A Model for Real−Time Traffic Signs Recognition Based on the YOLO Algorithm – A Case Study Using Vietnamese Traffic Signs

Chapter · November 2019

DOI: 10.1007/978-3-030-35653-8_8

An Tran Cong
Can Tho University
34 PUBLICATIONS **179** CITATIONS
SEE PROFILE

Lu-Dien Duong
Can Tho University
2 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Hiep Xuan Huynh
Can Tho University
199 PUBLICATIONS **831** CITATIONS
SEE PROFILE

A model for real-time traffic signs recognition based on the YOLO Algorithm – A case study using Vietnamese traffic signs

An Cong Tran[1], Duong Lu Dien, Hiep Xuan Huynh[1]
Nguyen Huu Van Long[1], and Nghi Cong Tran

[1]Can Tho University, Vietnam
{tcan, hxhiep, nhvlong}@ctu.edu.vn

Abstract. The rapid development of the automobile resulted in the traffic infrastructure becomes more and more complicated. Both type and the number of traffic signs on the streets are increasing. Therefore, there is a need for applications to support drivers to recognize the traffic signs on the streets to help them avoid missing traffic signs. This paper proposes an approach to detecting and classifying Vietnamese traffic signs based on the YOLO algorithm, a unified deep learning architecture for real-time recognition applications. An anchor boxes size calculation component based on the k-means clustering algorithm is added to identify the anchor boxes size for the YOLO algorithm. A Vietnamese traffic sign dataset including 5000 images containing 5704 traffic signs of types was collected and used for evaluation. The F1 score of the tiny model (the fastest but the most inaccurate model) achieved is about more than 92% and the detection time is approximately 0.17 seconds (in our testing environment: laptop CPU Intel 3520M, 8GB RAM, no GPU). In comparison with similar research in Vietnamese traffic sign recognition, the proposed approach in this paper shows a potential result that provides a good trade-off between the recognition accuracy and recognition time as well as the feasibility for real applications.

Keywords: Vietnamese traffic signs, detection, YOLO algorithm, deep learning, k-means clustering

1 Introduction

The growth of automobiles results in the changes of traffic infrastructure. The number and types of road signs have been increased to respond to the development of the new infrastructure. As a result, the drivers have to remember more and more road signs as well as pay more attention to traffic signs when they are driving, i.e. more observation is needed. Therefore, it is necessary to have real-time traffic sign detection and classification systems to support drivers, to help them in avoiding missing traffic signs, which may lead to dangerous situations or event accidents. Besides, traffic sign detection and classification can also be applied in many other potential applications such as in developing smart cars or self-driving cars, etc.

Due to the important applications of traffic sign recognition, i.e. including detection and classification, many studies on this problem have been pursued since the 90s. In Arturo de la Escalera et al. research conducted in 1997, the authors proposed a method for traffic sign recognition, which uses the color threshold to segment the image and shape analysis to detect the road signs. Then, the neural network is used to classify the detected signs. The proposed method had been evaluated using a small dataset, including 9 images for training the neural network and 10 images for testing. However, the images in this dataset are taken in the natural scenes and in ideal conditions, i.e. without noise. Such selection was affected by the disadvantage of the proposed method, that is, this method is sensitive to the noise. It is easy to mis-classify the objects that have similar shape or color to the road signs such as advertisement panels, which is very popular in the urban environment.

Similar to the above, some other studies on traffic sign recognition around that time were also based on image processing techniques combined with similarity measurement algorithm or on the combination of image processing techniques and machine learning algorithm using popular feature descriptors such as in [3, 11, 6, 2, 9, 5]. Gavrila [6] uses a template-based correlation method and distance transforms to identify potential traffic signs in images. Then, a radial basis function network is used for classifying the recognized traffic signs. The template-based correlation method has a high computational cost and thus it is not suited to real-time recognition systems. Barnes Nick and Zelinsky Alex [2] use the fast radial symmetry detector to the image stream from a camera mounted in a car eliminate almost all non-sign pixels from the image stream. Then, they apply normalized cross-correlation to classify the signs. This method is suitable for circular signs only and thus they evaluate the method on the Australian road signs only. In order to detect triangular, square and octagonal signs, Loy Gareth and Barnes Nick use a similar technique in [9]. The symmetric nature of these shapes, together with the pattern of edge orientations exhibited by equiangular polygons with a known number of sides is used to establish possible shape centroid locations in the image. This approach is invariant to in-plane rotation and returns the location and size of the shape detected. However, this increases the computational cost so it cannot work in real-time. Unfortunately, the color-based methods are sensitive to strong light, poor light and other bad weather condition.

