

NHẬN DIỆN BIỂN BÁO GIAO THÔNG

Trần Nguyễn Yến Nhi¹, Trịnh Tuấn Tú², Nguyễn Việt Quang³

^{1,2,3} Khoa Khoa học và Kỹ thuật thông tin, Trường Đại học Công nghệ Thông tin (UIT) - ĐHQG TPHCM, Việt Nam

*Correspondence: 21522429@gm.uit.edu.vn¹, 21522747@gm.uit.edu.vn², 21522515@gm.uit.edu.vn³

Abstract— Trên các tuyến đường bộ tại Đức, biển báo đóng vai trò rất quan trọng trong việc cung cấp cho người lái xe những thông tin quan trọng, giúp việc điều khiển phương tiện trở nên an toàn hơn. Tuy nhiên, ở Đức, biển báo cũng là trở ngại đối với người tham gia giao thông vì việc chú tâm lái xe khiến họ không để ý hoặc đôi khi không hiểu ý nghĩa của biển báo đó. Chính vì thế, chúng tôi đã nghiên cứu và đề xuất 3 phương pháp máy học nhận diện biển báo giao thông theo hai bước chính. Đó là chuyển đổi hình ảnh sang kênh màu RGB và nhận diện biển báo. Trong bài báo này, nhóm chúng tôi đã sử dụng các mô hình học máy Convolutional neural network (CNN), thuật toán Random Forest và Support Vector Machine (SVM).

Keywords: Object Detection, CNN, Traffic Sign, SVM

I. GIỚI THIỆU

Hiện nay, tình trạng vi phạm tham gia giao thông ngày càng phổ biến, thường diễn ra ở nhiều khu vực đông dân cư. Đặc biệt, tại Đức - quốc gia được mệnh danh có mức phạt giao thông nghiêm khắc nhất, trên các tuyến đường lớn, người điều khiển phương tiện tham gia giao thông vi phạm biển báo, biển cấm vẫn còn tồn tại nhiều. Điều này đang trở thành một vấn đề có thể gây nguy hiểm cho những người xung quanh. Để hỗ trợ cho các camera hình ảnh trên đường phố cũng như các ảnh chụp từ điện thoại ở Đức, nhóm chúng tôi đề xuất mô hình máy học nhận diện biển báo giao thông trên đường.

A. Chủ đề bài toán

Bài toán nhận diện biển báo giao thông.

- E: Gồm 3 tập dữ liệu: tập train, tập test và tập valid.
- P: Nhận diện biển báo và cho biết tên biển báo.
- Input: Ảnh của một biển báo bất kỳ.
- Output: Tên và hình ảnh của biển báo.

B. Mục tiêu:

Xây dựng được mô hình nhận diện biển báo giao thông có độ chính xác đạt trên 80%.

C. Ứng dụng:

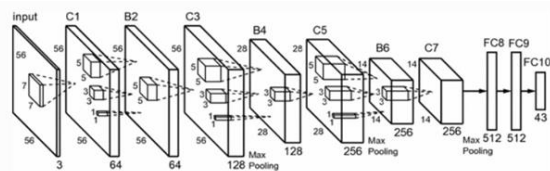
Ứng dụng của chúng tôi: Khi người dùng đưa một bức ảnh biển báo vào hệ thống để nhận diện biển báo thì hệ thống sẽ trả về nhãn và hình ảnh của biển báo.

D. Các nghiên cứu liên quan:

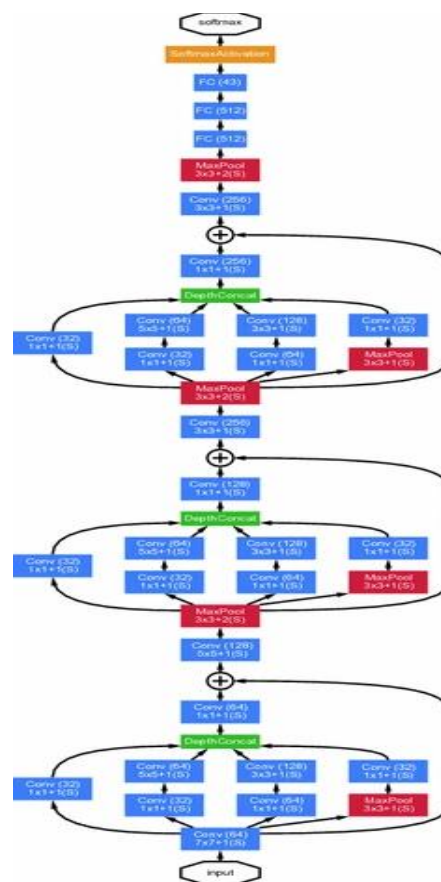
Trong các mô hình CNNs truyền thống, các layers thường được cấu trúc bằng các convolutional layers xếp chồng lên nhau và một số fully-connected layers, chẳng hạn như là LeNet-5 [1], AlexNet [2] và VGG [3].

