Paper No: SC-06 **Smart Computing**

Automatic road traffic signs detection and recognition using 'You Only Look Once' version 4 (YOLOv4)

W. H. D. Fernando*
Department of Mathematics
Eastern University, Sri Lanka
harshadfernando@gmail.com

S. Sotheeswaran
Department of Mathematics
Eastern University, Sri Lanka
sotheeswarans@esn.ac.lk

***Abstract* - This paper presents an approach to detect traffic signs using You Only Look Once version 4 (YOLOv4) model. The traffic sign detection and recognition system (TSDR) play an essential role in the intelligent transportation system (ITS). TSDR can be utilized for driver assistance and, eventually, driverless cars to reduce accidents. When driving an automobile, the driver's attention is usually drawn to the road. On the other hand, most traffic signs are situated on the side of the road, which may have contributed to the collision. TSDR allows drivers to view traffic sign information without having to divert their attention. Due to the existence of a large background, clutter, fluctuating degrees of illumination, varying sizes of traffic signs, and changing weather conditions, TSDR is an important but difficult process in intelligent transport systems. Many efforts have been made to find answers to the major issues that they face. The objective of this study addresses road traffic sign detection and recognition using a technique that initially detects the bounding box of a traffic sign. Then the detected traffic sign will be recognized for usage in a speeded-up process. Since safe driving necessitates real-time traffic sign detection, the YOLOv4 network was employed in this research. YOLOv4 was evaluated on our dataset, which consisted of manual annotations to identify 43 distinctive traffic signs classes. It was able to achieve an average recognition accuracy of 84.7%. Overall, the work adds by presenting a basic yet effective model for real-time detection and recognition of traffic signs.**

Keywords* - *Intelligent Transport systems, Traffic sign Detection, YOLOv4

I. INTRODUCTION

Traffic Sign Detection and Recognition (TSDR) is a critical work because detecting and accurately identifying traffic signs can alert drivers and pedestrians to the regulations they must observe, reducing the frequency of reckless accidents and, in some cases, deaths [1]. Due to factors such as different perspectives, degraded/damaged or discolored traffic signs, illumination on the traffic sign, and motion blur, traffic sign identification and recognition is a difficult process. The challenges of detection and classification of traffic signs are shown in Figure 1.

Fig. 1. Challenges of detecting traffic signs employing different lighting conditions, deformed signs, andvariation of illumination

Traditional approaches including Bag of features methods and Regional Convolutional Neural Networks were used for the detection of traffic signs in the past but were discarded due to the poor performances produced by those approaches compared with the newer approaches.

In this paper, we used the You Only Look Once version 4 (YOLOv4) technique to detect and recognize traffic signs. YOLOv4 is a state-of-the-art approach for detecting visual objects in a real-time environment. A dense block, a dense net, and CSPDarknet53 form the backbone of YOLOv4. A YOLOv4 model's neck is made up of feature pyramid networks and a spatial pyramid pooling layer. Finally, the output is generated by the Dense prediction layer. YOLOv4 has dense prediction at layers 139,150 and 161. These layers contribute directly to the ultimate output and their combined results are obtained [2].

The remainder of this article is laid out as follows. In section II, there are summaries of various methods used in previous works related to detection and recognition. The perspective on the terminologies used in this work is covered in section III. The fourth section is devoted to a detailed explanation of the proposed strategies. The experimental environment and testing results on traffic sign detection and recognition in section V. The suggested solution is discussed and concluded with future extensions in section VI.

