

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA TOÁN CƠ TIN HỌC



BÁO CÁO CUỐI KỲ  
MÔN HỌC THỊ GIÁC MÁY TÍNH

*Đề tài*

HỆ THỐNG NHẬN DIỆN BIỂN SỐ XE

Giảng viên: Hà Mạnh Toàn

Nhóm sinh viên thực hiện:

<i>Họ và tên</i>	<i>MSSV</i>
Nguyễn Văn Dũng	22001243
Hoàng Văn Dương	22001247

Hà Nội, Tháng 12/2024

# Mục lục

I.	Đặt vấn đề . . . . .	4
II.	Các khái niệm cơ bản . . . . .	4
1.	Biến số xe . . . . .	4
2.	Nhận diện biến số xe . . . . .	4
3.	Nhận diện đối tượng . . . . .	5
4.	Nhận diện ký tự quang học . . . . .	5
III.	Các phép xử lý ảnh được sử dụng . . . . .	5
1.	Gaussian Blur . . . . .	5
1.1.	Định nghĩa . . . . .	5
1.2.	Nguyên lý hoạt động . . . . .	6
1.3.	Hàm Gaussian và kernel . . . . .	6
2.	CLAHE . . . . .	7
2.1.	Định nghĩa . . . . .	7
2.2.	Nguyên lý hoạt động . . . . .	7
2.3.	Biểu diễn toán học . . . . .	8
3.	Erosion . . . . .	8
3.1.	Định nghĩa . . . . .	8
3.2.	Nguyên lý hoạt động . . . . .	9
3.3.	Biểu diễn toán học . . . . .	9
4.	Opening . . . . .	9
4.1.	Định nghĩa . . . . .	9
4.2.	Nguyên lý hoạt động . . . . .	10
4.3.	Biểu diễn toán học . . . . .	10
5.	Flood Fill . . . . .	10
5.1.	Định nghĩa . . . . .	10
5.2.	Nguyên lý hoạt động . . . . .	11
5.3.	Biểu diễn toán học . . . . .	11
IV.	Thiết kế hệ thống . . . . .	11
1.	Giới thiệu về hệ thống . . . . .	11
2.	Kiến trúc hệ thống . . . . .	12
V.	Triển khai hệ thống . . . . .	12
1.	Cài đặt môi trường và phần mềm . . . . .	12
2.	Tiền xử lý ảnh . . . . .	13
2.1.	Phương pháp tiền xử lý ảnh truyền thống . . . . .	13

2.2.	Phương pháp tiền xử lý ảnh với YOLOv8 . . . . .	17
3.	Huấn luyện mô hình YOLOv8 . . . . .	18
3.1.	Tham số đầu vào . . . . .	19
3.2.	Đánh giá mô hình . . . . .	19
4.	Nhận diện biển số xe với EasyOCR . . . . .	19
5.	Chuẩn hoá kết quả . . . . .	20
VI.	Đánh giá hệ thống . . . . .	20
1.	Đánh giá trên bộ dữ liệu . . . . .	20
2.	Đánh giá thực nghiệm . . . . .	21
3.	Nhận xét . . . . .	22
VII.	Tổng kết . . . . .	22
1.	Kết luận . . . . .	22
2.	Hướng phát triển . . . . .	22

# MỞ ĐẦU

Cùng với sự phát triển nhanh chóng của công nghệ, trí tuệ nhân tạo (AI) và thị giác máy tính (Computer Vision) đã trở thành nền tảng quan trọng trong việc xây dựng các hệ thống tự động và thông minh. Một trong những ứng dụng phổ biến của lĩnh vực này là nhận diện biển số xe, đóng vai trò quan trọng trong quản lý giao thông, điều tiết phương tiện và đảm bảo an ninh công cộng.

Trong bối cảnh đô thị hóa nhanh chóng, số lượng phương tiện giao thông tại Việt Nam không ngừng gia tăng, đặc biệt ở các thành phố lớn. Điều này đòi hỏi các hệ thống quản lý giao thông hiện đại, có khả năng xử lý lượng dữ liệu lớn và hoạt động trong thời gian thực. Tuy nhiên, đặc điểm riêng biệt của biển số xe Việt Nam, như định dạng ký tự và phong phú về kiểu dáng, đặt ra không ít thách thức trong việc xây dựng một hệ thống nhận diện tự động.

Đề án này tập trung vào việc nghiên cứu và phát triển hệ thống nhận diện ký tự biển số xe thời gian thực, ứng dụng các kỹ thuật xử lý ảnh và học máy tiên tiến. Mục tiêu là tạo ra một hệ thống hiệu quả, độ chính xác cao, phù hợp với điều kiện thực tế tại Việt Nam, góp phần thúc đẩy quá trình chuyển đổi số trong quản lý giao thông và dịch vụ công.

Trong báo cáo này, các bước thực hiện sẽ được trình bày rõ ràng, bao gồm: khảo sát và thu thập dữ liệu, xử lý ảnh để trích xuất vùng biển số, nhận diện ký tự sử dụng mô hình học sâu, và triển khai hệ thống hoạt động thời gian thực.

Trong thời gian thực hiện báo cáo không thể tránh khỏi thiếu sót. Nhóm rất mong nhận được những ý kiến đóng góp để có thể cải thiện và phát triển hơn nữa trong tương lai.

# I. Đặt vấn đề

Trong bối cảnh lượng phương tiện giao thông ngày càng tăng, đặc biệt tại các đô thị lớn, việc quản lý giao thông và đảm bảo an ninh công cộng đang trở thành thách thức cấp bách. Các giải pháp truyền thống, dựa trên kiểm tra thủ công hoặc hệ thống không tự động, thường không đáp ứng được yêu cầu về tốc độ và độ chính xác trong môi trường hiện đại. Điều này đặt ra nhu cầu cấp thiết cho các hệ thống tự động hóa, trong đó nhận diện biển số xe tự động là một công cụ hỗ trợ quan trọng.

