A Deep Learning Based Traffic Sign Detection for Intelligent Transportation Systems

Bao-Long Le[1,2], Gia-Huy Lam[1,2], Xuan-Vinh Nguyen[1,2], The-Manh Nguyen[1,2],
Quoc-Loc Duong[1,2], Quang Dieu Tran[3], Trong-Hop Do[1,2],
and Nhu-Ngoc Dao[4](✉)

[1] University of Information Technology, Ho Chi Minh City, Vietnam
{19521782,18520832,18521655,18521084,18521006}@gm.uit.edu.vn,
hopdt@uit.edu.vn

[2] Vietnam National University, Ho Chi Minh City, Vietnam

[3] Ho Chi Minh National Academy of Politics, Hanoi, Vietnam
dieutq@hcma.vn

[4] Department of Computer Science and Engineering, Sejong University,
Seoul, South Korea
nndao@sejong.ac.kr

Abstract. Automatic detection and classification of traffic signs [1] bring convenience and caution to drivers on the road. It provides drivers with accuracy and timeliness in compliance as well as notifications in the route they are on. In the field of computer vision, the problem of detecting and classifying traffic signs has attracted great attention from research communities, because of the consequences it can bring if any mistake is made. In this problem, we have built a highly realistic data set with many challenges for Vietnam's traffic. In addition, we also solve the problem of automatic detection [2] and classification of traffic signs on the dataset that we have built using YOLOv4 and YOLOv5 algorithms with fine-tuned parameters. The results obtained in this paper are that the accuracy in detecting and classifying signs is quite high and the error is very low compared to outside traffic in Vietnam. The article is expected to benefit the development of practical applications and bring certain contributions in further developing this issue.

Keywords: Traffic sign detection · Smart car · Intelligent transportation system

1 Introduction

In recent years, self-driving car have become a hot topic that receive lots of attention from both academic and industry. One of the key components in a self-driving car is the computer vision module used for obtaining various type of traffic data from environment. Traffic sign is among crucial data for self-driving

D. Mohaisen and R. Jin (Eds.): CSoNet 2021, LNCS 13116, pp. 129–137, 2021.
https://doi.org/10.1007/978-3-030-91434-9_12

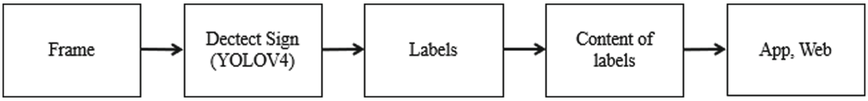


Fig. 1. System design.

car to operate properly. For example, based on the instruction on traffic signs, the vehicle know if it can turn left or right, or if it must reduce the speed. Therefore, a traffic sign detection system is a must for any self-driving car system [3].

Detecting a single traffic sign is not a difficult problem as most traffic signs has simple patterns with features easy to extract. Detecting and differentiating many traffic signs [4], however, is a challenging problem as many traffic signs have similar patterns. Beside accuracy, processing time is another factor to concern. For such an application like self-driving car, any mistake or delay in detecting and classifying a traffic sign might lead to serious consequences. The problem is exacerbated in developing countries with modest traffic infrastructure where traffic signs are usually blocked by many obstacles.

Thanks to the development of advance object detection algorithms, traffic sign detection has become a much approachable compare to it was just less than 10 years ago. Among possible approaches for traffic sign detection, deep learning based algorithms are likely to have the best performance in terms of accuracy and processing time. A tremendous number of experiments has shown that deep learning based techniques like You Only Look Once (YOLO) [5], Single Shot Detection (SSD) [6] perform very well in manky object detection tasks. However, compared to normal object detection tasks, traffic sign detection [7] is different in that the number of object class, which is the number of types of traffic signs, is much larger. The larger number of classes, the higher possibility of misclassifying the detected object [8].

This paper focuses on building a traffic sign detection application to detect popular traffic signs in Vietnam. This application receives a traffic video as input. It then locates the regions of the traffic signs in the videos and recognizes these signs. To train the traffic sign detection model, a large dataset consisting of 16770 images of 54 types of traffic signs has been built. The performance of the proposed application has been tested and evaluated in various metrics. Based on the experiment results, an analysis of detection errors in the application has also been provided.

