Penerbit Akademia Baru

Journal of Advanced Research in Computing and Applications

Journal homepage:
https://akademiabaru.com/submit/index.php/arca/index
ISSN: 2462-1927

Shape-based Single Stage Deep Neural Network for Traffic Sign Applications

Hossamelden Mohamed Elhawary[1,2,*], Mohd Ibrahim Shapiai[2], Upendra Suddamalla[3], Anthony Wong[3], Hairi Zamzuri[4]

[1] Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt
[2] Center of Artificial Intelligence & Robotics, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
[3] MooVita Pte. Ltd., Singapore
[4] eMooVit Technology Sdn Bhd., Cyberjaya, Malaysia

ARTICLE INFO

Article history:
Received 26 June 2022
Received in revised form 11 August 2022
Accepted 21 August 2022
Available online 20 September 2022

Keywords:
Object detection; Deep learning, YOLO; Artificial intelligence; Traffic sign detection and recognition

ABSTRACT

One of the main parts of the advanced driving assistant system (ADAS) is traffic sign detection and recognition, which seeks to detect and recognize street signs in real-time. However, real-world applications demanding high precision and instantaneous recall present difficulties for traffic sign identification. The tiny object size and the class imbalance are the causes of these difficulties. Recently, researchers have proposed several methods to enhance the detection quality, including adding attention techniques, spatial enhancing of small objects, and enriching the features using a multiscale network. Researchers are addressing the class imbalance by introducing different loss functions and cascaded networks. However, because of the existing techniques and systems, the current model becomes more complicated. Single-stage networks like YOLO are also impacted by the imbalance, which results in a reduced recall for tiny objects. We have proposed a new training method for a one-stage detection network called the Real Time-Shape Deep Neural Network (Real Time-Shape DNN). The YOLO detection head is expanded by our suggested method to include the four primary parameters of objectiveness, regression, class, and shape. We added an additional parameter to the loss and Non-Maximum Suppression (NMS) to reduce the class number. We train the network jointly between classes and shapes. With the German Traffic Sign Detection Benchmark (GTSDB) as the benchmark dataset, we validate our proposed methods. The findings indicate that our presented method increases average precision (AP) for Yolov4 from 69.49% to 76.12% while increasing the recall index from 88.07% to 99.30% and for Yolov4-tiny-3l increases the recall index from 84.95% to 97.73% while increasing the average precision from 46.45% to 49.44% without increasing the complexity of the primary network. In terms of recall and precision, the baseline in the German Traffic Sign Detection Benchmark dataset is not as good as our proposed method.

* *Corresponding author.*
E-mail address: hossam@aast.edu (Hossamelden Mohamed Elhawary)

1. Introduction

The industrial value and potential of traffic sign detection technology is substantial. Industry and academia have been diligently working in this field for several years. The technology for detecting traffic signs is a hot topic that has gained considerable attention. This technology is indispensable in applications like position sensing, autonomous vehicles, and road marking detection. In recent years, the development of deep learning technology has significantly advanced traffic recognition and detection technology. Deep learning has become the primary method for detecting and identifying road signs. Typical issues with traffic sign data include tiny size, noise, fluctuating light intensity, and imbalance. Similarly, traffic sign data is typically collected from natural scenes. Therefore, traffic sign data often consists of a few tiny objects. Size variation is a characteristic that poses considerable hurdles to the detection model.

Depending on their function, traffic signs can be divided into different types. Within each type, they can be further subdivided into subtypes with similar generic shapes and appearances but differing particulars. This indicates that traffic sign recognition should consist of two phases: detection and classification. The detection step utilizes shared information to provide bounding boxes that may contain traffic signs belonging to a particular category. In contrast, the categorization step utilizes differences to identify the type of sign present. We want to point out that "detection" and "classification" have different meanings in the general object recognition community, as shown by the ImageNet competition. Classification means giving an image a label instead of an object, while "detection" means finding the bounding box of an object in a certain category.

In recent years, the convolutional neural network (CNN)-based deep learning technique has significantly enhanced object detection performance. Generally speaking, traffic sign detection consists of two components: location and recognition. CNNs are generally capable of extracting image features and classifying images with an extraordinary degree of precision. CNNs are widely used in the ImageNet Large-Scale Visual Recognition Challenge (LSVRC) [1] and have become the standard approach for classifying images. Location has always been a tricky problem for object detection. The classical selective search approach is based on color, texture, size, etc., and searches for more than 2K regions for object detection, which consumes a large amount of time [2] . Region-based CNN (R-CNN) fully utilizes CNN's robust feature extraction ability [3]. The Faster R-CNN method makes direct use of CNNs' feature extraction capabilities to detect and identify objects [4]. Recently, the best method for detecting traffic signs has been an upgraded version of Faster RCNN and YOLO. All these approaches have insufficient effectiveness in detecting small objects. As the CNN does forward calculations on the image, the size of the image is gradually down-sampled, and the small object's features disappear from the high-level features. This makes it hard for the network to find small objects.