Hệ thống nhận diện biển số xe tự động không chỉ giúp giảm tải cho các cơ quan chức năng bằng cách tự động phát hiện, nhận diện và lưu trữ thông tin phương tiện, mà còn góp phần cải thiện hiệu quả quản lý giao thông, giảm thiểu ùn tắc, và nâng cao an ninh tại các khu vực nhạy cảm.

Sự phát triển của trí tuệ nhân tạo đã mở ra những khả năng mới cho các hệ thống nhận diện trong thời gian thực. Những công nghệ này không chỉ đảm bảo độ chính xác cao trong việc phát hiện và nhận diện biển số, mà còn cho phép áp dụng hiệu quả trong các lĩnh vực khác như giao thông thông minh, quản lý logistics, và tự động hóa công nghiệp.

Hệ thống được thiết kế nhằm đáp ứng các yêu cầu thực tiễn trong việc nhận diện biển số xe tự động, áp dụng trên cả hình ảnh tĩnh và video thời gian thực.

## II. Các khái niệm cơ bản

### 1. Biển số xe

Biển số xe là một dãy ký tự, thường bao gồm cả chữ và số, được gắn lên phương tiện giao thông nhằm mục đích nhận dạng và quản lý các phương tiện này trong các hệ thống giao thông. Biển số xe có thể bao gồm các yếu tố như mã vùng, mã số đăng ký xe, và thông tin khác về phương tiện.

Biển số xe không chỉ dùng để nhận diện phương tiện trong các tình huống giao thông mà còn đóng vai trò quan trọng trong việc giám sát và quản lý phương tiện, đặc biệt trong các hệ thống giám sát an ninh và ứng dụng quản lý giao thông. Đối với các phương tiện tham gia giao thông, biển số xe là yếu tố quan trọng để xác định quyền sở hữu và tình trạng pháp lý của phương tiện đó.

Việc nhận diện biển số xe trong các hệ thống tự động là một phần quan trọng trong các ứng dụng như thu phí tự động, giám sát giao thông, và bảo mật khu vực.

### 2. Nhận diện biển số xe

Nhận diện biển số xe là quá trình tự động xác định và nhận dạng biển số của các phương tiện từ hình ảnh. Mục tiêu chính của nhận diện biển số là phát

hiện vị trí của biển số trong ảnh và trích xuất các ký tự trên đó. Đây là một quá trình bao gồm hai bước chính: phát hiện đối tượng và nhận diện ký tự quang học (OCR).

### 3. Nhận diện đối tượng

Nhận diện đối tượng là quá trình xác định vị trí (bounding box) và phân loại các đối tượng trong hình ảnh hoặc video theo các lớp được định nghĩa trước, kết hợp giữa phát hiện đối tượng và phân loại đối tượng.

### 4. Nhận diện ký tự quang học

Nhận diện ký tự quang học (OCR) là công nghệ chuyển đổi các ký tự trong ảnh thành văn bản có thể chỉnh sửa được. Trong hệ thống nhận diện biển số xe, OCR được sử dụng để trích xuất các ký tự từ biển số xe sau khi đã phát hiện và cắt ra từ ảnh. Quá trình OCR giúp chuyển các ký tự trên biển số thành dạng văn bản mà hệ thống có thể xử lý và lưu trữ.

OCR có thể được thực hiện bằng cách sử dụng các mô hình học sâu được huấn luyện trên một lượng lớn dữ liệu để nhận diện các ký tự trong các điều kiện khác nhau, như kiểu chữ, kích thước và độ mờ của chữ. Các thư viện OCR như EasyOCR sử dụng các mạng nơ-ron sâu để nhận diện ký tự trên biển số xe, giúp hệ thống có thể nhận diện chính xác ngay cả khi biển số có chất lượng ảnh không tốt hoặc có sự biến dạng.

## III. Các phép xử lý ảnh được sử dụng

### 1. Gaussian Blur

#### 1.1. Định nghĩa

Gaussian Blur là một kỹ thuật làm mờ ảnh bằng cách sử dụng hàm Gaussian để làm trung bình giá trị của các pixel trong vùng lân cận. Kỹ thuật này giúp làm mịn ảnh, giảm nhiễu và chi tiết không mong muốn, đồng thời bảo toàn các cấu trúc lớn. Trong quy trình tiền xử lý ảnh để xác định biển số xe, Gaussian Blur đóng vai trò quan trọng trong việc:

- Giảm nhiễu (noise) trong ảnh đầu vào, đảm bảo các bước xử lý tiếp theo không bị ảnh hưởng bởi các chi tiết không mong muốn.
- Làm mịn các vùng đồng nhất trên ảnh, giúp cải thiện hiệu quả của các thuật toán phát hiện cạnh hoặc phân đoạn ảnh.

#### Ưu điểm

- Hiệu quả trong việc giảm nhiễu và làm mịn ảnh mà không làm mất nhiều chi tiết lớn.
- Kernel Gaussian có tính chất toán học tốt, giúp tránh các hiện tượng bất thường trong xử lý ảnh.

### Hạn chế

- Làm mờ các chi tiết nhỏ và cạnh sắc, có thể không phù hợp cho các bài toán yêu cầu bảo toàn chi tiết.
- Hiệu quả tính toán giảm khi kích thước kernel hoặc độ phân giải ảnh lớn.

## 1.2. Nguyên lý hoạt động

- Hàm Gaussian được sử dụng để tạo ra một *kernel* hoặc *filter* có hình dạng giống phân phối chuẩn (*Gaussian distribution*).
- Mỗi pixel trong ảnh được thay thế bằng giá trị trung bình có trọng số của các pixel trong vùng lân cận, với trọng số được xác định bởi kernel Gaussian.
- Trọng số lớn nhất nằm ở trung tâm của kernel và giảm dần khi khoảng cách đến tâm tăng lên.

## 1.3. Hàm Gaussian và kernel

Hàm Gaussian 2D được định nghĩa như sau:

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

Trong đó:

- $x, y$ : Tọa độ pixel trong không gian kernel.
- $\sigma$ : Độ lệch chuẩn của phân phối Gaussian, quyết định độ mờ (blur) của ảnh.

Kernel Gaussian là một ma trận được xây dựng từ hàm Gaussian, với các giá trị được chuẩn hóa để tổng các phần tử bằng 1.