In recent years, with the increase in the amount of data and the computational power of computers, deep learning has become a new trend in both research and application areas. Many new neural network models have been introduced to solve the traditional problems by using the data-oriented approach, i.e. the huge amount of data is used with deep neural networks, and this method produces promising results. Therefore, recent research in traffic sign recognition also apply deep learning methods and they archived outstanding performance [4, 7, 12, 17, 15]. Cireşan et al. [4] proposed a multi-column deep neural network for classification running on a graphical processing unit and obtained a better-than-human recognition rate. Jin et al. [7] used a convolution neural network (CNN) with a hinge loss stochastic gradient descent method, which achieved a high detection rate. Quian et al. [12] used CNN as a classifier to learn the discriminative feature of max pooling position to classify traffic signs and obtained a good performance in comparison with the state-of-the-art method. Ali et al. [17] used a procedure based on color segmentation, histogram of oriented gradient, and a CNN for traffic sign detection. This research achieves better recognition accuracy and computational speed. However, in this research line, it is important for the real-time recognition and thus it is necessary to find more network structures to improve the recognition accuracy and processing time.

There are also some studies on Vietnamese traffic sign recognition, which are more closely related to our work in this paper [10, 1]. Binh et al. [10] combined the color-based and shape-based method for image segmentation and localization. Then, the SIFT matching is applied to match the keypoints of the extracted pictogram with the keypoints of template images in the database to detect the sign. The recognition accuracy is approximately 95%. However, the dataset is rather small with 600 images that belong to six types of dangerous signs. Also, this method is rather slow with the detection time is more than 2 seconds, in which the segmentation and localization take about 0.34 seconds and classification takes more than 1.8 seconds. In Truong et al. research [1], the authors used the HOG feature and neural network for traffic sign detection. A color-based segmentation is used beforehand to segment the red and blue traffic signs. Then, a shape-based method is used to identify the boundary of the potential candidates. Finally, the HOG feature of detected candidates are fed into a 3-layer neural network for training. The reported accuracy is about 95% and the detection time is approximately 0.12 seconds. This is a good performance in terms of both accuracy and recognition time. However, in general, the color-based is sensitive to lighting changes while the shape-based cue is insensitive to lighting changes but could be distracted by cluttered background [16].

With the prominent representation capacity and outstanding performance of deep learning methods in traffic sign recognition, in this study, we propose to use YOLO, a state-of-the-art deep learning model to recognize Vietnamese traffic signs. This learning model introduces several architectures that provide different levels of the trade-off between recognition accuracy and recognition time. Therefore, it is very flexible and can be adjusted to adapt to a particular circumstance, such as the hardware limitation, constraint on the accuracy, and the like. We used the k-means clustering algorithm [8] to cluster the bounding boxes of the traffic signs to identify the anchors' size for the YOLO algorithm. This study also introduces a new Vietnamese traffic sign dataset including 5704 images of 22 types of the traffic signs. These images are extracted from the videos recorded by a phone camera that was attached to the front of a motorcycle. The videos were mainly recorded in the urban areas, there exist many noises and distractions.

The paper is structured as follows: the proposed model for Vietnamese traffic sign recognition is presented in Section 2, including the description of the Vietnamese traffic sign system, the overall model of the proposed method and the structure of the YOLO algorithm. Then, the evaluation dataset and evaluation result are reported in Section 3. Finally, the conclusion and future work of this research are discussed in Section 4.

2 Traffic Sign Detection using YOLO Algorithm

2.1 Vietnamese Traffic Signs

There are 144 traffic signs in the Vietnamese traffic sign system. They are grouped into 4 groups: prohibitory or restrictive signs (40 signs), warning signs (47 signs), mandatory signs (10 signs) and indication signs (47 signs). Some Vietnamese traffic signs are shown in Figure 1 and an image in which there exists a traffic sign is demonstrated in Figure 2.

