x}}$ qua
- Thực hiện tiếp tục với các điểm lấy mẫu khác.
- Những điểm có $\hat{\mathbf{x}}$ thỏa mãn (< 0.5) được thêm vào tập hợp mẫu tốt nhất, tiếp tục phân tích tiếp.
- Dùng $D(\hat{\mathbf{x}})$ để loại những điểm cực trị không ổn định (độ tương phản thấp).
- Thay $\hat{\mathbf{x}}$ vào $D(\hat{\mathbf{x}})$ ta được:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}}$$

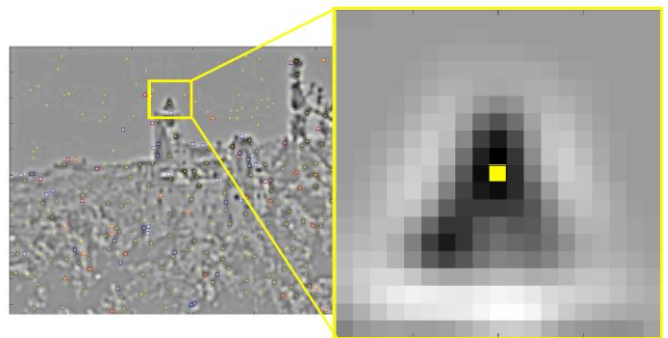
Nếu $D(\hat{\mathbf{x}}) < 0.03$ thì điểm lấy mẫu đó sẽ bị loại.



Hình 2.10 Định vị điểm hấp dẫn

➤ Xác định hướng cho các điểm hấp dẫn

- Dựa vào hướng của điểm hấp dẫn, biết được điểm hấp dẫn bất biến với sự quay ảnh.
- Tại mỗi điểm hấp dẫn, trích xuất một ảnh Gaussian (khung Gaussian) chứa các điểm lân cận điểm hấp dẫn đó.



- Keypoint location = extrema location
- Keypoint scale is scale of the DOG image

Hình 2.11 Trích ảnh Gaussian

- Tính toán hướng và độ lớn cho điểm hấp dẫn:

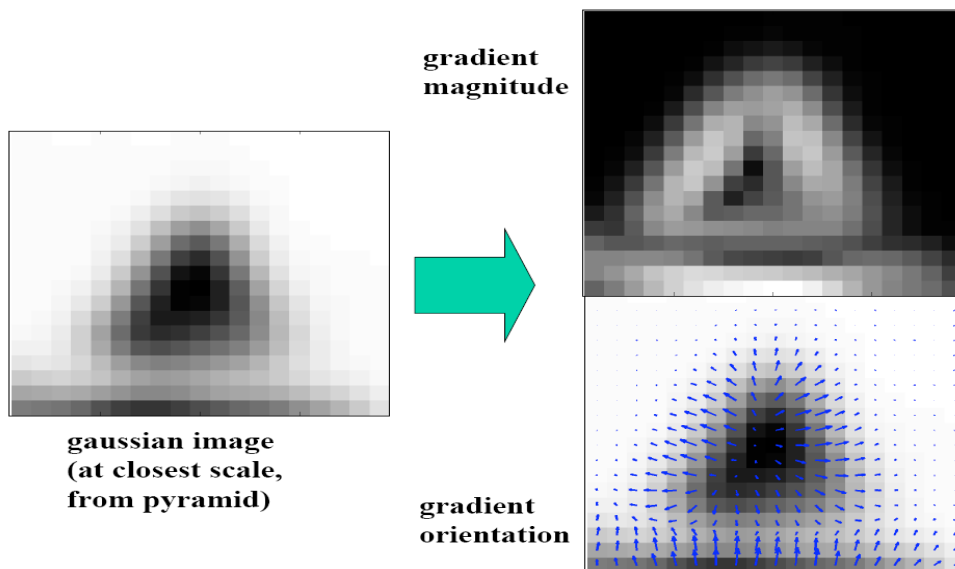
$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y)))$$

$m(x, y)$ Độ lớn của vector định hướng

$\theta(x, y)$ Hướng của vector định hướng (biểu diễn qua góc Θ)

$L(x, y)$ Ảnh Gaussian ở tỷ lệ nhỏ nhất

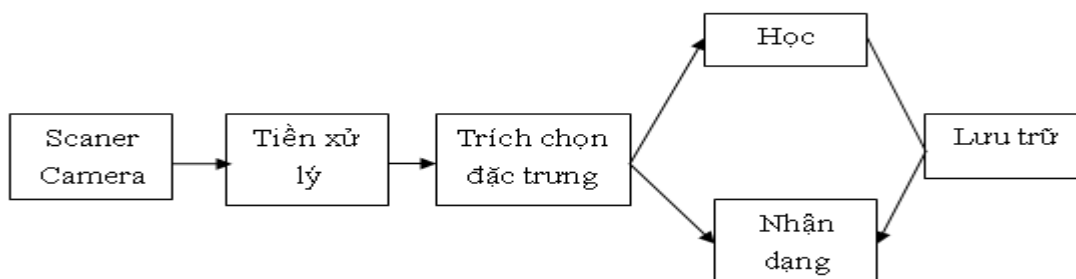


Hình 2.12 Tính độ lớn và hướng của gradient

2.4 NHẬN DẠNG ẢNH BIỂN SỐ XE

Bài toán nhận dạng biển số xe có nhiều ý nghĩa trong thực tế, nó giúp việc giám sát, quản lý, thống kê các phương tiện một cách dễ dàng, tiện lợi và nhanh chóng. Một số ứng dụng điển hình đã được triển khai trong thực tế như ứng dụng trong quản lý bãi đỗ xe thông minh, ứng dụng thu phí ở các trạm thu phí, ứng dụng phát hiện lỗi vi phạm giao thông một cách tự động. Trong phần này ta nghiên cứu về mặt kỹ thuật của một bài toán nhận dạng biển số xe, viết chương trình demo và các bước cần thiết để đưa bài toán vào ứng dụng thực tế. Giả sử ta đang xây dựng bài toán nhận dạng biển số xe ô tô, với đầu vào là một ảnh chứa biển số xe và đầu ra là một chuỗi kí tự của biển số đã được nhận dạng. Nếu quan sát bằng mắt người ta có thể dễ dàng nhận biết được một biển số, tuy nhiên với máy tính, đó là một điều không dễ dàng gì, nó rất dễ bị nhiễu, bị nhầm bởi các hình khối xung quanh tương tự, bởi các điều kiện thời tiết, góc độ ... Để hạn chế bớt được những khó khăn này, nhiều hệ thống nhận dạng trong thực tế thường giới hạn các điều kiện, chẳng hạn như camera thu ảnh được cố định ở một vị trí, xe được đi vào một khe hẹp nhất định... Có nhiều cách khác nhau để thực hiện bài toán nhận dạng này, ở các phần mềm thương mại hoàn chỉnh, nó là sự kết hợp và tối ưu của khá nhiều thuật toán phức tạp, trong bài này, ta đi theo hướng tiếp cận chia bài toán thành hai bài toán nhỏ: phát hiện biển số xe, cách ly kí tự và nhận dạng các kí tự.

Hệ thống nhận dạng gồm các khối chính là Tiền xử lý, Trích chọn đặc trưng, Học, Nhận dạng, Lưu trữ.



Hình 2.13 Sơ đồ khối của một hệ thống nhận dạng

2.4.1 Tiền xử lý

Trong khoa học máy tính, **tiền xử lý** là một chương trình xử lý các dữ liệu đầu vào thành các đầu ra. Các đầu ra này tiếp tục được sử dụng là đầu vào của một chương trình khác. Các đầu ra được coi là dạng tiền xử lý của dữ liệu đầu vào, thường được sử dụng bởi các chương trình tiếp theo như các trình biên dịch. Số lượng và các dạng xử lý làm được tùy thuộc vào tính tự nhiên của tiền xử lý; một số tiền xử lý chỉ có khả năng thực thi văn bản đơn giản và mở rộng vĩ mô, trong khi số khác có toàn quyền của một ngôn ngữ lập trình chính thức.

Tiền xử lý là giai đoạn đầu tiên trong xử lý ảnh số. Tùy thuộc vào quá trình xử lý tiếp theo trong giai đoạn này sẽ thực hiện các công đoạn khác nhau như:

- **Nắn chỉnh hình học:** Những biến dạng hình học thường là do các thiết bị điện tử và quang học gây ra. Do đó, phương pháp hiệu chỉnh ảnh dựa vào mô hình được mô tả dưới dạng biến đổi ảnh dạng $f(x, y)$ thành ảnh lý tưởng $f(x'', y'')$.