## 2. CLAHE

### 2.1. Định nghĩa

CLAHE (Contrast Limited Adaptive Histogram Equalization) là một kỹ thuật nâng cao dựa trên histogram Equalization, được thiết kế để cải thiện độ tương phản trong ảnh kỹ thuật số. CLAHE chia ảnh thành các vùng nhỏ gọi là *tiles* và thực hiện cân bằng histogram trên từng vùng. Kỹ thuật này giúp tránh các hiện tượng tăng cường quá mức độ tương phản, thường xảy ra trong Histogram Equalization thông thường.

Trong tiền xử lý ảnh để nhận diện biển số xe, CLAHE thường được sử dụng để:

- **Cải thiện nhận diện biển số xe:** Làm nổi bật các ký tự trên biển số, đặc biệt trong điều kiện ánh sáng yếu hoặc bóng tối.
- **Xử lý các vùng thiếu sáng:** Tăng cường chi tiết trong các vùng tối mà không làm mất thông tin ở vùng sáng.
- **Tăng độ tương phản cục bộ:** Nâng cao độ tương phản trong các vùng có chi tiết thấp, giúp cải thiện khả năng phát hiện ký tự trên biển số. Tăng cường các vùng có cường độ sáng yếu hoặc bị che khuất, đảm bảo thông tin đầy đủ cho các bước xử lý sau. Giúp cải thiện hiệu suất của các thuật toán phát hiện biên và trích xuất đặc trưng.

#### Ưu điểm

- Cải thiện độ tương phản cục bộ trong ảnh, làm nổi bật các chi tiết ở cả vùng sáng và vùng tối.
- Hạn chế hiện tượng tăng cường quá mức nhờ giới hạn độ tương phản (clip limit).

#### Hạn chế

- Yêu cầu thêm thời gian tính toán do thực hiện trên từng tile.
- Có thể gây nhiễu ở một số vùng đồng nhất nếu không chọn tham số hợp lý.

### 2.2. Nguyên lý hoạt động

- Ảnh được chia thành các vùng nhỏ (tiles).
- Histogram của mỗi tile được cân bằng cục bộ, làm tăng độ tương phản trong vùng đó.
- Giới hạn độ tương phản được áp dụng bằng cách cắt bớt các giá trị histogram vượt ngưỡng (clip limit) để tránh tăng cường quá mức.



- Các vùng lân cận được kết hợp lại bằng nội suy để tạo ra ảnh cuối cùng, đảm bảo chuyển tiếp mượt mà giữa các vùng.

### 2.3. Biểu diễn toán học

Histogram của một vùng ảnh được biểu diễn như một hàm phân phối xác suất:

$$P(i) = \frac{h(i)}{N}$$

Trong đó:

- $h(i)$ : Số lượng pixel có mức xám  $i$ .
- $N$ : Tổng số pixel trong vùng.

CLAHE giới hạn giá trị tối đa của  $P(i)$  dựa trên *clip limit*, sau đó tái phân phối các giá trị bị cắt vào các mức xám khác.

## 3. Erosion

### 3.1. Định nghĩa

Erosion (xói mòn) là một phép toán hình thái học (morphological operation) được áp dụng trên ảnh nhị phân hoặc ảnh mức xám, sử dụng một cấu trúc phần tử (structuring element/kernel). Phép toán này giúp loại bỏ các pixel biên, làm giảm kích thước của các đối tượng hoặc loại bỏ các chi tiết nhỏ không mong muốn.

Trong quy trình tiền xử lý ảnh, Erosion có vai trò quan trọng như:

- Loại bỏ các chi tiết nhỏ hoặc nhiễu không mong muốn.
- Thu nhỏ kích thước các đối tượng, hỗ trợ trong các bước phân tích hình thái học khác.
- Tách biệt các đối tượng dính liền nhờ làm mất đi các cầu nối mỏng.

#### Ưu điểm

- Hiệu quả trong việc làm giảm các vùng nhiễu nhỏ hoặc loại bỏ các chi tiết không liên quan.
- Dễ dàng kết hợp với các phép toán khác như Dilation để thực hiện các thao tác hình thái phức tạp (Opening, Closing).

#### Hạn chế

- Có thể làm mất đi các chi tiết nhỏ mong muốn trong đối tượng.
- Kết quả phụ thuộc nhiều vào kích thước và hình dạng của kernel.

### 3.2. Nguyên lý hoạt động

- Với mỗi pixel trong ảnh, Erosion kiểm tra các pixel lân cận trong vùng được xác định bởi kernel.
- Pixel trung tâm được giữ lại ( $value = 1$ ) nếu và chỉ nếu tất cả các pixel trong vùng kernel đều có giá trị 1. Nếu không, pixel trung tâm sẽ bị loại bỏ ( $value = 0$ ).
- Đối với ảnh mức xám, giá trị của pixel trung tâm được thay thế bằng giá trị tối thiểu trong vùng kernel.

### 3.3. Biểu diễn toán học

$$A \ominus B = \{z \mid (B_z \subseteq A)\}$$

Trong đó:

- $A$ : Ảnh gốc.
- $B$ : Kernel (*structuring element*).
- $B_z$ : Kernel  $B$  được dịch chuyển đến vị trí  $z$ .

## 4. Opening

### 4.1. Định nghĩa

Opening là một phép toán trong xử lý ảnh dựa trên kỹ thuật *morphological*, kết hợp hai phép cơ bản là *Erosion* (co rút) và *Dilation* (giãn nở). Cụ thể, Opening được định nghĩa là phép Erosion sau đó là phép Dilation, sử dụng cùng một *structuring element*. Phép toán này thường được sử dụng để loại bỏ các đối tượng nhỏ không mong muốn trong ảnh, đồng thời bảo toàn hình dạng chính của các đối tượng lớn hơn.