II. PREVIOUS WORK

In [3], authors have used a YOLO network to detect and identify Vehicles, trucks, pedestrians, traffic signs, and traffic lights. Traffic signs were then submitted to a CNN, which further categorized them into one of 75 groups. The entire solution was built on a pre-trained YOLO v3 model for class detection, whereas a CNN was trained from scratch and excellent results were displayed on input images for the classification. Detected Traffic signs were cropped and fed into the CNN for classification. They have obtained a classification accuracy of 99.2% for detected traffic signs in various weather conditions. The Berkley Deep Drive Dataset was used to train the YOLO network. The Belgian TS Dataset and the German Traffic Sign Recognition Benchmark have been compiled into a single large dataset with over 120000 images of traffic signs which were then divided into 75 categories. Images were augmented by performing Gaussian Blur, Median Filter, Max Filter, Min Filter, and some simple image rotations. Filtering false expected bounding boxes with coefficients less than 0.5 was achieved using the non-max suppression algorithm. In this study, they used three CNNs. YOLO v3 for object detection and localization, another CNN for a

vehicle, truck, pedestrian, traffic sign, and traffic light classification, and a third CNN for traffic sign classification into 75 classes. Using three CNNs has led to the increase in computational cost while training the models and during realtime object detection.

In [4], authors have proposed a novel YOLOv3 architecture. On pictures, real-time detection with mean average precision (mAP) exceeding 88% has been demonstrated. The model was trained on a broad dataset of 200 different classes. The testing set consisted of 25% of images from the total number of images. As part of the image augmentation process, randomized placement of narrowly cropped traffic signs was done, as well as distortions such as changes to the shape, scale, luminance, and contrast on the training photos. YOLOv3 detection was based on a publicly accessible implementation based on the Darknet network. Weights that were pre-trained on the ImageNet database were used as the initial weights for the model. The number of filters in YOLOv3 or Tiny YOLO's final layer was increased to enable the detection of 200 classes. The learning rate was set at 0.001 and reduced every 15000 iterations, with the input picture size set to 608 608 pixels. After 10000 repetitions in Tiny YOLO, the learning rate was decreased. Both models were trained over 400 epochs on a machine with two 1080ti GPU. A predefined threshold of value 50 was used to calculate Intersection over Union. An accuracy(mAP) of 84.1% was achieved without using image augmentation and 88.1% mAP was obtained by using image augmentation on the YOLOv3 model, and an accuracy(mAP) of 72.1% was achieved without using image augmentation and 71.3% mAP with using image augmentation on the tiny Yolo model. Results have proven that when compared to Tiny YOLO, YOLOv3 was much more precise. Non-maxima-suppression algorithm was used to eliminate unwanted detections and double bounding boxes. YOLO v3 had a lesser number of hidden layers compared to yolo v4. Therefore, yolov4 had better detection accuracy. The time it took to train a YOLO v3 model was about two weeks. Although YOLO models provided greater accuracy and real-time performance, the training time complexity was significant.

In [5], authors have introduced a traffic sign recognition approach based on deep learning, with the primary goal of detecting and classifying circular signs. Initially, images were preprocessed to highlight key details to increase detection and classification accuracy. Image Enhancement, color space conversion from RGB (Red, Green, and Blue) to HSV (Hue, Saturation and value) image noise filtering using mean and median filters were included in Preprocessing stage. The hough transform and segmentation were used to detect and locate traffic sign regions. Morphological operation Opening was used to reduce the noise introduced by segmentation. Finally, deep learning was used to classify the detected road traffic signs. A basic CNN of lent-architecture was used with two convolutional layers with a kernel size of 5×5, step one, and ReLU activation function which was able to learn complex features, two pooling layers with 2×2 kernel size, and two fully connected layers which contained 512 and 128 hidden nodes respectively. Finally, there were 43 hidden nodes in the output layer. The learning rate was set to 0.0001 at the beginning. German Traffic sign Recognition Benchmark (GTSRB) was used and the accuracy of detected circular symbols was 98.2%. The entire dataset was split into two parts, a training set, and a testing set. 90% of the dataset was considered as the training set, while the remaining 10% was taken as the test set.