Currently, researchers are attempting to improve the extracted feature map by adding more modules to the CNN and more scales, either spatial or feature scales, such as Spatial Pyramid Pooling [5] and Feature Pyramid Network [6] (FPN), which may include a trade-off between accuracy and speed. Another approach would be to develop a new learning mechanism that employs distinct loss functions. Traffic signs have small object sizes and imbalances between classes, causing the present work to have difficulty identifying them and produce more false positives. The remainder of the paper is structured as follows: Section 2 discusses relevant research. In Section 3, we describe the proposed method based on shapes based on YOLO architecture. In Section 4, the results and discussion will be given. Finally, the conclusion will be provided in Section 5.

2. Related Work

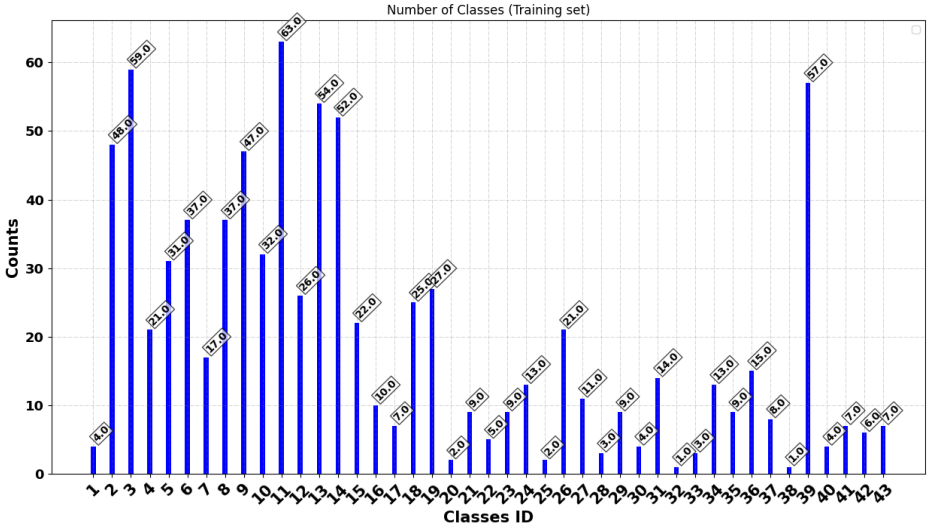
Modern traffic sign detection algorithms are primarily based on CNN. In [7], the color feature of the traffic sign is employed to extract possible regions and the fine results are generated by faster-RCNN from the possible regions. In reference, the region proposal network (RPN) network is utilized twice sequentially to identify coarse to fine traffic indicators [8]. Meng *et al.,* [9] trained a tiny object-sensitive network enhanced by a single-shot multibox detector (SSD) by subgraph division. From different points of view, these studies improve the accuracy of traffic sign detection, but they don't meet the requirement for real-time detection.

In recent years, traffic sign detection algorithms based on deep learning have produced high-quality results. Object detection is generally categorized as either single-stage or two-stage. Single-stage algorithms have a fast detection rate, such as the YOLO series [10-13] and SSD series [14]. The accuracy of two-stage object detection algorithms, such as the R-CNN series [3,4,15], is quite high. Traffic sign detection jobs must distinguish not only between the major categories of traffic signs (instructions, prohibitions, and warnings), but also between the minor categories (the meaning of each traffic sign), or their practical utility will be severely reduced. Yang *et al.,* [16] present an attention approach that relies on local context information to enhance the model's traffic sign identification. Wang *et al.,* [17] developed a high-accuracy, high-speed detector based on faster R-CNN and MobileNet, which refines the localizations of miniature traffic signs using colour and shape information.