Trong tiền xử lý ảnh để nhận diện biển số xe, Opening được áp dụng để:

- **Loại bỏ nhiễu nhỏ:** Xử lý các hạt nhiễu hoặc đốm nhỏ không liên quan đến biển số xe.
- **Tăng độ mượt của biên đối tượng:** Giảm các chi tiết răng cưa hoặc gồ ghề trên biên của biển số xe.
- **Làm sạch vùng nền:** Loại bỏ các đối tượng nhỏ không mong muốn trong vùng nền của ảnh biển số xe. Làm mịn biên của các đối tượng quan tâm, giúp cải thiện hiệu suất của các bước xử lý tiếp theo.

**Ưu điểm**

- Hiệu quả trong việc loại bỏ nhiễu nhỏ hoặc đối tượng không mong muốn mà vẫn giữ được cấu trúc chính.
- Cải thiện độ chính xác trong phát hiện biên và phân đoạn đối tượng.

### Hạn chế

- Có thể làm mất các chi tiết nhỏ cần thiết nếu không chọn kích thước structuring element hợp lý.
- Hiệu quả phụ thuộc nhiều vào hình dạng và kích thước của structuring element.

## 4.2. Nguyên lý hoạt động

- **Erosion:** Co rút vùng sáng bằng cách loại bỏ các pixel sáng nằm cạnh pixel tối, làm giảm kích thước các vùng sáng.
- **Dilation:** Giãn nở vùng sáng bằng cách mở rộng các pixel sáng, phục hồi hình dạng ban đầu của đối tượng lớn.
- **Kết hợp:** Sau khi co rút bằng Erosion để loại bỏ nhiễu nhỏ, phép Dilation phục hồi các đối tượng lớn bị giảm kích thước, đảm bảo kết quả cuối cùng sạch hơn nhưng vẫn bảo toàn hình dạng.

## 4.3. Biểu diễn toán học

Gọi  $A$  là ảnh gốc và  $B$  là structuring element, phép Opening được định nghĩa:

$$A \circ B = (A \ominus B) \oplus B$$

Trong đó:

- $\ominus$ : Phép Erosion.
- $\oplus$ : Phép Dilation.

Phép toán này sử dụng structuring element  $B$  để xác định vùng ảnh cần loại bỏ hoặc giữ lại.

## 5. Flood Fill

### 5.1. Định nghĩa

Flood Fill là một thuật toán được sử dụng để tô màu hoặc xác định một vùng liên thông trong ảnh, bắt đầu từ một điểm ban đầu (*seed point*). Vùng liên thông được xác định dựa trên sự tương đồng về giá trị pixel trong không gian ảnh.

Trong tiền xử lý ảnh để nhận diện biển số xe, Flood Fill được áp dụng để:

- **Lấp đầy các lỗ hổng trong vùng quan tâm.**
- **Xác định vùng chứa biển số xe:** Làm nổi bật vùng liên thông có giá trị pixel tương tự với biển số xe.
- **Lấp đầy vùng bị khuyết:** Xử lý các lỗ nhỏ trong vùng biển số xe, đảm bảo hình dạng hoàn chỉnh.
- **Loại bỏ nhiễu không mong muốn:** Làm rõ các vùng không mong muốn dựa trên liên thông về giá trị pixel từ đó giảm thiểu tính toán, nhiễu cho các bước tiếp theo

## 5.2. Nguyên lý hoạt động

- Thuật toán bắt đầu từ một pixel ban đầu (*seed point*) và kiểm tra các pixel lân cận (theo 4 hoặc 8 chiều kết nối).
- Các pixel lân cận sẽ được thêm vào vùng liên thông nếu chúng thỏa mãn điều kiện (ví dụ: giá trị pixel nằm trong ngưỡng cho phép).
- Quá trình lặp lại cho đến khi không còn pixel lân cận nào thỏa mãn điều kiện.
- Vùng liên thông sẽ được tô màu hoặc gán giá trị mới để làm nổi bật.

## 5.3. Biểu diễn toán học

Gọi  $I(x, y)$  là giá trị pixel tại vị trí  $(x, y)$  trong ảnh và  $T$  là ngưỡng liên thông. Vùng liên thông  $R$  được xác định như sau:

$$R = \{(x, y) \mid |I(x, y) - I(x_s, y_s)| \leq T\}$$

Trong đó:

- $x_s, y_s$ : Tọa độ của điểm ban đầu (seed point).
- $T$ : Ngưỡng liên thông, xác định độ tương tự giữa các pixel.

# IV. Thiết kế hệ thống

## 1. Giới thiệu về hệ thống

Hệ thống nhận diện biển số xe được thiết kế để tự động nhận diện và giải mã biển số xe từ các hình ảnh đầu vào. Mục tiêu của hệ thống là cải thiện độ chính xác và tốc độ nhận diện trong môi trường thực tế, bao gồm các điều kiện ánh sáng thay đổi, góc chụp không đồng nhất, và biển số bị che khuất một phần.

## 2. Kiến trúc hệ thống

Hệ thống được chia thành các thành phần chính:

- **Input:** Ảnh đầu vào từ camera hoặc nguồn dữ liệu hình ảnh.
- **Preprocessing:** Bước tiền xử lý ảnh, gồm hai phương pháp truyền thống và YOLOv8.
- **OCR:** Sử dụng EasyOCR để nhận diện các ký tự từ biển số xe.
- **Output:** Kết quả nhận diện biển số xe dưới dạng chuỗi ký tự.

Hệ thống sử dụng phần cứng GPU để tăng tốc quá trình xử lý, đặc biệt là trong bước phát hiện đối tượng và nhận diện ký tự.

