






Road Sign Detection and Recognition of Thai Traffic Based on YOLOv3

Paitoon Thipsanthia^(✉), Rapeeporn Chamchong,
and Panida Songram

POLAR Lab, Faculty of Informatics, Mahasarakham University,
Maha Sarakham, Thailand
{paitoon.thi, rapeeporn.c, panida.s}@msu.ac.th

Abstract. This paper aims to apply a YOLOv3 technique for detecting and recognizing Thai traffic signs in real-time environments. The Thai Traffic Sign Dataset (TTSD) was collected by car cameras to store the video images using the resolution of 1920×1080 pixels using 60 frames per second, and a 1280×720 pixels and 30 frames per second. In addition, the data was collected in the rural area of Maha Sarakham Province and Kalasin Province. The dataset was generated and distributed for general traffic sign detection and recognition. Two architectures (YOLOv3 and YOLOv3 Tiny) are compared with 50 classes of road signs and 200 badges in each class, containing 9,357 images. The experiment shows that the mean average precision (mAP) of YOLOv3 (88.10%) is better than YOLOv3 Tiny (80.84%) while the speed of YOLOv3 marginally is better than YOLOv3.

Keywords: Road sign detection · Deep learning · Machine learning

1 Introduction

The development of autonomous car technology has gained popularity in several developed countries with a number of reasons. There are many road traffic accidents resulting in serious injury or death because of failure to read road signs [1]. Nowadays, advance computer technology can be proceeded large amounts of data with real time processing. However, the recognition rate of road sign software is not promising. Although some of the most popular works have a high recognition rate, the processing cannot be done quickly enough [2].

Real-time traffic detection and recognition is one of the systems for driving unmanned vehicles that still has problems in time complexity and low accuracy. Yuan et al. [2] proposed the real-time traffic detection and recognition techniques using traffic detection techniques with Aggregated Channel method Features (ACF). This method has high traffic recognition efficiency but low detection rate. Time complexity of traffic sign detection is an average of 13.16 frames per second using CPU processing. It is not fast enough to be used in real-time processing but it provides good recognition performance. This technique applied the Support Vector Machines (SVM) to recognize traffic signs and achieve at 97.78%.

At present, traffic sign detection and recognition systems can be done faster in everyday life, it also comes with a high cost due to the problems of high speed processing. There are many techniques such as the Mask Regions with Convolutional Neural Network (Mask R-CNN) [3], Single Short Multibox Detector (SSD) [4] and You Only Look Once (YOLO) techniques [5]. These techniques provide high processing speed and efficient recognition so these techniques therefore are popular and wide uses. However, they still cannot classify the similarity object that has different class. Only the research of Zhang et al. [6], who introduced the YOLOv2 technique, categorized the labels according to the types which were divided into three types, forced labels, warning signs and banned signs. As these types have different shape characteristics, YOLO technique can classify with high accuracy. If the traffic signs such as the sign “forced to turn left” and the sign “forced to turn right” have the same size, similar shapes and symbols. It will be difficult to identify traffic sign types. This problem leads to challenge research in detection and recognition of traffic signs in real time for higher efficiency.

The proposal of this paper is road sign detection and recognition of Thai traffic based on YOLOv3. The Thai Traffic Sign Dataset (TTSD) has been collected and distributed for future work. The YOLOv3 and YOLOv3 Tiny has been compared in this study for applying the real time detection and recognition. The paper is organized as follows. In the next section, related works are described. Then, Sect. 3 explains the modification network architecture of YOLOv3. Section 4 presents the data collections and ground truth generation follows by experimental results in Sect. 5. The final section is a conclusion.

2 Related Work

Recently, Convolutional Neural Networks (CNN) is one of deep learning research. CNN has been used to extract and classify objects with promising results. One of the most common problems is how to design a neural network structure. In practice, the well-known structures of ImageNet have been widely used in these research areas. These tasks were developed by many researchers as follows.

