

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO: NHẬP MÔN KHOA HỌC DỮ LIỆU**

**ĐỀ TÀI: NHẬN DIỆN BIỂN SỐ XE**

Giảng viên: Vũ Hoài Nam

Sinh viên: Đặng Ngọc Anh B21DCCN141

Ngô Hải Đăng B21DCCN201

Nguyễn Văn Hùng B21DCCN417

Nhóm: 12 - 02

# LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn sâu sắc và chân thành nhất đến Thầy Vũ Hoài Nam, giảng viên môn *Nhập môn Khoa học Dữ liệu*, người đã luôn đồng hành và tận tình hướng dẫn chúng em trong suốt quá trình học tập và thực hiện đồ án này.

Trong suốt thời gian qua, thầy đã không ngừng truyền đạt những kiến thức nền tảng quan trọng của lĩnh vực khoa học dữ liệu, giúp chúng em xây dựng được nền móng vững chắc để tiếp cận một lĩnh vực đầy tiềm năng và thử thách. Không chỉ dừng lại ở việc cung cấp lý thuyết, thầy còn chia sẻ những kinh nghiệm thực tiễn quý giá, giúp chúng em hiểu rõ hơn cách áp dụng khoa học dữ liệu vào giải quyết các vấn đề trong thực tế.

Đặc biệt, trong quá trình thực hiện đồ án, thầy đã luôn theo sát, hỗ trợ chúng em với những góp ý chân thành, cụ thể và mang tính định hướng. Những lời khuyên của thầy không chỉ giúp chúng em vượt qua các khó khăn, bế ngõ ban đầu, mà còn mở ra cho chúng em những cách nhìn mới mẻ và sáng tạo hơn trong việc triển khai dự án. Chính nhờ sự động viên, khích lệ của thầy mà chúng em đã có thêm động lực để hoàn thành đồ án này với tinh thần trách nhiệm và tâm huyết cao nhất.

Chúng em xin kính chúc thầy luôn dồi dào sức khỏe, hạnh phúc và thành công trong sự nghiệp giảng dạy cũng như trong các lĩnh vực mà thầy đam mê. Hy vọng rằng thầy sẽ tiếp tục truyền cảm hứng và dẫn dắt nhiều thế hệ sinh viên khác đạt được những thành công to lớn trong tương lai.

Trân trọng,  
Nhóm 12, lớp 02

## **Lý do chọn đề án**

Hiện nay, số lượng xe cộ tham gia giao thông trên đường là rất lớn dẫn đến tiêu tốn rất nhiều nhân lực và vật lực cho việc quản lý phương tiện cá nhân trong bãi gửi xe. Nếu không có một công cụ thuận tiện thì việc quản lý phương tiện cá nhân rất mất thời gian, dễ gây nhầm lẫn, thiệt hại cho người sử dụng dịch vụ tại các bãi đỗ xe.

Để giảm tải cho các công việc như thu tiền, bảo hiểm xe, tìm xe cộ trong bãi đỗ xe, trên thế giới đã phát triển công nghệ giám sát tự động đối với các phương tiện giao thông, chính nhờ tính cá nhân của biển số xe mà nó đã trở thành đối tượng chính được sử dụng để nghiên cứu, phát triển trong công nghệ này.

Do đó em muốn chọn đề tài này như bước căn bản trong việc tìm hiểu các công cụ giám sát mạnh hơn như kiểm soát xe lưu thông trên đường hay nhận dạng khuôn mặt ... đang được thế giới rất chú trọng lúc này.

# Mục lục

GIỚI THIỆU .....	6
Tổng quan .....	6
Nhiệm vụ đề tài.....	6
TỔNG QUAN BÀI TOÁN NHẬN DIỆN BIỂN SỐ XE .....	6
Khái niệm biển số xe .....	6
Xử lý ảnh và Open CV .....	7
Hướng giải quyết.....	7
PHÁT HIỆN VỊ TRÍ VÀ TÁCH BIỂN SỐ XE .....	8
Hướng giải quyết.....	8
Chuyển ảnh xám.....	10
Tăng độ tương phản .....	10
Phép toán hình thái học.....	10
Tăng độ tương phản .....	14
Giảm nhiễu bằng bộ lọc Gauss.....	15
Nhiễu .....	15
Bộ lọc Gauss (Gauss filter).....	15
Nhị phân hóa với ngưỡng động (Adaptive Threshold) .....	16
Ảnh nhị phân.....	16
Nhị phân hóa.....	17
Nhị phân hóa với ngưỡng động .....	17
Phát hiện cạnh Canny (Canny Edge Detection).....	18
Giảm nhiễu .....	18
Tính toán Gradient .....	18
Loại bỏ những điểm không phải là cực đại .....	19
Lọc ngưỡng.....	20
Kết quả .....	20

Lọc biên số với contour .....	21
Một số phương pháp tìm contour .....	21
Lọc biên số.....	25
PHÂN ĐOẠN KÍ TỰ .....	25
Hướng giải quyết.....	25
Xoay biên số .....	26
Tìm vùng đối tượng.....	27
Tìm và tách kí tự.....	28
NHẬN DIỆN KÍ TỰ .....	29
Tổng quan về lý thuyết.....	29
AI (Artificial Intelligent) .....	29
Machine Learning.....	30
Hướng giải quyết.....	34
KẾT QUẢ THỰC HIỆN.....	36
Cách thức đo đạc, thử nghiệm.....	36
Kết quả và giải thích .....	37
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	42
Kết luận .....	42
Hướng phát triển .....	42

# GIỚI THIỆU

## Tổng quan

Nội dung :

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

## Nhiệm vụ đề tài

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

- Tìm hiểu khái quát về xử lý ảnh và bài toán nhận dạng biển số xe.
- Tìm hiểu thông tin về biển số xe và phân loại biển số xe của Việt Nam.
- Tìm hiểu các công đoạn chính của bài toán nhận dạng biển số xe gồm 3 khâu chính:
  - Phát hiện vị trí và tách biển số xe
  - Phân đoạn kí tự trong biển số xe
  - Nhận dạng kí tự
- Cài đặt thử nghiệm.

## TỔNG QUAN BÀI TOÁN NHẬN DIỆN BIỂN SỐ XE

### Khái niệm biển số xe

Ở Việt Nam, biển kiểm soát xe cơ giới (hay còn gọi tắt là biển kiểm soát, biển số xe) là tấm biển gắn trên mỗi xe cơ giới, được cơ quan công an cấp (đối với xe quân sự do Bộ Quốc phòng cấp) khi mua xe mới hoặc chuyển nhượng xe. Biển số xe được làm bằng hợp kim nhôm sắt, có dạng hình chữ nhật hoặc hơi vuông, trên đó có in số và chữ (biển xe dân sự không dùng các chữ cái I, J, O, Q, W. Chữ R chỉ dùng cho xe rơ-moóc, sơ-mi rơ-moóc) cho biết: Vùng và địa phương quản lý, các con số cụ thể khi tra trên máy tính còn cho biết danh tính người chủ hay đơn vị đã mua nó, thời gian mua nó phục vụ cho công tác an ninh, đặc biệt trên đó còn có hình Quốc huy Việt Nam dập nổi.

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

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

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

## Xử lý ảnh và Open CV

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

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOS lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Opencv có rất nhiều ứng dụng như:

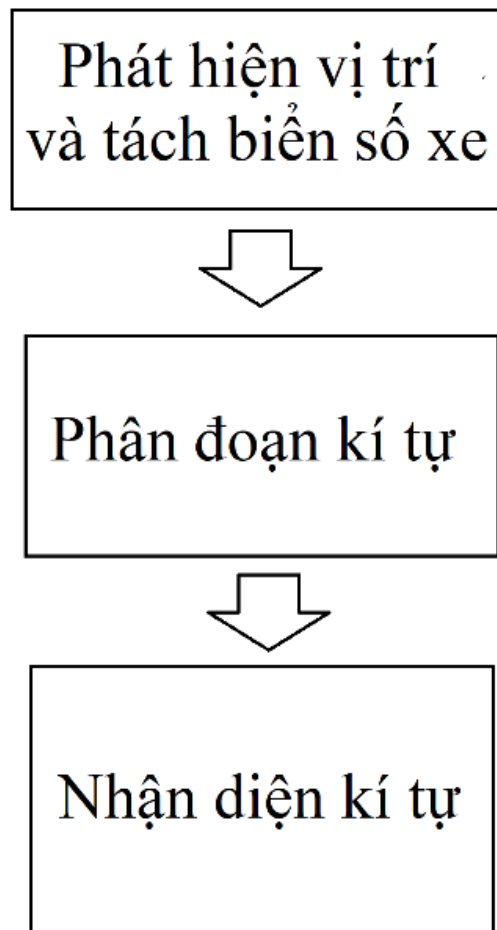
- Nhận dạng ảnh
- Xử lý hình ảnh
- Phục hồi hình ảnh/video
- Thực tế ảo
- Các ứng dụng khác