## V. Triển khai hệ thống

### 1. Cài đặt môi trường và phần mềm

Để triển khai hệ thống nhận diện biển số xe, các phần mềm và thư viện sau cần được cài đặt:

- **Python:** Là ngôn ngữ lập trình chủ đạo để phát triển hệ thống sử dụng phiên bản Python 3.12.6 và các phiên bản Python 3.8 hoặc cao hơn cũng có thể sử dụng được. Python thể được cài đặt thông qua trang chủ chính thức python.org. Python hỗ trợ nhiều thư viện mạnh mẽ cho xử lý ảnh và học sâu, làm cho nó trở thành lựa chọn lý tưởng cho phát triển hệ thống nhận diện biển số.
- **Thư viện EasyOCR:** Thư viện nhận diện ký tự quang học này sử dụng các mô hình học sâu để nhận diện các ký tự trong biển số xe. Sau khi phát hiện biển số, EasyOCR được sử dụng để nhận diện các ký tự từ biển số đã được cắt ra.
- **YOLOv8:** YOLOv8 là một mô hình phát hiện đối tượng mạnh mẽ, được sử dụng để phát hiện và phân vùng các biển số xe trong ảnh. YOLOv8 có thể nhận diện các đối tượng với độ chính xác cao và tốc độ xử lý nhanh, rất phù hợp cho các ứng dụng thời gian thực giúp phát hiện các vùng biển số trong ảnh và cắt chúng ra để xử lý thêm.
- **OpenCV:** Thư viện mã nguồn mở mạnh mẽ cho xử lý ảnh và video, OpenCV cung cấp các công cụ hữu ích để thao tác với hình ảnh như đọc, hiển thị, chỉnh sửa, và xử lý ảnh. OpenCV hỗ trợ nhiều chức năng, bao gồm chuyển đổi ảnh sang các không gian màu khác nhau, thao tác hình học ảnh, cắt và thay đổi kích thước ảnh, cũng như các kỹ thuật

nâng cao như phát hiện đối tượng và nhận diện khuôn mặt. Trong hệ thống này, OpenCV được sử dụng để cắt biên số từ ảnh, chuyển đổi ảnh thành ảnh xám (grayscale), vẽ bounding box lên ảnh, và thực hiện các phép toán hình thái học như lọc Gaussian và toán học morphology.

- **NumPy:** Thư viện mạnh mẽ hỗ trợ các thao tác với mảng số học (array) và tính toán số học. NumPy cung cấp các cấu trúc dữ liệu mảng đa chiều (ndarray) và các hàm toán học cần thiết để xử lý và phân tích dữ liệu số. Trong hệ thống nhận diện biển số, NumPy được sử dụng để thao tác với các mảng ảnh, chuyển đổi ảnh giữa các định dạng khác nhau (ví dụ: từ OpenCV sang các mảng numpy), và hỗ trợ các phép toán hình học ảnh. Nó cũng được sử dụng trong các tác vụ như xử lý ảnh và chuẩn bị dữ liệu cho các mô hình học máy.

## 2. Tiền xử lý ảnh

Hệ thống sử dụng hai phương pháp tiền xử lý: phương pháp truyền thống và phương pháp YOLOv8.

### 2.1. Phương pháp tiền xử lý ảnh truyền thống

Phương pháp truyền thống bao gồm các bước sau để phát hiện và phân vùng biển số xe:

- **Làm mờ ảnh:** Mục tiêu chính của việc làm mờ ảnh là làm mờ chi tiết của ảnh, ví dụ như giảm sự sắc nét của cạnh và chi tiết nhỏ, qua đó làm mềm các chi tiết sắc nét hoặc làm nổi bật các vùng lớn hơn.
- **Giảm nhiễu:** Các bộ lọc Gaussian hoặc Median Filtering được sử dụng để giảm nhiễu trong ảnh, giúp cải thiện chất lượng ảnh và giảm thiểu tác động của nhiễu trong quá trình nhận diện.
- **Cải thiện độ tương phản:** Sử dụng các kỹ thuật như histogram equalization hoặc CLAHE để làm nổi bật vùng biển số trong ảnh với giới hạn mức tăng cường nhằm tránh nhiễu. Bên cạnh đó, ta sử dụng toán tử mở (opening) để làm mờ các vùng nhiễu lớn, sau đó loại bỏ các vùng nhiễu này bằng cách dùng ảnh ban đầu trừ đi ảnh đã làm mờ bằng phép trừ pixel-wise. Các kỹ thuật này giúp các chi tiết của biển số trở nên rõ ràng hơn.
- **Chuyển ảnh thành đen trắng:** Mục tiêu của bước này là chuyển ảnh từ không gian màu (RGB hoặc Grayscale) thành ảnh nhị phân (đen trắng), giúp làm nổi bật các đối tượng quan trọng trong ảnh. Phương pháp này thường được sử dụng để chuẩn bị dữ liệu cho các bước phân đoạn hoặc nhận diện tiếp theo.

- **Loại bỏ nhiễu nhỏ và chi tiết không mong muốn:** Trong ảnh nhị phân, nhiễu thường xuất hiện dưới dạng các điểm nhỏ (chấm trắng) nằm rải rác trên vùng nền đen và không liên quan đến các đối tượng chính trong ảnh. Điều này gây cản trở các thuật toán phát hiện đối tượng, phân đoạn, hoặc nhận dạng, tăng độ phức tạp khi xử lý ảnh, làm giảm độ chính xác của các bước tiếp theo. Ở đây, ta sử dụng phép toán hình thái học (Morphological Operations) Erosion nhằm loại bỏ những điểm nhiễu này do kích thước nhỏ của chúng.
- **Lấp đầy các vùng trống trong ảnh:** Biển số xe thường là một vùng chữ nhật liên thông trong ảnh, với sự khác biệt rõ ràng giữa ký tự (trắng) và nền (đen) sau khi nhị phân hóa. Trong các trường hợp ảnh có nhiễu hoặc các chi tiết nền phức tạp, việc xác định vùng biển số trở nên khó khăn. Do đó, để xác định và làm nổi bật vùng liên thông đại diện cho biển số xe. Ở đây ta ứng dụng thuật toán flood fill nhằm lấp đầy các lỗ hổng không liên thông trên ảnh, qua đó giảm bớt các cạnh không cần thiết trong ảnh, từ đó dễ xác định cạnh của biển số xe hơn.
- **Cắt ảnh (ROI):** Việc cắt ảnh vùng biển số xe (Region of Interest - ROI) được thực hiện bằng cách sử dụng các contours để xác định vị trí và kích thước của vùng chứa biển số. Ta sử dụng các contours để xác định và cắt ROI dựa trên thông tin các tiêu chí như độ dài cạnh, diện tích contours, contours có khép kín hay không, ...