2 Proposed System Architecture

System Design: The design of the proposed system is described in Fig. 1. The input of the system are traffic video frames. A transfer learning model based on YOLOv4 is used for detecting the traffic signs in each video frames to obtain

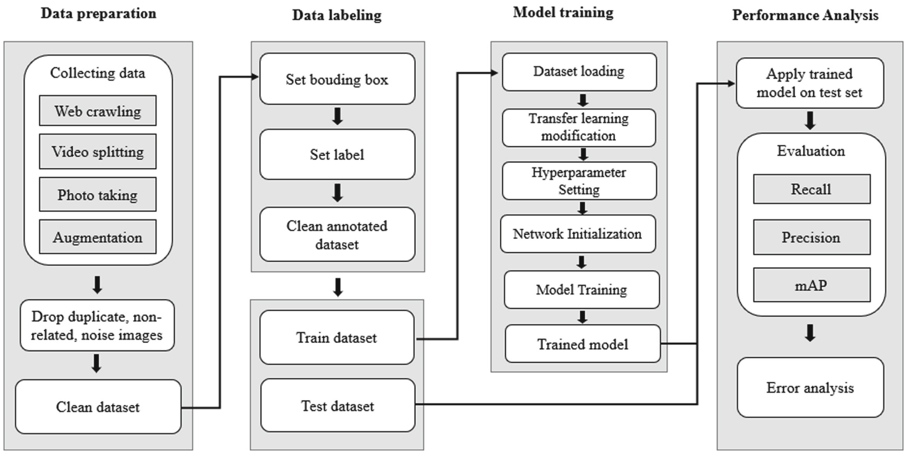


Fig. 2. Experimental procedure.

the labels of these signs. Then, the contents of these labels is shown to users through the web based interface of the system.

Transfer Learning Model Based on Yolov4: YOLOv4 [9] has many special enhancements that increase the accuracy and speed of its brother YOLOv3 [10] on the same COCO dataset and on the V100 GPU. The structure of v4 is divided into four parts: Backbone, Neck, Dense prediction, Sparse Prediction.

The backbone network for object recognition is usually pre-trained through the ImageNet classification problem. Pre-train means that the weights of the network have been adjusted to identify relevant features in an image, although they will be fine-tuned in the new task of object detection. The author considers using the backbone: CSPResNext50, CSPDarknet53, EfficientNet-B3.

Neck is responsible for mixing and matching feature maps learned through feature extraction (backbone) and identification process (YOLOv4 called Dense prediction).

YOLOv4 allows customization of Neck structures such as: FPN, PAN, NAS-FPN, BiFPN, ASFF, SFAM, SSP.

3 Experiment

The procedure of the experiment in this paper is described in Fig. 2. The experiment includes four steps: data preparation, data labeling, model training, and performance analysis.

Table 1. Correlation table between *class_id* and *label.*

Class	Label	Class	Label	Class	Label
0	No thoroughfare	18	No proceeding unspecified direction	36	Road/lane reserved for cars
1	No entry	19	Road junction	37	One way
2	No cars	20	Roundabout ahead	38	Parking
3	No motorcycles	21	Road junction with priority	39	U-turn permitted
4	No motor vehicles	22	Give way to main road	40	Direction allowed per lane
5	No trucks	23	Traffic signals ahead	41	Road/lane reserved for specified vehicles
6	No bus or trucks	24	Railway level crossing ahead that has-automatic gates or other barriers	42	Direction indicating
7	No motorcycles, tricycles	25	Railway level crossing ahead that does not have any automatic gate or other barriers	43	Alternative route where a turn is prohibited
8	No pedestrians	26	Pedestrian crossing ahead	44	Star/end of a built-up area
9	Gross vehicle weight limit	27	Children	45	Pedestrian crossing
10	Height limit	28	Construction	46	Overpass/underpass route for pedestrian
11	Stop	29	Slow	47	Hospital
12	No left/right turn	30	Direction	48	Traffic police station
13	No U-turn	31	Keep pass side	49	Gas station
14	No overtaking	32	Roundabout	50	Bus stop/station
15	No overtaking bytrucks	33	Walking path	51	Disabled parking only
16	Maximum speed limit	34	Minimum speed limit	52	Overpass route
17	No stopping/parking or waiting	35	Overpass route	53	Others

3.1 Datasets

In this paper, the dataset of traffic signs was collected in two ways: image collection from Google search page and video recording. Most of the data is collected by video recording because of its closeness to reality, the variety of contexts as well as the noise that the images available on Google rarely bring. The video recording is divided into two directions, once is the actual battle (out to the street to shoot traffic signs), the other is based on the image projected from the satellite on Google Maps and then back to the screen. For the first direction is collected images will more than for the second direction. However, The second direction is used to supplement data for signs that are difficult to encounter in real life because it is not possible to correctly locate the remaining signs. If this second direction still does not meet the quantity, the sample signs will be stitched

