



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN
MÔN HỌC: NHẬP MÔN THỊ GIÁC MÁY TÍNH
ĐỀ TÀI: NHẬN DIỆN BIỂN SỐ XE Ở BÃI GIỮ XE

Giảng viên hướng dẫn : TS. Mai Tiến Dũng
Sinh viên thực hiện : Hồ Trung Tín - 21521536
Mạc Hoàng Hà - 21522027
Hứa Bảo Duy - 21521993
Lớp : CS231.012.KHCL

Thành phố Hồ Chí Minh, ngày 10 tháng 01 năm 2024.

Mục lục

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	1
1.1. Lý do chọn đề tài	3
1.2. Giới thiệu bài toán	3
1.3. Quy trình thực hiện.....	4
1.3.1. Phát hiện biển số xe	4
1.3.2. Nhận diện biển số xe.....	4
CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN	5
2.1. Phát hiện biển số xe.....	5
2.2 Nhận diện biển số xe	5
2.2.1. Xoay và cắt biển số	5
2.2.2. Phân đoạn ký tự	6
2.2.3. Nhận diện ký tự.....	7
CHƯƠNG 3. Thực nghiệm	9
3.1. Dataset	9
3.2. Train mô hình với Dataset.....	10
3.2.1. YOLOv8	10
3.2.2. Convolutional Neural Network.....	10
3.2.3. Suport Vector Machine	11
3.2.4. Softmax Regression	11
3.3. Phát hiện biển số xe với YOLO	11
3.4. Nhận diện biển số xe	12
3.4.1. Xoay thẳng và crop ảnh	12
3.4.2. Phân đoạn ký tự.	14
3.4.3. Nhận diện ký tự.....	15
3.5. Độ đo đánh giá	15
3.6. Kết quả đánh giá.....	16
3.7. Hạn chế.....	16
3.8. Hướng phát triển	18
TÀI LIỆU THAM KHẢO	19

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong thời đại hiện nay, xe máy đã trở thành phương tiện giao thông chủ yếu ở Việt Nam, và sự gia tăng đáng kể về số lượng xe máy đã đặt ra nhiều thách thức trong việc quản lý và kiểm soát. Để giải quyết vấn đề này, việc áp dụng công nghệ nhận diện biển số xe tự động đã trở thành một giải pháp hữu ích và được tích hợp trong nhiều lĩnh vực khác nhau như quản lý giao thông, kiểm tra an ninh, thu phí giao thông, bãi giữ xe tự động. Ứng dụng của công nghệ nhận diện biển số xe tự động là không thể phủ nhận trong quản lý giao thông, giám sát an ninh, thu phí giao thông, và quản lý bãi giữ xe tự động.

1.2. Giới thiệu bài toán

Bài toán “Phát hiện và nhận diện biển số xe cho mô hình bãi giữ xe” là quá trình tự động xác định và trích các kí tự trên biển số xe từ hình ảnh hoặc video trong bãi giữ xe. Sau đây là mô tả Input, Output của bài toán.

Input:

- Hình ảnh có chứa biển số xe máy.
- Tập ảnh và nhãn có chứa biển số xe của xe máy. Trong đó:
 - Ảnh là hình ảnh có chứa biển số xe của xe máy được chụp bởi camera giám sát tại bãi giữ xe.
 - Nhãn gồm tọa độ bounding box của biển số xe và các kí tự nhận diện được từ biển số xe đó.

Output: Những kí tự đã nhận diện được từ biển số xe đã phát hiện.

Constraints:

- Hình ảnh chỉ có 1 biển số xe.
- Góc chụp chính diện, được chụp từ trên cao xuống.
- Xe không được quá gần hoặc quá xa camera cũng như biển số xe phải hoàn toàn nằm trong ảnh.

Input

- File ảnh: 2_59L206377.jpg



Output



59L206377

- File nhãn Bounding Box của ảnh:
2_59L206377.txt chứa:

```
0
0.4904661016949153
0.19306930693069307
0.17584745762711865
0.23432343234323433
```

Trong đó tên file nhãn sẽ được gán theo những kí tự của biển số đó. Bên trong file sẽ chứa class của object (luôn là 0 vì chỉ có 1 lớp để phát hiện đối tượng) cùng với tọa độ bounding box của biển số xe đó.