Đối với những bức ảnh tương đối rõ nét và không chứa nhiều vật thể gây nhiễu, phương pháp phân vùng hiện tại có thể hoạt động khá chính xác. Tuy nhiên, khi xử lý các bức ảnh biển số xe bị nghiêng, méo hoặc có sự xuất hiện của nhiều vật thể có hình dạng tương tự biển số xe, phương pháp này dễ dẫn đến việc phân vùng sai, chẳng hạn như cắt nhầm hoặc bỏ sót vùng biển số.

Dưới đây là ví dụ minh họa cho thấy phương pháp truyền thống hoạt động kém hiệu quả:



Hình 1: Biển số trước khi cắt bằng phương pháp truyền thống





Hình 2: Biển số sau khi cắt bằng phương pháp truyền thống

Ví dụ trên cho thấy biển số xe có thể bị nghiêng, khiến phương pháp nhận diện truyền thống dựa trên các contours hình chữ nhật không thể phát hiện chính xác, do biển số đã biến dạng theo góc nhìn thành hình bình hành.

Để khắc phục các hạn chế này, việc sử dụng các mô hình phát hiện vật thể tiên tiến như YOLOv8 là cần thiết. YOLOv8 với khả năng nhận diện nhanh chóng và chính xác trong nhiều điều kiện hình ảnh phức tạp sẽ hỗ trợ hiệu quả hơn trong việc xác định và phân vùng biển số xe, đảm bảo độ tin cậy cao hơn trong quá trình xử lý dữ liệu.



Hình 3: Biển số sau khi cắt bằng YOLOv8

Mặc dù các ký tự trên hình ảnh sau khi cắt quá mờ và không đủ rõ ràng để nhận diện, nhưng ví dụ này cho thấy phương pháp YOLOv8 vẫn hoạt động hiệu quả, ngay cả với những biển số xe được chụp từ các góc nghiêng khó nhận diện.

## 2.2. Phương pháp tiền xử lý ảnh với YOLOv8

YOLOv8 được sử dụng để phát hiện và phân vùng biển số xe trong ảnh. Quy trình tiền xử lý với YOLOv8 bao gồm:

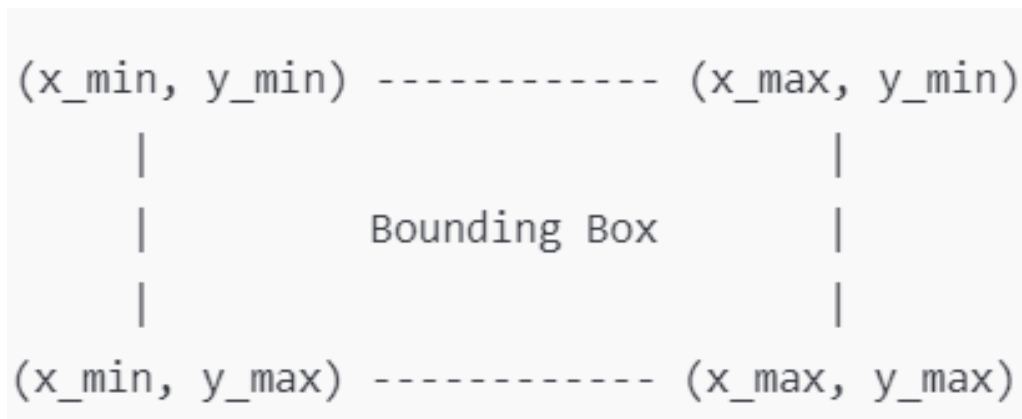
- **Phát hiện biển số bằng YOLOv8:** Mô hình YOLOv8 được huấn luyện để phát hiện biển số xe trong ảnh. Quá trình phát hiện này sẽ xác định chính xác vị trí của biển số trong ảnh, giúp tăng độ chính xác khi cắt biển số.
- **Cắt và chuẩn hóa ảnh:** Sau khi YOLOv8 phát hiện biển số, ảnh sẽ được cắt ra và chuẩn hóa kích thước về một kích thước cố định, phù hợp với yêu cầu của mô hình OCR. Việc này giúp giảm sự biến đổi trong ảnh và tạo điều kiện thuận lợi cho quá trình nhận diện ký tự.

- **Tận dụng GPU:** Để tối ưu hóa tốc độ phát hiện, mô hình YOLOv8 được chạy trên GPU. Việc này giúp giảm thời gian xử lý, đặc biệt trong môi trường thực tế khi cần nhận diện biển số xe một cách nhanh chóng.

### 3. Huấn luyện mô hình YOLOv8

Mô hình YOLOv8 được huấn luyện với bộ dữ liệu **Vietnam License Plate**, bao gồm các hình ảnh biển số xe ô tô và xe máy bao gồm:

- **Tập Images:** gồm 4578 hình ảnh, bao gồm các biển số xe ô tô và xe máy, trong đó tập train gồm 3433 ảnh, tập valid gồm 1145 ảnh
- **Tập Labels:** gồm các file .txt đánh dấu tọa độ 4 góc của biển số và phân nhóm thành 2 loại biển số (Định dạng <classId> <x1> <y1> <x2> <y2> <x3> <y3> <x4> <y4>)



Hình 4: Biểu diễn tọa độ tương ứng với tệp được gán nhãn

Các bước huấn luyện mô hình:

- **Cấu hình mô hình:** Các tham số của mô hình YOLOv8, bao gồm số lớp, kích thước ảnh đầu vào, và learning rate, được điều chỉnh để phù hợp với bài toán nhận diện biển số xe. Các tham số này ảnh hưởng trực tiếp đến hiệu quả và độ chính xác của mô hình.
- **Huấn luyện trên GPU:** Để giảm thời gian huấn luyện, mô hình YOLOv8 được huấn luyện sử dụng GPU. Quá trình huấn luyện này được thực hiện trên một máy chủ có GPU hỗ trợ CUDA.
- **Đánh giá và tối ưu hóa:** Sau khi huấn luyện, mô hình được đánh giá với các chỉ số như Precision, Recall, và mAP (mean Average Precision). Các chỉ số này giúp đánh giá độ chính xác của mô hình trong việc phát hiện biển số xe. Dựa trên các kết quả đánh giá, mô hình có thể được tối ưu hóa để cải thiện hiệu suất.