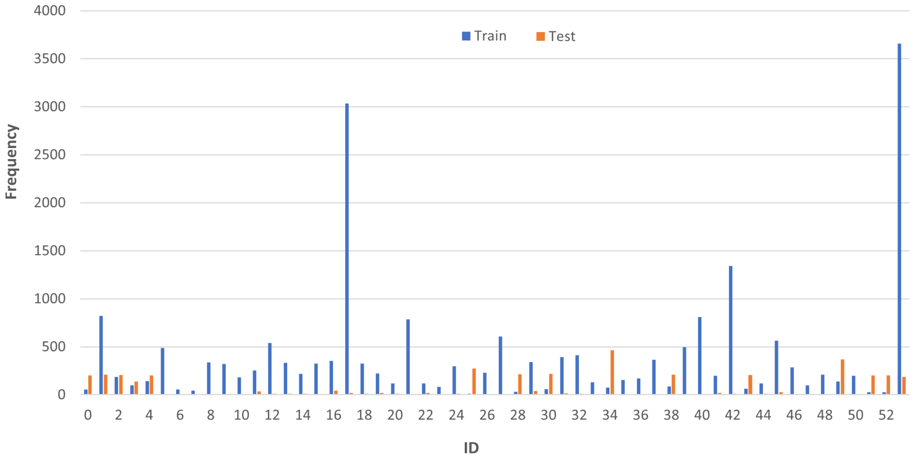


Fig. 3. Number of photos per set.

into the actual context to create a realistic image and ensure the quantity for the signs.

The collected signs are common signs that can be encountered in life with the label names based on the traffic manual, a total of about 54 labels of which 53 are single signs, one category contains images deemed complex or absent from the selected number of signs as shown in Table 1.

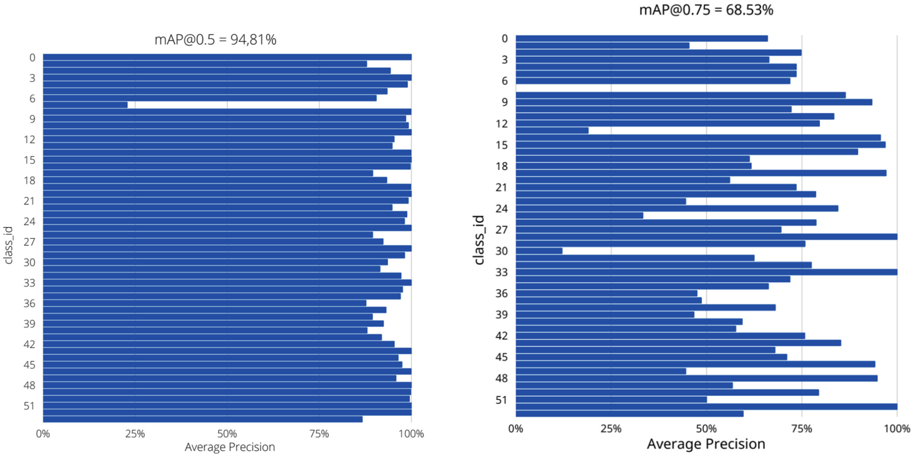
This label was added for the later developed problem. After labeling the images and videos, there are 16770 images in total, of which 13439 are for the training set and 3331 for the test set. Figure 3 illustrates the statistical chart of the number of each assigned label.

3.2 Data Preprocessing

Each image has many different features. Therefore, to be used in the model, the image data has to go through several preprocessing steps. Below are the preliminary preprocessing steps on the image dataset:

- Read the image, then convert the color channels of all images to RGB format to create consistency in the number of color channels for all images to match the model input.
- Resize the photo to the appropriate size - height: 416 pixels and width: 416 pixels. So all images have been converted to size 416 * 416 * 3.

After preprocessing, we use yolov4 to train labeled images from the dataset with the following parameters: Yolov4 using the model yolov4 Pre-trained. The parameters used are: batch = 64, subdivisions = 16, max_batches = 108000, steps = 86400,97200, filters = 177, classes = 54, width = 416, height = 416.



(a) 0.5

(b) 75

Fig. 4. mAP.