Lin và các đồng đội của mình [4] đã đưa ra đề xuất sử dụng một mạng nơ-ron sâu có tên là Network In Network (NIN). Họ đã xây dựng micro neural networks với các cấu trúc phức tạp hơn để trừu tượng hóa dữ liệu trong trường tiếp nhận.

Kết hợp với NIN, Yin và các đồng đội [5] đã đề xuất một phương pháp mới cho hệ thống nhận diện biển báo giao thông dựa trên mạng nơ-ron tích chập sâu (deep convolutional neural network) có tên là dubbed block-layer below. Cấu trúc hệ thống của họ có 10 lớp với các tham số. 7 layers đầu xen kẽ các convolutional layers và block-layers, 3 layers còn lại là fully-connected layers. Tổng quan và chi tiết hệ thống nghiên cứu được thể hiện dưới hình 1 và 2.



Hình 1: Tổng quan kiến trúc nghiên cứu



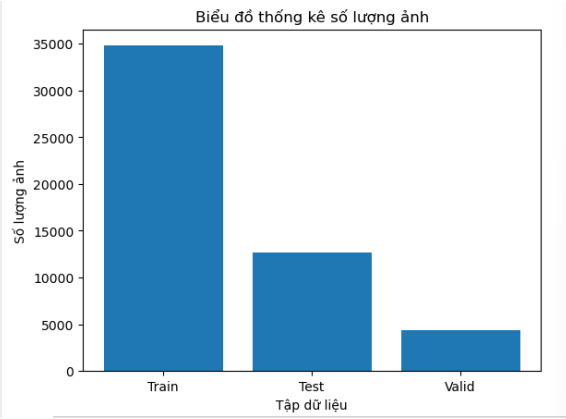
Hình 2: Mô tả chi tiết hệ thống nghiên cứu

II. DỮ LIỆU

A. Tổng quan về bộ dữ liệu

Bộ dữ liệu được nhóm chúng tôi sử dụng là bộ dữ liệu được lấy từ GitHub của tác giả *thibo73800*. Tác giả đã thu thập bộ dữ liệu với các thông số như sau:

- Bộ dữ liệu được sử dụng là bộ dữ liệu hình ảnh biển báo giao thông.
- Bộ dữ liệu được chia thành 3 tập: train, test và valid.
- Số lượng mẫu dữ liệu huấn luyện: 34799 hình ảnh.
- Số lượng mẫu dữ liệu test: 12630 hình ảnh.
- Số lượng mẫu dữ liệu valid: 4410 hình ảnh.
- Kích thước mỗi mẫu dữ liệu là 32x32 với số kênh màu là 3.



Hình 3: Biểu đồ thống kê số lượng ảnh chia theo tập dữ liệu.

B. Các thuộc tính có trong bộ dữ liệu

Dataset có 4 thuộc tính gồm:

- 'features' là mảng 4D chứa dữ liệu pixel thô của hình ảnh biển báo giao thông.
- 'labels' là mảng 1D chứa nhãn của biển báo giao thông.
- 'sizes' là danh sách chứa kích thước ảnh bao gồm chiều rộng, chiều cao và số kênh màu.
- 'coords' là biểu thị tọa độ của hộp giới hạn xung quanh dấu hiệu trong hình ảnh.