### 3.1. Tham số đầu vào

Mô hình xử dụng các tham số chính sau:

- **Epochs:** Số lần huấn luyện mà mô hình sẽ trải qua. Đối với mô hình hiện tại, mô hình sẽ huấn luyện trong 3 epochs. Mỗi epoch là một lần duyệt qua toàn bộ dữ liệu huấn luyện. Số lần huấn luyện tăng lên đồng nghĩa với mô hình sẽ được duyệt qua toàn bộ dữ liệu nhiều hơn làm cho mô hình càng hiểu dữ liệu huấn luyện tuy nhiên nếu tăng lên quá nhiều thì thời gian huấn luyện sẽ lớn và có khả năng bị overfitting.
- **Imgsz:** Kích thước hình ảnh đầu vào mà mô hình sẽ sử dụng trong quá trình huấn luyện. Tất cả ảnh sẽ được thay đổi kích thước trước khi đưa vào mô hình giúp tối ưu hóa tốc độ và hiệu suất huấn luyện. Dựa trên kích thước biến số xe thường thấy imgsiz được lựa chọn có giá trị 640 x 640.
- **Batch:** Số lượng mẫu trong mỗi bước huấn luyện. Đối với mô hình hiện tại, batch size là 16, có nghĩa là mỗi lần mô hình sẽ huấn luyện với 16 ảnh. Kích thước batch lớn hơn có thể giúp cải thiện độ chính xác nhưng cũng yêu cầu bộ nhớ GPU lớn hơn.

### 3.2. Đánh giá mô hình

Sau khi huấn luyện và lựa chọn mô hình tốt nhất các tham số để đánh giá mô hình trên tập kiểm tra các tham số trả về đem lại kết quả khá tốt. Cụ thể, độ chính xác (precision) và độ nhạy (recall) đều nằm trong khoảng 96 - 97 %. Điều này cho thấy mô hình có thể sử dụng để nhận diện biển số xe một cách hiệu quả

## 4. Nhận diện biển số xe với EasyOCR

Sau khi ảnh biển số xe được tiền xử lý và cắt ra bằng YOLOv8, ảnh sẽ được đưa vào thư viện EasyOCR để nhận diện ký tự. Quy trình nhận diện biển số xe với EasyOCR bao gồm:

- **Chuẩn bị ảnh đầu vào:** Các ảnh biển số sau khi được cắt ra và chuẩn hóa sẽ được chuyển vào thư viện EasyOCR. EasyOCR hỗ trợ nhận diện ký tự trong các ngôn ngữ khác nhau, bao gồm tiếng Việt.
- **Nhận diện ký tự:** EasyOCR sẽ thực hiện nhận diện ký tự trên biển số xe. Kết quả trả về là chuỗi ký tự chứa thông tin biển số xe, bao gồm các chữ cái và số.

## 5. Chuẩn hoá kết quả

Sau khi phân tích kết quả đầu ra, nhóm nhận thấy rằng các ký tự chữ và số trên biển số xe đã được nhận diện với độ chính xác cao. Tuy nhiên, đối với biển số xe Việt Nam, dấu "-" xuất hiện thường xuyên nhưng không ổn định trong quá trình nhận diện. Cụ thể, dấu "-" đôi khi được nhận diện chính xác, nhưng cũng có trường hợp bị nhận dạng sai thành các ký tự đặc biệt khác. Vì các ký tự này không mang giá trị phân tích thực tiễn, nhóm đã thực hiện chuẩn hóa kết quả bằng cách loại bỏ các ký tự không cần thiết và chỉ giữ lại các ký tự chữ và số.

Ngoài ra, nhóm cũng nhận thấy rằng ký tự thứ 3 trên biển số xe Việt Nam luôn là chữ cái. Do đó, các cặp ký tự dễ gây nhầm lẫn như 'B' và '8', 'Z' và '2' đã được chuẩn hóa thành chữ cái tương ứng để đảm bảo tính chính xác trong quá trình nhận diện và xử lý dữ liệu. Việc này giúp giảm thiểu sai sót và nâng cao hiệu quả của hệ thống.

## VI. Đánh giá hệ thống

### 1. Đánh giá trên bộ dữ liệu

Trong quá trình đánh giá hệ thống, nhóm đã chọn lọc ảnh kiểm thử sao cho phản ánh chính xác điều kiện thực tế. Các bức ảnh được chọn theo tiêu chí không quá mờ nhưng cũng không quá đẹp, nhằm đảm bảo tính đại diện cho các tình huống gặp phải trong thực tế. Bộ dữ liệu kiểm thử bao gồm 115 ảnh, bao gồm cả ảnh của ô tô và xe máy và độc lập hoàn toàn so với bộ dữ liệu dùng để huấn luyện mô hình YOLOv8. Nhóm đã ghi nhận kết quả về độ chính xác nhận diện như sau:

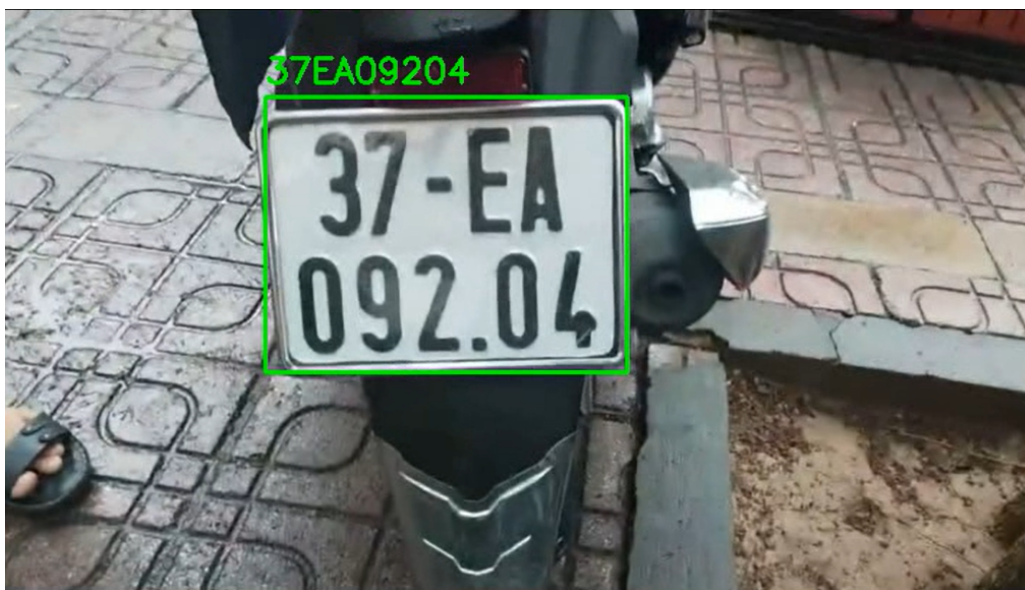
Loại kết quả	Tỷ lệ
Chính xác tuyệt đối	57.5 %
Sai 1 ký tự	18.33 %