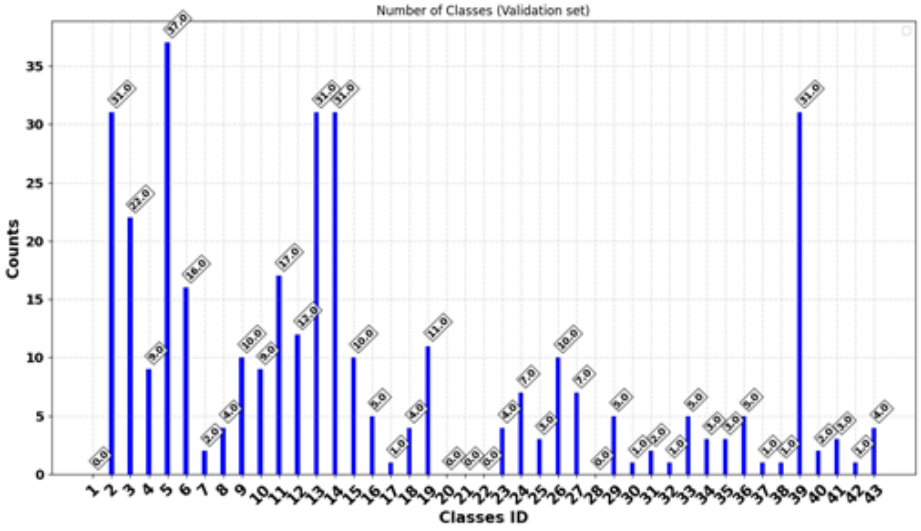
The YOLO (You Only Look Once) [10] approach is a traditional single-stage-based method that can nearly achieve real-time detection. The regional proposal technique obtains more detection precision at the expense of slower running speed, while the single-stage-based operating speed is faster. However, the detection performance of the comparative area proposal method is poor. Zhang *et al.,* [18] proposed an enhanced one-stage traffic sign detector based on YOLO-v2 by modifying the number of convolutional layers in the traditional YOLO-v2 network to make it efficient for the China Traffic Sign Dataset. Using the improved YOLOv2 algorithm, real-time detection was performed in the Chinese traffic sign data set (CTSD) [19]. However, the imbalance has an effect on one-stage networks like YOLO, resulting in lower recall performance for minor classes and remaining a difficult obstacle.

3. Methodology

Generally, we choose using GTSDB [20] dataset with the class distribution illustrated in Figure 1(a) for training set and Figure 1(b) for validation set. Unbalanced datasets make it difficult for one-stage deep networks such as YOLO to detect minor classes. In this paper, the Real Time Shape Based Deep Neural Network Real (Real Time–Shape DNN) training and inference method was proposed for a single-stage detection network. The suggested Real Time-Shape DNN is designed to learn concurrently the shape and class of each box. This allows classes with fewer samples to combine with classes with more samples, making it easier for the network to learn the common features of the group of classes during training and verify whether the predicted class is present in the group of shapes during inference.



(a)



(b)

Fig. 1. Frequency distribution per class for (a) GTSDB training set and (b) validation set

3.1 The Proposed Concept

Therefore, the shape parameter is added to the layer before the YOLO head to extend it. We select the class's shape as the additional parameter and alter the YOLO head to handle the extra parameter. Then, we jointly trained the network's classes and its shape. Thus, we ensure greater recall per shape parameter while maintaining the same precision per class. The output of the shape parameter helps in the elimination of false positives within the class, hence increasing the overall recall while virtually retaining precision and inference speed.

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{1}$$

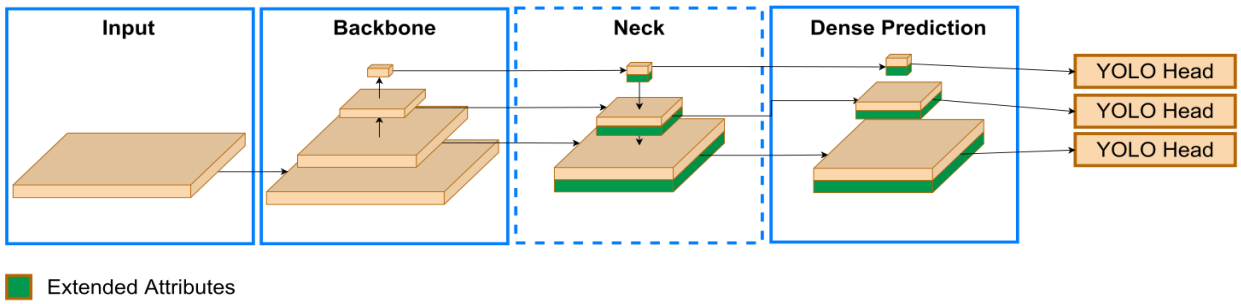


Fig. 2. Yolov4 and Yolov4-tiny-3l architecture

The architecture of Yolov4 [12] consists of four main parts, as shown in Figure 2: the input, the backbone, the neck, and the dense prediction. The input of the network is an image $608 \times 608 \times 3$ in 64 patches for training. The backbone used in Yolov4 is CSPDarknet53 and in Yolov4-tiny-3l is CSPDarknet53-tiny where they use the CSPBlock module in cross-stage partial network to divide the feature map into two parts and combine them by cross-stage residual edge to improve the network correlation. In Yolov4-tiny-3l LeakyReLU is used instead of the Mish activation function. In Yolov4, an additional module with modified PANet path-aggregation is used for the Neck Spatial Pyramid Pooling (SPP), whereas in Yolov4-tiny-3l Feature Pyramid Network (FPN) is used as the neck, taking from three different feature scales: 19×19, 38×38 and 76×76. The output channels are $B \times (4 + 1 + C)$ each. where B is the number of boxes in each scale, and each box has normalized offset coordinates, t_x, t_y, t_w, t_h, a single objectiveness score, and C is the number of classes. Finally, the dense prediction uses Yolov3 [13] heads with anchors to predict the boundary boxes, objectiveness, and class score for each scale. For each scale, the YOLO head divides the input image into $S \times S$ grid cells, predicting boundary boxes, objectiveness scores, and C class probability. The boundary box is calculated using Eq. (1) where the cell is offset from the top left corner of the image by (c_x, c_y) and the box's prior width and height are (p_w, p_h) where the box prior is predefined for each scale. The normalized offset coordinates for the ground truth can be calculated by inverting Eq. (1).