The first important task is a selective search to determine an area of an object instead of checking every pixel of an image. This work also offers an object classification system using the Scale Invariant Feature Transform (SIFT) [7] that is calculated from many color spaces. This feature extraction is called handcrafted features. The SVM is used to identify the selected objects. Girshick et al. [8] developed the Regions-based Convolutional Neural Network (R-CNN) by finding region proposals using a selective search then extracting features using the CNN instead of handcrafted features, follows by classification using SVM. Later, Grishick proposed the Fast R-CNN [9] using a selective search, and deep convolution networks to extract features, classify class and predict bounding boxes. Then, the Faster R-CNN [10] was applied by Ren et al. from the fast R-CNN. This technique composes of three parts; (1) feature extraction using CNN without selective search, (2) the Region Proposal Network (RPN) to determine the proposed objects, and (3) the classification of the proposed objects from RPN and the regression of bounding boxes. Although the Faster R-CNN is very accurate, the problem with the R-CNN family is the speed. Another improved

technique which is called YOLO. This technique can perform at the real time processing. YOLO will determine boxes and classes at once with a single stage of network. This object detector is significantly faster than the Faster R-CNN, obtaining 45 FPS on a GPU. Another structure can be seen as the extension of YOLO is the SSD. The SSD proposed to predict various boxes from many scales instead of a single scale like YOLO. This helps to solve any image problems with different scales. SSD only uses upper layer (low resolution) for detection because the bottom layer have high resolution. This therefore performs much worse for small objects. An interesting structure is the Feature Pyramid Network (FPN) developed by Lin et al. [11]. FPN applied the concept of the Faster R-CNN. Pyramid concept was used from bottom layer to upper layer in order to combines low-resolution, semantically strong features with high-resolution. FPN can extract data with high semantic content from every resolution. This principle has also been applied in YOLOv3.

From the study of real-time object detection and recognition problems that are state-of-the-art, for example, YOLOv2, SSD321, R-FCN, SSD513, DSSD513 and FPN FRCN found that the YOLO algorithm works the fastest. YOLOv3 has been used in this study for detecting and recognizing traffic signs in real-time.

You Only Look Once (YOLO) Algorithm

YOLO is a state-of-the-art, real-time object detection system. On a Pascal Titan X, it processes images at 30 FPS and has an mAP of 57.9% on COCO test-dev [12]. YOLOv3 is extremely fast and accurate in mAP measured at 0.5 IOU. YOLOv3 is on the same level as Focal Loss [13] but it is about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required.

Loss Function

The loss (\mathcal{L}) [5] consists of two parts, the localization loss (\mathcal{L}_{loc}) for bounding box offset prediction and the classification loss (\mathcal{L}_{cls}) for conditional class probabilities. Both parts are computed as the sum of squared errors. Two scale parameters are used to control how much we want to increase the loss from bounding box coordinate predictions (λ_{coord}) and how much we want to decrease the loss of confidence score predictions for boxes without objects (λ_{noobj}). Down-weighting the loss contributed by background boxes is important as most of the bounding boxes involve no instance. In the paper, the model sets $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$. The loss is calculated as shown in the equation below.

$$\mathcal{L} = \mathcal{L}_{loc} + \mathcal{L}_{cls} \quad (1)$$

$$\mathcal{L}_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \quad (2)$$

$$\mathcal{L}_{cls} = \sum_{i=0}^{S^2} \sum_{j=0}^B (1_{ij}^{obj} + \lambda_{coord} (1 - 1_{ij}^{obj})) (c_i - \hat{c}_i)^2 + \sum_{i=0}^{S^2} \sum_{ccC} (1_{ij}^{obj} (p_i(c) - \hat{p}_i(c))^2 \quad (3)$$

where,

- 1_i^{obj} : An indicator function of whether the cell i contains an object.
- 1_{ij}^{obj} : It indicates whether the j -th bounding box of the cell i is “responsible” for the object prediction (see Fig. 1).
- c_i : The confidence score of cell i , $\Pr(\text{containing an object}) * \text{IoU}(\text{pred}, \text{truth})$.
- \hat{c}_i : The predicted confidence score.
- C : The set of all classes.
- $p_i(c)$: The conditional probability of whether cell i contains an object of class $c \in C$.
- $\hat{p}_i(c)$: The predicted conditional class probability.