C. Thông tin về nhãn của bộ dữ liệu

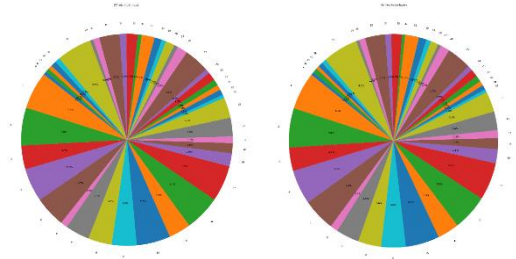
Bộ dữ liệu bao gồm 43 nhãn. Bảng 1 mô tả tên các nhãn ở cột SignName và liệt kê theo thứ tự từ 0 đến 42.

Bảng 1: Tên nhãn được gán bằng các số từ 0 đến 42

ClassId	SignName
0	Speed limit (20km/h)
1	Speed limit (30km/h)
2	Speed limit (50km/h)
3	Speed limit (60km/h)
4	Speed limit (70km/h)
5	Speed limit (80km/h)
6	End of speed limit (80km/h)
7	Speed limit (100km/h)
8	Speed limit (120km/h)

9	No passing
10	No passing for vehicles over 3.5 metric tons
11	Right-of-way at the next intersection
12	Priority road
13	Yield
14	Stop
15	No vehicles
16	Vehicles over 3.5 metric tons prohibited
17	No entry
18	General caution
19	Dangerous curve to the left
20	Dangerous curve to the right
21	Double curve
22	Bumpy road
23	Slippery road
24	Road narrows on the right
25	Road work
26	Traffic signals
27	Pedestrians
28	Children crossing
29	Bicycles crossing
30	Beware of ice/snow
31	Wild animals crossing
32	End of all speed and passing limits
33	Turn right ahead
34	Turn left ahead
35	Ahead only
36	Go straight or right
37	Go straight or left
38	Keep right
39	Keep left
40	Roundabout mandatory
41	End of no passing
42	End of no passing by vehicles over 3.5 metric tons

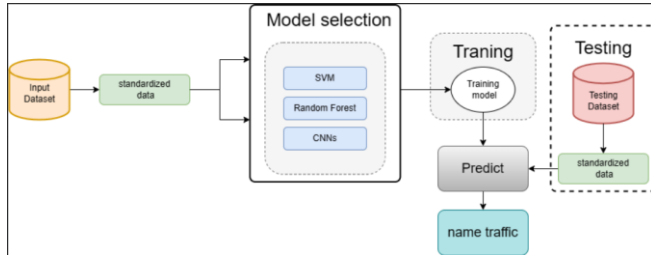
Theo thống kê của nhóm, nhãn biển báo chiếm tỉ lệ nhiều nhất trong bộ dataset là biển 2 (Speed limit (50km/h)) với tỉ lệ 5.9% và thấp nhất là 0.5% với 4 nhãn biển 0 (Speed limit (20km/h)), 19 (Dangerous curve to the left), 37 (Go straight or left), 42 (End of no passing by vehicles over 3.5 metric tons).



Hình 4: Biểu đồ tròn thể hiện tỉ lệ từng nhãn của tập dữ liệu train và test

III. PHƯƠNG PHÁP MÁY HỌC

Trong phần này, chúng tôi trình bày hệ thống nhận diện biển báo giao thông dựa trên ba phương pháp là SVM, Random Forest và CNNs. Hình 5 cung cấp kiến trúc tổng quan hệ thống nhận diện từ giai đoạn đầu vào đến giai đoạn đầu ra.

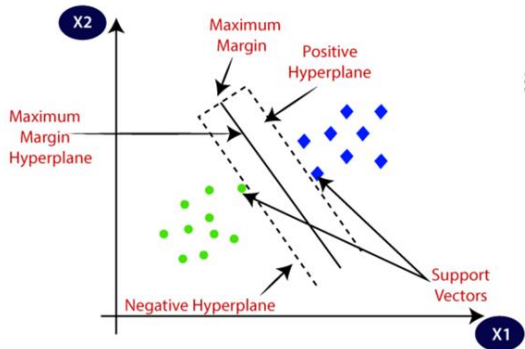


Hình 5: Tổng quan hệ thống

A. Mô hình máy học

1) Support Vector Machine (SVM)

Support Vector Machine (SVM)[6] là một thuật toán thuộc nhóm Supervised Learning dùng để phân chia dữ liệu thành các nhóm riêng biệt (bài toán phân lớp).