1.3. Quy trình thực hiện

1.3.1. Phát hiện biển số xe

Bài toán phát hiện biển số xe thuộc bài toán phát hiện đối tượng (Object Detection) nhằm mục đích xác định ra được vùng chứa biển số xe trong ảnh.

1.3.2. Nhận diện biển số xe

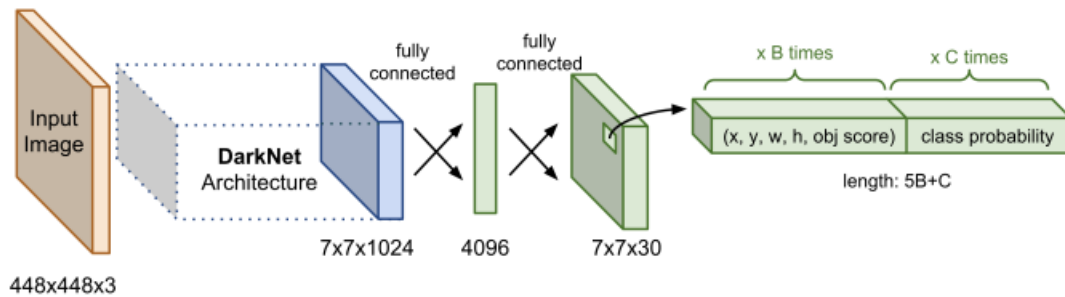
Bài toán nhận diện biển số là phần tiếp theo của bài toán phát hiện biển số xe. Biển số xe sau khi được phát hiện sẽ đưa qua mô hình để đưa ra kết quả cuối cùng là các kí tự trên biển số xe đó.

CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN

2.1. Phát hiện biển số xe.

❖ YOLOv8

YOLO (You Only Look Once) là mô hình phát hiện đối tượng phổ biến được biết đến với tốc độ nhanh và độ chính xác cao. Mô hình này lần đầu tiên được giới thiệu bởi Joseph Redmon và cộng sự vào năm 2016.



Kiến trúc mạng YOLO

Nhóm em sử dụng phiên bản YOLOv8 được ra mắt vào năm 2023 cho bài toán phát hiện biển số xe. YOLOv8 cũng có nhiều phiên bản với số lượng tham số và độ chính xác khác nhau.

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

2.2 Nhận diện biển số xe

2.2.1. Xoay và cắt biển số

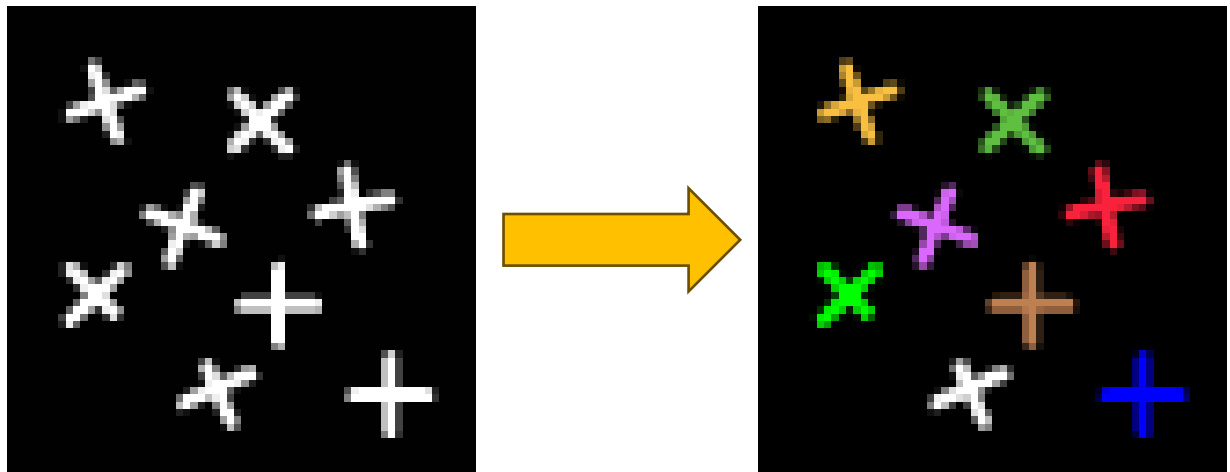
❖ HoughTransform

Biến đổi Hough(HT) (Hough, 1962) là một kỹ thuật trích xuất đặc trưng của một hình dáng cụ thể trong một ảnh, nó được sử dụng trong phân tích hình ảnh, thị giác máy tính và xử lý hình ảnh kỹ thuật số. Bên cạnh đó, thuật toán này hoạt động rất tốt trong việc phát hiện đường thẳng. Ý tưởng chung của việc phát hiện đường thẳng trong thuật toán này là tạo ánh xạ từ không gian ảnh (A) sang một không gian mới (B) mà mỗi đường thẳng trong không gian (A) sẽ ứng với một điểm trong không gian (B).