The loss function only penalizes classification error if an object is present in that grid cell, $1_{ij}^{obj} = 1$. It also only penalizes bounding box coordinate error if that predictor is “responsible” for the ground truth box, $1_{ij}^{obj} = 1$.

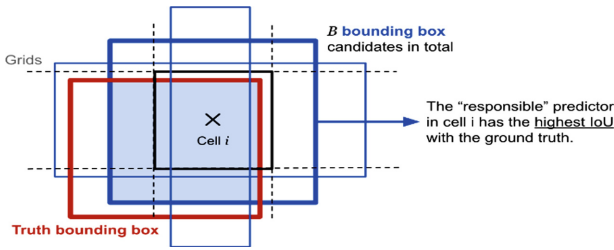


Fig. 1. At one location, in cell i , the model proposes B bounding box candidates and the one that has highest overlap with the ground truth is the “responsible” predictor.

Network Architecture of YOLO

The base model is similar to GoogLeNet with inception module replaced by 1×1 and 3×3 convolution layers. The final prediction of shape $S \times S \times (5B + K)$ is produced by two fully connected layers over the whole convolution feature map (Fig. 2).

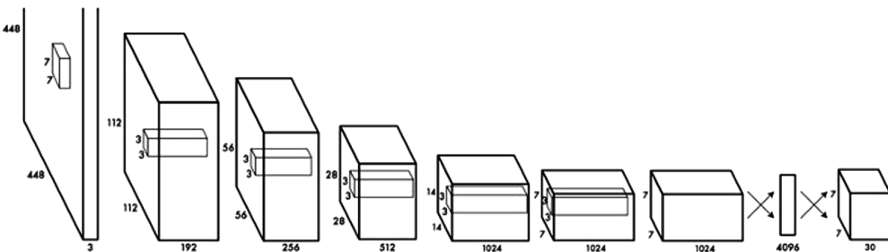


Fig. 2. The network architecture of YOLO [5].

3 The Modification Network Architecture of YOLOv3

The network structures based on YOLOv3 has been applied in this study as follows:

Determination of Neural Network Structures in YOLOv3

The network structure of this study consists of 106 neural networks. The input image size is adjusted using 416×416 pixels, batch size is 24, and learning rate is 0.001. The Darknet53.conv.74 are used for the training set as shown in Fig. 3(A).