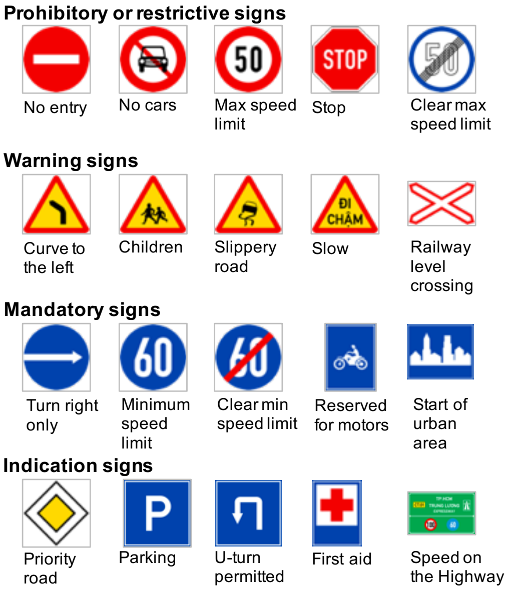

Fig. 1. Example of Vietnamese traffic signs

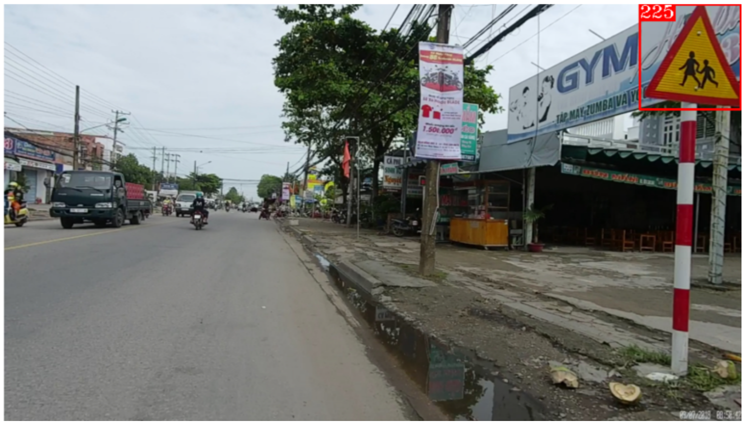

Fig. 2. Traffic sign in a dashcam image

2.2 Vietnamese Traffic Sign Recognition Model based on YOLO Algorithm

Traffic sign recognition includes two main tasks: i) detect the traffic signs within the images, and ii) classify the detected signs. In this research, we propose a model for recognizing the Vietnamese traffic signs based on the YOLO algorithm. We also employ the k-means clustering algorithm for identifying the anchors' size, one of the important hyper-parameters of the YOLO algorithm. Our proposed approach is presented in Figure 3.

First, the training data is labeled including the bounding box for traffic signs within the images and the label of the traffic signs. Then, we use the k-means clustering algorithm to cluster the bounding boxes to calculate the anchors' size. The labeled data together with the calculated anchors' size are put into the YOLO algorithm to build the traffic sign recognition model. This model then will be use to recognize the traffic signs inside images. Detail of the basic steps are described below.

2.3 The YOLO Network Architecture

YOLO (You Only Look Once) is an object detection approach which was first introduced in 2015 by Joseph Redmon and his colleagues [13]. This is a unified for real-time object detection in which a single neural network is used to predict bounding boxes and class probabilistic directly from full image in one inference. Therefore, it is known as an extremely fast deep learning model.