## Hướng giải quyết

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

1. Phát hiện vị trí và tách biển số xe từ một hình ảnh có sẵn từ đầu vào là camera

2. Phân đoạn các kí tự có trong biển số xe
3. Nhận diện các kí tự đó rồi đưa về mã ASCII



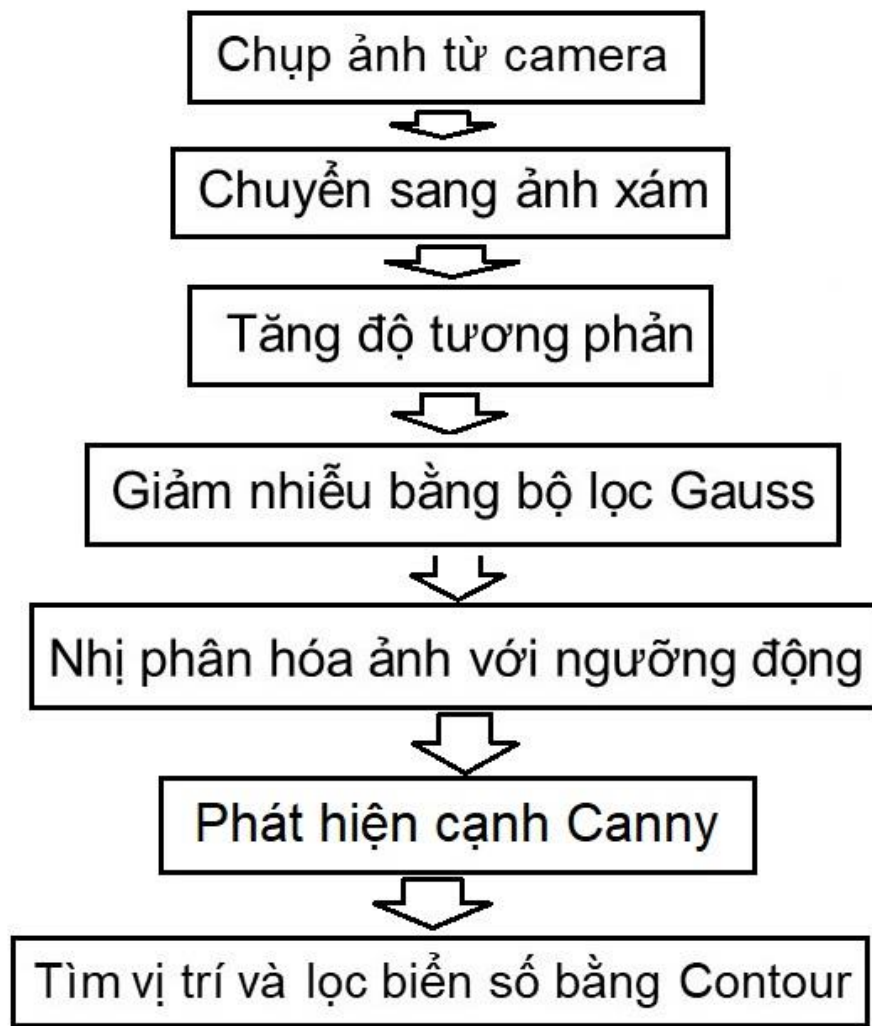
*Hình 2.3 - 1 Các bước chính trong nhận dạng biển số xe*

## PHÁT HIỆN VỊ TRÍ VÀ TÁCH BIỂN SỐ XE

### Hướng giải quyết

Sơ đồ dưới đây sẽ tóm gọn các bước để xác định và tách biển số xe từ clip:





*Hình 3.1 - 1 Xác định và tách biển số xe*

Đầu tiên từ clip ta sẽ cắt từng frame ảnh ra từ clip đầu vào để xử lý, tách biển số. Ở phạm vi đồ án này, ý tưởng chủ yếu là nhận diện được biển số từ sự thay đổi đột ngột về cường độ ánh sáng giữa biển số và môi trường xung quanh nên ta sẽ loại bỏ các dữ liệu màu sắc RGB bằng cách chuyển sang ảnh xám. Tiếp theo ta tăng độ tương phản với hai phép toán hình thái học Top Hat và Black Hat để làm nổi bật thêm biển số giữa phông nền, hỗ trợ cho việc xử lý nhị phân sau này. Sau đó, ta giảm nhiễu bằng bộ lọc Gauss để loại bỏ những chi tiết nhiễu có thể gây ảnh hưởng đến quá trình nhận diện, đồng thời làm tăng tốc độ xử lý.

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

giúp xác định được đúng biên số. Cuối cùng, ta sẽ xác định vị trí của biên số trong ảnh bằng cách vẽ Contour bao quanh.

## Chuyển ảnh xám

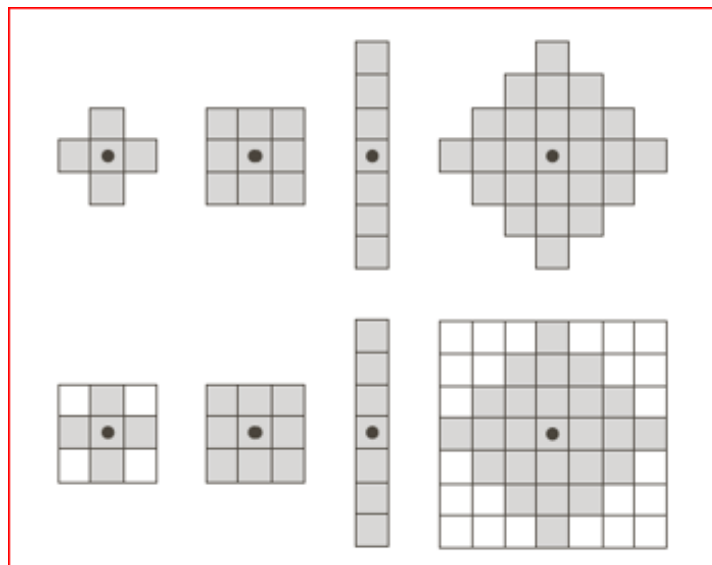
Ảnh xám (Gray Scale) đơn giản là một hình ảnh trong đó các màu là các sắc thái của màu xám với 256 cấp độ xám biến thiên từ màu đen đến màu trắng, nằm trong dải giá trị từ 0 đến 255, nghĩa là cần 8 bits hay 1 byte để biểu diễn mỗi điểm ảnh này. Lý do cần phải phân biệt giữa ảnh xám và các ảnh khác nằm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

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

## Tăng độ tương phản

### Phép toán hình thái học

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

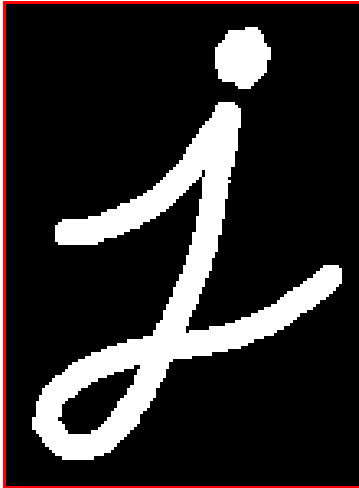


Hình 3.3 - 1 Ví dụ về phần tử cấu trúc

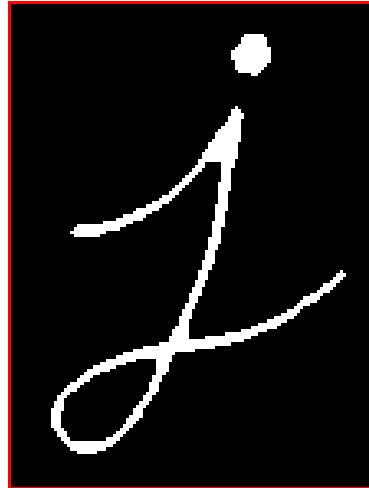
Hình thái học toán học đã được phát triển cho hình ảnh nhị phân, và sau đó được mở rộng cho ảnh xám,... Đây là một trong những kỹ thuật được áp dụng trong giai đoạn tiền xử lý. Hai phép toán thường dùng là phép giãn nở (Dilation) và phép co (Erosion). Từ hai phép toán cơ bản này người ta phát triển thành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

#### Phép co

Phép toán co có ứng dụng trong việc giảm kích thước của đối tượng, tách rời các đối tượng gần nhau, làm mảnh và tìm xương của đối tượng.



Ảnh gốc

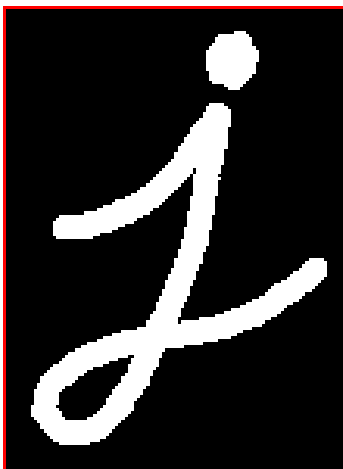


Ảnh sau khi dùng phép co

Hình 3.3 - 2 Phép co

#### Phép giãn nở

Phép toán này có tác dụng làm cho đối tượng ban đầu trong ảnh tăng lên về kích thước (giãn nở ra). Ứng dụng của phép giãn nở là làm cho đối tượng trong ảnh được tăng lên về kích thước, các lỗ nhỏ trong ảnh được lấp đầy, nối liền đường biên ảnh đối với những đoạn rời nhỏ.