- **Khử nhiễu:** Nhiễu được chia làm hai loại: nhiễu hệ thống và nhiễu ngẫu nhiên. Đặc trưng của nhiễu hệ thống là tính tuần hoàn. Do vậy, có thể tách được loại nhiễu này bằng việc sử dụng biến đổi Fourier và loại bỏ các điểm đỉnh (peaks). Đối với nhiễu ngẫu nhiên, trường hợp đơn giản là các vết bản tương ứng với các điểm rất sáng hay rất tối, có thể khử bằng phương pháp nội suy, lọc trung bình và trung vị.

- **Chỉnh mức xám:** Đây là kỹ thuật nhằm chỉnh sửa tính không đồng đều của các thiết bị thu nhận hoặc tăng độ tương phản giữa các vùng ảnh.

- **Chỉnh tán xạ:** Ảnh nhận được từ các thiết bị điện tử hay quang học có thể bị nhòe. Phương pháp Fourier dựa trên tích chập của ảnh với hàm tán xạ cho phép giải quyết được bài toán hiệu chỉnh này.

2.4.1.1 Trích chọn đặc trưng ảnh

Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Việc trích chọn hiệu quả các đặc điểm giúp cho

việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

Tác giả có thể nêu ra một số đặc điểm của ảnh sau đây:

- **Đặc điểm không gian:** Phân bố mức xám, phân bố xác suất, biên độ, điểm uốn...

- **Đặc điểm biên và vùng biên:** Đặc trưng đường biên của đối tượng, trích chọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng.

2.4.2 Nhận dạng

Nhận dạng tự động, mô tả đối tượng, phân loại và phân nhóm mẫu là các vấn đề quan trọng trong thị giác máy tính, được ứng dụng trong nhiều ngành khoa học khác nhau. Hệ thống nhận dạng tự động bao gồm ba module tương ứng với ba giai đoạn chủ yếu: Thu nhận dữ liệu và tiền xử lý, biểu diễn dữ liệu, nhận dạng, ra quyết định.

Trong các ứng dụng không thể chỉ dùng một cách tiếp cận đơn lẻ để phân loại “tối ưu”. Do vậy cần sử dụng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Các phương thức phân loại tổ hợp hay được dùng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybird system) bao gồm nhiều mô hình kết hợp.

CHƯƠNG III: CHƯƠNG TRÌNH NHẬN DẠNG BIỂN SỐ XE Ô TÔ CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO

3.1. MÔ TẢ VÀ GIỚI THIỆU VỀ BIỂN SỐ XE Ô TÔ CỦA CỘNG HÒA DÂN CHỦ NHÂN DÂN LÀO

Phần này tập trung trình bày về bài toán nhận dạng biển số xe của Cộng hòa Dân chủ Nhân dân Lào. Hệ thống này sẽ nhận dạng được các ký tự và số trên biển số xe, có thể nhận biết được xe đó là thuộc tỉnh, huyện và cơ quan nào, còn biết được cả thời gian phát hiện biển số xe, thời gian nhận dạng các ký tự và tổng kết cả hai thời gian cộng với nhau.

Trong chương trình nhận dạng này là nhận dạng ảnh biển số xe từ máy chụp camera trên đường vì bên Lào chưa có chương trình nhận dạng này chỉ có lấy ảnh chụp và tìm dữ liệu xe trong công an tỉnh. Chương trình này sẽ được sử dụng trong các khu vực như: chỗ gửi xe, trường học, bệnh viện và trong viễn thông để giúp cảnh sát giao thông tìm xe được dễ hơn.

Biển số xe của bên Lào trong thời gian tới sẽ có sự thay đổi về cách đánh mã số, ký hiệu trên biển số xe của nước Cộng hòa Dân chủ Nhân dân Lào, kiểu biển mới sẽ gồm các số và ký tự chữ cái A, B, C,... ký hiệu sẽ bảo tên các tỉnh trong đất nước Lào và có tất cả 4 màu như là: đỏ, trắng, vàng, xanh dương.

Màu đỏ (xe quân đội, xe công an).

Màu trắng chữ đen (xe tạm thời, doanh nghiệp)

Màu trắng chữ xanh (xe công ty tư nhân, đại sứ quán)

Màu vàng (xe tư nhân)

Màu xanh dương (xe cơ quan)

Biển số xe ô tô của Lào gồm 2 biển giống nhau cả chữ và số trong biển, một biển gắn phía trước và một biển gắn phía sau xe Có kích thước như sau:



Số hóa bởi Trung tâm Học liệu – 460 mm [tp://www.lrc.tnu.edu.vn](http://www.lrc.tnu.edu.vn)

Biển số xe mới sẽ được thay đổi có các ký hiệu, các số và ký tự chữ cái tiếng Anh như:

EP> Enterprise (Doanh nghiệp)

ES> Embassy (Đại sứ quán)

GM> Government (Cơ quan)

PC> Private company (Công ty tư nhân)

PL> Police (Công an)

PV> Private (Tư nhân)

SD> Soldier (Quân đội)

TP> Temporary (Tạm thời)

VD: Biển số xe của xe tư nhân.



Số **17** là cho biết một tên của tỉnh trong tất cả 18 tỉnh (là tỉnh Xiengkhaung), Chữ **PV** (private) là cho biết được biển số xe này thuộc trong ký hiệu nào (là xe tư nhân), Số **4** là cho biết một tên của huyện trong tỉnh đó (là huyện Khoun) và 4 số **2122** sau cùng là số của xe.

3.2 THƯ VIỆN OPENCV

3.2.1 OpenCV là gì?

OpenCV là viết tắt của Open Source Computer Vision, là một thư viện mở gồm các hàm được xây dựng phục vụ cho việc xử lý thị giác máy thời gian thực (Real time computer vision). Các thuật toán xử lý ảnh thông thường lẫn cao cấp đều được tối ưu hóa bởi các nhà phát triển thư viện thành các hàm đơn giản và rất dễ sử dụng. OpenCV hỗ trợ hai ngôn ngữ chính: C/C++ và python.. Thư viện của nó được viết bằng C và C++ và chạy trong nền Linux, Window và Mac OS X.

OpenCV được thiết kế cho hệ thống tính toán hiệu quả và tập trung lớn vào các ứng dụng thời gian thực. OpenCV được viết bằng C để có thể tận dụng tối đa và tối ưu bộ vi xử lý.

Một trong những mục tiêu của OpenCV là cung cấp một nền đơn giản sử dụng thị giác máy tính, giúp người xây dựng các ứng dụng phức tạp thực hiện nhanh hơn. Thư viện OpenCV chứa hơn 500 hàm là cầu nối cho nhiều phần trong thị giác máy tính bao gồm cả nhà máy kiểm tra sản phẩm, hình ảnh y tế, an ninh, giao diện người dùng, hiệu ứng máy ảnh, tầm nhìn stereo và robot. Bởi vì thị giác máy tính và máy học luôn bắt tay với nhau.

OpenCV cũng chứa đầy đủ các mục đích chung của thư viện máy học – MLL. Đây là thư viện nhỏ tập trung vào nhận dạng mẫu thống kê và phân nhóm. MLL rất hữu ích cho các tác vụ thị giác, đó là cốt lõi của OpenCV, nhưng nó đủ khái quát để được sử dụng cho bất kỳ vấn đề máy học nào.

3.2.2 Những ai sử dụng OpenCV

Hầu hết các nhà khoa học máy tính và lập trình nhận thức được một vài khía cạnh mà tầm nhìn máy tính đưa ra. Nhưng ít người nhận thức được đầy đủ các cách thức mà tầm nhìn máy tính sử dụng.

Thư viện mã nguồn mở của OpenCV đã được cấu trúc để bạn có thể xây dựng một sản phẩm thương mại bằng cách sử dụng tất cả hoặc một phần của OpenCV.

OpenCV phổ biến trên thế giới với đông đảo người sử dụng như ở Trung Quốc, Nhật Bản, Nga, Châu Âu, Israel.

Kể từ khi phát hành phiên bản đầu tiên vào năm 1999, OpenCV đã được sử dụng trong nhiều ứng dụng, trong nhiều sản phẩm và trong nhiều nghiên cứu. Các ứng dụng này bao gồm các hình ảnh quét liên kết, hình ảnh y tế giảm nhiễu, phân tích đối tượng, bảo mật và phát hiện xâm nhập hệ thống, giám sát tự động và hệ thống an toàn, hệ thống kiểm tra sản xuất, hiệu ứng máy ảnh và các ứng dụng quân sự. Nó thậm chí còn được sử dụng với âm thanh. OpenCV là một phần chính của thị giác máy tính.

3.2.3 Nguồn gốc của OpenCV

OpenCV được đưa ra từ một nghiên cứu của Intel để nâng cao các ứng dụng chuyên sâu CPU.