$$loss = loss_{reg} + loss_{obj} + loss_{attr} + loss_{class} \tag{2}$$

In Yolov4 and Yolov4-tiny-3l, the loss function has three primary terms for computing the loss function: regression loss $loss_{reg}$, objectiveness loss $loss_{obj}$, and class loss $loss_{class}$. To overcome the unbalance in the dataset, the shape parameter loss $loss_{shape}$ is used. as an addition to the loss function Eq. (2).

$$loss_{reg} = 1 - IoU_{pred}^{truth} + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{16}{\pi^4} \frac{\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^4}{1 - IoU_{pred}^{truth} + \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)} \tag{3}$$

The regression loss is using the CIoU [21] loss function Eq. (3) to consider the three important factors; the overlapping area, the distance between central points and the aspect ratio where IoU_{pred}^{truth} is the intersection over union between the ground truth bounding box and prediction bounding box, $\rho^2(b, b^{gt})$ is the Euclidean distance between the predicted box center b and the ground truth center b^{gt}. c is the diagonal distance of the intersection area between the ground truth and the predicted bounding box.

$$C_{ij} = P_{ij} * IoU_{pred}^{truth} \tag{4}$$

$$\begin{aligned} loss_{obj} = &-\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\hat{C}_{ij} log\left(C_{ij}\right) + \left(1-\hat{C}_{ij}\right) log\left(1-C_{ij}\right)\right] \\ &- \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\left(1-I_{ij}^{obj}\right)\left[\hat{C}_{ij} log\left(C_{ij}\right) + \left(1-\hat{C}_{ij}\right) log\left(1-C_{ij}\right)\right] \end{aligned} \tag{5}$$

The objectiveness loss $loss_{obj}$ uses the cross-entropy loss for the grid cells with and without objects as shown in Eq. (5) where $C^{ij}, \hat{C}_{ij}$ are the predicted and ground truth objectiveness confidence score, I_{ij}^{obj} is equal to one if there is an object in the j^{th} bounding box and i^{th} grid cell otherwise it will be zero and λ_{noobj} is weight parameter for non-objects. The object in the i^{th} grid cell and j^{th} box is considered as an object if the objectiveness confidence score is greater than a certain threshold otherwise it will be considered as not an object. The objectiveness confidence score is calculated as shown in Eq. (4) where the C_{ij} is the objectiveness confidence score of the j^{th} box in the i^{th} grid cell, P_{ij} is the objectiveness probability in j^{th} box and i^{th} grid cell and IoU_{pred}^{truth} is the intersection over union between the ground truth bounding box and the predicted bounding box.

$$loss_{class} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\sum_{c=0}^{C}\left[\hat{p}_{ij}(c) log\left(p_{ij}(c)\right) - \left(1-\hat{p}_{ij}(c)\right) log\left(1-p_{ij}(c)\right)\right] \tag{6}$$

Finally, the class loss $loss_{class}$ is as in Eq. (6), the binary cross-entropy with logistic activation where $p_{ij}(c), \hat{p}_{ij}(c)$ are the predicted probability and the ground truth label of class c of the j^{th} box and i^{th} grid cell and C is the total number of classes.

$$loss_{shape} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\sum_{a=0}^{A}\left[\hat{v}_{ij}(a) log\left(v_{ij}(a)\right) - \left(1-\hat{v}_{ij}(a)\right) log\left(1-v_{ij}(a)\right)\right] \tag{7}$$

We added extra parameter to the neck and the dense prediction as shown in Figure 2 by extending the channel to be $B \times (4 + 1 + C + A)$ where A is the number of the added shape parameters, and makes the total loss as in Eq. (2) where $loss_{shape}$ is the shape loss. In this study we choose to group the classes with shape, but it can be any parameter. The shape loss as in Eq. (7) is also using binary cross-entropy with logistic activation function for each added shape a where $v_{ij}(a), \hat{v}_{ij}(a)$ are the predicted and ground truth shape repeatedly in the j^{th} box and i^{th} grid cell.