Ảnh gốc

Ảnh sau khi dùng phép giãn nở

Hình 3.3 - 3 Phép giãn nở

#### Phép mở

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



Ảnh gốc

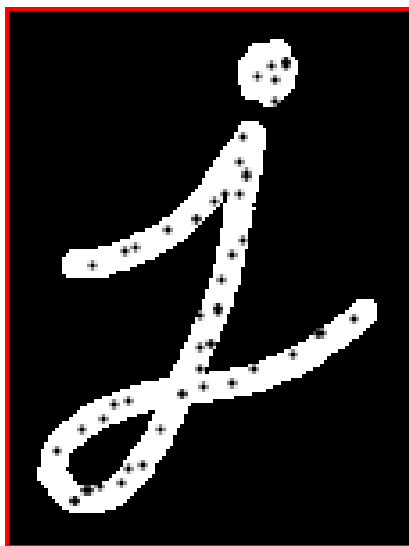


Ảnh sau khi dùng phép mở

Hình 3.3 - 4 Phép mở

#### Phép đóng

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



Ảnh gốc



Ảnh sau khi dùng phép đóng

Hình 3.3 - 5 Phép đóng

#### Phép Top Hat

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



Ảnh gốc



Ảnh sau khi dùng phép Top Hat

Hình 3.3 - 6 Phép Top Hat

#### Phép Black Hat

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



Ảnh gốc



Ảnh sau khi dùng phép Black Hat

*Hình 3.3 - 7 Phép Black Hat*

### Tăng độ tương phản

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



Ảnh gốc



Ảnh sau khi tăng độ tương phản

*Hình 3.3 - 8 Ảnh sau khi tăng độ tương phản*



## Giảm nhiễu bằng bộ lọc Gauss

### Nhiễu

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

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

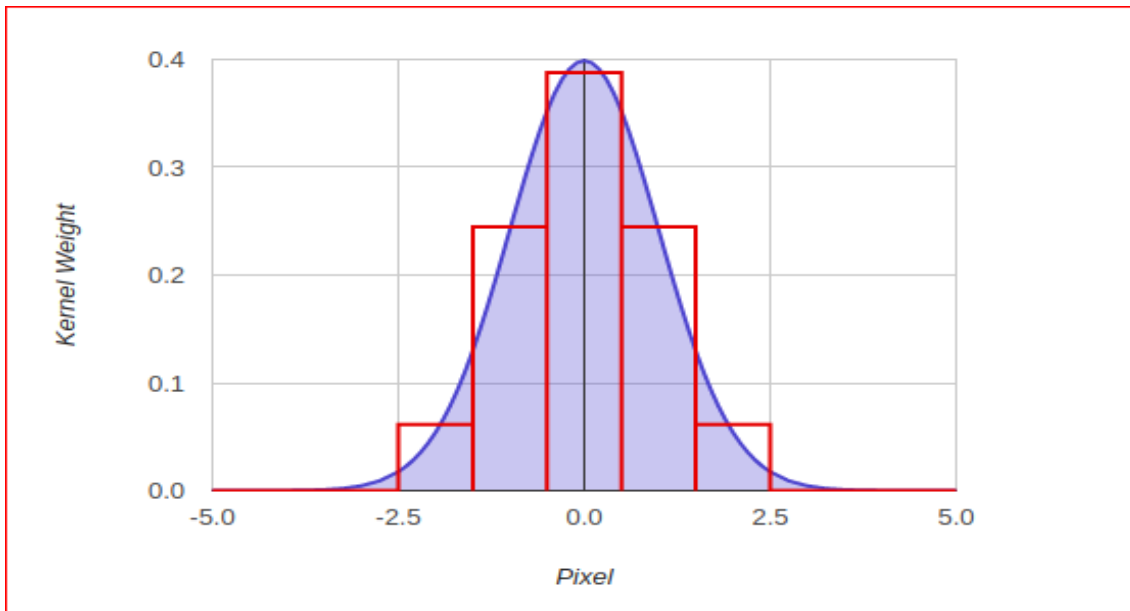


*Hình 3.4 - 1 Nhiễu*

### Bộ lọc Gauss (Gauss filter)

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

Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa. Trọng số của sự phụ thuộc được lấy theo hàm Gauss (cũng được sử dụng trong quy luật phân phối chuẩn).



Hình 3.4 - 2 Ma trận lọc Gauss

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



Ảnh gốc



Ảnh sau khi làm mờ, giảm nhiễu

Hình 3.4 - 3 Kết quả sử dụng bộ lọc Gauss

## Nhị phân hóa với ngưỡng động (Adaptive Threshold)

### Ảnh nhị phân

Là ảnh mà giá trị của các điểm ảnh chỉ được biểu diễn bằng hai giá trị là 0 (Đen) và 255 (Trắng)



## Nhị phân hóa

Là quá trình biến đổi một ảnh xám thành ảnh nhị phân.

- Gọi giá trị cường độ sáng tại một điểm ảnh là  $I(x,y)$  .
- $INP(x,y)$  là cường độ sáng của điểm ảnh trên ảnh nhị phân .
- (Với  $0 < x < \text{image.width}$  và  $(0 < y < \text{image.height})$ ).

Để biến đổi ảnh xám thành ảnh nhị phân. Ta so sánh giá trị cường độ sáng của điểm ảnh với một ngưỡng nhị phân  $T$ .

- Nếu  $I(x,y) > T$  thì  $INP(x, y) = 0$ .
- Nếu  $I(x,y) > T$  thì  $INP(x, y) = 255$ .

## Nhị phân hóa với ngưỡng động

Việc nhị phân hóa ảnh với ngưỡng toàn cục như thông thường sẽ rất khó khăn khi phải tự tính toán và chọn mức ngưỡng phù hợp cho từng ảnh khác nhau. Nhị phân hóa ảnh ngưỡng động sẽ giúp tính toán ngưỡng cho phù hợp với từng ảnh, lợi thế thứ hai chính là nó rất phù hợp khi ảnh có vùng bị quá chói hoặc quá tối dẫn đến có thể mất luôn hình ảnh tại vùng đó nếu sử dụng ngưỡng toàn cục.

Về ý tưởng chính sẽ theo 3 bước sau:

1. Chia tấm ảnh thành nhiều khu vực, cửa sổ (Region) khác nhau.
2. Dùng một thuật toán để tìm một giá trị  $T$  phù hợp với từng cửa sổ.
3. Áp dụng phương pháp nhị phân hóa cho từng khu vực, cửa sổ với ngưỡng  $T$  phù hợp.

Có rất nhiều phương pháp để tìm  $T$ , ở đây em sử dụng một kiểu thuật toán mà thư viện OpenCV hỗ trợ là `ADAPTIVE_THRESH_GAUSSIAN_C` tức lấy trung bình các giá trị xung quanh điểm ngưỡng động đang xét  $T(x,y)$  với phân phối Gauss rồi trừ đi hằng số  $C$ .



*Hình 3.5 - 1 Nhị phân hóa ảnh ngưỡng động*

### Phát hiện cạnh Canny (Canny Edge Detection)

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

1. Giảm nhiễu (Noise reduction)
2. Tính toán Gradient (Gradient calculation)
3. Loại bỏ những điểm không phải là cực đại (Non-maximum suppression)
4. Lọc ngưỡng (Double threshold)

#### Giảm nhiễu

Làm mờ ảnh, giảm nhiễu dùng bộ lọc Gauss kích thước 5x5. Kích thước 5x5 thường hoạt động tốt cho giải thuật Canny

#### Tính toán Gradient

Ta dùng 2 bộ lọc Sobel X và Sobel Y (3x3) để tính đạo hàm  $G_x$  và  $G_y$

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

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

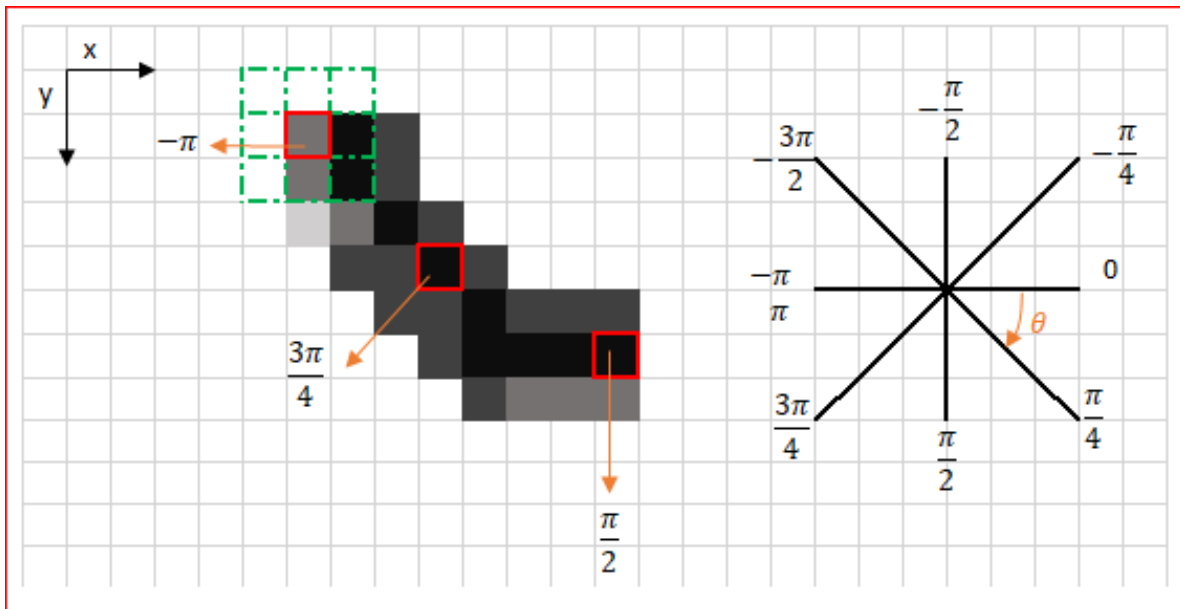
Tìm gradient và hướng được làm tròn về 4 hướng: hướng ngang (0 độ), hướng chéo bên phải (45 độ), hướng dọc (90 độ) và hướng chéo trái (135 độ).

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1} \left( \frac{G_y}{G_x} \right)$$

### Loại bỏ những điểm không phải là cực đại

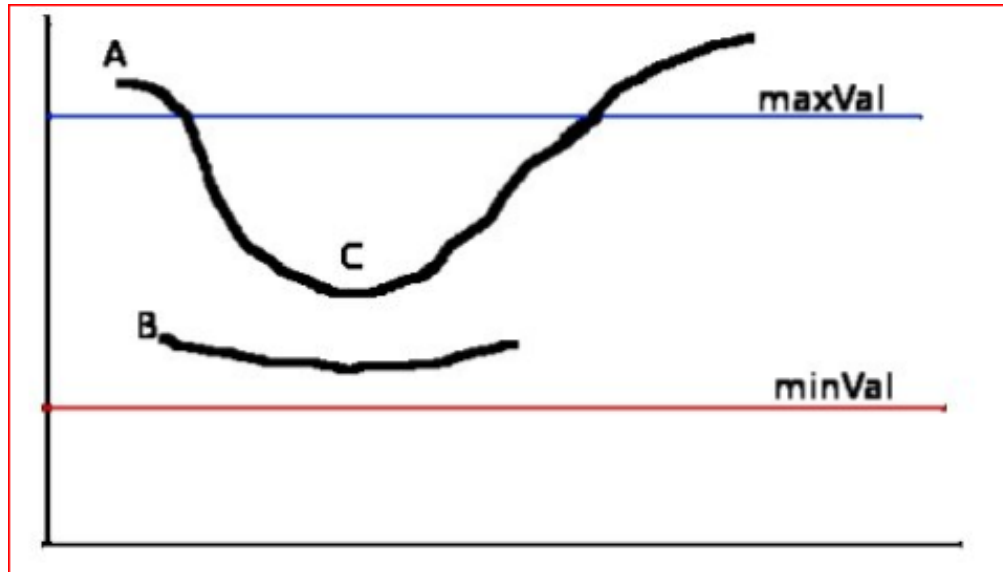
Ở bước này, dùng một filter 3x3 lần lượt chạy qua các pixel trên ảnh gradient. Trong quá trình lọc, xem xét xem độ lớn gradient của pixel trung tâm có phải là cực đại so với các gradient ở các pixel xung quanh. Nếu là cực đại, ta sẽ ghi nhận sẽ giữ pixel đó lại. Còn nếu pixel tại đó không phải là cực đại lân cận, ta sẽ set độ lớn gradient của nó về zero. Ta chỉ so sánh pixel trung tâm với 2 pixel lân cận theo **hướng gradient**. Ví dụ: nếu hướng gradient đang là 0 độ, ta sẽ so pixel trung tâm với pixel liền trái và liền phải nó. Trường hợp khác nếu hướng gradient là 45 độ, ta sẽ so sánh với 2 pixel hàng xóm là góc trên bên phải và góc dưới bên trái của pixel trung tâm



