

# ĐẠI HỌC XÂY DỰNG HÀ NỘI

*Khoa Công nghệ Thông tin*

## HỆ THỐNG NHẬN DIỆN BIỂN SỐ XE

### Sinh viên thực hiện:

Nguyễn Minh Năng	MSSV: 0211668
Đỗ Công Trí	MSSV: 0214268
Nguyễn Đức Mạnh	MSSV: 0210668

Hà Nội, 2025

## TÓM TẮT

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 chúng 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.

**Từ khóa:** License Plate Recognition, YOLOv8, Xử lý ảnh, Deep Learning

# 1 Giới thiệu

## 1.1 Tổng quan

Đề tài tập trung vào các nội dung chính sau:

- 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 ký tự ứng dụng trong việc nhận dạng biển số xe

## 1.2 Nhiệm vụ đề tài

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

1. Tìm hiểu khái quát về xử lý ảnh và bài toán nhận dạng biển số xe
2. Tìm hiểu thông tin về biển số xe và phân loại biển số xe của Việt Nam
3. 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 ký tự trong biển số xe
  - Nhận dạng ký tự
4. Thực nghiệm code và đánh giá kết quả

## 2 Tổng quan bài toán và các phương pháp

### 2.1 Tổng quan bài toán nhận diện biển số xe

#### 2.1.1 Khái niệm biển số xe

Ở Việt Nam, biển kiểm soát xe cơ giới (hay còn gọi tắt là biển kiểm soát, biển số xe) là tấm biển gắn trên mỗi xe cơ giới, được cơ quan công an cấp (đối với xe quân sự do Bộ Quốc phòng cấp) khi mua xe mới hoặc chuyển nhượng xe. Biển số xe được làm bằng hợp kim nhôm sắt, có dạng hình chữ nhật hoặc hơi vuông, trên đó có in số và chữ cho biết vùng và địa phương quản lý.

#### Tiêu chuẩn kích thước biển số xe

- **Loại biển số dài:** Chiều cao 110 mm, chiều dài 470 mm
- **Loại biển số ngắn:** Chiều cao 200 mm, chiều dài 280 mm
- **Tỉ lệ cao/rộng:**
  - Biển một hàng:  $3.5 \leq \text{cao/rộng} \leq 6.5$
  - Biển hai hàng:  $0.8 \leq \text{cao/rộng} \leq 1.5$
- **Số lượng ký tự:** Từ 7 đến 9 ký tự
- **Kích thước ký tự:** Chiều cao 80mm, chiều rộng 40mm

#### 2.1.2 Xử lý ảnh

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.

### 2.2 Tổng quan các nghiên cứu liên quan

Trong lĩnh vực nhận diện biển số xe (License Plate Recognition – LPR), các phương pháp đã phát triển mạnh mẽ qua nhiều giai đoạn, từ xử lý ảnh cổ điển, học máy truyền thống đến các mô hình học sâu hiện đại.

#### 2.2.1 Phương pháp Cổ điển (Gllavata et al., 2003)

Nghiên cứu của Gllavata là đại diện tiêu biểu cho hướng tiếp cận dựa trên các đặc trưng mức thấp của ảnh trong bài toán phát hiện văn bản.

**Mục tiêu:** Phát hiện văn bản dựa trên giả định rằng các vùng văn bản có độ tương phản cao so với nền và có các cạnh biên sắc nét.

**Ưu điểm:**

- Thuật toán đơn giản, dễ cài đặt.
- Tốc độ xử lý nhanh, không yêu cầu phần cứng mạnh.

**Nhược điểm:**

- Chỉ hoạt động hiệu quả với văn bản nằm ngang.
- Nhạy cảm với nhiễu, nền phức tạp và điều kiện ánh sáng thay đổi.

**Kết quả:** Phương pháp đạt Recall khoảng 88.7% và Precision khoảng 83.9% trên tập dữ liệu tự xây dựng do chưa tồn tại bộ benchmark chuẩn tại thời điểm nghiên cứu.

**Nhận xét:** Phương pháp cổ điển chỉ phù hợp cho tài liệu scan hoặc ảnh có nền đơn giản, không đáp ứng tốt các bài toán ảnh tự nhiên phức tạp hiện nay.

### 2.2.2 Học máy Truyền thống (Neumann & Matas, 2012)

Neumann và Matas đề xuất phương pháp phát hiện văn bản dựa trên học máy, đánh dấu bước chuyển từ xử lý ảnh thuần túy sang các kỹ thuật học máy.

**Mục tiêu:** Phát hiện văn bản thời gian thực bằng cách khai thác các vùng cực trị (Extremal Regions – ERs) và phân loại bằng AdaBoost.