2.2.2. Phân đoạn ký tự

Connected Components Labelling (CCL)

Gán nhãn thành phần liên thông (CCL) là một ứng dụng thuật toán của lý thuyết đồ thị, nơi các tập con của thành phần liên thông được gán nhãn một cách duy nhất dựa trên một tiêu chí cho trước. Gán nhãn thành phần liên thông được sử dụng trong thị giác máy tính để phát hiện các vùng liên thông trong ảnh nhị phân, mặc dù hình ảnh màu và dữ liệu có số chiều cao hơn cũng có thể được xử lý. Khi được tích hợp vào hệ thống nhận dạng hình ảnh hoặc giao thông tương tác người-máy, gán nhãn thành phần liên thông có thể hoạt động trên nhiều loại thông tin. Việc trích xuất blob thường được thực hiện trên hình ảnh nhị phân kết quả từ phép lấy ngưỡng, nhưng nó cũng có thể áp dụng cho ảnh xám và ảnh màu. Các blob có thể được đếm, lọc và theo dõi.



2.2.3. Nhận diện kí tự

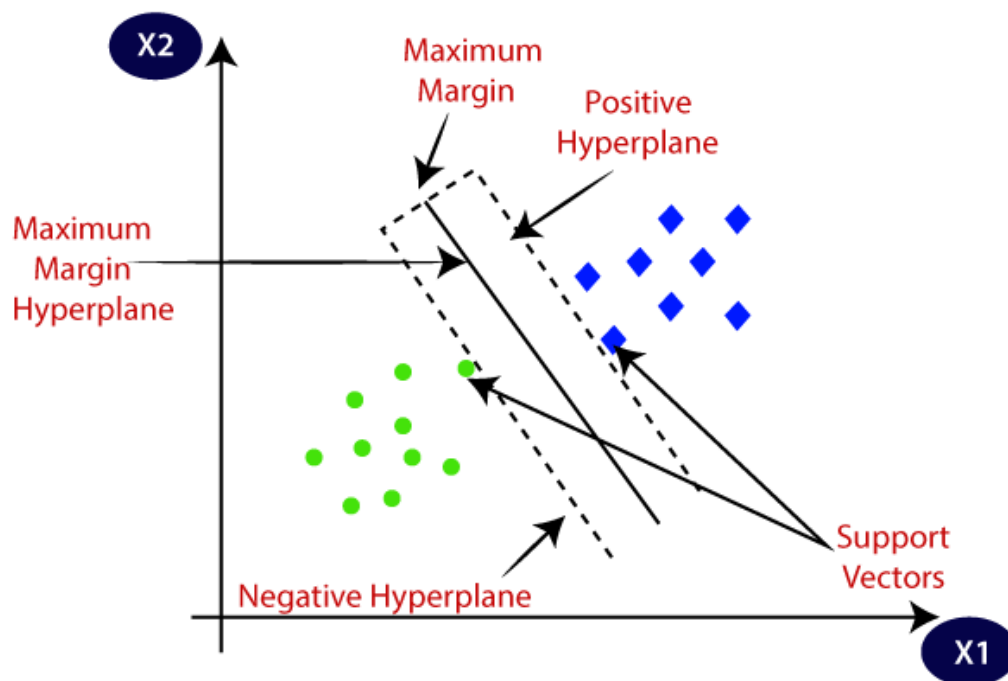
2.2.3.1. Thuật toán Support Vector Machine (SVM)

Thuật toán SVM bắt nguồn từ những năm 1960, khi Vladimir N. Vapnik, một nhà khoa học người Nga. Về mặt toán học, SVM tìm ra một siêu phẳng trong không gian N chiều, chia dữ liệu thành hai phần tương ứng với lớp của chúng. Siêu phẳng này phải có lề (margin) cực đại và cách đều chúng.