During evaluation, we use the extra added parameter shape to filter out all the predicted classes that do not belong to the shape category. First, we calculate the confidence score of the class is calculated as $obj_{ij}^{score} * (0.1 * v_{ij}(c \rightarrow a) + 0.9 * p_{ij}(c))$ where the obj_{ij}^{score} is the objectiveness score of the j^{th} box and i^{th} grid cell, v_{ij} is the predicted score of shape, $c \rightarrow a$ is a mapping function that map the class index to the shape index and $p_{ij}(c)$ is the predicted score of the class c. Then we modified the DIoU non-maximum suppression (NMS) to sort detection based on the calculated class confidence score. Second, all the boxes of classes and shapes with a DIoU value greater than 0.6 are eliminated.

3.2 Dataset

We validate our method using GTSDB. GTSDB has 741 images and 1213 annotated traffic signs, with each image cropped to a size of 1360×800 pixels and divided into 506 for training and 235 for validation, with 43 class categories. We use all 43 class categories for training and then filter out those with no instances in the validation set, leaving 38 categories used in validation. Even though the dataset is unbalanced, we have not applied any augmentation techniques to rectify this. Simply, we use it unbalanced. Only online color jitter has been used throughout training. Each traffic sign class is mapped to one of the following five shapes: circle, triangle, diamond, flipped triangle, and octagon, with the majority of classes being circular-shaped.

3.3 Parameter setup

The training hyper-parameters that have been used are shown in Table all training was using two GPUs but inference was using only one GPU.

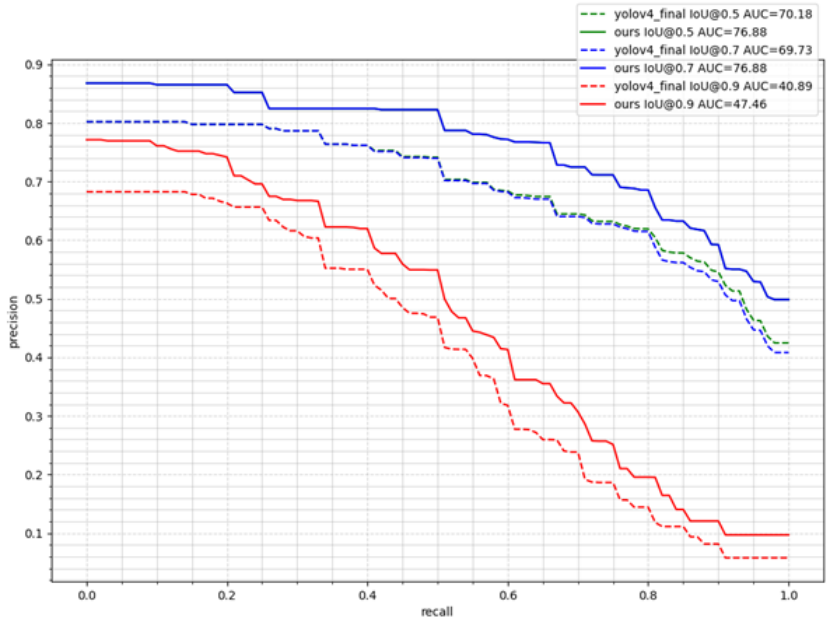
Table 1
Training Hyper Parameters

Parameter	Yolov4-tiny-3l	Yolov4
Pre-trained Model	COCO until layer 29	COCO until layer 137
Max batches	21758	21758
Learning Rate	0.0013	0.0005
Steps	17406, 19582	17406, 19582
Scales per step	0.1, 0.1	0.1, 0.1
Optimizer	SGD	
Momentum	0.9	0.949
Weight decay	0.0005	0.0005
batch	64	64
Subdivisions	2	32
Width × Height	608×608	
Classes	43	
IoU normalizer	0.07	
IoU Loss	CIoU	
Ignore box threshold	0.7	
NMS	DIoU NMS beta 0.6	