Bảng 1: Kết quả đánh giá độ chính xác nhận diện

Một số đặc trưng được rút ra từ kết quả rằng một số ký tự như "B" và "8", "1" và "7" dễ bị nhầm lẫn, đặc biệt khi phương tiện di chuyển nhanh. Mặc dù vậy, kết quả vẫn chấp nhận được, và nhóm sẽ tiếp tục cải thiện hệ thống, đặc biệt là trong việc giảm nhầm lẫn ký tự và tăng cường dữ liệu huấn luyện cho tình huống video động.

## 2. Đánh giá thực nghiệm

Hệ thống đã hoạt động ổn định khi camera di chuyển từ khoảng cách 3 đến 5 mét, với độ chính xác cao trong việc nhận diện biển số từ các phương tiện di chuyển chậm. Tuy nhiên, đối với các video quay từ camera di động khi phương tiện di chuyển nhanh hoặc khoảng cách quay xa dẫn đến độ phân giải của ảnh không cao, hệ thống đôi khi gặp khó khăn trong việc nhận diện chính xác, đặc biệt là trong việc nhận diện các ký tự nhỏ, bị mờ hoặc dễ bị nhầm lẫn.



Hình 5: Kết quả thực nghiệm chính xác



Hình 6: Kết quả thực nghiệm không chính xác

Trên hình 3 ta có thể thấy biển số xe xuất hiện hình tròn không mong muốn do đó kết quả nhận diện bị ảnh hưởng.

### 3. Nhận xét

Kết quả thu được từ hệ thống cho thấy mức độ nhận diện ổn định, với hơn một nửa số biển số được nhận diện chính xác tuyệt đối. Tuy nhiên, tỷ lệ sai một ký tự vẫn chiếm một phần đáng kể, điều này chỉ ra rằng hệ thống vẫn cần có sự cải tiến để giảm thiểu các lỗi nhỏ trong quá trình nhận diện ký tự.

## VII. Tổng kết

### 1. Kết luận

Hệ thống nhận diện biển số xe đã đạt được kết quả khả quan với độ chính xác ổn định trong điều kiện thử nghiệm. Mặc dù một số ký tự vẫn bị nhầm lẫn, đặc biệt trong video thời gian thực và dữ liệu ảnh không tốt, nhưng hệ thống đã thể hiện hiệu suất tốt. Việc cải thiện các bước tiền xử lý ảnh, tối ưu hóa thuật toán và mở rộng dữ liệu huấn luyện là các hướng phát triển tiếp theo để nâng cao độ chính xác và khả năng xử lý trong các tình huống thực tế.

Mặc dù kết quả chưa đạt mức tối ưu, nhưng đối với những người mới bắt đầu tìm hiểu về lĩnh vực thị giác máy tính nói riêng và nhận diện biển số xe nói chung, các kết quả này là khá khả quan và cho thấy hệ thống có tiềm năng phát triển. Nhóm nhận thấy rằng việc tối ưu hóa các bước tiền xử lý ảnh và cắt ảnh sẽ góp phần nâng cao hiệu suất nhận diện. Bên cạnh đó, việc bổ sung dữ liệu huấn luyện đa dạng và áp dụng các thuật toán hiện đại hơn sẽ là những phương hướng triển khai hiệu quả trong tương lai để cải thiện kết quả nhận diện.

### 2. Hướng phát triển

Trong tương lai, nhóm sẽ hướng đến việc tự xây dựng mô hình phát hiện vật thể và nhận diện ký tự quang học, thay vì phụ thuộc vào các thư viện có sẵn như YOLO và EasyOCR. Việc này sẽ giúp tăng cường khả năng kiểm soát quá trình phát triển hệ thống, đồng thời cải thiện hiệu quả và tính linh hoạt của mô hình. Việc xây dựng mô hình từ đầu sẽ đòi hỏi đội ngũ phải nghiên cứu và phát triển các thuật toán học sâu cho cả hai nhiệm vụ là phát hiện biển số và nhận diện ký tự quang học. Đây là một thách thức lớn nhưng cũng sẽ mở ra cơ hội phát triển một hệ thống mạnh mẽ và tùy chỉnh, phù hợp với các điều kiện thực tế khác nhau.

# Tài liệu tham khảo

- [1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788). DOI: <https://doi.org/10.1109/CVPR.2016.91>
- [2] JaiedAI. (2020). EasyOCR: A Python package for OCR (Optical Character Recognition). GitHub repository. Retrieved from <https://github.com/JaiedAI/EasyOCR>
- [3] Silva, S. M., & Jung, C. R. (2020). Real-time license plate detection and recognition using deep convolutional neural networks. *Journal of Visual Communication and Image Representation*, 102773. DOI: <https://doi.org/10.1016/j.jvcir.2020.102773>
- [4] Tao, L., Hong, S., Lin, Y., Chen, Y., He, P., & Tie, Z. (2024). A Real-Time License Plate Detection and Recognition Model in Unconstrained Scenarios. *Sensors*, 24(9), 2791. DOI: <https://doi.org/10.3390/s24092791>
- [5] Bùi Quang Mạnh. (n.d.). Nhận diện biển số xe Việt Nam.