Hình 3.6 - 1 Loại bỏ những điểm không phải cực đại

## Lọc ngưỡng

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



Hình 3.6 - 2 Lọc ngưỡng

## Kết quả

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



*Hình 3.6 - 3 Ảnh sau khi phát hiện biên Canny*

## Lọc biến số với contour

### Một số phương pháp tìm contour

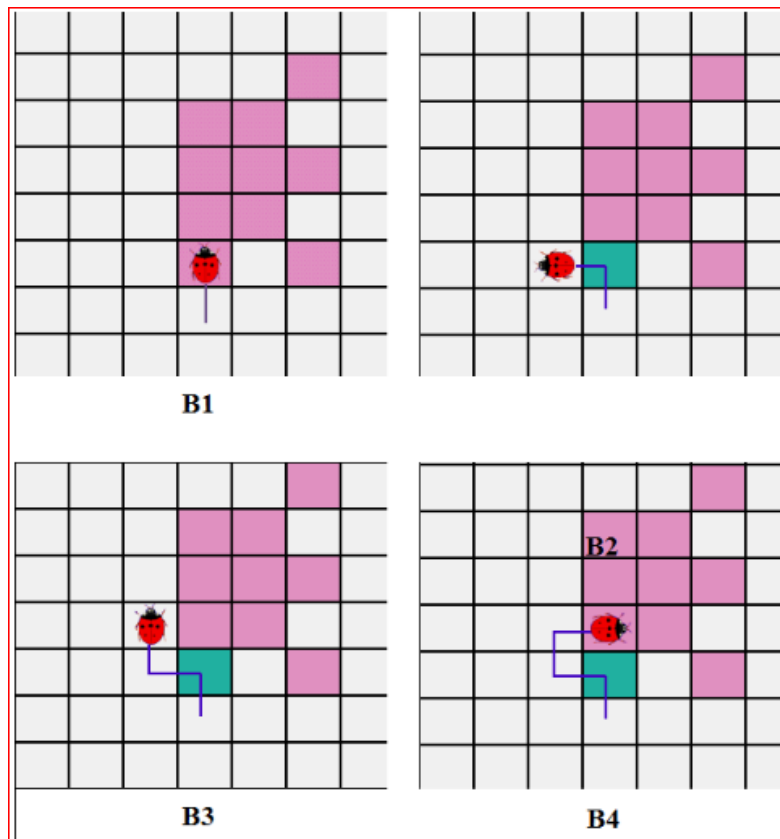
Có thể hiểu Contour là tập hợp các điểm tạo thành đường cong kín bao quanh một đối tượng nào đó. Thường dùng để xác định vị trí, đặc điểm của đối tượng. Có 4 thuật toán Contour Tracing chung nhất. Hai trong số đó có tên là: Square Tracing algorithm và Moore – Neighbor Tracing là dễ để thực hiện và thường xuyên được dùng để dò tìm contour của một mẫu. Với thư viện OpenCV người ta áp dụng thuật toán Suzuki's Contour tracing. Dưới đây em sẽ trình bày kĩ hơn về 3 phương pháp trên:

#### *Thuật toán Square Tracing*

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

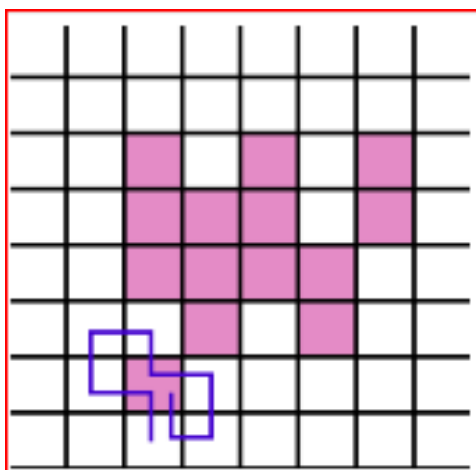
- Nếu gặp Pixel có giá trị bằng 255 thì rẽ trái.
- Nếu gặp Pixel có giá trị bằng 0 thì rẽ phải.
- Di chuyển cho tới khi quay lại pixel start thì dừng lại.

Hình sau mô tả cách hoạt động của thuật toán:

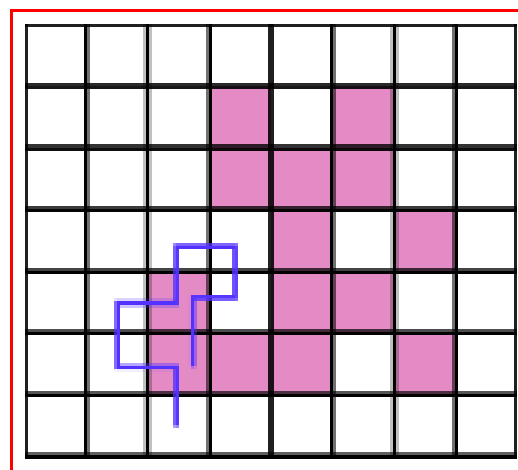


Hình 3.7 - 1 Thuật toán Square Tracing

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



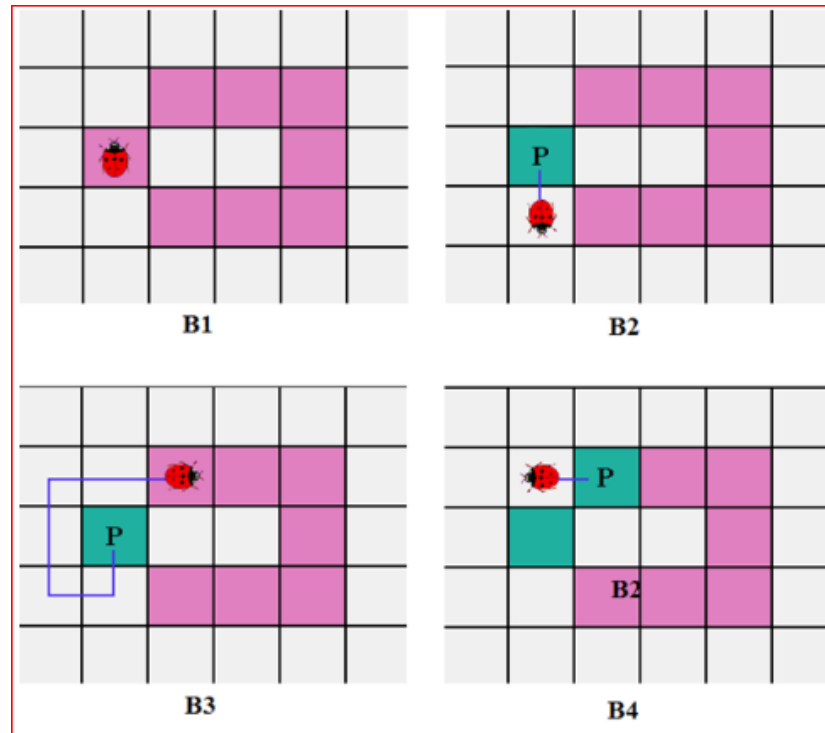
Hình 3.7 - 2 Thuật toán Square Tracing  
chạy đúng



Hình 3.7 - 3 Thuật toán Square Tracing  
chạy sai

### Thuật toán Moore – Neighbor

Thuật toán này có chút khác biệt so với thuật toán Square Tracking, cụ thể: khi gặp pixel có giá trị bằng 255 đầu tiên (pixel start) thì ta sẽ quay lại pixel trước đó, sau đó đi vòng qua các pixel thuộc 8-connected theo chiều kim đồng hồ cho tới khi gặp pixel khác có giá trị bằng 255. Và điều kiện kết thúc cũng giống như thuật toán Square Tracking



Hình 3.7 - 4 Thuật toán Moore - Neighbor.

### Thuật toán Suzuki's Tracing

Đây là thuật toán được thư viện OpenCV sử dụng, ngoài khả năng xác định được biên của vật thể như hai phương pháp trên. Phương pháp Suzuki's Tracing còn có khả năng phân biệt được đường biên ngoài (Outer) và đường biên trong (Hole) của vật thể.

Hàm trong OpenCV được biểu diễn như sau:

**findContours**(InputOutputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset=Point())

Các tham số:

**image** : hình ảnh cần tìm biên, là ảnh nhị phân.

**contours** : lưu trữ các đường biên tìm được, mỗi đường biên được lưu trữ dưới dạng một vector của các điểm.

**hierarchy** : chứa thông tin về hình ảnh như số đường viền, xếp hạng các đường viền theo kích thước, trong ngoài, ..



**mode :**

CV\_RETR\_EXTERNAL : khi sử dụng cờ này nó chỉ lấy ra những đường biên bên ngoài, nhưng biên bên trong của vật thể bị loại bỏ.

CV\_RETR\_LIST : Khi sử dụng cờ này nó lấy ra tất cả các đường viền tìm được.

CV\_RETR\_CCOMP : khi sử dụng cờ này nó lấy tất cả những đường biên và chia nó làm 2 level, những đường biên bên ngoài đối tượng, và những đường biên bên trong đối tượng.

CV\_RETR\_TREE : khi sử dụng cờ này nó lấy tất cả các đường biên và tạo ra một hệ thống phân cấp đầy đủ của những đường lồng nhau.