Từ mục đích đó, Intel tung ra nhiều dự án bao gồm các dự án ray tracing thời gian thực và hiển thị trong không gian 3D. Một trong những tác giả làm việc cho Intel ở thời điểm đó đã đến thăm trường đại học và nhận thấy rằng một số nhóm các trường đại học hàng đầu như MIT Media Lab phát triển tốt và có cơ sở hạ tầng về tầm nhìn máy tính – code mà đã được các sinh viên thông qua để các sinh viên có một sự khởi đầu mới có giá trị trong việc phát triển các ứng dụng thị giác cho riêng mình. Thay bằng việc làm lại từ đầu, một sinh viên mới có thể bắt đầu bằng cách xây dựng dựa trên những cái đã có sẵn.

Vì vậy, OpenCV đã được hình thành như là một cách để làm cho cơ sở hạ tầng của thị giác máy tính trở nên tiện ích hơn.

3.3 BỘ THƯ VIỆN MÃ NGUỒN MỞ OPENCV

3.3.1 Một vài thư viện của openCV

❖ CXCORE bao gồm các dạng dữ liệu cơ bản rõ ràng. Ví dụ cấu trúc dữ liệu về ảnh, điểm, hình chữ nhật được xác định trong file `cxtypes.h`. CXCORE chứa các phép toán đại số tuyến tính và thống kê, các hàm lưu trữ lâu dài (persistence fun...) và các lỗi thao tác. Các hàm đồ họa được dùng cho việc vẽ ảnh cũng được lưu trữ tại đây.

❖ CV chứa đựng quá trình xử lý ảnh và các phương pháp đánh giá sơ bộ kích thước ảnh. Những hàm tính toán hình học cũng được lưu trữ tại đây.

❖ CVAUX được mô tả trong OpenCV như là modul cũ và chỉ dùng để thí nghiệm. Tuy nhiên, giao diện đơn giản nhất cho nhận dạng mặt được nằm trong modul này. Những mã nguồn nằm trong module này rất phù hợp cho việc nhận dạng mặt và chúng được sử dụng rộng rãi cho mục đích này

❖ Những hàm còn lại được nằm trong HighGUI và CVCAM. Cả hai đều nằm ở thư mục có tên “otherlibs”, sử dụng chúng rất dễ gặp lỗi. Vì HighGUI chứa các thư viện vào ra cơ bản. Nó cũng chứa đựng nhiều cửa sổ đa chức năng.

❖ CVCAM là thư viện chứa các hàm truy nhập video thông qua DirectX trên môi trường Window 32 bit. Tuy nhiên, HighGUI cũng có các giao diện video. Nếu bạn sử dụng Window XP hoặc 2000, bạn có thể làm tăng hiệu quả

của nó bằng cách chuyển qua giao diện CVCAM, nhưng học OpenCV sẽ đơn giản hơn khi dùng HighGUI.

❖ ML: các giải thuật học máy và gom nhóm.

3.3.2 Các thao tác ảnh cơ bản

Load từ file ảnh

Lấy file ảnh hiển thị ra trong 1 cửa sổ

```
#include "highgui.h"
```

```
int main()
```

```
{
```

```
1. IplImage* img = cvLoadImage( "tu nhan copy.jpg" );
```

```
2. cvNamedWindow( "test 1", CV_WINDOW_AUTOSIZE );
```

```
3. cvShowImage( "test 1", img );
```

```
4. cvWaitKey(0);
```

```
5. cvReleaseImage( &img );
```

```
6. cvDestroyWindow( "test 1" );
```

```
}
```

❖ Dòng 1.

- **IplImage**: Kiểu cấu trúc lưu giữ thông tin file ảnh cần load, như: Định dạng, kích thước, dung lượng, màu sắc,...

- **Hàm cvLoadImage()** lấy thông số hàm là ảnh được load. Hàm này load và quyết định định dạng file dựa vào file name nên cần chỉ rõ định dạng file, vd: .jpg, và đường dẫn có thể là tương đối hoặc tuyệt đối để xác định vị trí file. Hàm này tự động cấp phát bộ nhớ cần thiết để lưu trữ ảnh. Hàm có thể đọc được nhiều định dạng: BMP, DIB, JPEG, JPE, PNG, PBM, PGM, PPM, SR, RAS, và TIFF

❖ **Dòng 2**: Hàm cvNamedWindow () thiết lập cửa sổ để hiển thị ảnh.

- **Thông số 1**: Tên cửa sổ

- **Thông số 2**: Thuộc tính cửa sổ, trong vd **CV_WINDOW_AUTOSIZE** tức là kích thước cửa sổ sẽ vừa khít với ảnh được hiển thị. Nếu thông số này thiết lập là 0 thì cửa sổ ở chế độ mặc định.

❖ **Dòng 3:** Biến `img` ở dòng 1 đang lưu giữ thông tin của ảnh, sẽ được hiển thị vào cửa sổ đã được tạo ở dòng 2 qua hàm `cvShowImage()`

❖ **Dòng 4:** Hàm `cvWaitKey()` yêu cầu chờ 1 phím bất kỳ do người dùng nhập để thoát, **Nếu đối số là dương** (vd: 3) thì chương trình chờ trong 3ms mà ko có phím nào được bấm thì chương trình tiếp tục. **Nếu đối số truyền vào ≤ 0** , chương trình **chờ vô hạn** đến khi có phím bấm thì thoát

❖ **Dòng 5, 6:** Hàm xóa rác.

Đối với chương trình xử lý ảnh, việc giải phóng bộ là cần thiết. Do đó, nên hình thành thói quen giải phóng bộ nhớ trước khi làm việc với chương trình.

3.4 HÀM OPENCV HỖ TRỢ XÂY DỰNG ỨNG DỤNG NHẬN DẠNG BIÊN SỐ XE.

3.4.1 Hàm `cvLoadImage`

Cú pháp

```
cvLoadImage(const char* filename, int  
flags=CV_LOAD_IMAGE_COLOR)
```

Trong đó:

+ `Filename`: Tên của file được load

+ `flags`: Chỉ ra màu sắc và độ sâu của ảnh được load:

- Màu sắc được chỉ ra trong ảnh được chuyển sang 3 kênh

(`CV_LOAD_IMAGE_COLOR`), 1 kênh (`CV_LOAD_IMAGE_GRAYSCALE`), hoặc phía trái như ảnh trong file đầu vào(`CV_LOAD_IMAGE_ANYCOLOR`).

- Độ sâu được chỉ ra trong ảnh được load được chuyển sang 8bit/pixel/color channel như đã quy định trong phiên bản trước của OpenCV hoặc bên trái như chúng ta thấy trong file đầu vào. Nếu hàm `CV_LOAD_IMAGE_ANYDEPTH` được chuyển sang định dạng pixel có thể là 8 bit unsigned, 16 bit unsigned, 32 bit signed hoặc 32 bit floating point.

Chức năng của hàm: Hàm `cvLoadImage` lấy thông số hàm là ảnh được load. Hàm này load và quyết định định dạng file dựa vào file name nên cần chỉ rõ

định dạng file, vd: .jpg, và đường dẫn có thể là tương đối hoặc tuyệt đối để xác định vị trí file.

3.4.2 Hàm *cvReleaseImage*

Cú pháp: void cvReleaseImage(IplImage** image);

Trong đó: Image: Con trỏ dạng double trỏ tới header của deallocate image.

Chức năng của hàm: là giải phóng header và dữ liệu ảnh. Đây là một form rút gọn:

```
if( *image )
{
cvReleaseData( *image );
cvReleaseImageHeader( image );
}
```

3.4.3 Hàm *cvCreateImage*

Cú pháp: cvCreateImage(CvSize size, int depth, int channels);

Trong đó:

+ Size: Chỉ kích thước chiều cao và chiều rộng của ảnh cần tạo

+ Depth: giá trị bit biểu diễn cho mỗi pixel. Có thể là:

IPL_DEPTH_8U - unsigned 8-bit integers

IPL_DEPTH_8S - signed 8-bit integers

IPL_DEPTH_16U - unsigned 16-bit integers

IPL_DEPTH_16S - signed 16-bit integers

IPL_DEPTH_32S - signed 32-bit integers

IPL_DEPTH_32F - single precision floating-point numbers

IPL_DEPTH_64F - double precision floating-point numbers

+ Channels: Số của kênh/mỗi phần tử ảnh (pixel). Có thể là 1, 2, 3, hoặc 4.

Các kênh được xen kẽ, ví dụ như cách bố trí dữ liệu thông thường của một ảnh màu là:

b0 g0 r0 b1 g1 r1 ...

Chức năng của hàm cvCreateImage: Là tạo header và phân bổ dữ liệu.