SVM có thể được sử dụng cho cả phân loại nhị phân và phân loại đa lớp. Trong phân loại nhị phân, SVM tìm ra một siêu phẳng chia dữ liệu thành hai phần, mỗi phần tương ứng với một lớp. Trong phân loại đa lớp, SVM sử dụng một tập các siêu phẳng để chia dữ liệu thành nhiều lớp.

SVM có thể áp dụng cho nhiều loại bài toán khác nhau, bao gồm:

- Phân loại email rác
- Phân loại hình ảnh
- Chuẩn đoán y tế



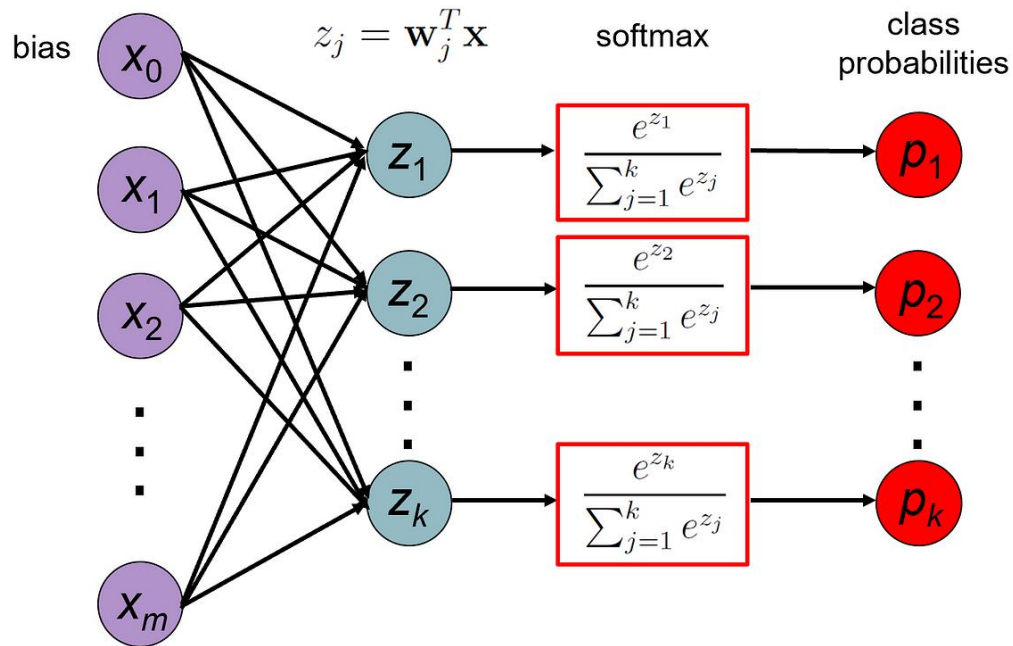
2.2.3.2. Thuật toán Softmax Regression

Softmax regression là một thuật toán học máy được sử dụng để phân loại dữ liệu thành các lớp rời rạc. Softmax regression là một mô hình tuyến tính, nghĩa là đầu ra của mô hình là một hàm tuyến tính của đầu vào.

Ý tưởng cơ bản của softmax regression là sử dụng hàm softmax để ánh xạ đầu ra của mô hình thành các xác suất. Hàm softmax sẽ tính toán xác suất của mỗi lớp, với tổng xác suất bằng 1. Sau đó, lớp có xác suất cao nhất sẽ được chọn là lớp dự đoán.