In [6], authors have proposed a method that addressed the problems of low detection and recognition accuracy of distant, small traffic signs and traffic signs which were affected by weather and illumination changes. YOLOv2 was used in real-time which had a fast-processing speed and few false detections to achieve the above goal. RGB images were obtained and used as input to the Yolo network. 22 convolutional layers and five pooling layers were used to build the YOLO v2 network. Each batch on the proposed system used a randomly selected image size from a selection of five. The YOLO network was used to estimate the bounding box and conditional class likelihood of each region in the input photos. Various image sizes were used to train a model that is resilient to scale shifts. A traffic sign dataset of 16 different types of traffic signs and 7160 annotations were created with an image size of 1093×615 pixels in JPEG format. The data volume was increased in this experiment by conducting high contrast, low contrast, noise, and flip horizontal data augmentations, which improved the generalization accuracy. Clear weather, night, and small objects were used in the test dataset which consisted of 123,241 and 140 images, respectively. During the test, 16 different kinds of traffic signs were discovered. An accuracy of 66.4 % and 60.0% was achieved as a result of data augmentation and training with various image sizes.

In [7], using cascade classifiers that were trained on HOG features authors have introduced a methodology to detect traffic signs. A CNN was used which ensured all traffic signs were identified. The CNN model was used to decide whether the candidate zone contained any traffic signs. The final decision was taken at the final stage of the cascade classifier. Image preprocessing was included in the HOG feature extraction to convert the image into a grayscale image. Then the gamma correction algorithm was used to normalize the grayscale image. Gradient components and ordinate coordinates were obtained separately by using Sobel and other edge detection filters with the original image. Cell segmentation and gradient histogram calculation was achieved by segmenting the image into several cells of the same size and counting the cell unit from the histogram. After that, several feature vectors were extracted, and cell units were grouped into a larger interval and feature vectors were superimposed to obtain the HOG features of the interval. Overlapping intervals were gathered and merged to get the final HOG features. Three convolutional layers, two max-pooling layers, and two fully connected layers were used to make the CNN model which was proposed in this study. 5×5, 3×3, and 3×3 filters kernels were used by each convolutional layer respectively.300 and 42 nodes were present in each fully connected layer. A CNN was adopted to extract object features while HOG-CNN was trained to acquire candidate object regions. Weight sharing was not performed among the nine regression variables. Therefore, bounding boxes of multiple scales were predicted using HOG-CNN. The dataset was divided into a training set and a testing set where three-fourth of the images in the dataset

was used for the training purposes while the remaining was used for testing. An accuracy of 90.12% was achieved was the detection rate on video.

III. BACKGROUND

A. Object detection

Object detection is the process of locating objects which are present in an image and marking the detected object coordinates by using a bounding box. Object detection is a technique for determining the location of objects in an image [8].

B. Bounding Box prediction

A bounding box is a method of representing a specific part of an image, such as an object within a region of interest. A bounding box is a rectangular box that surrounds an object. It's usually expressed as an array of coordinate pairs, with the first pair corresponding to the x and y-axis coordinates in the upper-left corner and the second pair corresponding to the x and y axis coordinates in the lower-right corner [8].

C. Intersection over Union

Intersection over Union is a way of measuring the precision of an object detector on a given dataset [9]. The Intersection over Union is calculated using the ground-truth bounding boxes and the projected bounding boxes from the used model [9].

D. Non-maximum Suppression

To reduce redundant bounding boxes of an object, many object detection systems employ the non-maximum suppression processing approach. When non-maximum suppression is utilized, the number of detections in a frame is limited to the total number of objects [10].

E. You Only Look Once (YOLO)

"You only look once" (YOLO) is an object detection system that uses a deep neural network as its foundation andis designed to detect general objects quickly and accurately. The YOLO detector has excellent detection efficiency and a short detection time. At the same time, it generates various anchor boxes and confidence scores for those boxes [5]. During training, YOLO considers the whole image, allowing it to consider contextual details about objects. YOLO breaks the input image into square grids and then estimates how many bounding boxes each grid will have. A confidence level is calculated for each bounding box to determine the likelihood that it contains an object. The object's class is then estimated using a conditional class likelihood for each grid containing an object. During testing, conditional class probabilities and box confidences are combined to convey the chance of a class existing in the box as well as the accuracy with which the box fits the object [5]. There are multiple versions of YOLO.