Hình 6: mô hình Support Vector Machine

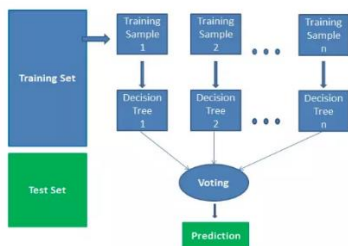
Xác định lớp cho một điểm dữ liệu mới sau khi tìm được mặt phân cách $w^T x + b = 0$, phân lớp của một điểm bất kỳ được xác định như sau:

$$f(x) = \text{class}(x) = \text{sgn}(w^T x + b)$$

Trong đó hàm $\text{sgn}()$ xác định dấu, nhận giá trị 1 nếu đối số là không âm, -1 nếu ngược lại và 0 nếu $w^T x + b = 0$. Với w là vector trọng số, x là vector đặc trưng của mẫu dữ liệu đang xét và hệ số điều chỉnh b .

2) Random Forest

Random Forest là một thuật toán máy học dựa trên việc xây dựng nhiều cây quyết định và kết hợp kết quả của chúng để đưa ra kết quả cuối cùng. Đây là mô hình học có giám sát có thể sử dụng cho cả phân lớp và hồi quy.

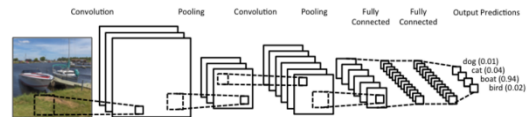


Hình 7: Mô hình Random Forest

3) CNNs

Convolution Neural Network (CNN)[7] là một mạng nơ-ron cho phép xử lý dữ liệu dạng lưới. CNNs được áp dụng phổ biến trong xử lý ảnh, xử lý tiếng nói, xử lý dữ liệu âm thanh. Trong mô hình học máy CNNs bao gồm các lớp convolution kết hợp với các hàm kích hoạt phi tuyến tính như hàm ReLU để tạo ra thông tin trừu tượng cho các lớp sau. Các lớp trong CNNs được liên kết với nhau bằng các phép tích chập, là một dạng đặc biệt của phép tuyến tính, tầng tiếp theo là kết quả tích chập từ tầng trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Ngoài mỗi lớp còn có các bộ lọc khác nhau, còn có một số lớp khác như pooling/subsampling dùng để chặn lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số cho các bộ lọc.

Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra các thông số tối ưu cho các bộ lọc tương ứng theo thứ tự raw pixel > edges > shapes > facial > higher-level features. Tầng cuối cùng dùng để phân lớp ảnh.



Hình 8: Mạng CNN

Công thức hàm ReLU:

$$\text{ReLU}(x) = \max(0, x)$$

Trong đó, x là giá trị đầu vào, \max là hàm lấy giá trị lớn nhất. Nếu x là dương, thì giá trị sẽ là x , còn nếu x là âm, thì giá trị sẽ là 0.

B. Công cụ sử dụng

Môi trường sử dụng: Google Colab.

Ngôn ngữ sử dụng: Python

Thư viện sử dụng:

- Wget: một công cụ dòng lệnh được sử dụng để tải xuống dữ liệu từ Internet.
- Os: cung cấp các hàm như tạo, xóa, di chuyển và đổi tên thư mục, quản lý biến môi trường, và nhiều thao tác khác.
- Pickle: là một module dùng để serialize và deserialize đối tượng Python.
- Matplotlib: dùng để vẽ biểu đồ hoặc đồ thị 2D
- Numpy: cung cấp các hàm, công cụ cho việc thao tác mảng và thực hiện các phép toán.
- Sklearn: cung cấp các công cụ cho việc xây dựng và đánh giá các mô hình máy học.
- Seaborn: cung cấp một cấu trúc cao cấp để tạo ra các biểu đồ thống kê đẹp mắt và dễ hiểu.
- Cv2: thực hiện các công việc thao tác đến ảnh

- TensorFlow: là một thư viện mã nguồn mở để xây dựng và huấn luyện mô hình máy học, đặc biệt là mạng nơ-ron (neural networks).