**method :**

CV\_CHAIN\_APPROX\_NONE : sử dụng cờ này sẽ lưu trữ tất cả các điểm của đường viền .

CV\_CHAIN\_APPROX\_SIMPLE : Ví dụ : một hình chữ nhật sẽ được mã hoá bằng toạ độ của 4 đỉnh.

CV\_CHAIN\_APPROX\_TC89\_L1 or CV\_CHAIN\_APPROX\_TC89\_KCOS : Áp dụng thuật toán xấp xỉ Tech-Chin.



*Hình 3.7 - 5 Vẽ Contour với OpenCV*

Trong ảnh, những đường màu hồng là đường contour bao quanh vật thể, nhưng vì có quá nhiều đường bao quanh các vật thể không phải biển số nên chúng ta sẽ áp dụng những đặc trưng riêng về tỉ lệ cao/rộng, diện tích trong khung hình cố định như ở mục 2.1 để lọc ra đúng biển số cần tìm.



## Lọc biên số

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



Hình 3.7 - 6 Contour chưa xấp xỉ đa giác



Hình 3.7 - 7 Contour đã xấp xỉ đa giác

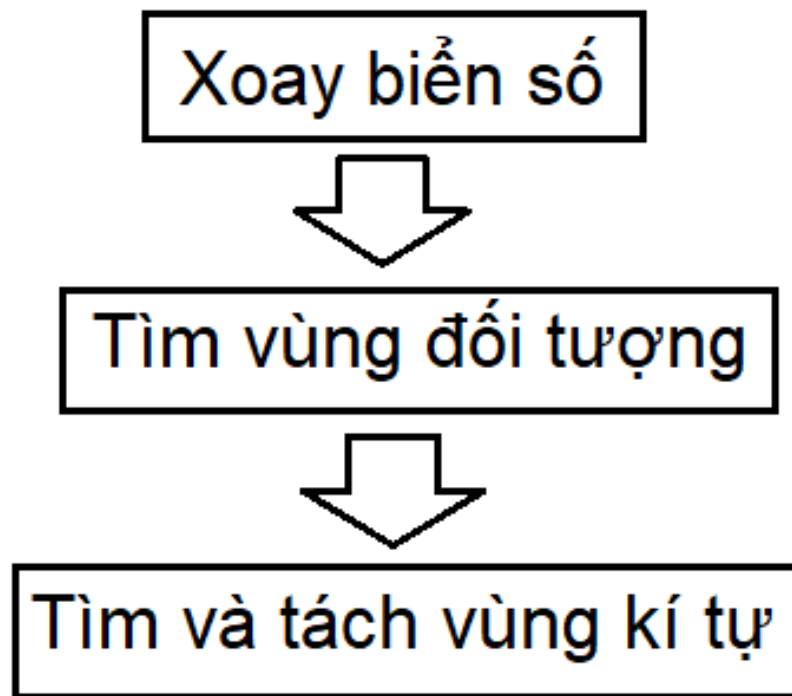
Tiếp theo ta tính toán tỉ lệ cao/rộng và diện tích của biên số phù hợp, sau đó ta lưu tất cả những biên số có trong hình dưới dạng tọa độ các đỉnh

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

## PHÂN ĐOẠN KÍ TỰ

### Hướng giải quyết

Ở giai đoạn này có những bước chính sau: Xoay biên số để tăng khả năng nhận diện, Tìm tất cả các vùng kín cho là kí tự và lọc ra những kí tự đúng. Tách hình ảnh nhưng kí tự đó ra và đưa vào bộ nhận diện



Hình 4.1 - 1 Các bước chính trong phân đoạn kí tự

### Xoay biến số

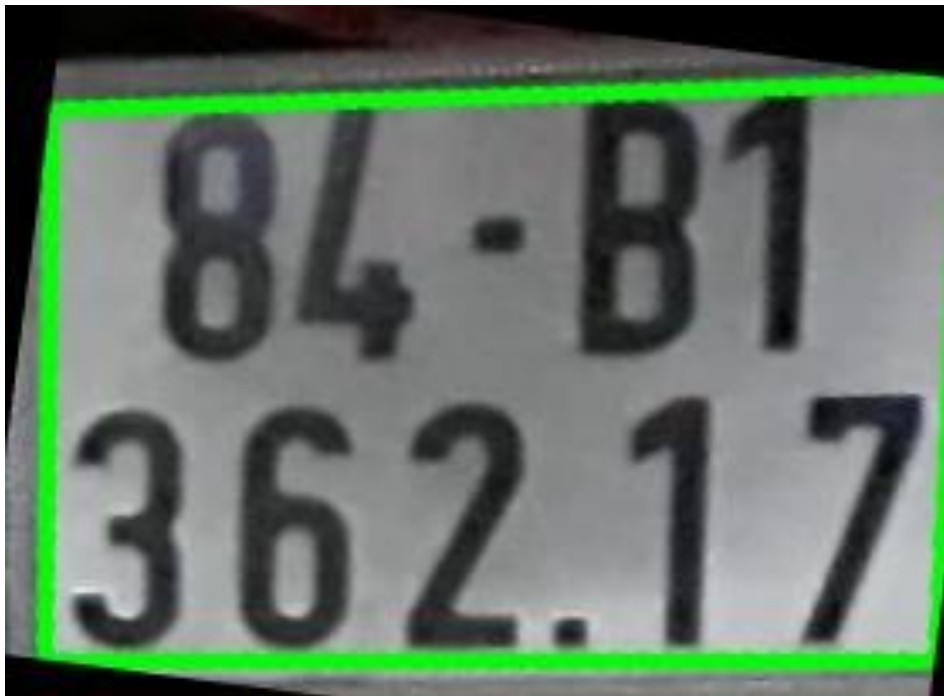
Khi chụp ảnh đầu vào, không phải lúc nào biển số cũng ở chính diện, có thể bị méo sang trái, sang phải, nghiêng góc dẫn đến nếu cứ sử dụng ảnh biển số đã cắt mà không điều chỉnh góc độ dẫn đến ảnh kí tự được cắt ra đưa vào bộ nhận diện rất dễ bị sai. Ví dụ giữa số 1 và số 7, số 2 và chữ Z, chữ B và số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



Hình 4.2 - 1 Ảnh biển số chưa xoay

Phương pháp xoay ảnh em sử dụng ở đây là:

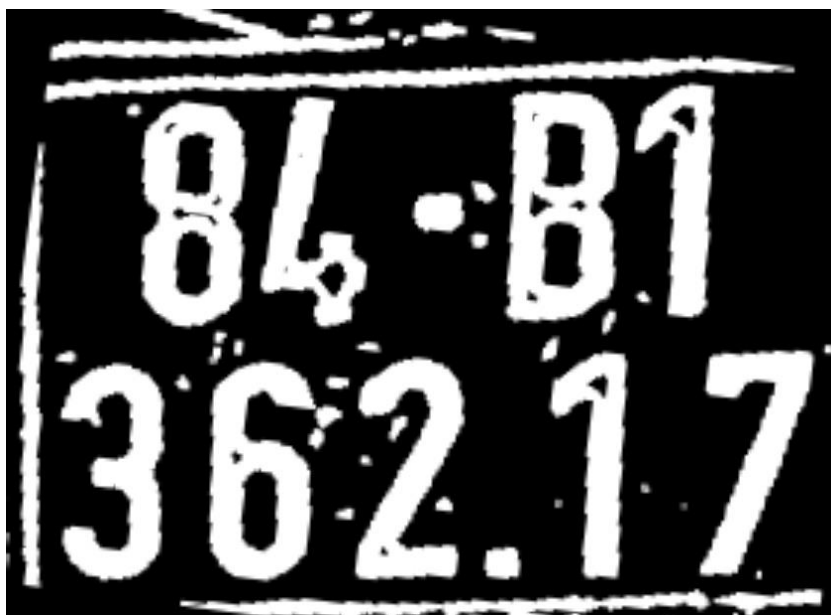
1. Lọc ra tọa độ 2 đỉnh A,B nằm dưới cùng của biển số
2. Từ 2 đỉnh có tọa độ lần lượt là A(x1, y1) và B(x2,y2) ta có thể tính được cạnh đối và cạnh kề của tam giác ABC
3. Ta tính được góc quay  $\widehat{BAC} = \tan^{-1} \left( \frac{\text{cạnh đối}}{\text{cạnh kề}} \right) = \tan^{-1} \left( \frac{BC}{AC} \right)$
4. Xoay ảnh theo góc quay đã tính. Nếu ngược lại điểm A nằm cao hơn điểm B ta cho góc quay âm



Hình 4.2 - 2 Ảnh biển số đã xoay

### Tìm vùng đối tượng

Từ ảnh nhị phân, ta lại tìm contour cho các kí tự (phần màu trắng). Sau đó vẽ những hình chữ nhật bao quanh các kí tự đó. Tuy nhiên việc tìm contour này cũng bị nhiễu dẫn đến việc máy xử lý sai mà tìm ra những hình ảnh không phải kí tự. Ta sẽ áp dụng các đặc điểm về tỉ lệ chiều cao/rộng của kí tự, diện tích của kí tự so với biển số



*Hình 4.3 - 1 Ảnh nhị phân*



*Hình 4.3 - 2 Tìm vùng đối tượng*

Trong ảnh 4.3 - 2 những đường màu vàng là đường contour và nếu so sánh với ảnh nhị phân 4.3 - 1 thì có rất nhiều đường nhiễu như đường viền biển số, dấu gạch, dấu chấm... Sau khi đã áp dụng các điều kiện thì sẽ vẽ ra những hình chữ nhật màu xanh bao quanh các ký tự.