- YOLOv1 [11] comprised two fully connected layers for likelihood prediction and 24 convolutional layers for extracting features.
- YOLOv2 [12] had the potential to train on large datasets and detect small objects with greater accuracy.
- YOLOv3 [13] architecture had 106 layers including residual blocks, skip connections, and upsampling, which had a slower detection speed compared with the other versions. YOLOv3 detects at three distinct scales and from three separate network places, as well as a larger number of border boxes [5]. A simpler variant, known as Tiny YOLO, with a total of 22 layers, can be used for faster detection at the expense of lower detection accuracy [5]. Previous studies used YOLOv3 models, which had a lower detection rate, a larger computational cost, and a lower real-time performance.
- YOLOv4 [2] is an object detector that can be trained with a smaller mini-batch scale on a single GPU. This allows a single GPU to train an extremely fast and reliable object detector.

IV. METHODOLOGY

First, we manually labeled the dataset that was utilized to train the YOLOv4 detector for this study using the labeling image annotation tool and uploaded it to Google Drive. Next, a YOLOv4 model was trained on Google collaborators using the annotated dataset. The RGB images in our annotated dataset were not subjected to any form of preprocessing during model training. This model generates cropped photos of identified traffic signs, which are saved to Google Drive. The model was trained for 10000 epochs and achieved an average accuracy of 84.7%.

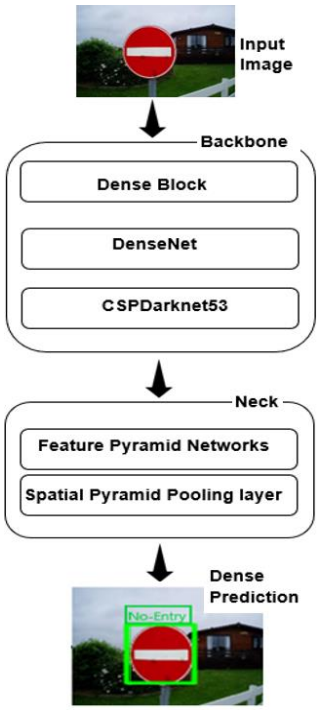


Fig. 2. The architecture of the YOLOv4 network

YOLOv4 considers the entire image during training, allowing it to consider contextual characteristics about objects. YOLOv4 divides the source image into rectangular grids and calculates the number of bounding boxes in each grid. For each bounding box, a confidence level is calculated to evaluate the possibility that it contains an item. For each grid containing an item, the object's class is then estimated using a conditional class probability.

Conditional class probabilities and box confidences are combined during testing to encode both the likelihood of a class being in the box and how well the box matches the object [5].

The overall framework of YOLOv4 is illustrated in Figure 2. The backbone architecture was used to describe the feature extraction architecture. YOLOv4's backbone was CSPDarknet53. A Dense block in the YOLOv4 backbone features multiple convolution layers, each of which has batch normalization, ReLU, and convolution. Dense Net is made up of many dense blocks connected by convolution and pooling layers in the middle. The Dense Block's input feature maps are separated into two parts by Cross-Stage-Partial connections (CSP), one of which will travel through a block of convolutions and the other will not. Following that, the outcomes are combined. This approach is used in the CSPDarknet53 backbone design.

FPN is a prominent methodology for producing object detection predictions at several scale levels. FPN up samples the preceding top-down stream and adds it with the adjoining layer of the bottom-up stream when producing predictions for a certain scale. Figure 3 shows how YOLOv4 uses feature pyramids to detect traffic signs at different scales. The output is passed through a 33% convolution filter to reduce upsampling artifacts and crevices [5]. Spatial Attention Module (SAM), Path Aggregation Network (PAN), and Spatial pyramid pooling layer (SPP) are implemented or replaced with the FPN approach in YOLOv4. Maximum and average pools are applied to input feature maps individually in SAM to produce two sets of feature maps. To produce spatial attention, the feature maps are sent into a convolution layer followed by a sigmoid function. This method is used to gather data and improve accuracy. The preceding layer's input is used by each subsequent layer.