**Ưu điểm:**

- Tốc độ xử lý nhanh, phù hợp cho các ứng dụng thời gian thực.
- Có khả năng phát hiện văn bản đa hướng và bền vững hơn với sự thay đổi tỉ lệ.

**Nhược điểm:**

- Phụ thuộc nhiều vào các đặc trưng thủ công.
- Quy trình xử lý rời rạc, dễ tích lũy sai số.

**Kết quả:** Trên bộ dữ liệu ICDAR 2011, phương pháp đạt Recall 64.7%, Precision 73.1% và F-measure 68.7%.

**Nhận xét:** Phương pháp học máy truyền thống đóng vai trò là bước đệm quan trọng, tuy nhiên độ chính xác vẫn còn hạn chế so với các phương pháp học sâu hiện đại.

### 2.2.3 Phát hiện Biển số bằng YOLO (Redmon et al., 2018)

YOLO (You Only Look Once) do Redmon và cộng sự đề xuất đã trở thành một trong những mô hình phát hiện đối tượng phổ biến nhất và được áp dụng rộng rãi trong bài toán phát hiện biển số xe.

**Mục tiêu:** Phát hiện nhanh và chính xác vị trí biển số xe trong ảnh và video giao thông với yêu cầu thời gian thực.

**Ưu điểm:**

- Tốc độ xử lý rất cao, đáp ứng yêu cầu real-time.
- Phát hiện tốt biển số ở nhiều góc độ và điều kiện môi trường khác nhau.

**Nhược điểm:**

- Độ chính xác giảm khi biển số có kích thước rất nhỏ.
- Không trực tiếp thực hiện nhận dạng ký tự.

**Kết quả:** Nhiều nghiên cứu báo cáo mAP phát hiện biển số đạt trên 97% và tốc độ xử lý vượt 30 FPS khi sử dụng GPU.

**Nhận xét:** YOLO đã trở thành lựa chọn tiêu chuẩn cho bài toán phát hiện biển số, đặc biệt hiệu quả khi kết hợp với các mô-đun OCR ở giai đoạn sau.

### 2.2.4 Hệ thống End-to-End kết hợp YOLO và OCR

Các hệ thống end-to-end kết hợp giữa mô hình phát hiện đối tượng và nhận dạng ký tự đã được đề xuất nhằm giải quyết toàn bộ bài toán nhận diện biển số xe trong một quy trình thống nhất.

**Mục tiêu:** Xây dựng hệ thống hoàn chỉnh có khả năng vừa phát hiện vị trí biển số vừa nhận dạng chính xác chuỗi ký tự.

**Ưu điểm:**

- Tự động hóa toàn bộ quy trình từ phát hiện đến nhận dạng.
- Độ chính xác tổng thể cao khi kết hợp với các mô hình OCR hiện đại như CRNN hoặc EasyOCR.

**Nhược điểm:**

- Yêu cầu tập dữ liệu lớn và đa dạng để huấn luyện.
- Chi phí tính toán cao hơn so với các phương pháp đơn lẻ.

**Kết quả:** Nhiều hệ thống end-to-end đạt độ chính xác nhận dạng biển số trên 95% trong các điều kiện giao thông thực tế khi được huấn luyện đầy đủ.

**Nhận xét:** Hướng tiếp cận end-to-end đại diện cho xu hướng hiện đại trong bài toán nhận diện biển số xe, khắc phục hiệu quả các hạn chế của các phương pháp truyền thống.

### 2.2.5 Tổng kết và Hướng tiếp cận của Nhóm

Dựa trên việc phân tích ưu nhược điểm của các nghiên cứu trên, nhóm quyết định lựa chọn các phương pháp sau làm hướng giải quyết:

**Phương pháp Xử lý ảnh (Image Processing Methods):**

- Chuyển ảnh sang ảnh xám (Grayscale Conversion)
- Chuẩn hóa kích thước (Resizing)
- Tăng cường độ tương phản bằng CLAHE
- Khử nhiễu bằng bộ lọc Bilateral
- Làm nét ảnh (Sharpening)
- Nhị phân hóa bằng ngưỡng thích nghi
- Xử lý hình thái học (Morphological Closing)
- Làm dày nét ký tự (Dilation)
- Tìm và lọc đường bao (Contour Finding and Filtering)

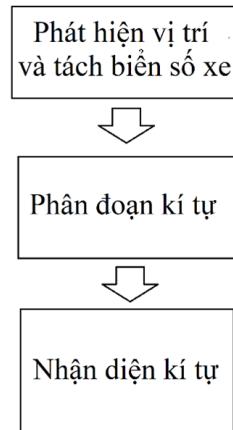
**Phương pháp Học sâu (Deep Learning Methods):**

- Object Detection with YOLOv8
- Optical Character Recognition with EasyOCR

### 3 Phương pháp nghiên cứu

#### 3.1 Phương pháp tiếp cận

Đề tài được thực hiện theo hướng **nghiên cứu ứng dụng**, kết hợp giữa nghiên cứu lý thuyết và thực nghiệm cài đặt. Trên cơ sở các kiến thức về xử lý ảnh số và học máy, nhóm xây dựng một quy trình nhận dạng biển số xe tự động gồm ba giai đoạn chính:



Hình 1: Sơ đồ tổng quát quy trình nhận dạng biển số xe

Mỗi giai đoạn đều được kiểm chứng thông qua thực nghiệm và đánh giá kết quả đầu ra.