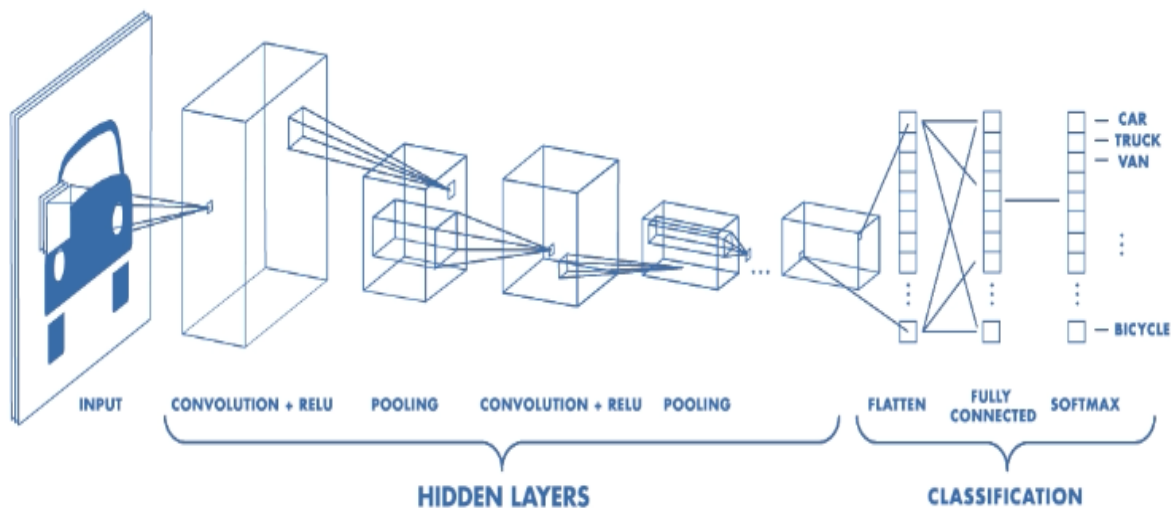
Softmax regression được sử dụng rộng rãi trong các bài toán phân loại dữ liệu, chẳng hạn như:

- Phân loại hình ảnh
- Phân loại văn bản
- Phân loại email
- Phân loại tiếng nói



2.2.2.3 Convolutional Neural Network (CNN)

Mạng nơ-ron tích chập, còn được gọi là ConvNets, được giới thiệu lần đầu vào những năm 1980 bởi Yann LeCun, một nhà nghiên cứu sau tiến sĩ khoa học máy tính. Kiến trúc CNN đặc biệt hữu ích cho nhận dạng hình ảnh và phân loại hình ảnh, cũng như các tác vụ thị giác máy tính khác vì chúng có thể xử lý lượng lớn dữ liệu và tạo ra dự đoán chính xác cao. CNNs có thể học các đặc trưng của một đối tượng thông qua nhiều lần lặp, loại bỏ nhu cầu cho các tác vụ kỹ thuật lấy đặc trưng thủ công.



CHƯƠNG 3. Thực nghiệm

3.1. Dataset

❖ Dataset dùng để train YOLOv8

Dữ liệu được dùng để train mô hình YOLOv8 đã được gán nhãn từ RoboFlow:

- Training set: Bộ dữ liệu huấn luyện có 1244 ảnh, chiếm 70% tổng số ảnh trong dataset.
- Validation set: Bộ dữ liệu đánh giá có 175, chiếm 10% tổng số ảnh trong dataset.
- Testing set: Bộ dữ liệu kiểm tra có 349 ảnh, chiếm 20% tổng số ảnh trong dataset.
- Link dataset: <https://app.roboflow.com/hamac03/-license-plate-detection/1>

❖ Dataset dùng để train các mô hình nhận diện ký tự

Dữ liệu được dùng để train mô hình YOLOv8 đã được tách trực tiếp từ Connected Components Labelling gồm 3722 ảnh có tổng cộng 31 ký tự.

- Training set: Bộ dữ liệu huấn luyện có 3347 ảnh, chiếm 90% tổng số ảnh trong dataset.
- Testing set: Bộ dữ liệu kiểm tra có 375 ảnh, chiếm 10% tổng số ảnh trong dataset.

Link dataset: https://drive.google.com/drive/folders/1nJ2ceEj31-JlD-Dp9udkc-02X3GY_lzR?fbclid=IwAR2tpfgZfaWMo7-4EtH2QrFnspsXg6Sxzquayzo4AmZ3Hxj7XkYtAeXdRwOY

3.2. Train mô hình với Dataset

3.2.1. YOLOv8

```
from ultralytics import YOLO
!yolo task=detect mode=train model=yolov8n.pt
data={'/content/gdrive/MyDrive/cv/bien_so_xe'}/data.yaml
epochs=50 batch=32 imgsz=640
```

Nhóm em sử dụng YOLOv8n để thực hiện train với mục đích tiết kiệm chi phí huấn luyện cũng như thời gian thực thi nhanh.