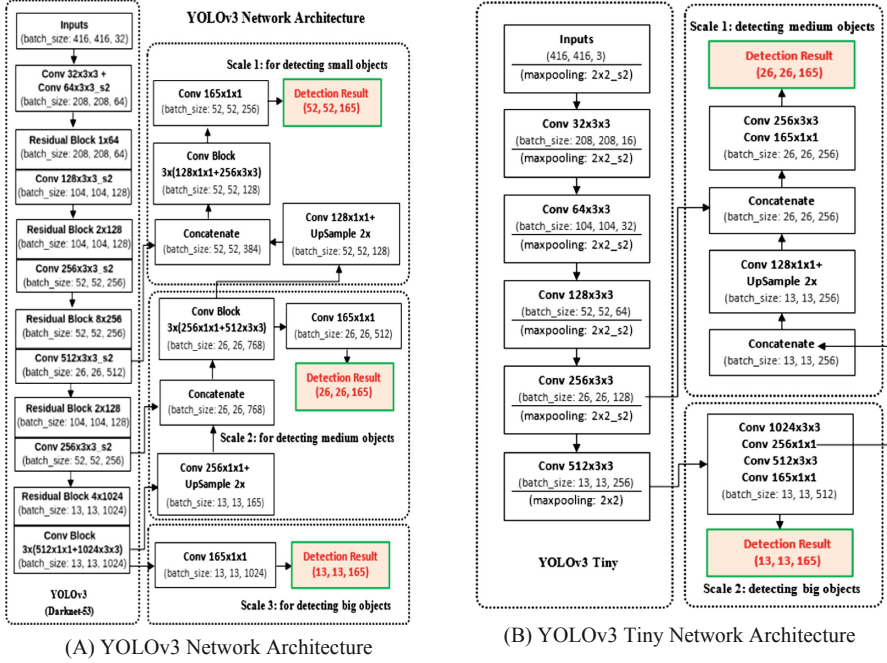


Fig. 3. YOLOv3 and YOLOv3 Tiny network architecture.

Determination of Neural Network Structures in YOLOv3 Tiny

The YOLOv3 Tiny has also been applied to compare to YOLOv3. The network structure consists of 23 artificial neural networks. The input image size is adjusted using 416×416 pixels, batch size is 24, and learning rate is 0.001. The Darknet53.conv.74 are used for this study as shown in Fig. 3(B).

4 Data Collections

The collected data was recorded from the car camera by storing video images with the resolution of 1920×1080 pixels using 60 frames per second, and a 1280×720 pixels using 30 frames per second. The recording was collected from 08:00am to

05:00pm on the route from Pathumrat District, Roi Et Province to Yang Talat District, Kalasin Province, and within the area of Mahasarakham University, Khamriang Sub-district, Kantharawichai District of Mahasarakham Province. The driving speed during video recording is between 50 and 70 km per hour, and it records in two minutes of video per a clip.

Characteristics of Collected Data: The dataset was collected as a video image, which was separated into frame by frame. The image frame of traffic signs was selected as a training set and test set. The total classes of signs is at 50. There are 200 badges in each class. The total badges of all classes is 10,000 from 9,357 images. The training set is 90% and test set is 10% by randomizing. The label of each image class is shown in Fig. 4.



Fig. 4. The 50 classes of the Thai traffic sign dataset.

The Ground Truth of Dataset: The program BBox-Label-Tool [14] was used to determine the actual area of traffic signs that need to detect. The ground truth is stored in a text file (.txt) as shown in Table 1.

Table 1. The ground truth of traffic signs.

| No. | File name | Class name | x1 | y1 | x2 | y2 |
|-------|------------------------|------------------|-----|-----|-----|-----|
| 1 | 20181220-19-00744.txt | 01-One_way | 376 | 284 | 395 | 299 |
| 2 | 20181220-19-00747.txt | 01-One_way | 348 | 285 | 370 | 303 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 9,357 | 20190109-183-06279.txt | 50-Narrow_bridge | 348 | 263 | 379 | 287 |

5 Experimental Results

The experiment was run on GPU Nvidia GeForce GTX 1060 with 6 Gb on Ubuntu 16.04 LTS. Fifty classes from 991 files were tested and the evaluation results of each detection and recognition are presented as below.

Thai Traffic Sign Detection Efficiency of YOLOv3

The result of testing was found that the detection and recognition of traffic signs with YOLOv3 obtained $mAP = 88.10\%$ (shown in Fig. 5(A)). The average time processing of an image per second is 0.0926, representing an average rate of 10.80 frames per second. This preview real-time processing is at <http://tiny.cc/1e5c6y>.

