Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V5

1st Aofan Liu
School of Computing and Data Science
Xiamen University Malaysia
Sepang, Malaysia
SWE2009510@xmu.edu.my

2nd Yutong Liu
School of Computing and Data Science
Xiamen University Malaysia
Sepang, Malaysia
SWE2009512@xmu.edu.my

3rd Saif Kifah*
School of Computing and Data Science
Xiamen University Malaysia
Sepang, Malaysia
saif.kifah@xmu.edu.my

***Abstract*—In today's era, deep learning neural networks with multiple hidden layers have been widely used in many fields. The deep learning method has more powerful features that enhance the method's performance by a learning process. With the development of the logistics industry and the prevalence of autonomous driving, traffic sign recognition has gained rising attention. This study uses a YOLO CNN to classify traffic signs. To improve model performance, we used MSRCR image augmentation during pre-processing. In the improvement phase, we used YOLOv5 to automate traffic sign categorization and improved training methods and network architecture. GTSRB and CCTSDB were used to assess the proposed technique. The experimental results show that the YOLOv5 model outperforms other methods. It has a 99.8% accuracy rate in the GTSRB dataset and 98.4% precision in the CCTSDB.**

***Index Terms*—Deep learning, YOLO algorithm, Traffic signs, Object recognition, CNN.**

I. Introduction

As as early as the 1960s, researchers have already shown high interest in conducting research in vision-based target inspection. Early researchers achieved robust detection of targets through cascade classifiers, Support Vector Machines (SVM), etc. These models were limited by technology restrictions [1]. Meanwhile, the traditional feature extraction method was not efficient. This could affect the quality of the model which leads to obtaining low-accuracy results. Therefore, the generalization ability of the model is relatively poor, and it is difficult to apply in the industrial and even commercial fields.

A deep learning neural network that incorporates numerous hidden layers possesses a highly potent feature learning function. This function enables the network to extract features from the initial input data by iteratively training the model to acquire increasingly abstract and fundamental representations. This method of training neural networks through deep learning was first applied to the field of speech recognition [2]. Compared with the traditional method, the accuracy, precision, and recall have been greatly improved. The improvement was significant reaching a 20%-30% improvement. Just less than a year later, Convolutional Neural Networks (CNN) have attracted the attention of researchers. This has drawn the interest of Internet giants such as Google and Microsoft, who have also invested significant resources to deploy deep learning.

Transportation is considered an important pillar in the basic industry of a country. The proliferation of autonomous driving technologies and the enhancement of living conditions have rendered autos a crucial mode of transportation for individuals' everyday mobility. Consequently, the aforementioned circumstances prompted the emergence and subsequent proliferation of intelligent transportation systems, which garnered increasing levels of scrutiny and interest [3]. Traffic signs play a vital role in intelligent transportation networks, and these signs show drivers the current traffic conditions of the road segment with words and symbols. Imagine you are driving on a highway and you see a sign that says "Exit 2 Miles". Without knowing the location of the sign, you may not know how much time you have to get off the highway or which lane you need to be in to exit.

However, due to the diversity of traffic signs, as well as the diversity of roads and weather conditions, the problem becomes more challenging. Furthermore, brightness, color, occlusion, and other issues, complicate the problem even further. Traffic lights are usually recorded in small images by occupying a very small part of the picture. In some cases, the weather conditions are very complex due to clouds, rain, sunny, and other conditions. On the other hand, images might be blocked by billboards, which has brought considerable difficulties [3]. Recognition of traffic signs through deep learning technology is a very challenging field [2].

At present, most related algorithms are only developed to detect a small number of categories, and it is difficult to overcome the influence of natural environmental factors such as nature, lightning, wind, rain, etc. In addition, the quality of the picture captured by the camera is not taken into account, which is seriously inconsistent with the actual situation [4]. Moreover, some algorithms only focus on the classification problem and ignore the problem of predicting the location of traffic signs, which is difficult to apply to industry and even commerce.

This paper conducts a series of empirical analyses on the application of deep learning YOLO (You Only Look Once). We propose the application of YOLOv5 in traffic detection and establish a CNN-based traffic sign recognition model. Additionally, it implements matching measures to enhance the accuracy and precision of real-time detection. The objective of

this study is to create a deep-learning neural network with the capability to accurately identify traffic signs. To achieve this goal, we have developed a fine-tuned model in GTSRB, and CCTSDB and achieved good results.