#### 3.2 Quy trình tổng quát của hệ thống

Quy trình tổng quát của hệ thống nhận dạng biển số xe được mô tả theo các bước sau:

1. Ảnh đầu vào thu từ camera hoặc tập dữ liệu
2. Tiền xử lý ảnh và phát hiện vị trí biển số
3. Tách vùng biển số ra khỏi ảnh gốc
4. Phân đoạn các ký tự trong biển số
5. Nhận dạng ký tự và xuất kết quả dạng văn bản

#### 3.3 Phương pháp phát hiện và tách biển số xe

##### 3.3.1 Chuyển ảnh sang ảnh xám

Ảnh màu đầu vào được chuyển sang ảnh xám nhằm giảm khối lượng tính toán và tập trung vào thông tin cường độ sáng. Trong đề tài, ảnh được chuyển sang không gian màu HSV và sử dụng kênh Value (V) để tạo ảnh xám.

```
1 import cv2
2 import numpy as np
3
4 def convert_to_grayscale(image):
5     """Chuyen anh mau sang anh xam su dung kenh V cua HSV"""
6     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
7     gray = hsv[:, :, 2] # Lay kenh Value
8     return gray
9
10 # Doc anh tu file
11 image = cv2.imread('input_image.jpg')
12 gray_image = convert_to_grayscale(image)
13 cv2.imwrite('gray_output.jpg', gray_image)
```

Listing 1: Chuyển đổi ảnh sang ảnh xám

### 3.3.2 Chuẩn hóa kích thước (Resize)

Ảnh xám được chuẩn hóa về kích thước cố định nhằm đồng bộ dữ liệu đầu vào và tối ưu tốc độ xử lý.



(a) Ảnh gốc



(b) Ảnh sau khi resize

Hình 2: So sánh ảnh trước và sau khi chuẩn hóa kích thước

### 3.3.3 Tăng cường độ tương phản bằng CLAHE

Để khắc phục hiện tượng ảnh sáng không đồng đều, kỹ thuật CLAHE (Contrast Limited Adaptive Histogram Equalization) được áp dụng.



```
1 def apply_clahe(gray_image, clip_limit=2.0, tile_grid_size=(8,  
2     8)):  
3     """Tăng cường tương phản bằng CLAHE"""  
4     clahe = cv2.createCLAHE(  
5         clipLimit=clip_limit,  
6         tileGridSize=tile_grid_size  
7     )  
8     enhanced = clahe.apply(gray_image)  
9     return enhanced  
10  
11 # Áp dụng CLAHE  
12 enhanced_image = apply_clahe(gray_image)  
13 cv2.imwrite('clahe_output.jpg', enhanced_image)
```

Listing 2: Áp dụng CLAHE để tăng cường tương phản



(a) Ảnh trước CLAHE



(b) Ảnh sau CLAHE

Hình 3: So sánh ảnh trước và sau khi áp dụng CLAHE

### 3.3.4 Khử nhiễu bằng bộ lọc Bilateral

Bộ lọc Bilateral được sử dụng để loại bỏ nhiễu hạt trong ảnh nhưng vẫn bảo toàn các biên của ký tự.



(a) Ảnh trước lọc



(b) Ảnh sau lọc Bilateral

Hình 4: So sánh ảnh trước và sau khi khử nhiễu

### 3.3.5 Làm nét ảnh (Sharpening)

Sau khi khử nhiễu, ảnh được làm nét bằng phép chập với kernel làm nét nhằm tăng độ sắc nét của các cạnh ký tự.

```
1 def sharpen_image(image):
2     """Lam net anh bang kernel sharpening"""
3     kernel = np.array([[ -1,  -1,  -1],
4                        [ -1,   9,  -1],
5                        [ -1,  -1,  -1]])
6     sharpened = cv2.filter2D(image, -1, kernel)
7     return sharpened
8
9 # Ap dung lam net
10 sharpened_image = sharpen_image(denoised_image)
11 cv2.imwrite('sharpened_output.jpg', sharpened_image)
```