Đây là một form rút gọn của

```
header = cvCreateImageHeader(size,depth,channels);  
cvCreateData(header);
```

3.4.4 Hàm cvCvtColor

Cú pháp: void cvCvtColor(const CvArr* src, CvArr* dst, int code);

Trong đó:

+ Src: ảnh nguồn có thể là 8 – bit(8u), 16 – bit(16u), hoặc single-precision floating-point (32f)

+ Dst: kiểu dữ liệu của ảnh nguồn và ảnh đích giống nhau. Số kênh có thể khác nhau.

+ Code: Để chuyển đổi màu sắc có thể sử dụng:

CV_*<src_color_space>*2*<dst_color_space>* constants (bên dưới).

Chức năng của hàm cvCvtColor: Là chuyển ảnh đầu vào từ không gian màu này sang không gian màu khác. Hàm này không quan tâm đến trường colorModel và channelSep của phần đầu IplImage, vì vậy không gian màu của ảnh nguồn nên được chỉ rõ (gồm thứ tự của các kênh trong trường hợp không gian màu RGB, ví dụ BGR có nghĩa là định dạng 24 bit với layout B₀ G₀ R₀ B₁ G₁ R₁ ... , trong khi đó RGB nghĩa là định dạng 24 bit với layout R₀ G₀ B₀ R₁ G₁ B₁ ...).

Quy ước khoảng giá trị cho kênh R, G, B là:

- 0..255 cho ảnh 8 - bit
- 0..65535 cho ảnh 16 - bit và
- 0..1 cho ảnh floating-point.

Thật vậy, trong trường hợp biến đổi tuyến tính phạm vi có thể tùy ý, nhưng để có được kết quả chính xác trong trường hợp biến đổi phi tuyến tính, ảnh đầu vào nên được thu nhỏ (nếu cần).

3.4.5 Hàm *cvNamedWindow*

Cú pháp:

```
int cvNamedWindow( const char* name, int  
flags=CV_WINDOW_AUTOSIZE );
```

Trong đó:

+ Name: Tên của cửa sổ được dùng như nhận diện cửa sổ và xuất hiện trong cửa sổ chú thích.

+ flags

Flags của cửa sổ. Hiện tại, chỉ được hỗ trợ flag là CV_WINDOW_AUTOSIZE. Nếu nó được thiết lập, kích thước cửa sổ được tự động điều chỉnh để phù hợp với hình ảnh hiển thị (xem *cvShowImage*), trong khi người dùng không thể thay đổi kích thước của cửa sổ bằng tay.

Chức năng của hàm *cvNamedWindow*: Là thiết lập cửa sổ để hiển thị hình ảnh. Cửa sổ được tạo được gọi bằng tên của chúng. Nếu cửa sổ với tên như vậy đã tồn tại, hàm *cvNamedWindow* sẽ không tạo ra cửa sổ nào.

3.4.6 Hàm *cvShowImage*

Cú pháp: void *cvShowImage*(const char* name, const CvArr* image);

Trong đó:

+ Name: Tên của cửa sổ.

+ Image: ảnh hiển thị.

Chức năng của hàm *cvShowImage*: là hiển thị hình ảnh trong cửa sổ chỉ định. Nếu cửa sổ được tạo ra với flag CV_WINDOW_AUTOSIZE thì ảnh được hiển thị với kích thước ban đầu của nó, nếu không ảnh là thu nhỏ để phù hợp với cửa sổ.

3.4.7 Hàm *cvResizeWindow*

Cú pháp: void *cvResizeWindow*(const char* name, int width, int height);

Trong đó:

+ Name: Tên của cửa sổ được định lại kích thước.

+ Width: Chiều rộng của cửa sổ mới.

+ Height: Chiều cao của cửa sổ mới.

Chức năng của hàm cvResizeWindow: Là thay đổi kích thước của cửa sổ.

3.4.8 Hàm cvThreshold

Cú pháp: void cvThreshold(const CvArr* src, CvArr* dst, double threshold, double max_value, int threshold_type);

Trong đó:

+ Src: mảng nguồn (kênh đơn, 8 – bit của 32 – bit floating point)

+ Dst: mảng đích; mảng đích phải cùng kiểu với mảng nguồn hoặc mảng đích có 8 – bit.

+ Threshold: giá trị phân ngưỡng

+ Max value

Giá trị tối đa để phân ngưỡng sử dụng với hàm

CV_THRESH_BINARY_INV.

+ Threshold_type: Kiểu phân ngưỡng type .

Chức năng của hàm cvThreshold: là áp dụng một mức phân ngưỡng cố định cho mảng kênh đơn. Hàm này thường được sử dụng để đưa ra ảnh nhị phân từ một ảnh xám (hàm cvCmpS cũng có thể sử dụng để chuyển từ ảnh xám sang ảnh nhị phân) hoặc để xóa nhiễu. Ví dụ để lọc ra các điểm ảnh có giá trị quá lớn hoặc quá nhỏ. Có một vài kiểu phân ngưỡng hỗ trợ các chức năng được xác định bởi threshold_type:

- threshold_type=CV_THRESH_BINARY:

$dst(x,y) = max_value$, if $src(x,y) > threshold$

0, otherwise

- threshold_type=CV_THRESH_BINARY_INV:

$dst(x,y) = 0$, if $src(x,y) > threshold$

max_value , otherwise

- threshold_type=CV_THRESH_TRUNC:

$dst(x,y) = threshold$, if $src(x,y) > threshold$

$src(x,y)$, otherwise

- threshold_type=CV_THRESH_TOZERO:

dst(x,y) = src(x,y), if src(x,y)>threshold

0, otherwise

- threshold_type=CV_THRESH_TOZERO_INV:

dst(x,y) = 0, if src(x,y)>threshold

src(x,y), otherwise

3.4.9 Hàm cvFindContours

Cú pháp: int cvFindContours(CvArr* image, CvMemStorage* storage, CvSeq** first_contour, int header_size=sizeof(CvContour), int mode=CV_RETR_LIST, int method=CV_CHAIN_APPROX_SIMPLE, CvPoint offset=cvPoint(0,0));

Trong đó:

+ Image: Ảnh nguồn kênh duy nhất 8 – bit. Các pixel trong ảnh chỉ có giá trị 0 hoặc 1 - ảnh đó được gọi là ảnh nhị phân. Để tạo ra được một ảnh nhị phân từ một ảnh xám, người ta có thể sử dụng hàm cvThreshold, hàm cvAdaptiveThreshold hoặc hàm cvCanny. Các hàm có chức năng sửa đổi nội dung ảnh nguồn.

+ Storage: Chứa các đường biên đã tìm được

+ Header_size: Kích thước của tiêu đề trình tự.

+ mode: Hình thức tìm kiếm:

o CV_RETR_EXTERNAL - kiểm với các đường biên ngoài nhất.

o CV_RETR_LIST - tìm kiếm tất cả các đường biên và đưa chúng vào danh sách.

o CV_RETR_CCOMP - tìm kiếm tất cả các đường biên và tổ chức chúng theo cây phân cấp có 2 mức: mức cao nhất là đường biên ngoài cùng của các thành phần, các mức thứ hai là đường biên bounda của các hole.

o CV_RETR_TREE - tìm kiếm tất cả các đường biên và dựng lại các cây phân cấp đầy đủ của các đường biên lồng nhau.

+ method

Phương pháp tính xấp xỉ (cho tất cả các phương thức, trừ CV_RETR_RUNS, trong đó sử dụng tích hợp tính xấp xỉ)

◦CV_CHAIN_CODE - đường biên ở đầu ra trong mã chuỗi Freeman. Tất cả các phương pháp khác đầu ra đa giác (trình tự của các đỉnh).

◦CV_CHAIN_APPROX_NONE – dịch tất cả các điểm từ mã chuỗi thành điểm.

+ offset: Offset, theo đó mỗi điểm biên là chuyển dịch. Điều này rất hữu ích nếu các đường biên được chiết xuất từ các hình ảnh ROI và sau đó chúng nên được phân tích trong toàn bộ hình ảnh.

Chức năng của hàm cvFindContours: là lấy đường nét với ảnh nhị phân và trả về số các đường biên tìm thấy. Các con trỏ first_contour được hàm điền. Nó sẽ chứa các con trỏ tới hầu hết các đường biên bên ngoài nhất hoặc là NULL nếu không có đường biên nào được phát (nếu ảnh là hoàn toàn đen). Các đường biên khác có thể đạt được từ first_contour sử dụng các liên kết h_next và v_next. Các mẫu trong phần thảo luận về hàm cvDrawContour cho thấy làm sao để sử dụng các đường biên để phát hiện ra các thành phần kết nối. Đường biên cũng có thể được sử dụng để phân tích hình dạng và tổ chức đối tượng – xem trong squares.c trong thư mục mẫu của OpenCV.