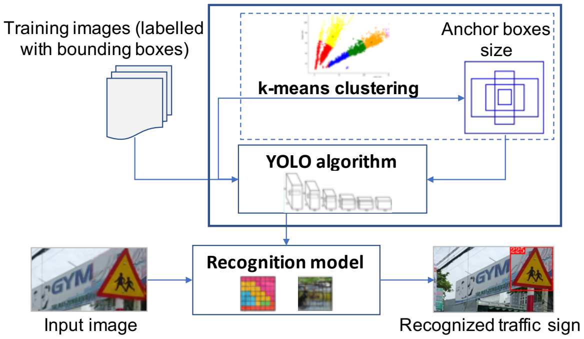

Fig. 3. Architecture of the Vietnamese traffic sign recognition system

The features from the entire image are used to predict each bounding box. All bounding boxes across all classes are also predicted simultaneously. This enables end-to-end training and real-time speed while maintaining high average precision [13]. To do so, YOLO divides the input image into S x S grids. Each cell in the grid predicts B bounding boxes and confidence score of bounding boxes. Each bounding box consists of five prediction (b_x, b_y, b_w, b_h, p). The (b_x, b_y) coordinates are the center of the box relative to the bound of the grid cell. The b_w and b_h are the the width and height of the bounding box which are predicted relative to the whole image. Finally, the predicted confidence p is calculated as follow:

$$p = Pr(Object) \times IOU_{pred}^{truth}$$

$Pr(Object) = 1$ if the grid cell contains a part of a ground truth box, otherwise it is zero. The IOU_{pred}^{truth} is the intersection over union between the predicted bounding box and the ground truth box.

Each grid cell also has conditional class probabilities $C = Pr(Class_i \mid Object)$, which represent the conditioned probabilities on the grid cell containing an object. Details of the calculation of C can be found in [13]. Figure 4 demonstrates the above process of YOLO.

At the detection time, the multiplication of C and p: $Pr(Class_i \mid Object) \times Pr(Object) \times IOU_{pred}^{truth}$ gives us the class-specific confidence scores for each box. These scores represent both the probability of that class appearing in the box and how well the predicted box fits the object.

The above model is implemented as a convolutional neural network (CNN). The initial convolution layers extract features from the images while the fully connected layers predict the output probabilities and coordinates. The YOLO

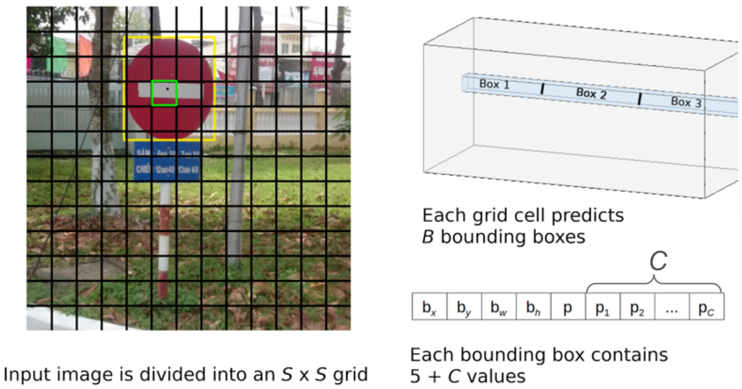

Fig. 4. Basic calculations in the YOLO Algorithm

algorithm has several versions and each of them has different numbers of convolutional layers and fully connected layers. The first version of YOLO has 24 convolutional layers followed by 2 fully connected layers. as shown in Figure 5.

An incremental version of YOLO, named YOLOv3, is introduced in [14] with faster detection time and better detection accuracy. In our study, we focus on the real-time detection with limited hardware condition, that is suited to the embedded systems. Therefore, so we use the tiny version of the YOLO v3 algorithm. This version has 13 convolutional layers followed by multiple subsampling layers to extract the image features. The YOLOv3 has an important improvement to the previous versions that is the capability to make detection at three different scale. It can also predict more bounding boxes than the previous ones. So, this algorithm is better than the previous version at detecting small objects.

2.4 Identify the Anchor Boxes Size

To identify the appropriate sizes of anchor boxes for the YOLO algorithm, we use the k-means clustering algorithm [8] as described in Figure 3. The YOLOv3 tiny version detect boxes at 2 different scales: 13 × 13 and 26 × 26. For each scale, YOLOv3 tiny uses three anchor boxes. Therefore, the number of anchor box sizes needed by the YOLOv3 tiny is 6. To calculate these required values, the size of all bounding boxes in the training data are clustered by the k-means algorithm into 6 clusters, i.e. the number of anchor boxes' size needed by YOLO. The centroids of 6 clusters are the six anchor sizes to be provided to the YOLOv3 tiny algorithm.