C. Các phương pháp đánh giá

Confusion matrix: Là một cách tốt để đánh giá hiệu quả các bộ phân lớp. Đếm số điểm dữ liệu đáng lẽ có nhãn A nhưng bị dự đoán nhầm sang nhãn B, điền vào ma trận với dòng là lớp đúng và cột là lớp dự đoán. Qua đó, dễ dàng phát hiện lớp nào phân loại đúng nhiều nhất, lớp nào thường bị phân loại sai vào lớp khác.

		ACTUAL	
		Negative	Positive
PREDICTION	Negative	TRUE NEGATIVE	FALSE NEGATIVE
	Positive	FALSE POSITIVE	TRUE POSITIVE

Hình 9: Tổng quan ma trận nhầm lẫn

- True Positive (TP): Số lượng các giá trị mô hình dự đoán đúng là positive.
- True Negative (TN): Số lượng các giá trị mô hình dự đoán đúng là negative.
- False Positive (FP): Số lượng các giá trị mô hình dự đoán là positive nhưng thực tế lại là negative.
- False Negative (FN): Số lượng các giá trị mô hình dự đoán negative nhưng thực tế lại là positive.

Dựa vào confusion matrix, chúng tôi sử dụng các độ đo để tính hiệu suất của mô hình:

- Accuracy đo khả năng mô hình dự đoán chính xác trong toàn bộ các dự đoán.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision dùng để đo độ chính xác của dự đoán dương tính (tỉ lệ dự đoán dương tính đúng).

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall là tỉ lệ những điểm dữ liệu dương tính (positive) được dự đoán đúng. Còn gọi là độ nhạy (sensitivity) hoặc (true positive rate).

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Độ đo F1 là độ đo kết hợp hai độ đo precision và recall dùng để so sánh các bộ phân lớp. F1 càng cao, bộ phân loại càng tốt. Vì vậy, độ đo F1 phù hợp với nghiên cứu này.

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

D. Đánh giá và tinh chỉnh hệ thống

Sau khi huấn luyện ba phương pháp máy học SVM, Decision Tree và CNNs với tham số mặc định, chúng tôi thu được các kết quả ở bảng 2.

Bảng 2: Kết quả của các phương pháp

Phương pháp	Accuracy	Precision	Recall	F1-score
SVM	0.68	0.66	0.66	0.65
Random Forest	0.761	0.771	0.751	0.756
CNNs	0,888	0,885	0,886	0,887

Với các kết quả trên, ta có thể thấy phương pháp CNNs cho ra kết quả khá tốt. Phương pháp SVM và Random Forest đưa ra kết quả thấp hơn so với kỳ vọng. Chúng tôi tiến hành tinh chỉnh các phương pháp bằng cách điều chỉnh giá trị các tham số.

Đối với mô hình SVM, chúng tôi tiếp tục tiến huấn luyện trên 3 kernel là rbf, poly, linear và điều chỉnh các tham số để tăng độ chính xác cũng như tăng hiệu suất của mô hình.

- Đầu tiên, ở kernel poly, chúng tôi điều chỉnh tham số C thay đổi theo từng giá trị 0.1, 1.0, 10.0 để tối ưu hóa việc phân loại đúng các điểm dữ liệu, giá trị degree thay đổi lần lượt là 2, 3, 4, 5 thì thu được accuracy cao nhất là 0.68 với C=10.0 và degree=2.
- Tiếp đến, ở kernel rbf, chúng tôi điều chỉnh tham số C thay đổi theo từng giá trị 0.1, 1.0, 10.0, giá trị gamma = 'scale' để tự động chia giá trị của gamma theo nghịch đảo của số lượng đặc trưng của dữ liệu 1/features thì thu được accuracy cao nhất là 0.74 với C=10.0.
- Cuối cùng, ở kernel linear, chúng tôi điều chỉnh tham số C thay đổi theo từng giá trị 0.1, 1.0, 10.0 thì accuracy cao nhất là 0.8 với C=0.1.

Đối với mô hình Random Forest, sau khi tinh chỉnh các tham số cơ bản của mô hình chúng tôi thấy kết quả mô hình không tăng lên đáng kể.

Đối với mô hình CNNs, chúng tôi xây dựng phương pháp CNNs với các lớp Convolutional 1 - Convolutional 2 - Flattening - Fully Connected (FC1) - Fully Connected (FC2) - Fully Connected (LastLayer).