3.4.10 Hàm cvBoundingRect

Cú pháp: CvRect cvBoundingRect(CvArr* points, int update=0);

Trong đó:

+ points: Một tập hợp điểm 2D, biểu diễn như một chuỗi (*CvSeq, *CvContour) hoặc vector (cvMat *) của các điểm, hoặc kênh đơn mặt nạ ảnh 8 – bit (* CvMat, IplImage), các điểm ảnh khác 0 được xem xét.

+ update: Cập nhật flag. Đây là danh sách các kết hợp có thể có của các giá trị flag và loại đường biên:

•Các điểm là Cvcontour *, cập nhật = 0: ranh giới của hình chữ nhật không được tính, nhưng nó được đọc từ trường rect của header của đường biên.

• Các điểm là Cvcontour *, cập nhật = 1: các hình chữ nhật ranh giới được tính và ghi vào trường rect của header của đường biên. Ví dụ, hình thức này được sử dụng bởi hàm cvFindContours.

• Các điểm là CvSeq * hoặc CvMat *: cập nhật bị bỏ qua, ranh giới của hình chữ nhật được tính toán và trả lại.

Chức năng của hàm cvBoundingRect: Là trả về ranh giới hình chữ nhật phía trên, biên phải cho điểm thiết lập trong không gian 2d.

3.4.11 Hàm cvCircle

Hàm này dùng để vẽ đường tròn.

Cú pháp:

```
void cvCircle( CvArr* img, CvPoint center, int radius, CvScalar color,  
              int thickness=1, int line_type=8, int shift=0 );
```

Trong đó:

Img: Ảnh.

Center: Tâm của đường tròn.

Radius: Bán kính.

Color: Màu

Thickness: Độ dày của đường tròn.

Line_type: Kiểu vẽ

Shift: Số bit phân kênh giữa tâm tọa độ và bán kính.

Đường tròn này sẽ bị cắt bởi hình chữ nhật ROI. Để vẽ màu viền của đường tròn ta sử dụng macro CV_RGB(r , g , b).

3.4.12 Hàm cvMemStorage

Cú pháp: CvMemStorage* cvCreateMemStorage(int block_size = 0);

Chức năng: Hàm này có chức năng như một đối số có kích thước khối, hàm này đưa ra kích thước của khối nhớ bên trong vùng lưu trữ. Nếu đối số này được đặt thành 0 thì kích thước khối mặc định 64kB sẽ được sử dụng. Hàm này trả về một con trỏ trỏ đến một vùng lưu trữ mới.

3.4.13 Hàm *cvBoundingRect*

Cú pháp:

`Cv Rect cvBoundingRect (CvArr* points, int update = 0);`

Trong đó:

+ `point`: Các điểm sử dụng cho các đối số đầu tiên có thể là một đường biên (`CvContour*`) hoặc ma trận hai chiều (`CvMat*`) chứa các điểm tuần tự.

+ `update`: Cập nhật. Nếu bạn gọi hàm `cvBoundingRect` với `update = 0` thì bạn chỉ nhận được nội dung của dữ liệu thành phần. Nhưng nếu bạn gọi hàm `cvBoundingRect` với `update = 1`; Đường bao quanh sẽ được tính và các thành phần dữ liệu liên quan cũng sẽ được cập nhật.

Chức năng: Hàm này có chức năng là tạo ra đường biên hình chữ nhật. Nó sẽ trả về một `cvRect`.

Một vấn đề với đường biên hình chữ nhật thu được từ hàm `cvBoundingRect` là nó là một `cvRect` và như vậy cũng có thể biểu diễn cho một hình chữ nhật mà các cạnh được định hướng theo chiều ngang và chiều dọc.

3.4.14 Hàm *cvDrawContours*

Một trong những nhiệm vụ cơ bản là vẽ đường biên trên mà hình. Để thực hiện điều này chúng ta có hàm `cvDrawContours`:

`void cvDrawContours(CvArr* img, CvSeq* contour, CvScalar external_color, CvScalar hole_color, int max_level, int thickness = 1, int line_type = 8, CvPoint offset = cvPoint(0,0));`

+ `img`: Nó là hình ảnh để vẽ đường biên.

+ `contour`: Đường biên thì không phải là đơn giản như ta thấy. Trong trường hợp cụ thể nó được xem như là nút gốc của cây đường biên.

Đối số khác như `primarily max – level` sẽ phát hiện được những việc cần thực hiện với phần còn lại của cây.

+ `external_color`: Màu sắc để vẽ đường biên.

+ `hold_color`: Khi có đường biên bất kỳ được đánh dấu được xem như một “hold” thì `hold_color` đưa với màu sắc thay thế.

+ `max_level` của hàm `cvDrawContours` cho ta biết làm thế nào để xử lý đường biên theo nghĩa là nút biến trên cây. Tham số này có thể được thiết lập để cho biết độ sâu tối đa có thể sử dụng trong bản vẽ. Vì vậy, nếu `max_level=0` nghĩa là tất cả các đường biên cùng cấp là cùng đầu vào (chính xác hơn là các đường biên và các đường biên bên cạnh) được đưa ra. Nếu `max_level = 1` nghĩa là các đường biên cùng cấp như cùng đầu vào và con cái của chúng được đưa ra.

+ `thickness`: chỉ ra độ dày của đường biên được vẽ.

+ `line_type`: là kiểu của đường biên được đưa ra bởi hàm `cvDrawContours`.

3.5 VISUAL C/C++

Microsoft Visual C++ là một sản phẩm Môi trường phát triển tích hợp (IDE) cho các ngôn ngữ lập trình C, C++, và C++/CLI của microsoft. Nó có các công cụ cho phát triển và gỡ lỗi mã nguồn C++, đặc biệt là các mã nguồn viết cho Microsoft Windows API, DirectX API, và Microsoft.NET Framework.

Các chức năng của Visual C++ như tô sáng cú pháp, IntelliSense (chức năng về tự động hoàn thành việc viết mã) và các chức năng gỡ lỗi tiên tiến.

Ví dụ, nó cho phép gỡ lỗi từ xa sử dụng một máy tính khác và cho phép gỡ lỗi bằng cách duyệt qua từng dòng lệnh tại một thời điểm. Chức năng "biên tập và tiếp tục" cho phép thay đổi mã nguồn và dịch lại chương trình trong quá trình gỡ lỗi, mà không cần phải khởi động lại chương trình đang được gỡ lỗi.

Đặc trưng biên dịch và xây dựng hệ thống, tính năng tiên biên dịch các tập tin đầu đề (*header files*) và liên kết tĩnh tiến (*incremental link*) - chỉ liên kết những phần bị thay đổi trong quá trình xây dựng phần mềm mà không làm lại từ đầu: Những đặc trưng về tính năng này thuyên giảm tổng thời gian biên tập, biên dịch và liên kết chương trình phần mềm, đặc biệt đối với những đề án phần mềm lớn.

3.6. GIAO DIỆN CỦA CHƯƠNG TRÌNH NHẬN DẠNG BIỂU SỐ XE.

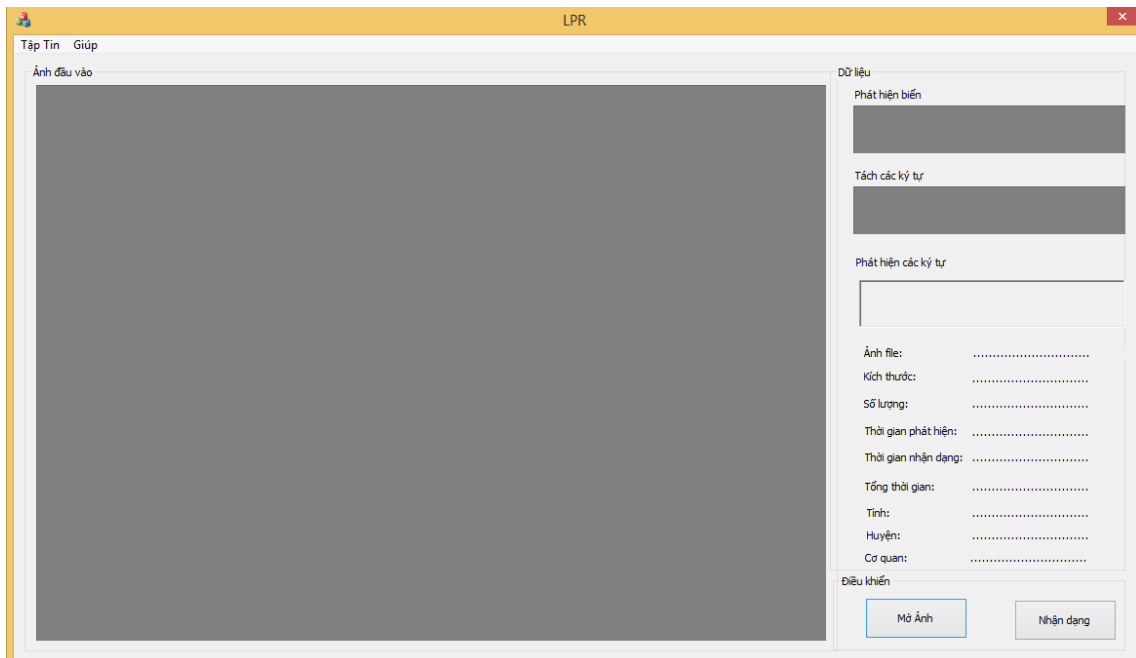
Trong phần này, trình bày một số giao diện chính của chương trình nhận dạng ảnh biển số xe. Với một ảnh đầu vào biển số xe sau:



Hình 3.1 Ảnh biển số xe đầu vào

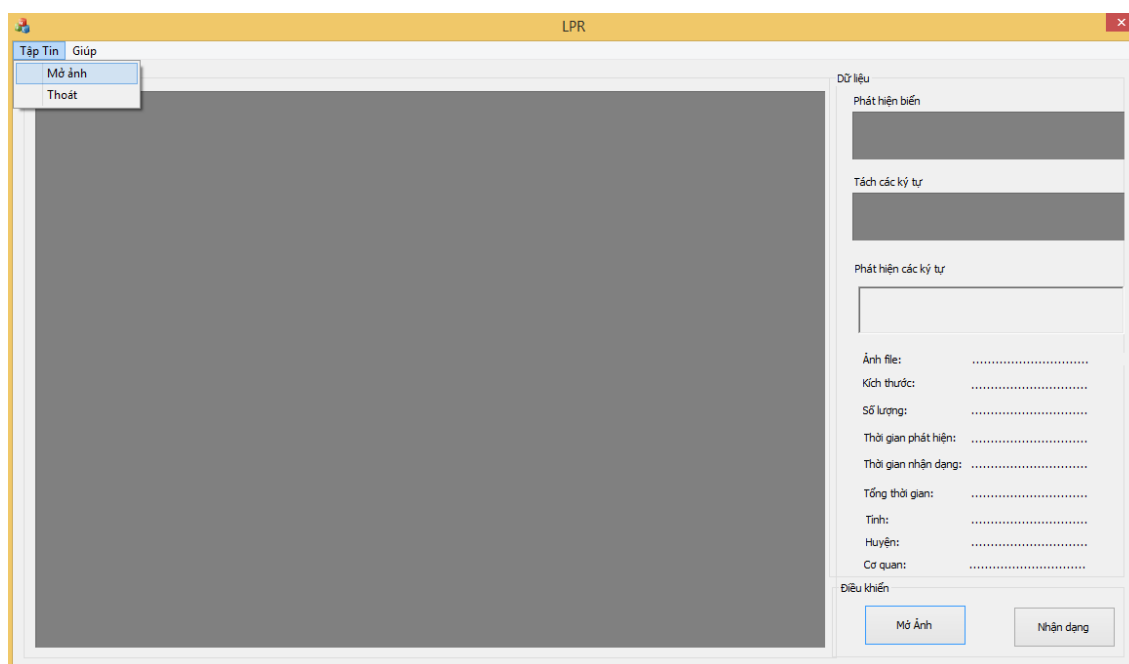
Trên cơ sở ảnh biển số xe, tiến hành đưa ảnh vào chương trình phần mềm nhận dạng:

1) Mở chương trình LPR ra là được hình sau:

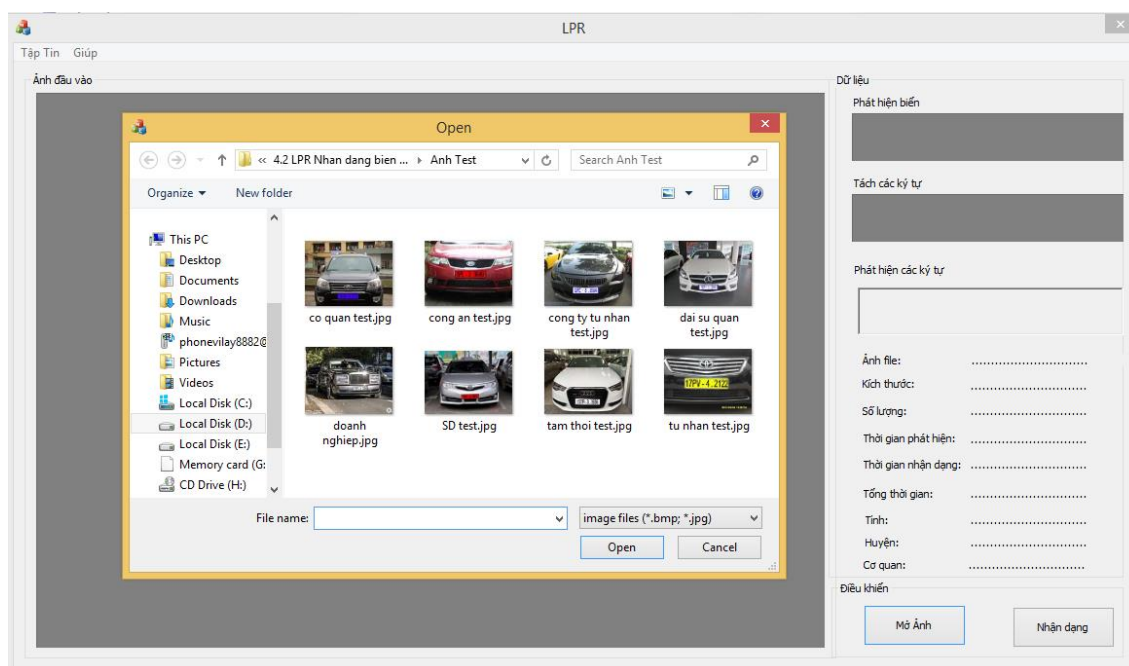


Hình 3.2 Giao diện của chương trình

2) Sau đó chọn ảnh trên Menu (Tập Tin)>(Mở ảnh) hay chọn ảnh trên button (Mở Ảnh)

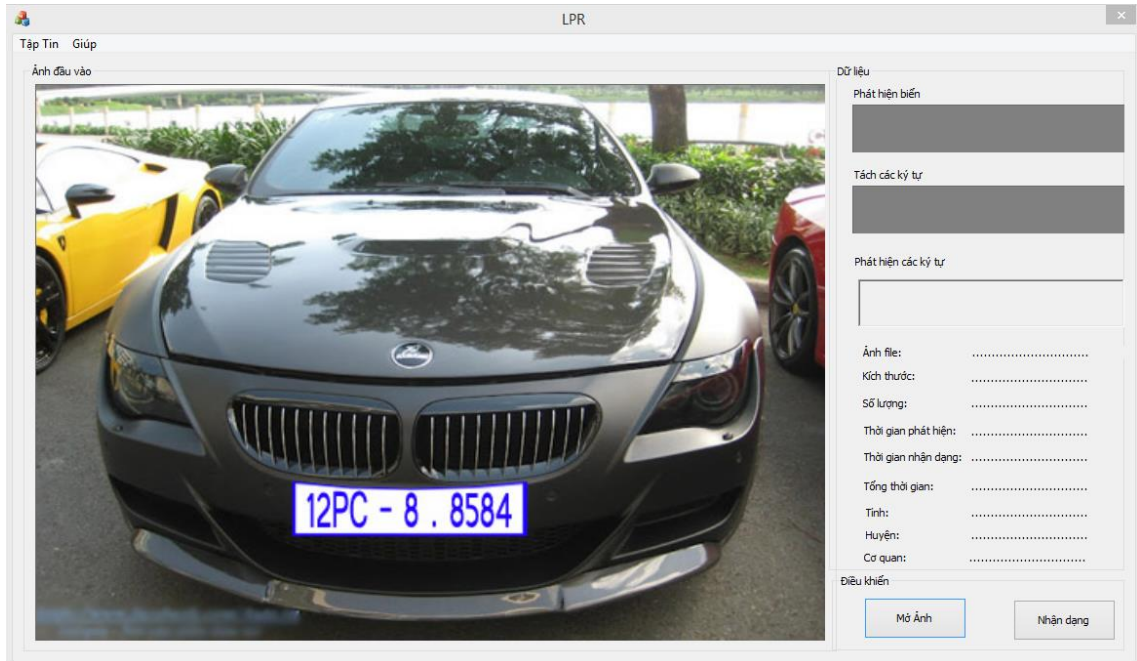


Hình 3.3 Chọn ảnh trên Menu(Tập Tin)