V. EEXPERIMENTAL SETUP

We examine the performance of our proposed YOLOv4 model for traffic sign detection and experimental findings using a set of calculated parameters and a dataset. The model was tested on 43 different traffic sign classes to gather all of the dat

A. Dataset

The YOLOv4 model was trained and tested using our dataset [14], which was manually annotated. It was separated into a train set of 835 images with 1393 annotations and a test set of 133 images with 225 annotations, with a total of 968 images and 1618 annotations. Figure 4 illustrates several examples of our dataset's images.

B. Google Colab

Google Colaboratory is a cloud-based tool that mimics the functionality of Jupyter Notebooks. Colab requires no setup and offers unrestricted access to computing resources.

C. Darknet repository

The model is trained by using the Darknet framework from AlexeyAB's repository. Darknet is a C and CUDA-based open-source neural network framework. It is easy to set up and supports both CPU and GPU computing. GPU backend was used to train the model.

D. Parameter calculation

Darknet repository was configured to match a batch size of 64 and 16 subdivisions. The learning rate was set to 0.001. The width and height of input images were set to 416×416. This YOLO v4 model consists of 161 layers which give detections at layers 139, 150, and 161. Max batches, Steps, and Filters used in this YOLOv4 model are given in equations (1), (2), and (3) respectively.

$$\textit{Max batches} = \textit{number of Classes} \times 2000 \quad (1)$$

$$\textit{Steps= from (80\% of max batches) to (90\% of max batches)} \quad (2)$$

$$\textit{Filters} = (\textit{number of classes} + 5) \times 3 \quad (3)$$

E. Testing results

Precision, mean average precision and Intersection over Union were computed using the equations (4), (5), and (6).

$$Precision = \frac{True\ Positives}{True\ Positives + Flase\ Positives} \quad (4)$$

$$Mean\ Avearage\ Precision = \frac{1}{n}\sum_{K=1}^{K=n}(Precision_K) \quad (5)$$

$$Intersection\ over\ Union = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (6)$$

In this study, the overall average accuracy of detection and recognition of the traffic sign over the test set for various situations was 84.7%. Detection and recognition accuracy achieved for each distinctive class is illustrated in Figure 5.

VI. CONCLUSION AND FUTURE EXTENSION

Because it was trained on Google Colab, the YOLOv4 model, which was used for traffic sign detection and recognition, was discovered to have a comparatively higher level of accuracy while saving a substantial amount of computing cost and time. The 161 layers in YOLOv4 contribute directly to the improved accuracy over prior YOLO versions. Higher results may have been obtained if the model had been trained on a larger number of epochs and images, as this results in a greater range of image contexts and image quality. 18 out of 43 classes got 100% accuracy and only two classes such as speed limit 80 and road work got less than 50% accuracy. This study was able to attain a mean average precision of 84.7 % for 10000 epochs when it came to concluding its conclusions. Overall, this study was able to confirm that YOLOv4 outperforms its predecessors in terms of traffic sign detection. It may be inferred that the detection works effectively in a range of situations, such as distorted input images and lighting fluctuations. In the future, the extended work of traffic sign recognition to be improved the performance by using skipped layer architecture and vocabulary voting technique [15].