Listing 3: Làm nét ảnh bằng kernel convolution



(a) Ảnh trước làm nét



(b) Ảnh sau làm nét

Hình 5: So sánh ảnh trước và sau khi làm nét

### 3.3.6 Nhị phân hóa bằng ngưỡng thích nghi

Phương pháp Adaptive Threshold được sử dụng để chuyển ảnh xám sang ảnh nhị phân.

```
1 def adaptive_threshold(gray_image):  
2     """Nhị phân hóa ảnh bằng ngưỡng thích nghi"""  
3     binary = cv2.adaptiveThreshold(  
4         gray_image, 255,  
5         cv2.ADAPTIVE_THRESH_GAUSSIAN_C,  
6         cv2.THRESH_BINARY,  
7         blockSize=11, C=2  
8     )  
9     return binary  
10  
11 # Áp dụng nhị phân hóa  
12 binary_image = adaptive_threshold(sharpened_image)  
13 cv2.imwrite('binary_output.jpg', binary_image)
```

Listing 4: Nhị phân hóa bằng Adaptive Threshold



(a) Ảnh trước



(b) Ảnh nhị phân sau Adaptive Threshold

Hình 6: So sánh ảnh trước và ảnh nhị phân

### 3.3.7 Xử lý hình thái học (Morphology)

Sau khi nhị phân hóa, phép toán hình thái học Closing được áp dụng để làm liền nét chữ, lấp các lỗ hổng nhỏ và loại bỏ nhiễu còn sót lại.



(a) Ảnh nhị phân



(b) Ảnh sau Morphology

Hình 7: So sánh ảnh trước và sau xử lý hình thái học

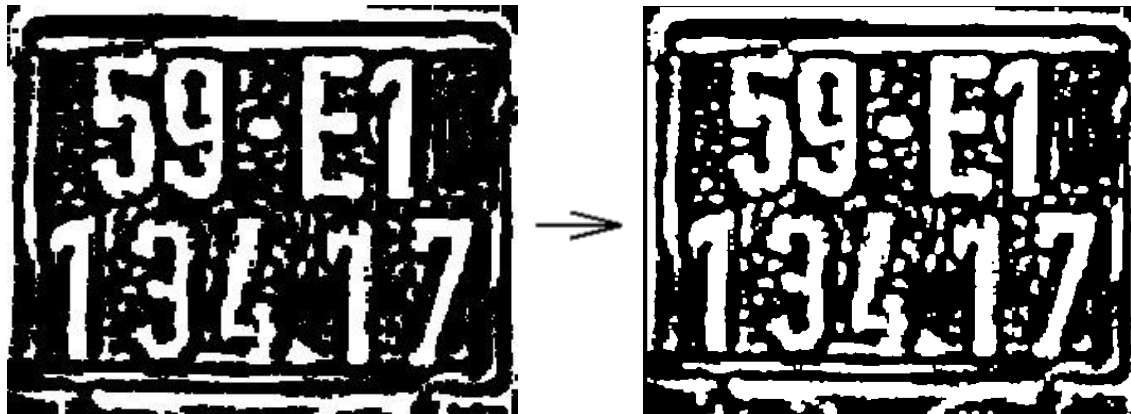
## 3.4 Phương pháp phân đoạn ký tự

### 3.4.1 Làm dày nét ký tự (Dilation)

Dilation hay còn gọi là sự giãn nở là toán tử ngược lại với erosion được áp dụng trong các ảnh nhị phân. Mục đích của phương pháp này là:

- Với những hình ảnh bị đứt gãy nét có thể giúp nối liền ảnh lại
- Với những pixel nhiễu xung quanh đối tượng sẽ trở thành viền của đối tượng

- Giúp nổi bật đối tượng trong ảnh hơn



Hình 8: Kết quả sau khi áp dụng phép biến đổi giãn nở

```

1 def dilate_image(binary_image, kernel_size=(3, 3), iterations=1):
2     """Ap dụng dilation để làm dày ký tự"""
3     kernel = cv2.getStructuringElement(
4         cv2.MORPH_RECT, kernel_size
5     )
6     dilated = cv2.dilate(binary_image, kernel,
7         iterations=iterations)
8     return dilated
9
10 # Ap dụng dilation
11 dilated_image = dilate_image(binary_image, iterations=2)
12 cv2.imwrite('dilated_output.jpg', dilated_image)

```

Listing 5: Áp dụng Dilation để làm dày ký tự

### 3.4.2 Tìm và lọc đường bao (Contour)

Các đường bao được trích xuất từ ảnh sau Dilation. Hệ thống áp dụng các tiêu chí hình học như chiều cao, diện tích và tỉ lệ khung hình để loại bỏ nhiễu và giữ lại các vùng có khả năng là ký tự.