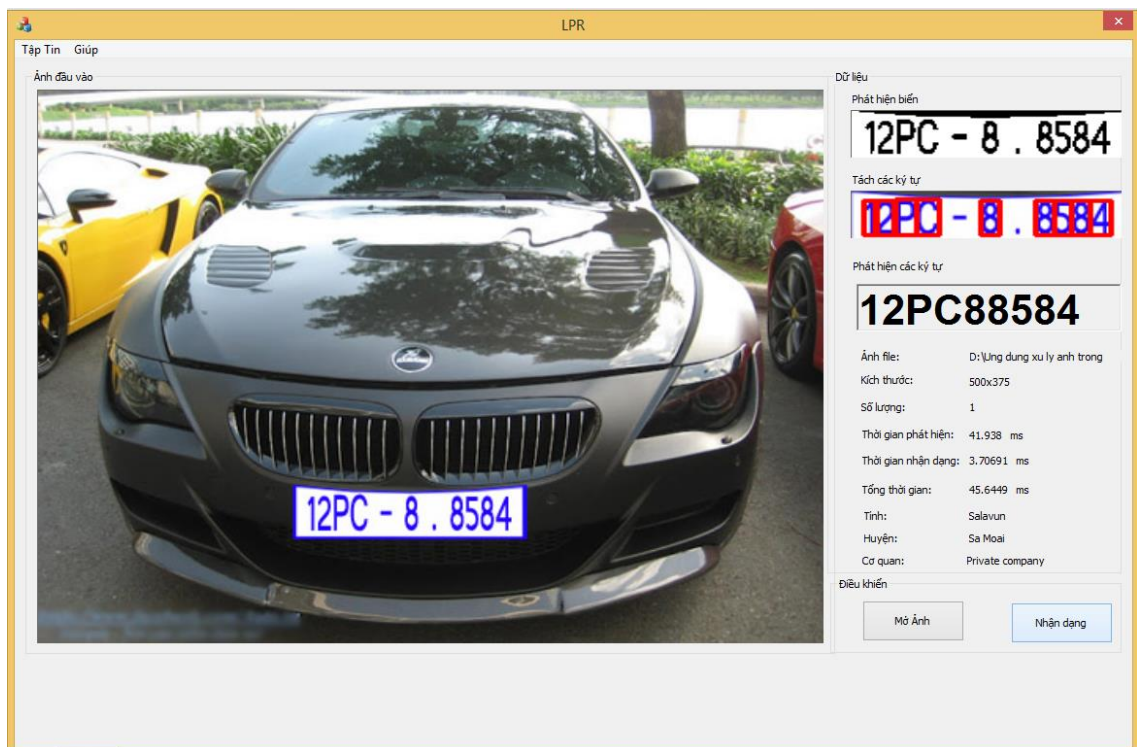
Hình 3.4 Chọn ảnh trong button (Mở Ảnh)

ta sẽ được ảnh đã chọn như hình 3.5



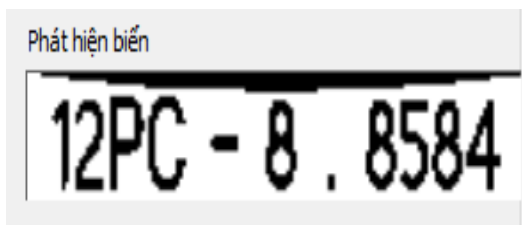
Hình 3.5 Ảnh đã được chọn

3) Sau chọn được ảnh nào đó rồi mình sẽ bấm trên button (Nhận dạng) để nhận dạng biển số xe là được kết quả trong hình 3.6

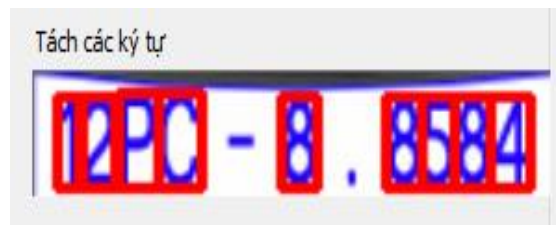


Hình 3.6 Nhận dạng biển số xe trong button (Nhận dạng)

4) Khi nhận dạng xong rồi sẽ có kết quả phát hiện biển số xe, tách các ký tự, nhận dạng các ký tự và các dữ liệu trong nhận dạng như: thời gian phát hiện biển số xe, thời gian nhận dạng các ký tự, tổng kết cả hai thời gian cộng với nhau, nhận dạng tỉnh, huyện và cơ quan.



Hình 3.7 Phát hiện biển số xe



Hình 3.8 Tách các ký tự



Hình 3.9 Nhận dạng các ký tự

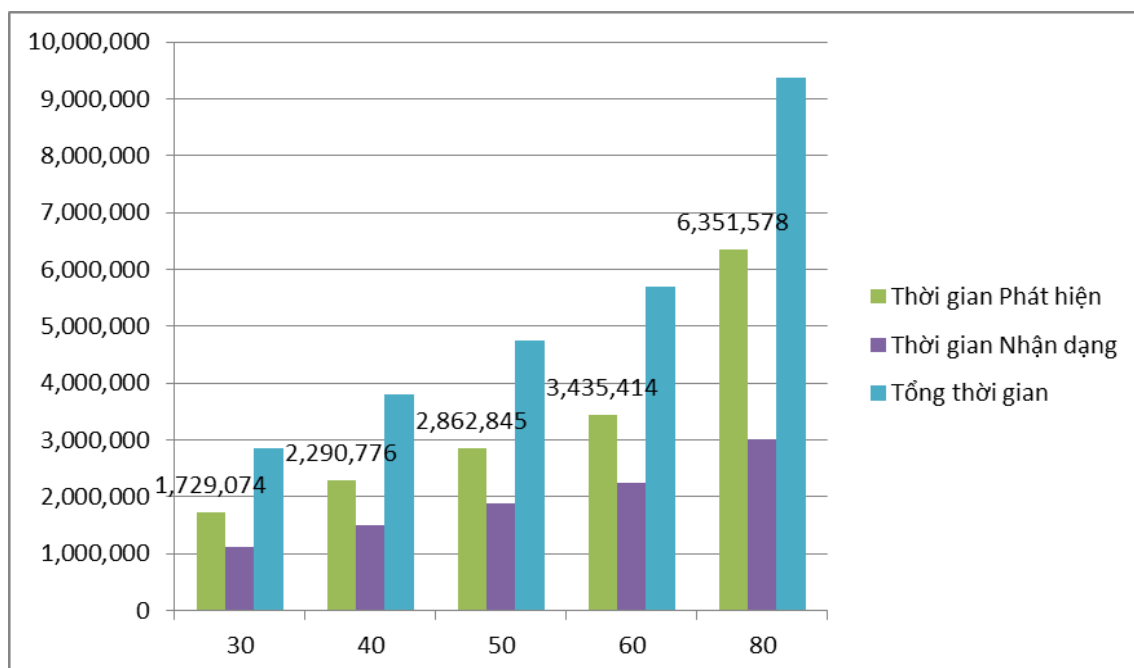
Ảnh file:	D:\Ứng dụng xử lý ảnh trong
Kích thước:	500x375
Số lượng:	1
Thời gian phát hiện:	45.3575 ms
Thời gian nhận dạng:	3.67869 ms
Tổng thời gian:	49.0362 ms
Tỉnh:	Salavun
Huyện:	Sa Moai
Cơ quan:	Private company

Hình 3.10 Các Dữ liệu sau nhận dạng

❖ Bảng kết quả độ chính xác trung bình của 80 biển số xe

Số	Tập ảnh dữ liệu	Độ chính xác	Thời gian Phát hiện	Thời gian Nhận dạng	Tổng thời gian
1	30	100%	1.729.074 ms	112.3764 ms	2.852.838 ms
2	40	100%	2.290.776 ms	150.5956 ms	3.796.732 ms
3	50	100%	2.862.845 ms	187.7445 ms	4.740.290 ms
4	60	100%	3.435.414 ms	225.2934 ms	5.688.348 ms
5	80	100%	6.351.578 ms	301.3912 ms	9.365.490 ms