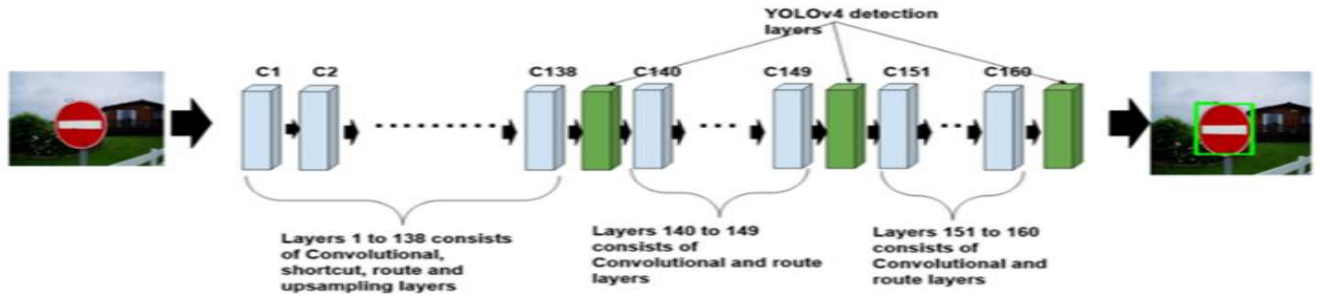


Fig. 3. How detections are found in feature pyramid network

Fig. 4. Some ample images of our dataset

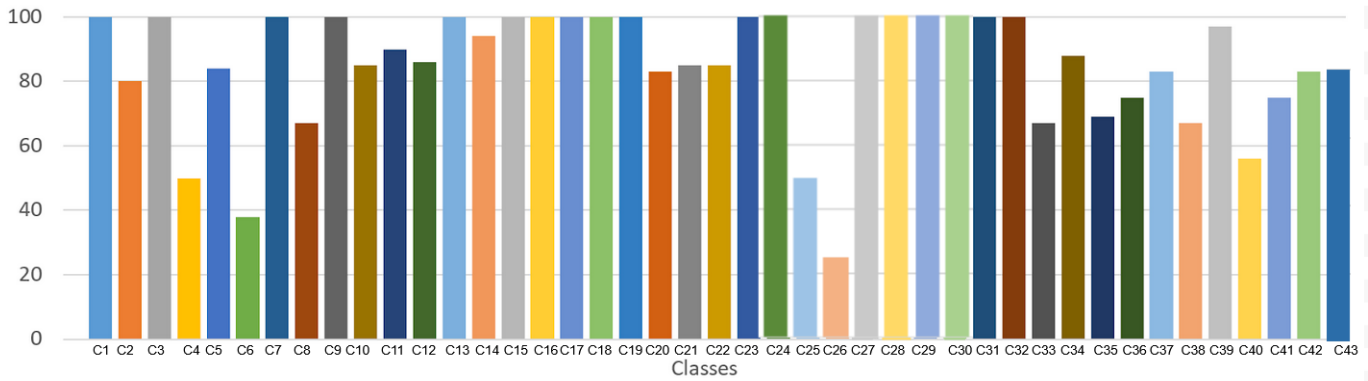
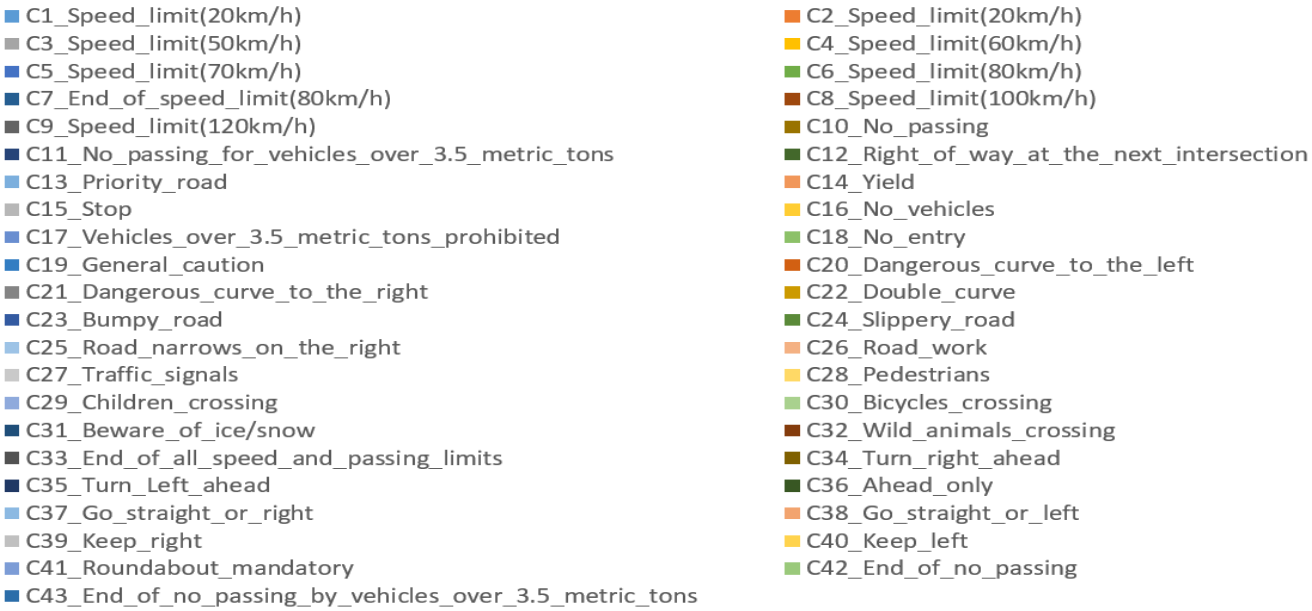