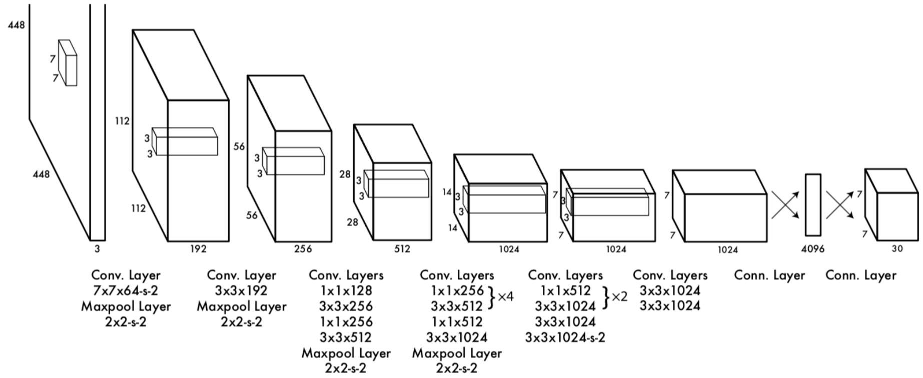

Fig. 5. The YOLO architecture [13]

In this study, the k-means clustering on our training data set returns six anchor boxes size, those are (20, 39), (51, 28), (37, 72), (89, 48), (116, 61), and (180, 95). This calculation is described in Figure 6.

The anchor box sizes are also used in the detection step. In the detection layer of each scale, the size of bounding boxes (b_x, b_y, b_w, b_h) are computed based on the values of the feature map (t_x, t_y, t_w, t_h). The object width and height are calculated by the linear function:

$$b_w = p_w + e^{t_w}$$

$$b_h = p_h + e^{t_h}$$

p_w and p_h are the width and height of the anchor box respectively. The object center (b_x, b_y) is the sum of the sigmoid of (t_x, t_y) and an offset (c_x, c_y), which is the coordinator of gird cell containing the object. For example, if the center of an object falls into the grid cell, which is the intersection of the 1^{st} row and the 2^{nd} column, then we have $c_x = 1$ and $c_y = 2$,

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

Figure 7 describe the calculation of the final bounding boxes.

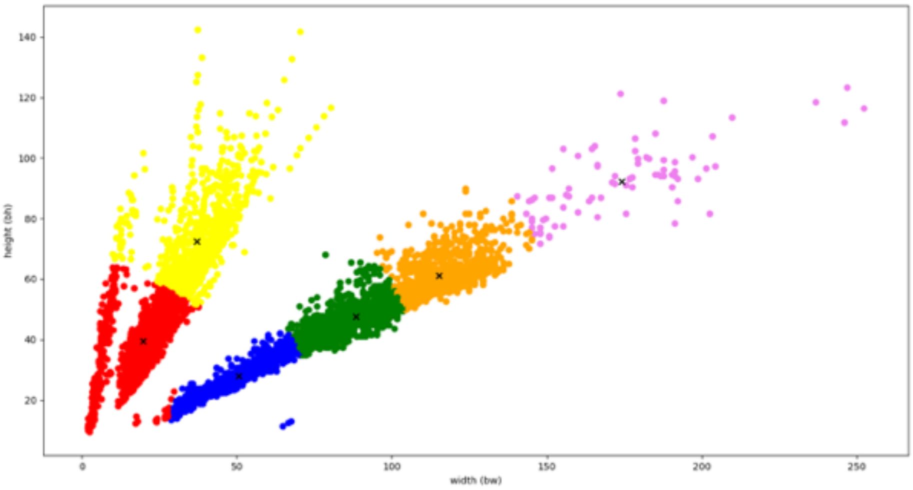

Fig. 6. The bounding box sizes clustering by k-means algorithm

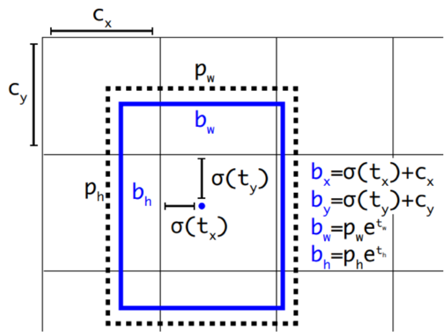

Fig. 7. Calculate the final bounding boxes [14]