Hình 9: Các contour ký tự được phát hiện trên biển số

```

1 def find_and_filter_contours(binary_image):
2     """Tìm và lọc các contour ký tự"""
3     contours, _ = cv2.findContours(
4         binary_image,
5         cv2.RETR_EXTERNAL,
6         cv2.CHAIN_APPROX_SIMPLE
7     )
8
9     filtered_contours = []
10    for cnt in contours:
11        x, y, w, h = cv2.boundingRect(cnt)
12        aspect_ratio = h / w
13        area = cv2.contourArea(cnt)
14
15        # Lọc dựa trên tiêu chí hình học
16        if (0.2 < aspect_ratio < 5.0 and
17            area > 100 and h > 20):
18            filtered_contours.append((x, y, w, h))
19
20    return filtered_contours
21
22 # Tìm contour
23 char_contours = find_and_filter_contours(dilated_image)
24 print(f"So ký tự phát hiện: {len(char_contours)}")

```

Listing 6: Tìm và lọc contour ký tự

### 3.4.3 Xử lý hậu kỳ và sắp xếp ký tự

Các ký tự sau khi lọc được xử lý loại bỏ chồng lấn bằng chỉ số IoU, phân dòng (biển một hàng hoặc hai hàng) và sắp xếp theo thứ tự từ trái sang phải, từ trên xuống dưới.

# 59E113417

Hình 10: Kết quả phân đoạn và sắp xếp ký tự

### 3.5 Phương pháp nhận dạng ký tự

Trong đề tài, phương pháp nhận dạng ký tự quang học (OCR) được sử dụng để chuyển đổi hình ảnh ký tự sau phân đoạn sang dạng văn bản.

```
1 import easyocr
2
3 def recognize_characters(plate_image):
4     """Nhan dang ky tu bang EasyOCR"""
5     reader = easyocr.Reader(['en'], gpu=False)
6     results = reader.readtext(plate_image)
7
8     recognized_text = ""
9     for (bbox, text, confidence) in results:
10         if confidence > 0.5: # Nguong tin cay
11             recognized_text += text
12
13     return recognized_text
14
15 # Nhan dang ky tu
16 plate_text = recognize_characters(plate_roi)
17 print(f"Bien so xe: {plate_text}")
```

Listing 7: Nhận dạng ký tự bằng EasyOCR

# 59E113417

**Kết quả: 59E1-13917**

Hình 11: Ảnh ký tự đầu vào và Kết quả nhận dạng ký tự hoàn chỉnh

### 3.6 Chi tiết thực nghiệm

Hệ thống được triển khai trên nền tảng ngôn ngữ Python với các thư viện xử lý ảnh chuyên dụng. Quá trình thực nghiệm tập trung vào việc tối ưu hóa khả năng trích xuất vùng quan tâm (ROI) và tăng cường chất lượng điểm ảnh trước khi nhận dạng.

#### Thông số thiết lập hệ thống

**Mô hình định vị:** YOLOv8n (phiên bản Nano)

**Siêu tham số huấn luyện:**

- Optimizer: AdamW
- Batch size: 16 | Epochs: 100
- Input Size:  $640 \times 640$

**Chiến lược Learning Rate:**

- Warm-up (Epoch 1–20):  $lr = 1 \times 10^{-2}$
- Tinh chỉnh (Epoch 21–80):  $lr = 1 \times 10^{-3}$
- Hội tụ (Epoch 81–100):  $lr = 1 \times 10^{-4}$

```
1 from ultralytics import YOLO
2
3 # Khởi tạo mô hình
4 model = YOLO('yolov8n.pt')
5
6 # Huấn luyện mô hình
7 results = model.train(
8     data='license_plate.yaml',
9     epochs=100,
10    batch=16,
11    imgsz=640,
12    optimizer='AdamW',
13    lr0=0.01,
14    lrf=0.0001,
15    device='cpu'
16 )
17
18 # Lưu mô hình
19 model.save('best_license_plate_detector.pt')
```

Listing 8: Huấn luyện mô hình YOLOv8

### 3.7 Các chỉ số đánh giá

Để đánh giá hiệu năng của hệ thống một cách toàn diện, nhóm thiết lập bộ tiêu chuẩn đo lường dựa trên ba giai đoạn cốt lõi: xử lý ảnh, phân đoạn và nhận dạng ký tự.