Các lớp Convolutional 1, 2 và Fully Connected 1, 2 đều sử dụng hàm kích hoạt ReLU để giữ lại mọi giá trị dương và loại bỏ giá trị âm với công thức:

$$f(x) = \max(0, x)$$

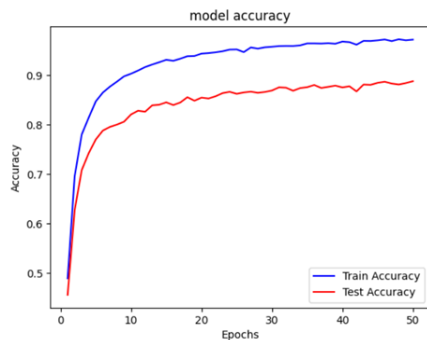
Lớp Convolutional còn sử dụng hàm max pooling giúp giảm kích thước của tensor, giữ lại các đặc trưng quan trọng và giảm độ phức tạp của mô hình. Lớp Flattening chuyển tensor đầu ra từ pooling cuối cùng thành vector 1D để dùng cho lớp Fully Connected. Ngoài ra, lớp Fully Connected (FC1) có áp dụng drop out để giảm overfitting. Lớp Fully Connected (LastLayer) là lớp đầu ra nên không dùng hàm kích hoạt ReLU mà chúng tôi sử dụng hàm mất mát softmax cross entropy với công thức sau:

$$H(y, p) = -\sum_i y_i \log(p_i)$$

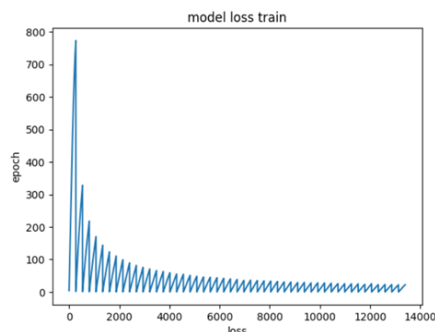
Chọn số Epochs là 50 và kích thước của mỗi batch là 130 để huấn luyện mô hình nhằm giúp mô hình được học đủ,

không bị overfitting hoặc underfitting. Ngoài ra, chúng tôi còn sử dụng thuật toán tối ưu hóa Adam để cải thiện hiệu suất.

Sau quá trình huấn luyện, accuracy cao nhất mà mô hình ở đạt được trên tập huấn luyện là 0.97 và tập test là 0,888. Kết quả tổng quan sau khi huấn luyện mô hình được thể hiện trên hình 10 và hình 11 thể hiện thông tin về hàm mất mát.

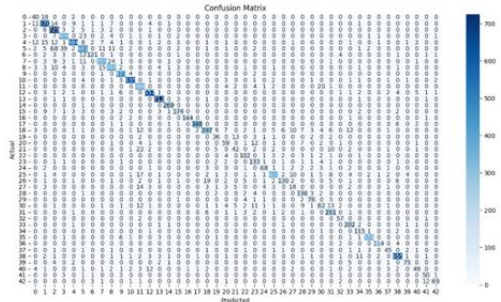


Hình 10: Accuracy CNNs



Hình 11: Loss CNNs

Ngoài ra, chúng tôi dùng ma trận nhầm lẫn để đánh giá mô hình trên dữ liệu test (Hình 12)



Hình 12: Confusion matrix CNNs

IV. PHÂN TÍCH LỖI

Sau quá trình tinh chỉnh mô hình, chúng tôi nhận thấy cả ba mô hình đều hoạt động tương đối đạt yêu cầu. Hầu hết các mô hình đều dự đoán chính xác các nhãn. Tuy nhiên, kết quả mô hình sử dụng Random Forest vẫn còn quá thấp. Để tìm hiểu nguyên nhân tại sao phương pháp Random Forest lại có kết quả thấp như vậy, chúng tôi đã tiến hành phân tích lỗi mô hình này.

Qua quá trình nghiên cứu, chúng tôi nhận thấy mô hình Random Forest vẫn còn nhầm lẫn do dữ liệu không cân bằng nên khi dự đoán sẽ bị nhầm lẫn giữa các nhãn. Bên cạnh đó,

việc tinh chỉnh mô hình Random Forest còn tương đối thô sơ, chưa xử lý qua các phương pháp tinh chỉnh nâng cao dẫn đến kết quả không đạt cao như mong muốn.