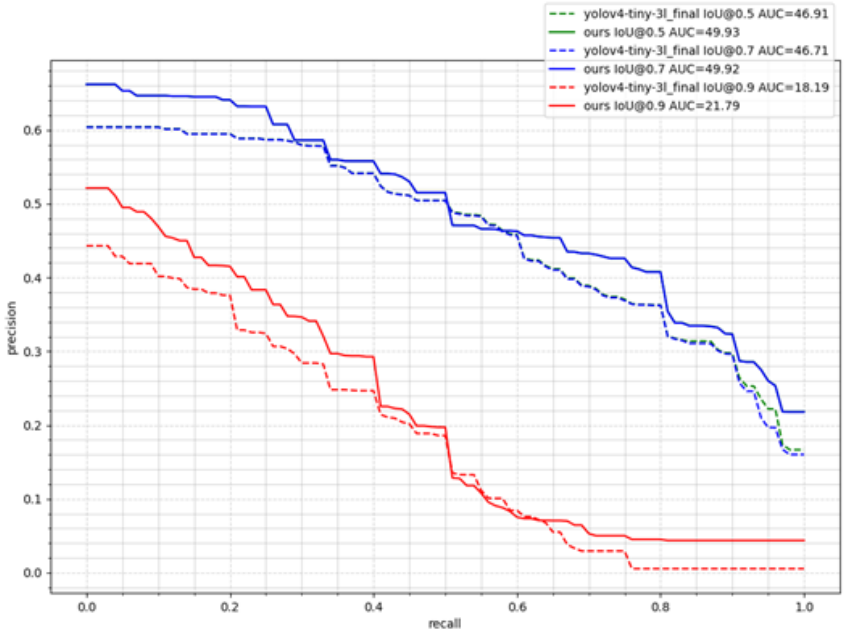
3.4 Performance Measure

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}$$
$$AP = \frac{1}{N}\sum^{C} P\,, AR = \frac{1}{N}\sum^{C} R\,, AUC = \sum_{i=1}^{101} PR_i \tag{8}$$

In this study, precision (P), recall (R), average precision (AP), average recall (AR), and area under the curve (AUC) are used as formula in Eq. (8), where C is the number of classes, N is the total number of images, and PR is the precision recall for 101 recall levels. We use an IoU threshold of 0.5 and a confidence threshold of 0.25.



(a)



(b)

Fig. 1. Precision Recall curve at 0.5, 0.7 and 0.9 IoU compare between baseline and ours for Yolov4 and Yolov4-tiny-3l

4. Results and Discussion

All our experiments were trained using 2x NVIDIA GPUs V100 with 16GB HBM2 memory. CPU IBM POWER9 PowerNV 8335-GTH. We evaluate using Microsoft Common Objects in Context (MS COCO) evaluation metrics as proposed in [22]. We use our modified darknet framework to implement our proposed idea. We evaluate our proposed idea on YOLOv4-tiny-3l and YOLOv4 with precision (P), recall (R), average recall (AR), average precision (AP), frames per second (FPS), and Billion Floating-point Operations (BFLOPS) in the quantitative evaluation in this section to show the performance comparison.

4.1 Results

Table 2
Evaluation comparison between our and baseline confidence threshold of 0.25 on GTSDB

	AP	AR	FPS	BFLOPS
Yolov4-tiny-3l	46.45%	84.95%	422	17.469
Our Yolov4-tiny-3l	49.44%	97.73%	400	17.5
Yolov4	69.49%	88.07%	77	127.915
Our Yolov4	76.12%	99.30%	77	127.977

As shown in Table 2, our modified model in both Yolov4 and Yolov4-tiny-3l obtains a superior average recall at IoU 0.5 with about the same FPS, where it improves yolov4-tiny-3l from 84.95 % to 97.73 % while increasing the average precision from 46.45 % to 49.44 %. Regarding Yolov4, our model raises the average recall from 88.07 % to 99.30 %, while increasing the average precision from 69.49 % to 76.12 % at an IoU of 0.5.

The dashed line in Figure 3 represents the baseline model, while the solid line represents our model. With a recall index of less than 0.2, our method improves the precision for Yolov4 in Figure 3(a) by 7%. From recall index 0.25, the improvement in precision is 3% and increases to a high of 9% at recall index 0.65, before remaining at 7% for both IoU values of 0.5 and 0.7 at recall index 0.8. However, for IoU 0.9, the gain in precision is 9% until the recall index hits 0.1, while the improvement of 7% is maintained. Precision fluctuates between 4% and 8% improvement from recall index 0.1 to 0.8, reaches a minimum of 1% improvement at recall index 0.85, and then plateaus at 4% improvement until recall index 1.0. In contrast, Figure 3(b) demonstrates that our model improves the precision of yolov4-tiny-3l by 6% and 4% from the recall index 0.0 to 0.25, whereas it did not improve from the recall index 0.3 to 0.6. Furthermore, our model continues to improve from the recall index above 0.6 to 1.0 with a maximum improvement of 6% at the recall index of 0.75 and a minimum improvement in precision of 2% for both IoU values of 0.5 and in contrast to the IoU 0.9, which exhibits a maximum improvement of 8% in precision between recall indices of 0.4 and 0.65, our model exhibits no improvement over the baseline. Then, our model increases from a recall index of 0.65 to 1.0 with a consistent 3% increase from a recall value of 0.75 to 1.0.

As shown in Table 3, the precision and recall per class for IoU 0.5 where shape C represents a circle, T represents a triangle, FP represents a Flip Triangle, O represents an octagon, and D represents a diamond. In Yolov4-tiny-3l (YT3L), our method has 21 class greater precision with two ties and 15 class higher recall with 21 ties compared to the baseline (BL), which has 15 class higher precision and two classes higher recall. Yolov4 baseline (BL) is higher in precision by 6 classes compared to our method, which is superior in precision by 26 classes with 6 ties and in recall index by 10 classes with 28 ties.