3.3 Evaluation Methods

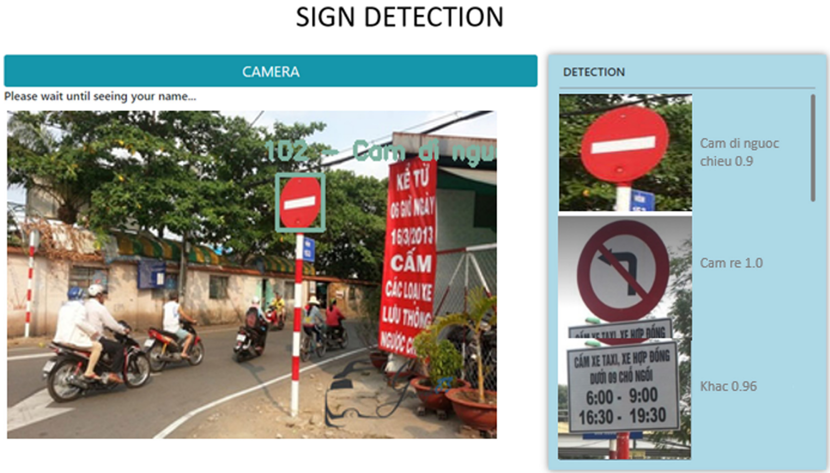
Performance metrics of object detection problem include:

- **IoU (Intersection over union)** is the ratio between measuring the degree of intersection between two contours (usually the predicted contour and the actual contour) to determine if two frames are overlapping. This ratio is calculated based on the area of intersection of 2 contours with the total area of intersection and non-intersection between them.
- **Precision** measure how accurate is the model's prediction i.e. percentage of model's prediction is correct.
- **Recall** measure how well the model finds all positive patterns.

From the precision and recall defined above, we can also evaluate the model based on changing a threshold and observing the values of Precision and Recall. The concept of Area Under the Curve (AUC) is similarly defined. With Precision-Recall Curve, the AUC has another name, **Average precision (AP)**. Suppose there are N thresholds for precision and recall, with each threshold for a pair of precision values, recall is $R_n, n = 1, 2, \ldots, N$. Precision-Recall curve is drawn by drawing each point with coordinates (P_n) on the coordinate axis and connecting them together. AP is defined by:

$$AP = \sum_{n=0}^{N} [\mathrm{R}_n - \mathrm{R}_{n-1}] * \mathrm{P}_n \tag{1}$$

In multiple-classes object detection, mAP is the average of AP calculated for all classes.



Please wait until seeing your name...

Cam di nguoc chieu 0.9

Cam re 1.0

Khac 0.96

Fig. 5. SIGN detection demo.

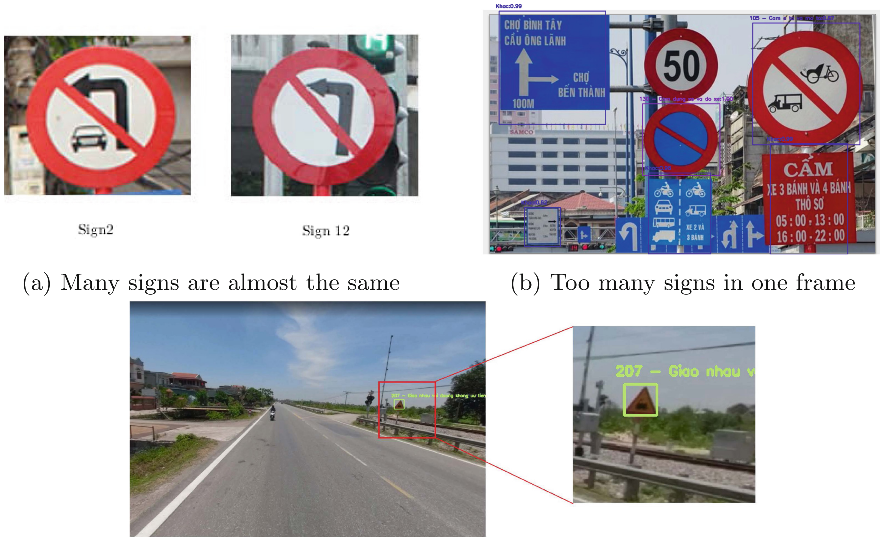
3.4 Results

During the long training period (specifically, training around 4000 rounds/day, with approximately 27 days for training to complete), there were many models saved at rounds 10000, 20000, ... 10000; along with the models saved from the calculation of mAP at each small round. And we compared the obtained models. In the end, the best model is the one with mAP@0.5 = 94.81% and mAP@0.75 = 68.53%.