### **Tìm và tách ký tự**

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



*Hình 4.4 - 1 Ảnh kí tự sau khi cắt*

## NHẬN DIỆN KÍ TỰ

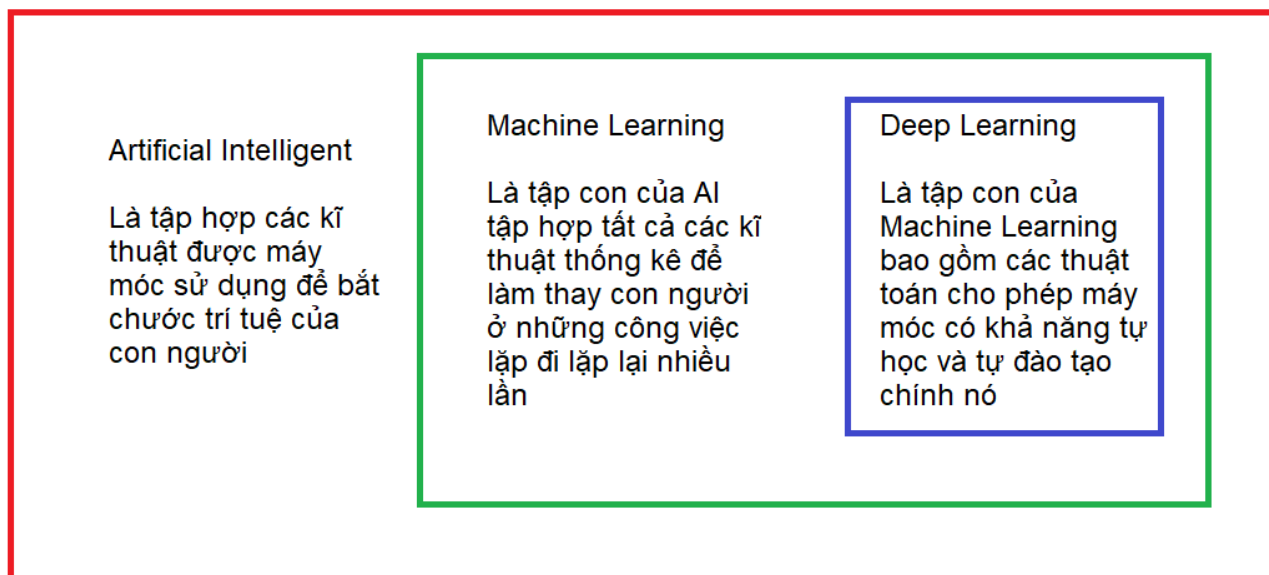
Thực chất quá trình nhận diện kí tự chính là quá trình chuyển đổi từ một hình ảnh là ma trận giá trị các điểm ảnh về một dạng thông tin khác như trong đề tài này là mã ASCII để có thể giao tiếp với người dùng. Để hiểu hơn về nhận diện, chúng ta cần đi ngược lại về ngành khoa học trí tuệ nhân tạo (Artificial Intelligent) hay còn gọi là AI

### Tổng quan về lý thuyết

#### AI (Artificial Intelligent)

Trí tuệ nhân tạo là dạng trí tuệ được biểu hiện ở máy móc, không giống như những dạng trí tuệ tự nhiên ở con người hay động vật. Thông thường cụm từ “trí tuệ nhân tạo” sẽ được hiểu như máy móc có khả năng bắt chước hành vi và nhận thức của con người như việc học tập hay xử lý vấn đề. Trong vài năm trở lại đây, AI thật sự bùng nổ, đặc biệt là từ năm 2015. Phần lớn trong số đó liên quan đến những tiện ích sẵn có của GPU khiến cho việc xử lý song song nhanh, rẻ và mạnh mẽ hơn.

Bên trong AI bao gồm hai khối nhỏ là Máy học (Machine learning) và Học sâu (Deep learning)



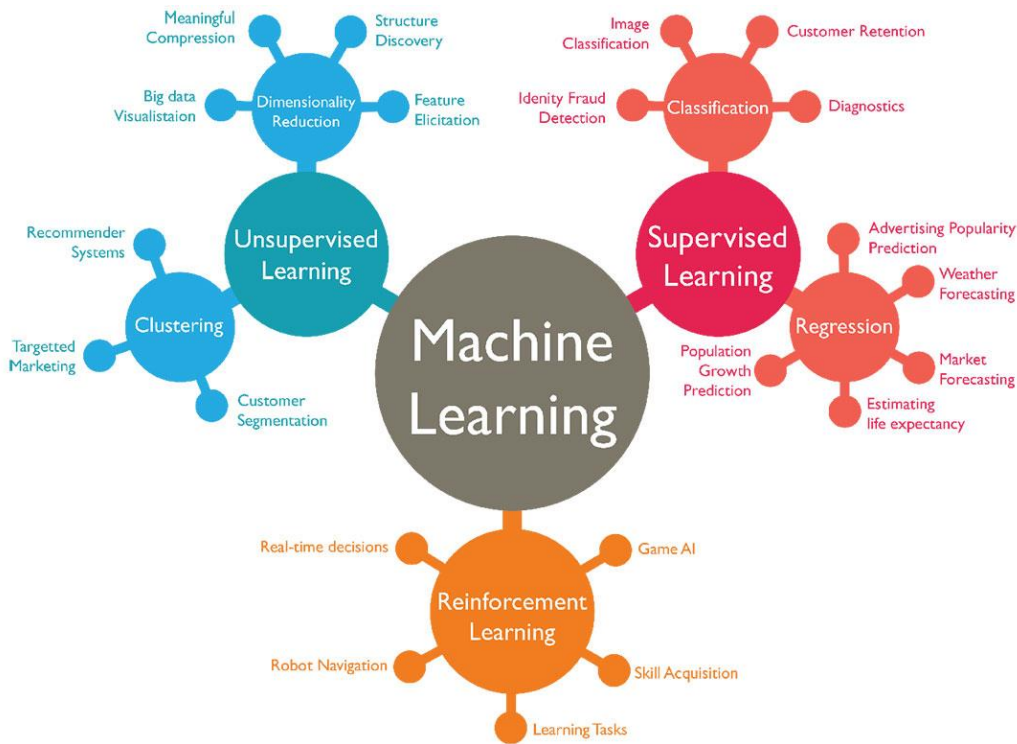
*Hình 5.1 - 1 Các lớp trong AI*

## Machine Learning

Machine learning là một lĩnh vực sử dụng các thuật toán cho phép máy tính có thể học từ dữ liệu để thực hiện các công việc thay vì được lập trình một cách rõ ràng. Một số ứng dụng có thể kể đến như xử lý hình ảnh (gắn thẻ, nhận diện ký tự, ô tô tự lái), phân tích văn bản (lọc mail spam, phân tích ngữ nghĩa và khai thác thông tin),...

Về cách hoạt động của Machine learning nói nôm na ta cần một tập dữ liệu huấn luyện bao gồm nhiều mẫu huấn luyện. Mỗi mẫu huấn luyện sẽ là một thể hiện của bài toán (có đầu vào và lời giải). Machine learning sẽ học từ các thể hiện đó để tìm ra lời giải phù hợp với từng đầu vào mới. Nó giống như khi dạy một đứa trẻ cách tắm cầu, ta sẽ tắm cầu vài lần cho đứa trẻ quan sát. Sau đó đứa trẻ sẽ bắt đầu học để tự tắm cầu. Trong đề tài đồ án này, em sẽ đưa vào tập dữ liệu mẫu là hình ảnh các chữ số, ký tự của biển số Việt Nam đã được gắn nhãn là một mã ASCII tương ứng để máy tự học.

Machine Learning có nhiều thuật toán khác nhau nhưng nổi bật nhất là hai thuật toán **Học có giám sát** (Supervised learning) và **Học không giám sát** (Unsupervised learning)



Hình 5.1 - 2 Phân loại Machine Learning

#### Học có giám sát.

Là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Ví dụ như thuật toán dò các khuôn mặt trong một bức ảnh đã được phát triển từ rất lâu. Thời gian đầu, Facebook sử dụng thuật toán này để chỉ ra các khuôn mặt trong một bức ảnh và yêu cầu người dùng tag friends - tức gán nhãn cho mỗi khuôn mặt. Số lượng cặp dữ liệu (khuôn mặt, tên người) càng lớn, độ chính xác ở những lần tự động tag tiếp theo sẽ càng lớn.

Thuật toán học có giám sát lại tiếp tục được chia ra hai phần nhỏ gồm bài toán phân loại (Classification) khi các nhãn của dữ liệu đầu vào chia thành hữu hạn các nhóm và bài toán hồi quy (Regression) khi nhãn không được chia thành các nhóm mà là một giá trị thực cụ thể. Ví dụ: một cô gái x tuổi, cao y m, thu nhập hàng tháng z triệu đồng thì sẽ kết hôn khi nào?

#### Học không giám sát

Trong thuật toán này, chúng ta không biết được đầu ra hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán học không giám sát sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Bao gồm hai loại chính: Phân nhóm (clustering) toàn bộ dữ liệu đầu vào thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm ví dụ như chia nhóm khách hàng dựa trên hành vi mua hàng, độ tuổi, giới tính của họ. Còn lại là bài toán kết hợp (Association)



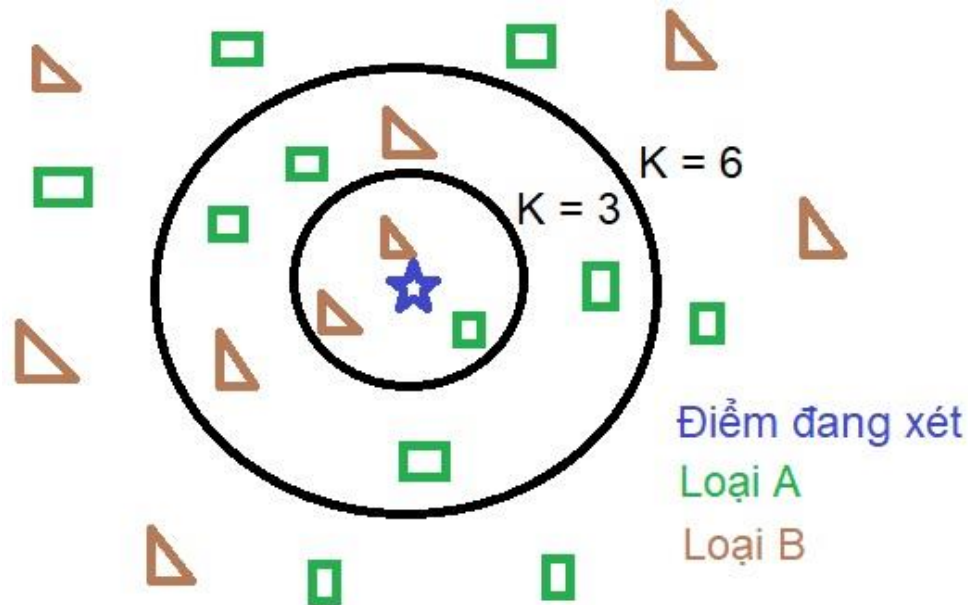
được dùng khi ta muốn khám phá ra một quy luật nào đó dựa trên tập dữ liệu cho trước như những khách hàng mua laptop thường có xu hướng mua thêm con chuột hay chó thường trung thành với chủ hơn mèo.