Nhóm em sử dụng các tham số như sau:

- task=detect: YOLO sẽ được dùng cho bài toán object detection, output sẽ gồm các bounding box và nhãn tương ứng với bounding box đó
- mode=train: dùng để train mô hình.
- model=yolov8n: sử dụng phiên bản YOLOv8n để tiết kiệm chi phí tính toán.
- data={'/content/gdrive/MyDrive/cv/bien_so_xe'}/data.yaml: file “data.yaml” chứa các thông tin về các lớp, nhãn, và thông tin đường dẫn đến file train, validation phục vụ cho việc huấn luyện.
- epochs = 50: chọn epoch bằng 50.
- imgsz = 640: ảnh sẽ được resize thành ảnh 640 x 640 khi đưa vào train.

3.2.2. Convolutional Neural Network

Ảnh kí tự sẽ được resize về kích thước 32x32 khi đưa vào mô hình để train. Output sẽ là 1 trong 31 kí tự.

- Kiến trúc của mô hình CNN:

conv2d (Conv2D)	(None, 32, 32, 32)
conv2d_1 (Conv2D)	(None, 32, 32, 32)
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)
conv2d_2 (Conv2D)	(None, 16, 16, 64)
conv2d_3 (Conv2D)	(None, 16, 16, 64)
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)
conv2d_4 (Conv2D)	(None, 8, 8, 64)
conv2d_5 (Conv2D)	(None, 8, 8, 64)
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 64)
flatten (Flatten)	(None, 1024)
dense (Dense)	(None, 512)
dense_1 (Dense)	(None, 31)

- Framework mà nhóm sử dụng là Tensorflow.
- Optimizer: Adam
- Epoch: 100
- EarlyStop ở epoch: 14
- Best epoch: 4
- Batchsize: 32

3.2.3. Suport Vector Machine

Ảnh của từng kí tự sẽ được resize về tỉ lệ 20x20 trước khi đưa vào mô hình
Output là 1 trong 31 kí tự.

- Kernel: “**Linear**”
- Framework được sử dụng là Sklearn

3.2.4. Softmax Regression

Ảnh của từng kí tự sẽ được resize về tỉ lệ 20x20 trước khi đưa vào mô hình
Output là 1 trong 31 kí tự.

- Max Iteration: 250
- Multi class: “**multinomial**”
- Framework được sử dụng là Sklearn

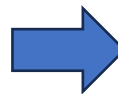
3.3. Phát hiện biển số xe với YOLO

```
model = YOLO ('/content/drive/MyDrive/runs/detect/train/weights/best.pt')
res = model.predict(img)
```

Nhóm em tiến hành load model từ bộ trọng số có hàm loss thấp nhất và truyền ảnh vào hàm `model.predict`

```
df = res.pandas().xyxy[0]
LpRegion = 0
for obj in df.iloc:
    xmin = float(obj['xmin'])
    xmax = float(obj['xmax'])
    ymin = float(obj['ymin'])
    ymax = float(obj['ymax'])
    coord = np.array([[xmin, ymin], [xmax, ymin], [xmin, ymax], [xmax, ymax]])
    LpRegion = perspective.four_point_transform(img, coord)
cropped_LP_copy = LpRegion
```

Sau đó nhóm em sẽ lấy ra bounding box biển số xe có confidence cao nhất thông qua các tọa độ của bounding box trả về.



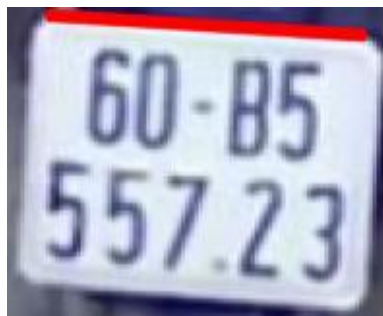
3.4. Nhận diện biển số xe

3.4.1. Xoay thẳng và crop ảnh

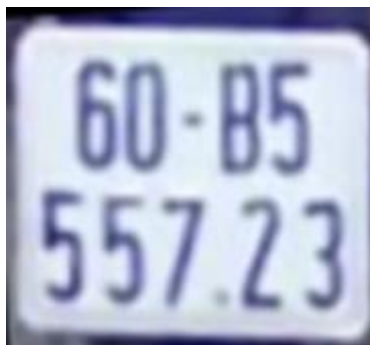
Đầu tiên, ảnh biển số sau khi đã được phát hiện từ mô hình YOLOV8 sẽ được đưa về ảnh xám bằng `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)` sau đó ảnh sẽ được đưa qua bộ lọc nhiễu 2 tầng `cv2.bilateralFilter(res,10,150,130)` để lọc nhiễu có trong ảnh. Tiếp theo, dùng phương pháp Otsu `cv2.threshold(res,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)` để thực hiện việc lấy ngưỡng cho ảnh và sử dụng thuật toán Canny để tìm cạnh của biển số có trong ảnh (`cv2.Canny(res,250,255)`):