Fig. 5. Detection accuracy of the YOLOv4 model according to each class



REFERENCES

[1] M. Galvani, "History and future of driver assistance", IEEE Instrumentation Measurement Magazine, vol. 22, no. 1, pp. 11–16, 2019.

[2] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao “YOLOv4: Optimal Speed and Accuracy of Object Detection”, Google Scholar, pp. 1-17, 2020.

[3] B. Novak, V. Ilić and B. Pavković, “YOLOv3 Algorithm with additional convolutional neural network trained for traffic sign recognition”, Zooming Innovation in Consumer Technologies Conference (ZINC), pp.1-4, 2020.

[4] A. Avramović, D. Tabernik and D. Skočaj, “Real-time Large Scale Traffic Sign Detection”, 14th Symposium on Neural Networks and Applications (NEUREL), pp.1-4, 2018.

[5] Y. Sun, P. Ge and D. Liu, “Traffic Sign Detection and Recognition Based on Convolutional Neural Network”, Chinese Automation Congress (CAC), pp.1-4, 2020.

[6] R. Hasegawa, Y. Iwamoto and Y. Wei Chen, “Robust Detection and Recognition of Japanese Traffic Sign in the Complex Scenes Based on Deep Learning”, IEEE 8th Global Conference on Consumer Electronics (GCCE), pp.1-4, 2020.

[7] L. Shangzheng, “A Traffic Sign Image Recognition and Classification Approach Based on Convolutional Neural Network”, 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp.1-4, 2019.

[8] Z. Zhong-Qiu “Object Detection with Deep Learning: A Review”, IEEE Transactions on Neural networks and learning systems, pp. 1-21, 2019.

[9] A. Kumar, Z.J. Zhang, and H. Lyu, “Object detection in real time based on improved single shot multi-box detector algorithm”, EURASIP Journal on Wireless Communications and Networking, pp. 1-18, 2020.

[10] S. Dasiopoulou, "Knowledge-assisted semantic video object detection", IEEE Transactions on Circuits and Systems for Video Technology, pp. 1210 - 1224, 2005.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, " You Only Look Once: Unified, Real-Time Object Detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1 - 10, 2016.

[12] X. Huang, X. Wang, and W. Lv, "YOLOv2: A Practical Object Detector", Cornell University Computer vision and pattern Recognition, pp. 1-7, 2017.

[13] J. Redmon, and A. Farhadi, "YOLOv3: An Incremental Improvement Practical Object Detector", Cornell University Computer vision and pattern Recognition, pp. 1-6, 2018.

[14] https://www.fsc.esn.ac.lk/mathematics.

[15] S. Sotheeswaran and A. Ramanan, "Front-view car detection using vocabulary voting and mean-shift search," Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), 2015, pp. 16-20, doi: 10.1109/ICTER.2015.7377660.