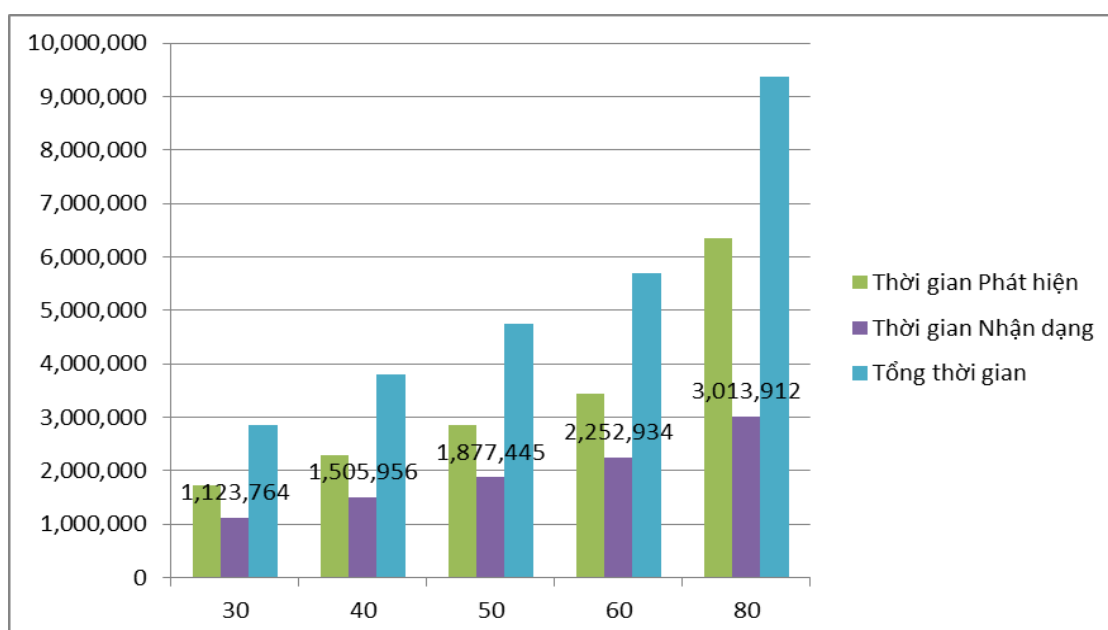
Thời gian phát hiện



Hình 3.11 biểu đồ của thời gian phát hiện

Biểu đồ của thời gian phát hiện là có các tập ảnh dữ liệu 30, 40, 50, 60, 80 ảnh và có thời gian phát hiện bằng ms. Nếu tập ảnh dữ liệu nhiều thì thời gian phát hiện càng lớn.

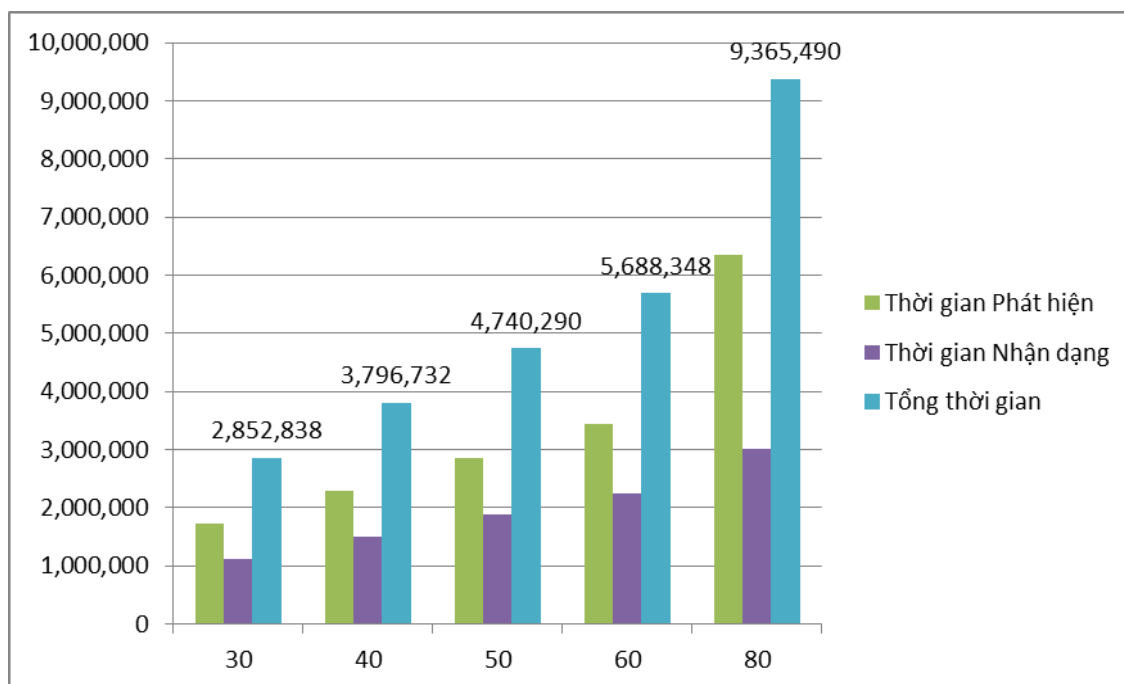
Thời gian nhận dạng



Hình 3.12 biểu đồ của thời gian nhận dạng

Trong biểu đồ này có các tập ảnh dữ liệu 30, 40, 50, 60, 80 ảnh và có thời gian nhận dạng là ms. Nếu tập ảnh dữ liệu nhiều thì thời nhận dạng càng tăng, thời gian nhận dạng sẽ ít hơn thời gian phát hiện.

Tổng thời gian



Hình 3.13 biểu đồ của tổng thời gian

Biểu đồ tổng thời gian là được dùng thời gian nhiều nhất vì ta lấy cả thời gian phát hiện và thời gian nhận dạng cộng với nhau.

KẾT LUẬN

Báo cáo đã nêu được một số nội dung:

- Nghiên cứu tổng quan về hệ thống xử lý ảnh số.
- Nghiên cứu một số kỹ thuật xử lý ảnh trong ứng dụng tự động nhận dạng biển số xe trong ảnh chụp từ camera.
- Tìm hiểu về thư viện OpenCV – thư viện hỗ trợ cho việc viết chương trình.
- Demo một số kỹ thuật trong xử lý ảnh phục vụ cho ứng dụng nhận dạng biển số xe ô tô của Lào.
- Tìm hiểu về một số vấn đề trong nhận dạng ảnh biển số xe ô tô Cộng hòa dân chủ nhân dân Lào.

Hướng nghiên cứu phát triển của đề tài:

- Xây dựng chương trình hoá các chức năng nhận dạng biển số xe.
 - Nghiên cứu chương trình nhận dạng biển số xe này để dùng với xe máy.
- Vì khoảng thời gian có hạn nên một số chức năng còn hạn chế và chưa hoàn thiện. Trong thời gian sắp tới em sẽ cố gắng nghiên cứu để tìm hiểu sâu hơn, em rất mong nhận được nhiều ý kiến đóng góp của thầy cô và các bạn!

TÀI LIỆU THAM KHẢO

- [1] Đỗ Năng Toàn, Phạm Việt Bình (2007), “*Giáo trình xử lý ảnh*”, Nhà xuất bản Khoa học Kỹ thuật, Hà Nội.
- [2] Lương Mạnh Bá, Nguyễn Thanh Thủy (2007), “*Nhập môn xử lý ảnh số*”, Nhà xuất bản Khoa học Kỹ thuật, Hà Nội
- [3] Nguyễn Thị Ngọc Mai (2011), “*Nghiên cứu một số kỹ thuật trích chọn đặc trưng biên ảnh và ứng dụng trong nhận dạng biển số xe*”, Đồ án Tốt nghiệp ĐH ngành Công nghệ thông tin và Truyền thông, Thái Nguyên.
- [4] Nguyễn Minh Tuấn (2009), “*Tự động nhận dạng ký tự trên biển số xe*”, học viện công nghệ bưu chính viễn thông khoa công nghệ thông tin, Hà Nội.
- [5] Nguyễn Phạm Anh Tuấn (2010), “*Nhận dạng biển số xe*”, Trường đại học Lạc Hồng Khoa Điện - Điện Tử, Biên Hòa
- [6] Nguyễn Thị Hoàn (2010), “*Phương pháp trích chọn đặc trưng ảnh trong thuật toán học máy tìm kiếm ảnh áp dụng vào bài toán tìm kiếm sản phẩm*”, Trường đại học Công nghệ- Đại học quốc gia Hà Nội.
- [7] Doãn Đạt Phước, Nguyễn Đồng Hải Phương (2008), “*Tự động nhận dạng biển số đăng ký xe trong ảnh chụp từ camera*”, Trường Đại học Bách Khoa, Đà Nẵng.
- [8] Phạm Thị Thanh Thủy (2009), “*tìm hiểu bài toán nhận dạng biển số xe*”, Trường đại học Dân Lập Hải Phòng.
- [9] <http://luanvan.co/luan-van/tong-quan-ve-xu-ly-anh-va-thu-vien-mã-nguồn-mở-Opencv-35612/>, Tổng quan về xử lý ảnh và thư viện mã nguồn mở Opencv.
- [10] Gary Bradski, Adrian Kaehler (2003), “*Learning OpenCV*”, O’Reilly Media, Inc, Cambridge, USA.