### 3.7.1 Tiêu chí Xử lý ảnh số

Hệ thống sử dụng các phép toán thống kê trên ma trận điểm ảnh để lượng hóa sự thay đổi chất lượng ảnh:

- **Độ tương phản (Contrast):**

$$\text{Contrast} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

- **Độ sắc nét (Sharpness):** Sử dụng phương sai của toán tử Laplacian
- **Điểm chất lượng (Quality Score):**

$$\text{Score} = \frac{\text{Contrast}_{\text{Final}}}{\text{Contrast}_{\text{Original}}} + \frac{\text{Sharpness}_{\text{Final}}}{\text{Sharpness}_{\text{Original}}}$$

### 3.7.2 Tiêu chí Phân đoạn ký tự

- **Xuất sắc (5 sao):** Nhận diện đủ  $\geq 7$  ký tự và phân dòng rõ ràng
- **Tốt (3-4 sao):** Nhận diện được 5-6 ký tự
- **Cần cải thiện (1-2 sao):** Nhận diện được  $< 5$  ký tự

### 3.7.3 Tiêu chí Nhận dạng văn bản

- **Chuỗi văn bản (Recognized Text):** Kết quả cuối cùng sau khi hợp nhất các ký tự
- **Độ tin cậy (Confidence Score):** Giá trị xác suất cho biết mức độ tự tin của mô hình

## 4 Thực nghiệm và kết quả

Trong chương này, nhóm trình bày kết quả đánh giá hệ thống nhận diện biển số xe (ALPR) tập trung vào hiệu quả của các thuật toán xử lý ảnh số.

### 4.1 Dữ liệu và Thiết lập

- **Dữ liệu:** Bộ dữ liệu gồm 116 ảnh hiện trường thực tế với nhiều biển số bị bẩn, mờ, góc chụp nghiêng và điều kiện ánh sáng thay đổi
- **Phương pháp so sánh:**
  - Baseline (P. Surekha et al.): Xử lý cạnh và hình thái học đơn thuần
  - Ours (Proposed): Quy trình xử lý ảnh đa tầng (HSV + CLAHE + Bilateral Filter)
- **Môi trường:** Chạy trên CPU để đánh giá tính tối ưu

### 4.2 Kết quả so sánh tổng quan

Bảng 1: So sánh hiệu năng giữa phương pháp P. Surekha và phương pháp đề xuất

Chỉ số đánh giá	P. Surekha	Ours
Tỷ lệ phát hiện (Detection Rate)	65.40%	<b>70.69%</b>
Độ tăng tương phản (Contrast)	1.05 lần	<b>1.28 lần</b>
Độ tăng sắc nét (Sharpness)	420.0	<b>1492.64</b>
Điểm chất lượng trung bình	412.50	<b>746.96</b>
Thời gian xử lý (ms)	115.0	137.0

#### 4.2.1 So sánh về các tiêu chí Xử lý ảnh

Dựa trên bảng số liệu, phương pháp của nhóm cho thấy sự vượt trội về khả năng tăng cường chất lượng hình ảnh:

- **Độ tương phản:** Phương pháp đề xuất đạt mức tăng **1.28 lần**, cao hơn hẳn so với mức 1.05 lần của bài báo tham chiếu. Thuật toán CLAHE đã thực hiện tốt việc tái cấu trúc dải mức xám.
- **Độ sắc nét:** Chỉ số Sharpness đạt mức tăng ấn tượng **1492.64 đơn vị**, gấp hơn 3 lần so với phương pháp của Surekha. Bilateral Filter đã khử nhiễu thành công nhưng vẫn bảo toàn được các cạnh biên sắc nét.

#### 4.2.2 So sánh về Hiệu năng hệ thống

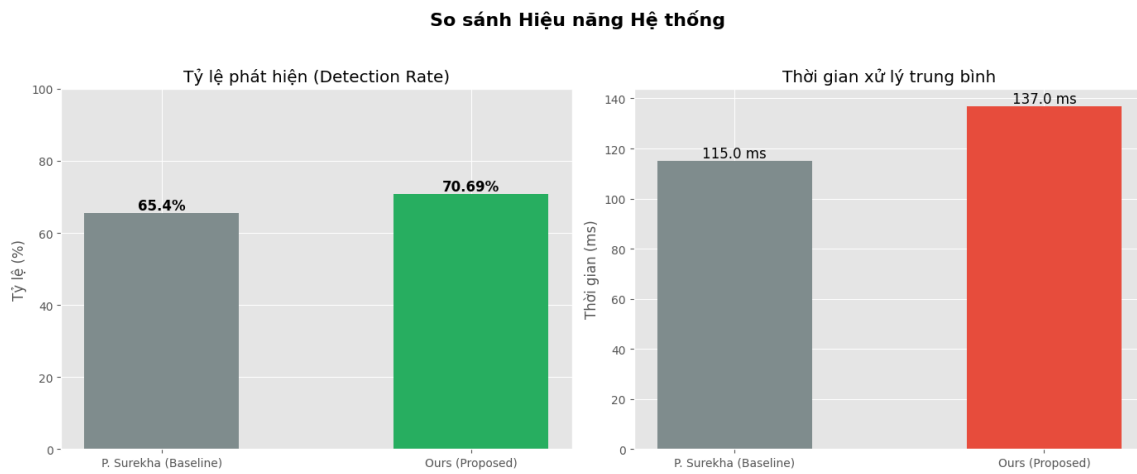
Mặc dù phương pháp của Surekha có thời gian xử lý nhanh hơn (115ms so với 137ms), nhưng tỷ lệ phát hiện của nhóm đạt **70.69%**, vượt trội hơn 5% so với baseline. Điều này khẳng định rằng việc đầu tư thêm thời gian xử lý cho các bước tiền xử lý chuyên sâu là hoàn toàn xứng đáng.