II. Related Work

A. Convolutional Neural Network

In the past 10 years, CNN networks have achieved groundbreaking breakthroughs in many fields. CNN is a feedforward neural network [5]. In order to handle two-dimensional input data, a multi-layer artificial neural network is specifically engineered, wherein each layer inside the network consists of numerous autonomous neurons.

Convolutional neural networks map the pixels of the original image into spatial data that can distinguish dimensions, a crucial step in breaking down the semantic gap between low-level pixels and high-level semantics [6]. Simultaneously, the model's capacity can be modified by altering the depth and width of the network. The features that are obtained through the convolutional layer are subsequently utilized as input for the classifier, ultimately leading to the attainment of the final prediction outcome [7].

B. Object Detection Algorithm

The domain of object detection, which relies on deep learning techniques, encompasses two primary algorithmic approaches: Two-Stage and One-Stage methods. R-CNN and Faster-RCNN are examples of the former, whereas YOLO-series and SSD-series represent the latter [8].

The detecting task is completed in two phases via two-stage approaches. After obtaining regional suggestions, characteristics in the regional proposals are utilized to locate and classify the objects. R-CNN is the first proposed Two-Stage algorithm that can achieve industrial-grade accuracy, but it has slow detection and cannot meet the requirements of a fast response. With the introduction of the One-Stage algorithm, the speed of target detection has been greatly improved, such as in YOLO [2]. YOLOv3 employs a feature pyramid network topology to perform multi-scale detection. YOLOv5 increases detection performance even further by fine-tuning the network topology, activation function, and loss function, and utilizing abundant data augmentation [7].

C. YOLOv5

The YOLO models represent a comprehensive technique for real-time object detection. In real-time object identification applications, the models consistently strive to achieve the optimal equilibrium between speed and accuracy [9].

In the field of object detection, we most likely need to identify the location and category of objects in the image and introduce bounding boxes to solve this problem. YOLO is one of the algorithms that uses bounding boxes. Assuming that the top-left corner of the grid can be represented with C_x and C_y while the network outputs are represented with O_w and O_h. Meanwhile, the anchor dimension can be expressed with P_w and P_h. At the same time, B_x, B_y, B_w and B_h are the core coordinates, width, and height of estimation.

Roughly speaking, object detection is the process of obtaining target information after processing the input picture/video, including coordinates, the predicted category of the target, and the predicted confidence of the target [10]. It separates the images into S by S grids, with every grid performing a distinct detection job as shown in Fig 1.

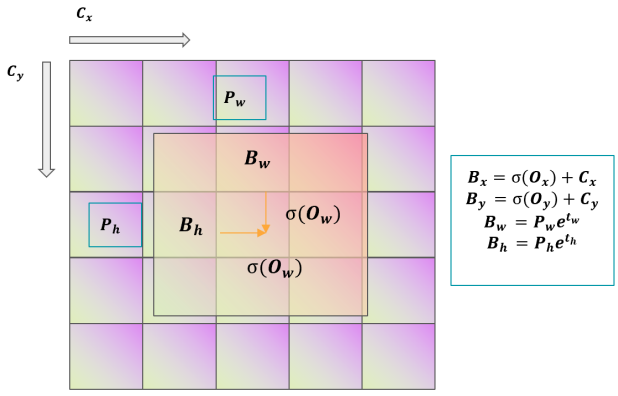

Fig. 1. Bounding boxes graph.

The whole network structure is shown in Fig 2. While the YOLO algorithm is good at detecting targets quickly, it is ineffective at detecting tiny targets. YOLOv5 algorithm transmits each batch of training data through a data loader while augmenting the training data. There are three ways for a data loader to perform data enhancement: scaling, color space adjustment, and mosaic enhancement.