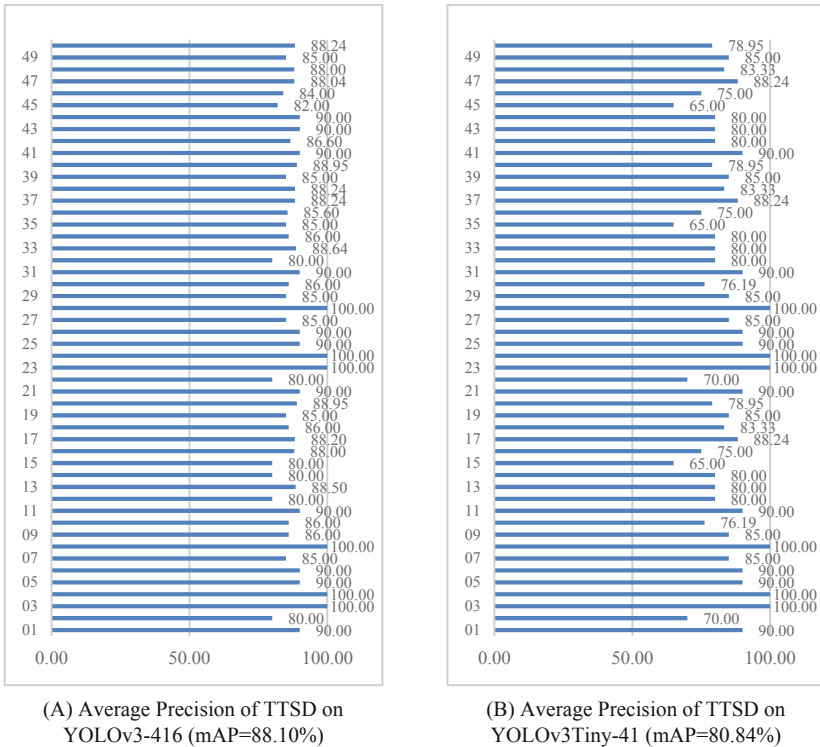


Fig. 5. A result of Thai traffic sign detection with YOLOv3 and YOLOv3 Tiny.

Thai Traffic Sign Detection Efficiency of YOLOv3 Tiny

The result of testing was found that the detection and recognition of traffic signs with YOLOv3 tiny obtained $mAP = 80.84\%$ (shown in Fig. 5(B)). The average time processing of an image per second is 0.0478, representing an average rate of 20.92 frames per second. This preview real-time processing is at <http://tiny.cc/nn5c6y>.

A comparison of the results of the Thai traffic sign detection and recognition with YOLOv3 and YOLOv3 Tiny are shown in Table 2. It is found that the detection and recognition with YOLOv3 is better performance than the YOLOv3 Tiny, while speed of YOLOv3 tiny is better than YOLOv3.

Table 2. The results of the TTSD with YOLOv3 and YOLOv3 Tiny.

| No. | Model | mAP | Processing time per images |
|-----|------------|--------|----------------------------|
| 1 | YOLOv3 | 88.10% | 10.80 fps |
| 2 | YOLOv3Tiny | 80.84% | 20.92 fps |

6 Conclusion

The objective of this research is to develop the detection and recognition with Thai traffic sign dataset in real time based on YOLOv3 technique. In terms of the efficiency of the experiment, it was found that the detection and recognition efficiency of YOLOv3 is better than YOLOv3 Tiny. However, the speed of YOLOv3 Tiny can process faster than YOLOv3. Because YOLOv3 Tiny has a number of neural network layers (23 layers) less than YOLOv3 (106 layers). In addition, we have collected a new of Thailand Traffic Signs Dataset (TTSD) that was published at <https://gitlab.com/bombomstory/ttsdb> for the future work.

Acknowledgement. We gratefully acknowledge Asst. Prof. Dr. Thawatchai Chomsiri for supporting the PC and GPU for the experiment.

References

1. Sivak, M.: Mortality from road crashes in 193 countries: a comparison with other leading causes of death (2014)
2. Yuan, Y., Xiong, Z., Wang, Q.: An incremental framework for video-based traffic sign detection, tracking, and recognition. *IEEE Trans. Intell. Transp. Syst.* **18**(7), 1918–1929 (2017)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988. IEEE (2017)
4. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
5. 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)
6. Zhang, J., Huang, M., Jin, X., Li, X.: A real-time Chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms* **10**(4), 127 (2017)

7. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, vol. 99, pp. 1150–1157 (1999)
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
9. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
11. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
12. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767) (2018)
13. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
14. Qiu, S.: A simple tool for labeling object bounding boxes in images. <https://github.com/puzzledqs/BBox-Label-Tool>. Accessed 01 Apr 2018