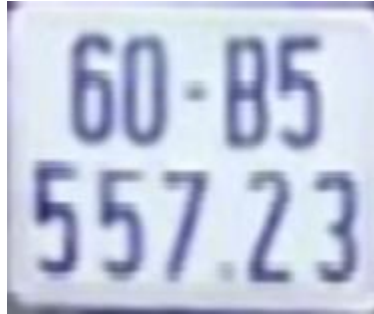
Để có thể thực hiện được việc xoay ảnh, nhóm em sử dụng thuật toán Hough transform với mục đích là phát hiện ra được các đường thẳng **cv2.HoughLinesP(temp,1,np.pi/180,50,Nonzer, 50, 1)**. Thuật toán này sẽ trả về rất nhiều cạnh nhưng nhóm chỉ chọn cạnh lớn nhất để sử dụng cho việc xoay ảnh. Ngoài ra, những cạnh nào trong danh sách lọc nhỏ hơn 65% hoặc lớn hơn 90% theo chiều lớn nhất của ảnh sẽ bị loại bỏ ($d \leq 0.65 * \max(h,w)$ or $d \geq 0.9 * \max(h,w)$):



Sau khi đã chọn được cạnh lớn nhất, cần phải tính góc để thực hiện phép xoay dựa vào tọa độ của điểm đầu và điểm cuối của cạnh vừa tìm được ($\text{math.atan}(\text{doi} / \text{ke}) * (180.0 / \text{math.pi})$) và thực hiện phép xoay:



Để thực hiện việc cắt ảnh nhằm mục đích loại bỏ phần bên ngoài cũng như viền của biển số, ảnh sau khi thực hiện phép xoay sẽ được chuyển về ảnh xám và lấy ngưỡng . Sau đó, tìm contour có diện tích lớn nhất có trong ảnh và lấy tọa độ x,y ở góc trái trên cùng và chiều rộng, chiều dài của bounding box nhỏ nhất bao quanh contour vừa tìm được để thực hiện việc cắt ảnh:



3.4.2. Phân đoạn kí tự.

Phân đoạn kí tự có trong ảnh sẽ được thực hiện bằng thuật toán Connected Components Labelling (CCL) để tách các kí tự có trong ảnh biển số. Đầu tiên, ảnh cần được chuyển về ảnh xám và thực hiện phép lấy ngưỡng để chuyển về ảnh nhị phân. Để thuật toán CCL có thể tách các kí tự chính xác hơn, nhóm có sử dụng bộ lọc nhiễu median có kernel size = 15x15 (`cv2.medianBlur(thresh,15)`):



Sau đó, sử dụng thuật toán CCL do thư viện OpenCV hỗ trợ (`cv2.connectedComponents(thresh)`). Để loại các thành phần có thể không là kí tự, nhóm có sử dụng các ngưỡng theo hoành độ, tung độ, chiều rộng và chiều cao của các bounding box của các thành phần liên thông vừa tìm được (`x >= 3 and x <= 170 and h >= 20 and h <= 55 and y >= 2 and h <= w`):





3.4.3. Nhận diện kí tự.

3.4.3.1. Convolutional Neural Network

Model CNN sau khi khởi tạo sẽ được load từ weights đã được huấn luyện bằng lệnh **model.load_weights('CNN.h5')**. Sau đó, mô hình sẽ nhận diện từng kí tự bằng lệnh **model.predict(character)**

3.4.3.2. Suport Vector Machine và Softmax Regression

Sau khi đã được huấn luyện, mô hình sẽ được lưu vào 2 file “svm.yml” và “softmax.yml” bằng đoạn mã **pickle.dump()**. Sau đó, mô hình sẽ được tái sử dụng sử dụng bằng lệnh **model1 = pickle.load(“svm.yml”)** và **model2 = pickle.load(“softmax.yml”)**. Cuối cùng là dùng lệnh **model1.predict()** và **model2.predict()** để dự đoán kí tự.