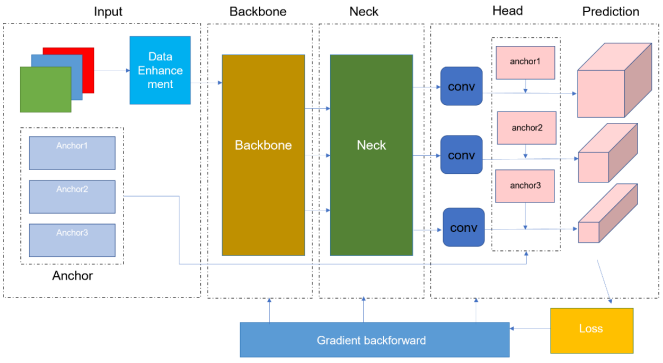

Fig. 2. CNN architecture of the present model

Additionally, YOLOv5 refers to a set of object detection models that have been trained on the COCO dataset. These models are designed with various scales and possess features that facilitate tasks such as test-time augmentation, model composition, hyperparameter tuning, and exportation to formats including ONNX, CoreML, and TFLite. The YOLOv5 model currently exhibits the most favorable balance between

performance and efficiency, achieving an average precision (AP) of 48.2% on the COCO dataset while maintaining a processing time of 13.7 milliseconds.

III. The Architecture of Proposed Network

In the new global economy, traffic sign reconviction has been a central issue for both autonomous transportation and urban traffic management systems [11]. Most previous research is based on a two-stage model. In some cases, the two-stage model is not efficient and cannot meet the requirements of the current industry. Therefore, the main content of this paper is to propose an efficient optimized convolutional neural network that can solve this issue [12]. As a typical single-stage algorithm, YOLO is also an end-to-end network structure. The prediction time of this network structure is obviously better than that of algorithms such as R-CNN.

One of the challenges before employing the proposed model is data preprocessing. The process includes data cleaning, data specification, and data transformation [13]. In the present study, synthetic data and mosaic augmentation were employed as techniques to augment the dataset and enhance the performance of the model. In addition, data augmentation techniques were employed, including rotation, scaling, and flipping, to generate supplementary versions of the original photos. Moreover, the usage of the mosaic augmentation approach allowed us to create new images that retain the appearance and characteristics of the original images while introducing more diversity and variability to the dataset.

In this network, we mainly train two models. Model-1 was trained on GTSRB, which has over 50,000 RGB images in total, including 32,909 in the train set and 12,631 in the test set. The images in this dataset can be classified into 43 categories and contain the same images under multiple conditions. Class 43 traffic signs include all traffic signs defined by German law [14].

The annotation for this dataset is given in a single text file and we use Python's Numpy and Pandas libraries to convert it to YOLO format. Following the conversion process, we received the corresponding photos and comments. They are kept in two separate files (images, labels), each with subfolders for training and testing. As a result, a text file containing data from each image's bounding box is associated with the dataset that has been prepared.

Next, our model's network is discussed. Our YOLO network consists of Input, Backbone, Neck, and Head. On the input side, to achieve a more complex picture background, Mosaic data augmentation is used to combine four pictures. The purpose is that the network can deal with a more complex natural environment and the environment where traffic signs are located.

The Backbone part mainly includes BottleneckCSP and Focus modules [15]. The former can greatly reduce the computational load of the network while maintaining the accuracy of the network almost unchanged or even reduced. Afterward, the Focus module slices the image and obtains the downsampling volume through the convolution layer which can also reduce the amount of computation and speed up the network. The convolution operation of the YOLO model is different from the convolution in the conventional sense but uses CBL to act to generate convolution. The above operations allow us to extract feature layers from YOLO. The following graph shows one of the feature layers:

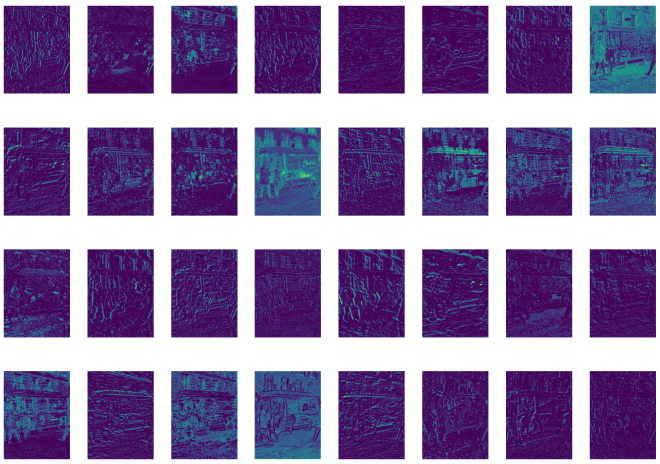

Fig. 3. Feature map 3/24 of the layers

Compared with the Backbone, the components of the Neck part are very single. It consists of CBS, UpperSample, and Concat. Simultaneously, the structure of FPN+PAN is used. The components perform a wave of mixing and combining of features and pass these features to the prediction layer [16], [17]. The Head section generates a vector that comprises the probability of the category, the score of the object, and the position of the bounding box for the target item. Subsequently, the feature vector of the detection layer is returned to its initial form. The activation functions employed in this study consist of leakyReLU and Sigmoid. The middle-hidden layer utilizes the Leaky Rectified Linear Unit (ReLU) activation function, whereas the final detection layer employs the Sigmoid activation function [18]. The CNN architecture of our model is adapted from the YOLOv5 paper. It can be expressed with the following graph.