3 Dataset and Evaluation

3.1 Evaluation Dataset

As there is no open Vietnamese traffic sign dataset, we collected a new one using a phone camera. We use a phone camera attached in the front of a motorcycle and run around the street in Can Tho City, Vietnam. Then, we extract the image from recorded videos. Our dataset finally has 5704 images of 33 traffic sign categories. Then, we labeled the image using the image labeling software labelImg[1] and grouped the similar traffic signs into one group to reduce the number of classes. For example, the road sign "Series of curves, the first curve to the left" and "Series of curves, the first curve to the right" are group into the "Series of curves". We used 80% of the dataset for training the recognition model and the remaining 20% for testing.

The final dataset has 5704 images with 22 classes as described in Table 1.

Table 1: Description of evaluation dataset

No	Label	Description	Train	Test
1	102	No entry	204	22
2	130	No stopping or parking or waiting	268	66
3	131	No parking or waiting	180	54
4	201a	Curve to the left	100	36
5	201b	Curve to the right	130	45
6	202	Series of curves	180	41
7	203	Road narrows	54	35
8	205	Road junction	307	79
9	207	Road junction with priority	773	182
10	208	Road junction with priority	145	29
11	209	Traffic signals ahead	125	25
12	221	Rough road surface	110	16
13	224	Pedestrian crossing ahead	428	100
14	225	Children	425	116
15	233	Other danger	111	24
16	245	Slow	62	12
17	302	Keep right	136	54
18	303	Roundabout	108	23
19	423	Pedestrian crossing	310	90
20	crowded	Start of a crowded area	50	11
21	end_crowded	End of a crowded area	29	10
22	traffic_light	Traffic light	304	95
Total			**4539**	**1165**

[1] https://github.com/tzutalin/labelImg

3.2 Evaluation Result

Evaluation metrics We use the following evaluation metrics to evaluate the proposed model:

- True positive (TP): model recognizes the object correctly, i.e. the model predict the class name correctly and the IoU value between predicted bounding box and ground truth box must be greater than or equal to 0.5.
- False Negative (FN): the traffic sign exists in the image but the model cannot recognize it.
- False Positive (FP): the model recognizes wrong object or correct object class but the IoU value between predicted bounding box and ground truth box is less than 0.5.
- Precision $= \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$
- Recall $= \dfrac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$
- F1 score: $\mathrm{F1} = 2 \times \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Evaluation result After using 80% of the dataset to train the recognition model using the proposed approach, we used the remaining 20% to evaluate the model. The evaluation result is given in Table 2.

Table 2: Evaluation result

No	Traffic sign	True Positive	False Negative	False Positive
1	102	22	0	0
2	130	61	4	1
3	131	54	0	0
4	201a	21	0	15
5	201b	36	0	9
6	202	41	0	0
7	203	12	1	22
8	205	78	0	1
9	207	180	1	1
10	208	29	0	0
11	209	25	0	0
12	221	16	0	0
13	224	99	1	0
14	225	115	0	1
15	233	23	0	1
16	245	12	0	0
17	302	40	14	0
18	303	20	2	1

19	423	90	0	0
20	crowded	9	0	2
21	end_crowded	5	1	4
22	traffic_light	1	94	0
Total		**989**	**118**	**58**

- $\text{Precision} = \dfrac{\text{TP}}{\text{TP} + \text{FP}} = \dfrac{1005}{1005 + 58} \approx 0.95$
- $\text{Recall} = \dfrac{\text{TP}}{\text{TP} + \text{FN}} = \dfrac{1005}{1005 + 118} \approx 0.89$
- $\text{F1} = 2 \times \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \dfrac{0.95 \times 0.89}{0.95 + 0.89} \approx 0.92$

The recognition time on a laptop CPU Intel 3520M, 8GB RAM (no GPU) is about 0.17 seconds per image. This result provides a good trade-off between detection time and detection accuracy and the detection time is suited to the real-time applications. It should be noted that the YOLO algorithm used in this evaluation is the tiny version, which is optimized to detection time. Therefore, if we need a better detection accuracy, we may use the full model of the YOLO algorithm.