3.5. Độ đo đánh giá

3.5.1. YOLOv8:

Các độ đo sử dụng:

- Precision
- Recall
- F1-Score
- mAP@.5
- mAP@[.5:.95]

3.5.2. Nhận diện kí tự

Các độ đo sử dụng:

- Accuracy
- Precision
- Recall
- F1-Score

3.5.3. Toàn bộ mô hình (YOLOv8 + Các mô hình nhận diện kí tự):

Độ đo sử dụng: Accuracy

$$Accuracy = \frac{\text{Tổng số biển số nhận diện chính xác}}{\text{Tổng số biển số}}$$

Trong đó: Một biển số nhận diện chính xác là biển số nhận diện chính xác tất cả các kí tự đã được gán nhãn cho biển số đó.

3.6. Kết quả đánh giá

Model	Precision	Recall	F1-Score	mAP@0.5	mAP@0.5:0.95
YOLOv8	99.98%	100%	100%	99.5%	94.1%

Kết quả đánh giá YOLOv8

Model	Accuracy	Precision	Recall	F1-Score
CNN	91.73%	93.08%	92.40%	91.75%
Softmax Regression	91.64%	93.08%	92.18%	92.03%
SVM	93.86%	94.96%	94.19%	94.09%

Kết quả đánh giá các mô hình nhận diện

Model	YOLO + CNN	YOLO + SVM	YOLO + Softmax Regression
Accuracy	72.99%	68.39%	65.8%

Kết quả đánh giá toàn bộ mô hình

3.7. Hạn chế

❖ Bị ảnh hưởng bởi ánh sáng:

Đối với những ảnh biển số bị loé sáng, khi dùng CCL để tách kí tự sẽ làm cho kí tự bị tách ra, hoặc bị che khuất khiến kí tự đó bị loại bỏ hoặc khi đưa vào mô hình nhận diện kí tự sẽ bị nhầm thành số khác.



Ảnh gốc	CNN	SVM	SOFTMAX
59D100444	59D1UL444	59D1UL444	E9D1UL444

Trong trường hợp trên ảnh, ánh sáng cho số “00” khi dùng CCL bị mất đi một phần khiến kí tự bị nhận diện thành “UL”.

❖ Kí tự bị dính vào biển số xe:

Đối với những kí tự bị dính vào viền biển số xe, khi dùng CCL để tách kí tự, nó sẽ bỏ qua kí tự đó do kí tự đó có kích thước không phù hợp với quy tắc của kí tự nhóm đã đặt ra từ trước.



Ảnh gốc	CNN	SVM	SOFTMAX
5F1126626	5126626	5126626	5126626

Trong trường hợp này, kí tự bị dính vào biển số xe, khiến số “9” và chữ “F” khi qua CCL sẽ bị loại bỏ dẫn đến biển số bị nhận diện thiếu.

❖ Kí tự bị nhầm lẫn:

Kí tự số “8” và chữ “B” đôi khi còn bị nhầm lẫn.



Ảnh gốc	CNN	SVM	SOFTMAX
638963040	638963040	63B963040	63B963040

❖ Có vật cản trong biển số:

Trong trường hợp này Kí tự số “5” khi có cây đinh, sẽ có hình dạng tương tự số “6” khiến mô hình còn bị nhầm lẫn.



Ảnh gốc	CNN	SVM	SOFTMAX
53X66011	63X66011	83X66011	53X66011

3.8. Hướng phát triển

- Có thể phát hiện và nhận diện các loại xe khác.
- Phát triển mô hình có thể hoạt động tốt hơn trên các tác nhân của môi trường xung quanh.
- Tìm cách phân đoạn kí tự tốt hơn.
- Có thể triển khai thành phần mềm để sử dụng.

TÀI LIỆU THAM KHẢO

- <https://arxiv.org/pdf/2304.00501.pdf>
- <https://github.com/ultralytics/ultralytics>
- <https://aicurious.io/blog/2019-10-24-hough-transform-phat-hien-duong-thang>
- <https://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm#:~:text=Connected%20component%20labeling%20works%20by,set%20of%20intensity%20values%20V.>
- <https://learnopencv.com/implementing-cnn-tensorflow-keras/>