#### Thuật toán KNN (K - Nearest Neighbor)

KNN là một trong những thuật toán học có giám sát đơn giản nhất trong Machine Learning, có thể sử dụng cho cả bài toán phân loại và hồi quy. Về ý tưởng là gán kết quả với dữ liệu training gần giống với mẫu nhất. Ví dụ như khi đi câu cá, ta không biết cá câu lên là cá rô hay cá chép, nhưng khi so sánh những đặc điểm về mắt, mang, vây,... từ những con cá rô, cá chép đã thấy thì cuối cùng có thể quyết định xem con cá mình câu được thuộc nhóm cá nào.

KNN hoạt động theo quy trình gồm 4 bước chính:

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



Hình 5.1 - 3 Ví dụ về KNN

Việc điểm đang xét thuộc loại nào còn phụ thuộc vào hệ số K hay trọng số khoảng cách... mà người dùng đặt sao cho phù hợp với bài toán đang xét. Chẳng hạn ở hình trên nếu ta xét  $K = 3$  thì điểm đang xét sẽ thuộc loại B, ngược lại nếu  $K = 6$  thì nó thuộc loại A. Ngoài ra người ta



có thể để trọng số cao hơn cho những điểm gần hơn hay ít khi sử dụng  $K = 1$  để đảm bảo kết quả đầu ra được tối ưu.

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

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

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

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Để dễ hiểu hơn em xin lấy một ví dụ về một công ty muốn đưa dòng xe ô tô mới của mình vào thị trường và muốn biết xe sẽ bán chạy hay ế. Họ tổng hợp các dữ liệu về dung tích xi lanh, giá tiền, màu sơn, diện tích xe của các dòng xe đã xuất ra thị trường. Đặt  $Z$  là khả năng tiêu thụ cụ các dòng xe.  $Z = 0$  nghĩa là xe ế,  $Z = 1$  nghĩa là xe sẽ bán chạy

	A	C	E	F	H	I
1	STT	Dung tích xi lanh	Giá tiền	Màu Sơn	Diện tích	Khả năng tiêu thụ Z
2	1	550	7800	Đỏ	25	1
3	2	498	6300	Đen	27	0
4	3	367	5100	Trắng	32	1
5	4	483	8600	Đỏ	28	0
6	5	542	3200	Trắng	23	0
7	6	430	5500	Trắng	31	??
8						

Hình 5.1 - 4 Xét khả năng tiêu thụ của xe

Sau đó ta chuẩn hóa dữ liệu:

	A	B	D	F	G	I
1	STT	Dung tích xi lanh	Giá tiền	Màu Sơn	Diện tích	Khả năng tiêu thụ Z
2	1	1.00	0.85	Đỏ	0.22	1
3	2	0.72	0.57	Đen	0.44	0
4	3	0.00	0.35	Trắng	1.00	1
5	4	0.63	1.00	Đỏ	0.56	0
6	5	0.96	0.00	Trắng	0.00	0
7	6	0.34	0.43	Trắng	0.89	??
8						

Hình 5.1 - 5 Dữ liệu sau chuẩn hóa

Sau đó ta tính khoảng cách giữa điểm đang xét đến các dữ liệu cho trước. Lưu ý ở các biến có giá trị định tính nếu khác nhau khoảng cách sẽ bằng 1, giống nhau sẽ bằng 0. Ví dụ ta tính khoảng cách giữa dòng xe 6 và 1:

$$d(6,1) = \sqrt{(0.34 - 1)^2 + (0.43 - 0.85)^2 + 1^2 + (0.89 - 0.22)^2}$$

	A	B	D	F	G	I	J	K
1	STT	Dung tích xi lanh	Giá tiền	Màu Sơn	Diện tích	Khả năng tiêu thụ Z	Khoảng cách	Xếp hạng
2	1	1.00	0.85	Đỏ	0.22	1	1.43	5
3	2	0.72	0.57	Đen	0.44	0	1.17	3
4	3	0.00	0.35	Trắng	1.00	1	0.37	1
5	4	0.63	1.00	Đỏ	0.56	0	1.23	4
6	5	0.96	0.00	Trắng	0.00	0	1.16	2
7	6	0.34	0.43	Trắng	0.89	??		
8								

Hình 5.1 - 6 Sau khi tính khoảng cách và xếp hạng

Nhìn hình trên nếu chọn K=1 thì kết quả Z=1, nếu K=3 thì Z=0, nếu K=5 thì Z=0.

Tuy nhiên phương pháp này có một số ưu và nhược điểm như sau:

- Ưu điểm:
  - Dễ sử dụng và cài đặt
  - Độ phức tạp tính toán nhỏ
  - Việc dự đoán kết quả rất đơn giản
- Nhược điểm:
  - Với K nhỏ, khi gặp nhiều dễ đưa ra kết quả không chính xác
  - Cần nhiều thời gian lưu training set và nếu test tăng lên sẽ tốn rất nhiều thời gian

## Hướng giải quyết

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

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

#### 4. In ra kết quả biến số

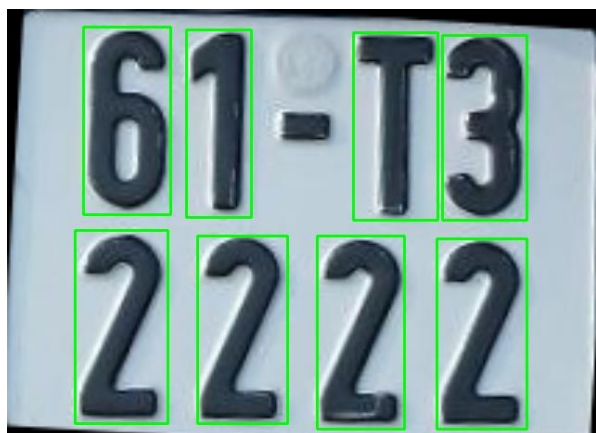
Bước 1 và 2 ta sẽ tạo ra mô hình KNN riêng biệt với code chính. Để khi cần nhận diện kí tự ta không cần phải làm lại các bước từ đầu. Đầu tiên em tạo tập dữ liệu (tập hình ảnh của các chữ số và kí tự) để train từ phần mềm paint. Trong phần mềm Paint ta viết các chữ số và kí tự (trừ kí tự O, I, J) với phong chữ “Biển số xe Việt Nam”, có thể xoay các kí tự này lần lượt với các góc  $-5^\circ$ ,  $5^\circ$ ,  $-10^\circ$ ,  $10^\circ$ . Kết quả có dạng như sau:

A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3  
4 5 6 7 8 9  
A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3  
4 5 6 7 8 9  
A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3  
4 5 6 7 8 9  
A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3  
4 5 6 7 8 9  
A B C D E F G H K L M N P Q R S T U V X Y Z 0 1 2 3  
4 5 6 7 8 9

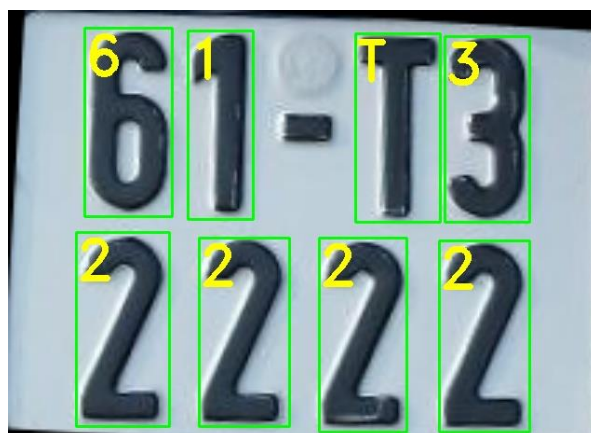
*Hình 5.2 - 1 Tập dữ liệu huấn luyện*

Tiếp theo ta lấy ngưỡng, vẽ contour và cắt từng kí tự. Vì mỗi kí tự có kích thước khác nhau xử lý phức tạp nên cần chuẩn hóa hình ảnh lại với kích thước cao:rộng là 30:20 pixels. Thay vì mỗi kí tự đưa vào mô hình để máy nhận diện thì những kí tự này sẽ được ta gắn nhãn bằng những phím bấm trên máy tính. Sau khi gắn nhãn hết các kí tự ta sẽ lưu hai file .txt là classifications.txt và flattened\_images.txt. File classifications.txt có nhiệm vụ lưu các mã ASCII của các kí tự đó và file flattened\_images.txt sẽ lưu giá trị các điểm ảnh có trong hình ảnh kí tự (hình 20x30 pixel có tổng cộng 600 điểm ảnh có giá trị 0 hoặc 255)

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



Hình 5.2 - 2 Biển số trước khi nhận diện



Hình 5.2 - 3 Biển số sau khi nhận diện



Hình 5.2 - 4 Biển số xe được in ra trên hình gốc

## KẾT QUẢ THỰC HIỆN

### Cách thức đo đạc, thử nghiệm

Clip đầu vào được quay bởi camera từ điện thoại Samsung J7 Prime có các thông số:

- Kích thước clip: FHD 1920x1080
- Camera: 9.6 MP

- 8 ảnh/s

Sau đó clip sẽ được chương trình xử lý cắt từng frame ảnh ra để nhận diện và in ra biển số. Chương trình xử lý được viết trên giao diện ứng dụng Visual Studio 2017 bằng ngôn ngữ Python (3.7), bao gồm các thư viện OpenCV, Numpy, Math.