4.2 Discussion

(a) Sample 1 (c) Sample 3

(b) Sample 2 (d) Sample 4

Fig. 2. Detection result comparison between left the baseline and right our method for four sample images

In Figure 4(a), the baseline model yolov4-tiny-3l predicts that the traffic sign belongs to the class id 5 circular group. Our method not only eliminates it but also predicts the proper class 13, as the additional shape parameter enables the network to assign a lower score for any circular shape and a higher score for any diamond shape, resulting in improved precision and recall. In Figure 4(b), although while the baseline of yolov4-tiny-3l recognizes it as class id 39, which corresponds to the circle shape, our method successfully eliminates the false detection because the circle-shaped object has a low score, our method was unable to predict any triangular shapes that match the object class.

On the other hand, none of the model yolov4's baseline predictions mismatch the class shape in the validation set. Nevertheless, our technique improves overall precision and recall for yolov4 as well. In Figure 4(c), the yolov4 baseline failed to recognize the object class id 24, but our technique not only discovered the class id 24 but also did it with 95 percent confidence. In Figure 4(d), the baseline yolov4 identifies the object as belonging to class id 42 when it should be class id 33. As seen in Table 3, class id 33, our model is more precise than the baseline in yolov4, which explains why our model was able to predict the correct class id of 33 with a high degree of confidence (92%) in sample 4.

Table 3
Evaluation comparison per class Precision (P) and Recall (R) between ours and baseline (BL) IoU@0.5 on GTSBD

Class ID			2	3	4	5	6	7	8	9	10	11	12	13	14
Shape			C	C	C	C	C	C	C	C	C	C	T	D	FT
YT3L	Our	P	0.85	0.57	0.23	0.66	0.55	1.00	0.56	0.93	0.86	0.88	0.65	0.98	0.96
		R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97
	BL	P	0.79	0.62	0.44	0.56	0.37	1.00	0.26	0.80	0.87	0.87	0.69	0.95	0.97
		R	0.97	1.00	1.00	0.97	0.88	1.00	1.00	1.00	1.00	1.00	0.92	0.97	1.00
Yolov4	Our	P	0.94	0.83	0.77	0.91	0.98	1.00	0.88	0.98	0.90	0.90	0.88	1.00	1.00
		R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	BL	P	0.93	0.83	0.60	0.88	0.91	1.00	1.00	0.96	0.84	0.89	0.83	1.00	1.00
		R	0.94	1.00	0.89	0.97	1.00	1.00	1.00	1.00	1.00	0.94	1.00	1.00	1.00
Class ID			15	16	17	18	19	23	24	25	26	27	29	30	31
Shape			O	C	C	C	T	T	T	T	T	T	T	T	T
YT3L	Our	P	0.84	0.84	0.03	1.00	0.62	0.16	0.41	0.08	0.36	0.11	0.17	0.06	0.53
		R	0.90	1.00	1.00	1.00	0.82	1.00	1.00	1.00	1.00	0.71	1.00	1.00	1.00
	BL	P	0.89	0.74	0.04	1.00	0.54	0.05	0.26	0.18	0.25	0.20	0.05	0.13	0.29
		R	1.00	0.80	1.00	1.00	0.82	0.75	0.71	1.00	0.90	0.71	0.60	1.00	1.00
Yolov4	Our	P	1.00	0.97	1.00	1.00	0.75	0.41	0.98	0.13	0.92	0.74	0.51	1.00	0.67
		R	1.00	1.00	1.00	1.00	0.91	1.00	1.00	1.00	1.00	0.86	1.00	1.00	1.00
	BL	P	1.00	1.00	1.00	1.00	0.68	0.23	0.82	0.02	0.90	0.70	0.51	0.13	0.65
		R	1.00	1.00	1.00	1.00	0.82	0.75	1.00	0.33	1.00	0.86	1.00	1.00	1.00
Class ID			32	33	34	35	36	37	38	39	40	41	42	43	
Shape			T	C	C	C	C	C	C	C	C	C	C	C	
YT3L	Our	P	0.02	0.35	0.51	0.67	0.54	0.00	0.13	0.90	0.03	0.64	0.01	0.11	
		R	1.00	0.80	1.00	1.00	1.00	1.00	1.00	0.94	1.00	1.00	1.00	1.00	
	BL	P	0.00	0.52	0.23	0.67	0.83	0.00	0.17	0.90	0.00	0.44	0.04	0.04	
		R	0.00	0.60	1.00	1.00	1.00	0.00	1.00	0.94	0.00	1.00	1.00	0.75	
Yolov4	Our	P	0.08	0.94	0.92	1.00	0.90	0.08	0.02	0.96	0.70	0.54	0.00	0.71	
		R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00	1.00	1.00	1.00	
	BL	P	0.00	0.79	0.56	0.83	0.94	0.00	0.00	0.96	0.83	0.50	0.20	0.50	
		R	0.00	1.00	1.00	1.00	1.00	0.00	0.00	0.97	1.00	1.00	1.00	1.00	