V. HƯỚNG PHÁT TRIỂN

Từ những kết quả thu được, chúng tôi đã đề xuất hướng đi cụ thể trong tương lai cho nghiên cứu này.

- Tự xây dựng bộ dữ liệu biển báo giao thông của Việt Nam. Tập trung thời gian cho phần thu thập đa dạng biển báo trong mọi điều kiện ánh sáng và thời tiết. Sử dụng các kỹ thuật như phóng to, thu nhỏ, xoay ảnh để tăng khả năng nhận diện của mô hình. Đặc biệt, tiền xử lý dữ liệu, điều chỉnh dữ liệu cho phù hợp với các mô hình, giúp tăng cường khả năng nhận diện của mô hình.
- Sử dụng các phương pháp học máy hoặc học sâu khác để tăng tính chính xác của mô hình. Tối ưu hóa thời gian nhận diện và tốn ít tài nguyên của hệ thống nhất. Liên tục cập nhật mô hình với dữ liệu mới để cải thiện độ chính xác.
- Áp dụng mô hình vào thực tế bằng phương pháp xây dựng ứng dụng di động cho người điều khiển giao thông.

VI. KẾT LUẬN

A. Kết quả đạt được

Sau quá trình tìm hiểu, huấn luyện cũng như tinh chỉnh mô hình, chúng tôi đã đạt được mục tiêu ban đầu đề ra, đó là xây dựng mô hình nhận diện biển báo giao thông có độ chính xác trên 80%. Chúng tôi đã ứng dụng được các kỹ thuật xử lý ảnh cơ bản, các phương pháp điều chỉnh nhằm tối ưu hóa mô hình. Có cái nhìn được toàn diện về mô hình học máy nhận diện biển báo giao thông.

B. Khó khăn gặp phải

Chưa có kiến thức về phương pháp CNNs nên mất rất nhiều thời gian trong việc tìm hiểu và xây dựng mô hình với phương pháp này.

Chưa biết nhiều kỹ thuật xử lý ảnh dẫn đến nhiều khó khăn trong việc xử lý dữ liệu. Huấn luyện các phương pháp với bộ dữ liệu ảnh tốn rất nhiều thời gian và công sức của cả nhóm.

VII. TÀI LIỆU

- [1] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11), 2278–2324 (1998)
- [2] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
- [3] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: The IEEE

Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

[4] Lin, M., Chen, Q., Yan, S.: Network in network. In: Proceedings of ICLR (2014)

[5] Yin, S., Deng, J., Zhang, D., Du, J. (2017). Traffic Sign Recognition Based on Deep Convolutional Neural Network. In: Yang, J., et al. Computer Vision. CCCV 2017.

[6] Aurélien Géron, Hands on Machine Learning with Scikit-Learn & TensorFlow, O'Reilly Media, 2017

[7] Đoàn Xuân, Dũng. Tóm tắt văn bản sử dụng các kỹ thuật trong deep learning. 2018. PhD Thesis.

[8] Nguyễn Duy Sim, Phân lớp bằng Random Forests trong Python (2018).

[9] “Tổng hợp các tài liệu về Convolutional Neural Network – chuẩn bị cho Quiz 02 Kambria Code Challenge”, Tổng hợp các tài liệu về Convolutional Neural Network – chuẩn bị cho Quiz 02 Kambria Code Challenge, [Tổng hợp các tài liệu về Convolutional Neural Network - chuẩn bị cho Quiz 02 Kambria Code Challenge | TopDev](#)

BẢNG PHÂN CÔNG CÔNG VIỆC			
MSSV	Họ và tên	Nội dung công việc	Đánh giá
21522429	Trần Nguyễn Yến Nhi	1.Tìm hiểu và xây dựng mô hình Random Forest. 2. Hỗ trợ viết báo cáo 3. Làm slide	100%
21522747	Trịnh Tuấn Tú	1.Tìm hiểu và xây dựng mô hình SVM 2. Viết chính báo cáo	100%
21522515	Nguyễn Việt Quang	1.Tìm hiểu và xây dựng mô hình CNN. 2. Hỗ trợ viết báo cáo	100%