In compare with former research on Vietnamese traffic sign recognition system, our system produces a good balance between performance and detection accuracy. Our system is better than the model proposed in [10] in both detection accuracy and detection time. However, we achieved a lower accuracy than [1]. This can be explained by two reasons. First, our dataset was collected in the city roads and thus it has more noise (i.e. the objects that are similar to traffic signs within the images) than the dataset used in [1]. Second, we are using the tiny version, which is mainly optimized for detection time by using a shallow network architecture. Therefore, if we have strong hardware, we can use deeper network architecture to improve the detection accuracy.

4 Conclusion and Future Work

In this study, we proposed a model for the Vietnamese recognition system. This model bases on the YOLOv3 algorithm, particularly the tiny version for real-time applications. We use the k-means clustering algorithm to compute the anchor box sizes for the YOLO algorithm. The evaluation result shows that our model gives a good trade-off between detection accuracy and detection time and can be used for real-time applications.

However, the detection accuracy is still lower than some of the state-of-the-art studies in traffic size recognition. As explained above, this may be caused by the shallow network architecture used or some other potential reasons such as the difference of the evaluation dataset, etc. This, together with investigations on the network parameters, will require further research.

References

1. Bao, T.Q., Chen, T.H., Dinh, T.Q.: Road traffic sign detection and recognition using hog feature and artificial neural network. Can Tho University Journal of Science 15, 47–54 (2015)
2. Barnes, N., Zelinsky, A.: Real-time radial symmetry for speed sign detection. In: IEEE Intelligent Vehicles Symposium, 2004. pp. 566–571. IEEE (2004)
3. Besserer, B., Estable, S., Ulmer, B., Reichardt, D.: Shape classification for traffic sign recognition. IFAC Proceedings Volumes 26(1), 487–492 (1993)
4. Cireşan, D., Meier, U., Masci, J., Schmidhuber, J.: Multi-column deep neural network for traffic sign classification. Neural networks 32, 333–338 (2012)
5. Garcia-Garrido, M.A., Sotelo, M.A., Martin-Gorostiza, E.: Fast traffic sign detection and recognition under changing lighting conditions. In: 2006 IEEE Intelligent Transportation Systems Conference. pp. 811–816. IEEE (2006)
6. Gavrila, D.M.: Traffic sign recognition revisited. In: Mustererkennung 1999, pp. 86–93. Springer (1999)
7. Jin, J., Fu, K., Zhang, C.: Traffic sign recognition with hinge loss trained convolutional neural networks. IEEE Transactions on Intelligent Transportation Systems 15(5), 1991–2000 (2014)
8. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory 28(2), 129–137 (1982)
9. Loy, G., Barnes, N.: Fast shape-based road sign detection for a driver assistance system. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). vol. 1, pp. 70–75. IEEE (2004)
10. Mai, B.Q.L., Dao, P.T., Huynh, H.T., Doan, D.A.: Recognition of vietnamese warning traffic signs using scale invariant feature transform. In: International Conference on Communications and Electronics (2014)
11. Piccioli, G., De Micheli, E., Parodi, P., Campani, M.: Robust method for road sign detection and recognition. Image and Vision Computing 14(3), 209–223 (1996)
12. Qian, R., Yue, Y., Coenen, F., Zhang, B.: Traffic sign recognition with convolutional neural network based on max pooling positions. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). pp. 578–582. IEEE (2016)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
14. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
15. Shao, F., Wang, X., Meng, F., Rui, T., Wang, D., Tang, J.: Real-time traffic sign detection and recognition method based on simplified gabor wavelets and cnns. Sensors 18(10), 3192 (2018)
16. Yang, M., Lv, F., Xu, W., Gong, Y., et al.: Detection driven adaptive multi-cue integration for multiple human tracking. In: 2009 IEEE 12th International Conference on Computer Vision (ICCV). pp. 1554–1561. IEEE (2009)
17. Youssef, A., Albani, D., Nardi, D., Bloisi, D.D.: Fast traffic sign recognition using color segmentation and deep convolutional networks. In: International Conference on Advanced Concepts for Intelligent Vision Systems. pp. 205–216. Springer (2016)