5. Conclusions

This research provides a shape-based parameter to enhance the YOLO head model. The proposed shape parameter enhances the detection network by reducing the total number of classes from 43 to five. In the loss function-based YOLO architecture for traffic sign detection, the shape term is introduced as a term. The addition of shape information as a parameter enhances the precision and recall performance of the model. On the GTSDB dataset, the proposed model is validated and compared to the baseline technique YOLO and tiny YOLO. It provides significant research value and reference value for automatic driving and traffic signal detection.

In the future, we will examine the benefits of grouping various criteria that can improve our system to avoid more false detections, particularly for similar traffic signs such as speed limits.

Acknowledgement
This paper is funded by Ministry of Higher Education (MOHE) under Fundamental Research Grant Scheme (FRGS), Registration Proposal No: FRGS/1/2022/ICT11/UTM/02/2 (Attention-Based Fully Convolutional Networks for Lane Detection of Different Driving Scenes - No Vote: R.K130000.7843.5F563)

References

[1] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. "Imagenet large scale visual recognition challenge." *International journal of computer vision* 115 (2015): 211-252. https://doi.org/10.1007/s11263-015-0816-y

[2] Uijlings, Jasper RR, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. "Selective search for object recognition." *International journal of computer vision* 104 (2013): 154-171. https://doi.org/10.1007/s11263-013-0620-5

[3] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "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. https://doi.org/10.1109/CVPR.2014.81

[4] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[5] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition." *IEEE transactions on pattern analysis and machine intelligence* 37, no. 9 (2015): 1904-1916. https://doi.org/10.1007/978-3-319-10578-9_23

[6] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117-2125. 2017. https://doi.org/10.1109/CVPR.2017.106

[7] Yang, Tingting, Xiang Long, Arun Kumar Sangaiah, Zhigao Zheng, and Chao Tong. "Deep detection network for real-life traffic sign in vehicular networks." *Computer Networks* 136 (2018): 95-104. https://doi.org/10.1016/j.comnet.2018.02.026

[8] Lu, Yifan, Jiaming Lu, Songhai Zhang, and Peter Hall. "Traffic signal detection and classification in street views using an attention model." *Computational Visual Media* 4 (2018): 253-266. https://doi.org/10.1007/s41095-018-0116-x

[9] Meng, Zibo, Xiaochuan Fan, Xin Chen, Min Chen, and Yan Tong. "Detecting small signs from large images." In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 217-224. IEEE, 2017. https://doi.org/10.1109/IRI.2017.57

[10] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "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. https://doi.org/10.1109/CVPR.2016.91

[11] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271. 2017. https://doi.org/10.1109/CVPR.2017.690

[12] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).

[13] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[14] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21-37. Springer International Publishing, 2016. https://doi.org/10.1007/978-3-319-46448-0_2

[15] Girshick, R. "Fast r-cnn.,(pp. 1440–1448)." *Google Scholar Google Scholar Digital Library Digital Library* (2015). https://doi.org/10.1109/ICCV.2015.169

[16] Yang, TingTing, and Chao Tong. "Real-time detection network for tiny traffic sign using multi-scale attention module." *Science China Technological Sciences* 65, no. 2 (2022): 396-406. https://doi.org/10.1007/s11431-021-1950-9

[17] Wang, Fei, Yidong Li, Yunchao Wei, and Hairong Dong. "Improved faster rcnn for traffic sign detection." In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/ITSC45102.2020.9294270

[18] Zhang, Weiwei, Xiaolin Song, and Xiaojuan Cao. "Multiscale matched filter-based lane detection scheme of the constant false alarm rate for driver assistance." *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 229, no. 6 (2015): 770-781. https://doi.org/10.1177/0954407014547929

[19] Zhang, Jianming, Manting Huang, Xiaokang Jin, and Xudong Li. "A real-time chinese traffic sign detection algorithm based on modified YOLOv2." *Algorithms* 10, no. 4 (2017): 127. https://doi.org/10.3390/a10040127

[20] Houben, Sebastian, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark." In *The 2013 international joint conference on neural networks (IJCNN)*, pp. 1-8. Ieee, 2013. https://doi.org/10.1109/IJCNN.2013.6706807

[21] Zheng, Zhaohui, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. "Distance-IoU loss: Faster and better learning for bounding box regression." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, pp. 12993-13000. 2020. https://doi.org/10.1609/aaai.v34i07.6999

[22] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740-755. Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-10602-1_48