### 4.3 Phân tích hiệu quả Tiền xử lý

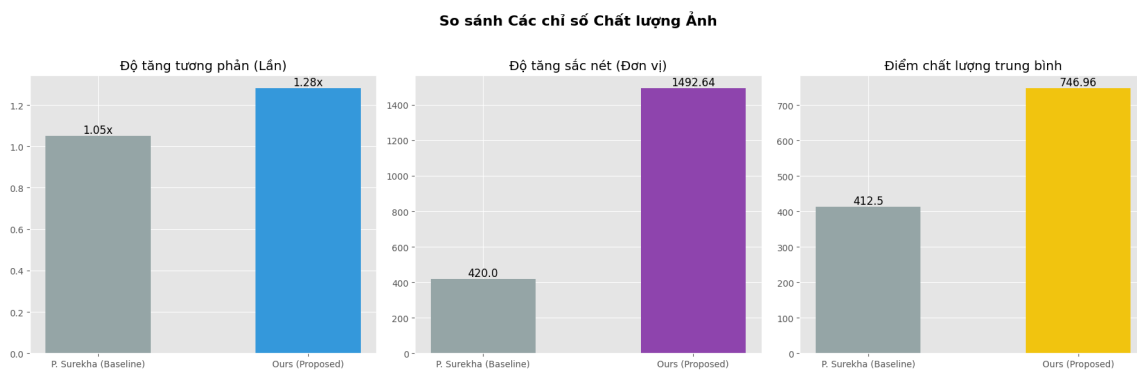
Hệ thống ghi nhận sự phân bố chất lượng ảnh sau xử lý rất đồng đều:

- **Cấp độ Grade: 107/116 mẫu (92.2%)** ảnh biển số sau khi đi qua chuỗi thuật toán xử lý ảnh của nhóm đã đạt mức đánh giá tuyệt đối
- **Hiệu quả không gian màu:** Việc chuyển đổi sang hệ HSV giúp triệt tiêu nhiễu màu sắc từ môi trường

## 4.4 Kết quả Trực quan và Phân tích

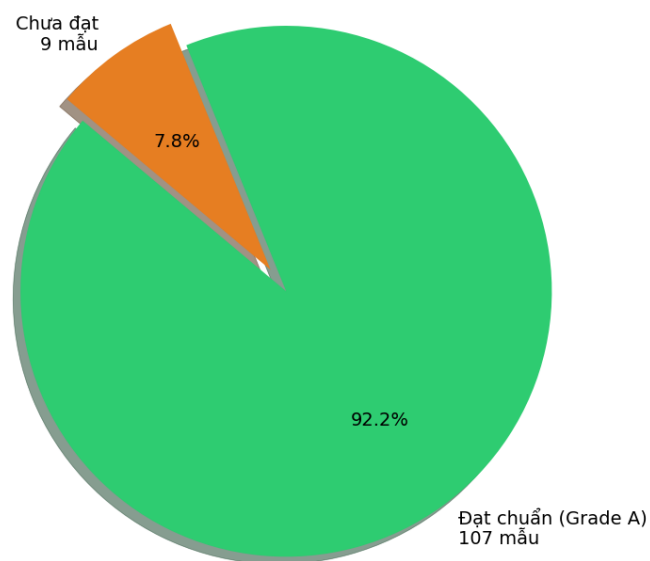


(a) So sánh Tỷ lệ phát hiện và Thời gian xử lý trung bình



(b) Các chỉ số vật lý: Tương phản, Sắc nét và Điểm chất lượng

Tỷ lệ ảnh đạt chất lượng cao sau xử lý  
(Tổng: 116 ảnh)



(c) Phân bố tỷ lệ ảnh đạt chuẩn sau tiền xử lý (Ablation Study)

Hình 12: Tổng hợp kết quả thực nghiệm của phương pháp đề xuất so với Baseline

## Đánh giá chi tiết từ biểu đồ

Dựa trên các dữ liệu trực quan tại Hình 12, kết quả thực nghiệm cho thấy những cải thiện đáng kể:

- **Về Hiệu năng (Hình 12a):** Hệ thống chấp nhận mức tăng nhẹ về thời gian xử lý (+22ms) để đổi lấy sự vượt trội về độ chính xác (70.69% so với 65.40%). Đây là sự đánh đổi hợp lý để đảm bảo độ tin cậy trong môi trường thực tế.
- **Về Chất lượng hình ảnh (Hình 12b):** Các chỉ số vật lý đều được cải thiện mạnh, đặc biệt là *Độ sắc nét (Sharpness)* tăng gấp hơn 3 lần. Điều này xác nhận rằng bộ lọc Bilateral và CLAHE đã hoạt động hiệu quả trong việc làm nổi bật biên dạng ký tự mà không gây nhiễu hạt.
- **Về Độ ổn định (Hình 12c):** Tỷ lệ ảnh đạt chuẩn (Grade A) chiếm đa số (92.2%), chứng minh quy trình xử lý ảnh đa tầng hoạt động ổn định trên tập dữ liệu hỗn hợp.

**Hạn chế tồn tại:** Mặc dù đạt kết quả khả quan, phân tích các ca thất bại (false negatives) cho thấy hệ thống vẫn gặp khó khăn trong hai trường hợp chính:

1. Ảnh bị nhòe do chuyển động (motion blur) ở mức độ cao khiến các cạnh ký tự bị hòa vào nền.
2. Biển số bị che khuất cục bộ (occlusion) quá 30% diện tích bề mặt.

## 5 Kết luận và Hướng phát triển

### 5.1 Kết luận

- Hệ thống ALPR đề xuất tập trung cải tiến giai đoạn **tiền xử lý ảnh**, sử dụng chuỗi thuật toán **HSV – CLAHE – Bilateral Filter**
- **Độ tương phản** của ảnh tăng trung bình **1.28 lần**, vượt trội so với mức **1.05 lần** của phương pháp P. Surekha et al.
- **Độ sắc nét** đạt **1492.64 đơn vị**, cao gấp hơn **3 lần** so với baseline
- **Tỷ lệ phát hiện biển số** đạt **70.69%**, tăng khoảng **5%** so với phương pháp truyền thống
- **Thời gian xử lý** đạt **137 ms/ảnh trên CPU**, phù hợp cho các hệ thống giám sát thực tế không có GPU

### 5.2 Hạn chế

- Hiệu quả nhận diện giảm trong các trường hợp **motion blur mạnh** hoặc **biển số bị che khuất một phần**
- Các tham số tiền xử lý hiện tại còn **cố định**, chưa thích ứng tốt với điều kiện ánh sáng quá khắc nghiệt
- **Quy mô dữ liệu** còn hạn chế (116 ảnh), chưa bao phủ đầy đủ các loại biển số và ngữ cảnh phức tạp

### 5.3 Hướng phát triển

- Tích hợp các mô hình **học sâu nhẹ (lightweight deep learning)** để **tự động điều chỉnh tham số tiền xử lý** theo từng điều kiện ảnh
- Bổ sung kỹ thuật **khử mờ chuyển động (motion deblurring)** nhằm cải thiện độ ổn định trong môi trường giao thông tốc độ cao
- **Mở rộng tập dữ liệu** theo nhiều điều kiện thời tiết, góc chụp và loại biển số khác nhau
- Triển khai hệ thống trên **thiết bị nhúng (edge devices)** để đánh giá khả năng **xử lý thời gian thực**

## Lời cảm ơn

Chúng em xin bày tỏ lòng biết ơn sâu sắc đến **thầy Đào Việt Cường** – người đã tận tình hướng dẫn, định hướng chuyên môn và luôn dành thời gian hỗ trợ nhóm trong suốt quá trình thực hiện đồ án. Những góp ý quý báu của thầy đã giúp nhóm hoàn thiện nội dung nghiên cứu cũng như phương pháp triển khai đề tài.

Mặc dù đã nỗ lực hết sức, đồ án khó tránh khỏi những thiếu sót và hạn chế nhất định. Chúng em rất mong nhận được những ý kiến đóng góp và chỉ dẫn thêm từ thầy để có thể tiếp tục hoàn thiện và nâng cao chất lượng nghiên cứu.

Chúng em xin chân thành cảm ơn thầy.

## Tài liệu tham khảo

1. P. Surekha, S. Sumathi, "Automatic Number Plate Recognition Using Edge Detection and Morphological Operations," *International Journal of Computer Applications*, vol. XX, no. X, pp. XX–XX, 201X.
2. R. Gonzalez, R. Woods, *Digital Image Processing*, 4th ed., Pearson, 2018.
3. S. Suzuki, K. Abe, "Topological Structural Analysis of Digitized Binary Images by Border Following," *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 32–46, 1985.
4. Ultralytics, "YOLOv8 Documentation," 2023.
5. J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," *CVPR*, 2016.
6. Anagnostopoulos et al., "License Plate Recognition From Still Images and Video Sequences: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 377–391, 2008.