Derived from Figs. 4a, 4b and Table 1, it can be seen that the overall model evaluation results for the dataset are very good with mAP@0.5 up to 94.81% and mAP@0.75 = 68.53%, only a few cases are not high, like class_id = 7 is a sign that prohibits motorcycles and tricycles with very low accuracy AP = 22.85% at rating mAP@0.5 and AP = 0 at rating mAP@0.75.

3.5 SIGN Detection Application

To build this application, we use python language with the main library Flask, that capable of creating an interface that can be accessed by the website. After detecting the signs and determining their classes, the parts containing the signs in the video frames will be shown in the right panels to show the signs and their contents as shown in Fig. 5. Next, transmit to the web the cropped image with the category of the image after the model predicts it. To transmit information on the device interface so that the driver can observe. This application builds for users a list of 20 consecutive signs that help drivers have more information about signs and the next section.



(a) Many signs are almost the same

(b) Too many signs in one frame

(c) Incorrect label detection cause small scale sign

Fig. 6. Typical error of detecting and labeling.

3.6 Error Analysis

Some pairs of figures have a lot of detail look similar to each other, which is explained in Fig. 6a. This issue easily causes confusion for detection model. Frames with a large number of objects (signs) or objects with overlap as shown in Fig. 6b also make the model difficult to detect and recognize. Another issue is about detecting small objects in large scenes. In figures c, because the object is so small in proportion to the frame, it is mistaken as sign #21 instead of sign #25 (Figs. 6c). There are a few road signs that are not ordinarily utilized, so collecting data will be a troublesome issue for us as class imbalance also affects the predictive performance of the model. Some of the reasons for the difference in accuracy of identifying signs is due to the unevenness in the complexity of them, or some signs have a highly correlated appearance with others, or the rate involved in data set construction.

4 Concluding Remarks

In the future, we plan to improve and expand the dataset by recording more videos of different routes to make it closer to real life. We also develop other methods to improve the quality of identifying QR codes by combining more frames. In this article, we used the method Yolov4. The method gave a high result of 94.18% for mAP @0.5, but it gave a quite low result of 68.53% for mAP@75 due to some cases of unable to identify such as prohibit motorcycle signs or prohibit three-wheeled vehicle sign. The accuracy of identifying those mentioned signs were extremely low: AP = 22.85% for mAP@0.5, and AP = 0 for mAP@0.75. This happened because the amounts of different signs in the dataset are quite uneven. In the future, we plan to improve and expand the dataset for those types of signs that have a small number of images. In addition to identifying the signs, we would develop the problem to also provide instructions or warnings based on the collected images from the dash camera. We hope to contribute this dataset to the community in order to motivate the research of identifying traffic signs, improve the efficiency of identifying with better methods.

Acknowledgment. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1G1A1008105).

References

1. Shneier, M.: Road sign detection and recognition. In: Unmanned Systems Technology VIII, vol. 6230, p. 623016. International Society for Optics and Photonics (2006)
2. Wali, S.B., Hannan, M.A., Hussain, A., Samad, S.A.: An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and SVM. Math. Probl. Eng. **2015**, 1–11 (2015)
3. Zaki, P.S., William, M.M., Soliman, B.K., Alexsan, K.G., Khalil, K., El-Moursy, M.: Traffic signs detection and recognition system using deep learning. arXiv preprint arXiv:2003.03256 (2020)
4. Tabernik, D., Skočaj, D.: Deep learning for large-scale traffic-sign detection and recognition. IEEE Trans. Intell. Transp. Syst. **21**(4), 1427–1440 (2019)
5. Huang, R., Pedoeem, J., Chen, C.: YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 2503–2510. IEEE (2018)
6. Zuo, Z., Yu, K., Zhou, Q., Wang, X., Li, T.: Traffic signs detection based on faster R-CNN. In: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 286–288. IEEE (2017)
7. Zhang, F., Zeng, Y.: D-FCOS: traffic signs detection and recognition based on semantic segmentation. In: 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), pp. 287–292. IEEE (2020)
8. Yu, J., Liu, H., Zhang, H.: Research on detection and recognition algorithm of road traffic signs. In: 2019 Chinese Control and Decision Conference (CCDC), pp. 1996–2001. IEEE (2019)
9. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
10. Zhang, B., Wang, G., Wang, H., Xu, C., Li, Y., Xu, L.: Detecting small Chinese traffic signs via improved YOLOv3 method. Math. Probl. Eng. **2021**, 1–10 (2021)