Em sẽ tạo hai clip test dành riêng cho biển số một hàng và hai hàng được quay với đủ các góc độ nghiêng trái, phải, trên dưới, xa gần, và các môi trường có phong nền, ánh sáng khác nhau. Với mỗi clip test có độ dài X frame ảnh, mỗi frame ảnh có chứa Y biển số, ta sẽ được:

Tổng biển số =  $X \times Y$ . (Y = 3 với biển 2 hàng, Y = 1 với biển một hàng).

Tỉ lệ tìm thấy biển số xe =  $100 \times \frac{\text{Số biển tìm thấy}}{\text{Tổng số mẫu}} (\%)$

Ta sẽ tìm tỉ lệ biển sai n kí tự trên tổng số biển bắt được trong clip. Lưu ý sai ở đây nghĩa là kí tự bị nhận diện sai, không khoanh được vùng hoặc khoanh vùng kí tự sai vị trí.

Tỉ lệ nhận diện =  $100 \times \frac{\text{Số biển sai}}{\text{Số biển tìm thấy}} (\%)$

## Kết quả và giải thích

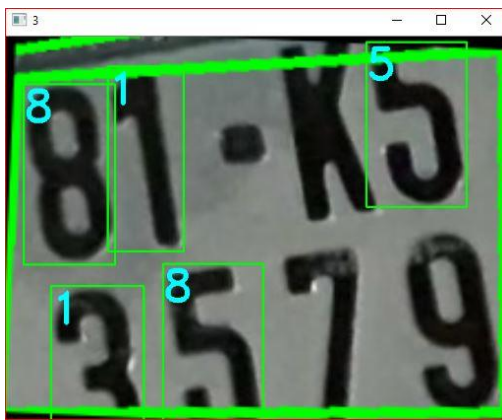
Loại biển	Tổng biển số	Số biển tìm thấy	Tỉ lệ tìm thấy biển số xe (%)
Biển 1 hàng	370	182	49,2%
Biển 2 hàng	2349	924	39,3%

Bảng 6.1 - 1 Tỉ lệ tìm thấy biển số xe trong hình

Ở biển số 1 hàng có tỉ lệ nhận diện cao hơn vì tập mẫu còn ít và mỗi frame ảnh chỉ có 1 biển, dẫn đến biển số thường nằm ở vị trí trung tâm nên dễ nhận diện hơn. Ngược lại, với biển 2 hàng có tập mẫu lớn, và số biển số trong 1 frame nhiều hơn. Chưa kể, biển số ở đây được tính là vùng hình bình hành có tối thiểu 7 kí tự mới được tính là biển số, dẫn đến có nhiều biển số đã cắt đúng nhưng vẫn không được tính vào. Tuy nhiên cách làm này lại giúp loại bỏ đáng kể những chi tiết, biển số sai lệch từ môi trường ngoài.



Hình 6.2 - 1 Không tìm thấy ký tự



Hình 6.2 - 3 Tìm thấy 5 ký tự



Hình 6.2 - 4 Tìm thấy 9 ký tự



Hình 6.2 - 2 Ảnh gốc



Ở hình 6.2 - 2 ta thấy chương trình xử lý tìm ra 3 vùng được cho là biển số, tuy nhiên số kí tự tìm ra ở hình 6.2 - 1 và 6.2 - 3 không đủ nên sẽ bị loại dẫn đến trong ảnh chỉ còn lại duy nhất biển số hình 6.2 - 4.

Khi ta quay theo nhiều góc độ, nhiều vị trí dẫn đến khi tính toán diện tích, tỉ lệ cao/rộng của biển số không còn thỏa điều kiện đặt ra nên đã bị loại. Biển số có thể bị ảnh hưởng bởi những chi tiết ngoài nên khi xấp xỉ contour không ra hình tứ giác, dẫn đến cũng gây mất biển số. Lỗi này đặc biệt xảy ra ở những xe ô tô vì ô tô thường có nền xung quanh biển số là những vật liệu phản chiếu ánh sáng mạnh, gây ảnh hưởng lớn đến quá trình xác định vùng biển số.



Hình 6.2 - 5 Lấy ngưỡng cao/rộng 1.5



Hình 6.2 - 6 Lấy ngưỡng cao/rộng 1.4



Hình 6.2 - 7 Lỗi xấp xỉ Contour

Ở hình 6.2 - 7 mặc dù đường contour màu hồng có bao quanh biển số, tuy nhiên sau khi xấp xỉ chỉ còn lại hình 2, 3 cạnh, để khắc phục được các lỗi tỉ lệ, xấp xỉ contour ta cần điều chỉnh mức ngưỡng cho tối ưu.

Trong quá trình xử lý, việc xử lý nhị phân cũng đóng vai trò quan trọng, ở hình 6.2 - 9 ta thấy ảnh bị nhiễu và bản thân biển số bị tối, dính nhiều bụi dẫn đến khi xử lý nhị phân sẽ bị đứt đoạn và vẽ contour bị sai, để khắc phục cần sử dụng những phép toán hình thái học như phép nở, phép đóng để làm liền những đường màu trắng trong ảnh nhị phân.



Hình 6.2 - 8 Ảnh nhị phân bị đứt



Hình 6.2 - 9 Đường contour bị đứt đoạn

Dưới đây ta xét khả năng khoanh vùng và nhận diện ký tự tương ứng với giai đoạn “Phân đoạn ký tự” và “Nhận diện ký tự” đã đặt ra ở đầu bài toán.

Loại biển	Số biển tìm thấy	Không sai	Sai 1 ký tự	Sai 2 ký tự	Sai 3 ký tự trở lên
Biển 1 hàng	182	61	88	19	14
Tỉ lệ (%)		33,5	48,4	10,4	7,7

Bảng 6.1 - 2 Tỉ lệ nhận diện sai ký tự ở biển 1 hàng

Loại biển	Số biển tìm thấy	Không sai	Sai 1 ký tự	Sai 2 ký tự	Sai 3 ký tự trở lên
Biển 2 hàng	924	286	273	175	190
Tỉ lệ nhận diện (%)		31	29,5	18,9	20,6

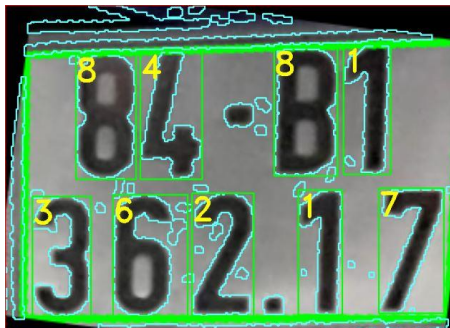
Bảng 6.1 - 3 Tỉ lệ nhận diện sai ký tự ở biển 2 hàng

Nhìn chung mô hình nhận diện KNN cũng khá tốt, có những ký tự dù bị mờ, bị nghiêng vẫn nhận diện đúng. Điều này một phần nhờ vào chương trình đã xoay biển số lại cho dễ tăng khả năng nhận diện, cho dù nghiêng thì ký tự cũng chỉ nghiêng từ  $3^\circ$  đến  $7^\circ$ . Tuy nhiên vẫn còn nhầm lẫn nhiều giữa các ký tự như số 1 với số 7. Chữ G, chữ D, số 6 với số 0. Chữ B với số 8...





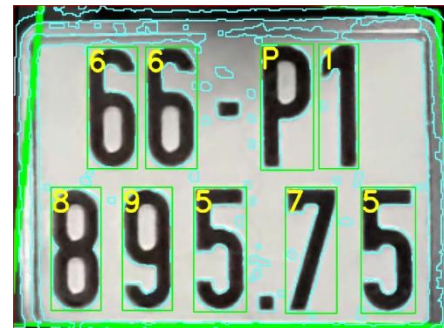
Hình 6.2 - 10 Ảnh gốc nhận diện 3 biển số



Hình 6.2 - 11 Biển số 1



Hình 6.2 - 12 Biển số 2



Hình 6.2 - 13 Biển số 3

Theo như hình gốc 6.2 - 10 ta nhận diện đúng hết biển số 2 và 3. Còn biển số 1 bị sai giữa chữ B và số 8



Hình 6.2 - 14 Không khoanh được vùng kí tự

Ở ảnh 6.2 - 14 chỗ số 0 bị dính con ốc dẫn đến không thể khoanh vùng được, Tuy nhiên ở chữ F mặc dù vẫn dính con ốc trong ảnh cắt ra để nhận diện, mô hình KNN vẫn cho ra đáp án đúng.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Từ những kết quả thu được ở chương trên em nhận thấy phương pháp nhận diện biển số xe bằng xử lý ảnh và thuật toán KNN có những ưu và khuyết điểm sau:

- Ưu điểm:
  - Dễ cài đặt và sử dụng.
  - Khá nhẹ nên máy tính với cấu hình yếu cũng có thể xử lý mượt mà so với các thuật toán khác như CNN, SVM.
  - Phù hợp cho đối tượng sinh viên muốn tìm hiểu căn bản về xử lý ảnh hay trí tuệ nhân tạo.
- Khuyết điểm:
  - Khả năng nhận diện của KNN còn thấp, khi tập dữ liệu quá nhiều sẽ tăng thời gian xử lý vì phải quét hết tập dữ liệu train.
  - Nhận diện kém với sự phản chiếu của biển số, sự di ảnh, chói sáng từ môi trường ngoài, những biển có phần chữ số không rõ ràng, với biển số xe ô tô

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

## Hướng phát triển

Cần thay đổi thuật toán nhận diện KNN sang những thuật toán khác tinh vi và phức tạp hơn như CNN, SVM hoặc có thể sử dụng những bộ thư viện đã có sẵn trên thế giới như YOLO, YOLOv3 ....

Sử dụng camera chuyên dụng cho việc nhận diện biển số xe vì có khả năng chống chịu với sương mù, đêm tối, chói sáng,...

Sử dụng các thuật toán xử lý ảnh khác để xác định vị trí biển số tốt hơn như phương pháp biến đổi Hough để nhận diện đường thẳng, xác định bằng màu sắc, những thuật toán làm hạn chế sự di ảnh khi xe đang di chuyển.

Kết hợp với những chương trình khác để quản lý kho bãi, quản lý các phương tiện đang tham gia giao thông, tìm xe thất lạc, theo dấu,...