Model-2 was trained on CCTSDB which contains more than 15 thousand images for training purposes. Since GTSRB is already 10k-level data, in order to achieve diversity, we extract data from it. About 3k pieces of data were randomly selected for thousands-level data training.

All the images contained in the CCTSDB dataset are accompanied by corresponding labels, constituting labeled data. Consequently, the process of dividing the dataset into a training set and a test set is the only need. In the training dataset, we conducted a train-test split with a ratio of 8:2. Since the data in CCTSDB is also all stored in a txt file. We apply the above formula again to generate the Label corresponding to the Image. The batch size of the model is trained from 16 to 256 (n times 16), however, we got the best accuracy when the batch size was 32.

Through the application of the above two datasets, we can preliminarily believe that our model is more effective in the field of traffic sign recognition and can meet the challenges of the industry to a certain extent. Meanwhile, the proposed method is a one-stage method.

IV. Model Analysis

Model-1 is trained using the YOLOv5 framework, with v5s serving as the initial weight. The parameters utilized for the training process are presented in Table 1. The table illustrates that the dataset was trained using a batch size of 16.0, and the learning rate employed was 0.1. The learning rate scheduler is implemented during the training process to optimize the model's performance by determining the most suitable parameter values. We test the model with all the learning rates ranging from 0.005 to 0.025 and test each learning rate three times. It was observed that the performance of the model in YOLOv5 is optimized when the learning rate is configured to 0.01. The parameters utilized in our study are as follows.

TABLE I
HYPER PARAMETERS USED IN THE MODEL TRAINING PROCESS.

Parameter	Value
Box	0.05
Scale	0.5
Shear	0.0
Batch size	16.0
Anchor T	4.0
Momentum	0.937
Learning Rate	0.01.
Warmup Epoch	3.0

Moreover, contrary to what is believed (the larger model will have better performance), v5s achieve the best result among various initial weights (v5s, v5m, v5l, v5x). We think other large models prefer generalized identification rather than this traffic sign-oriented situation. After training with our oriented dataset, the v5s model fit well. The hyper parameter for this model is listed below.

Model size and performance are two critical factors in designing deep learning models. In general, larger models tend to achieve higher accuracy, while smaller models are faster and more efficient in terms of computational resources. However, there is a trade-off between model size and performance, and finding the right balance between the two is crucial, especially for real-time applications with limited computational resources. In case higher accuracy is preferred, we did not limit the size of the model. However, due to the small size of the initial model, the size of the final model is only dozens of Mb.

At the same time, the environment contained in the image is diverse, which means that the richness of the image can better increase the robustness of the model. The following label correlation matrix (Fig 4) shows the distribution of labels and images for the model .

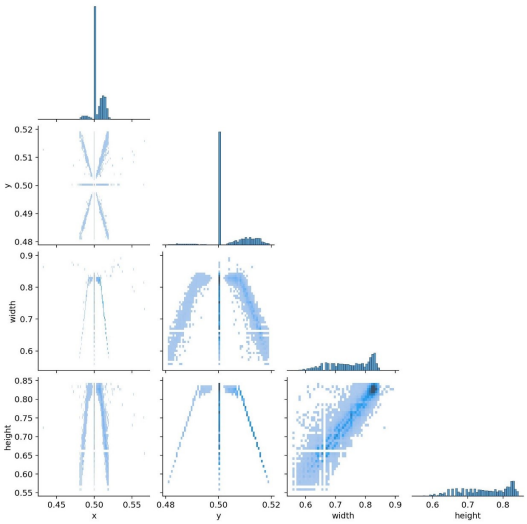

Fig. 4. Label correlation between x, y, width and height, which showing the frequency of label co-occurrence.

The model is evaluated from the following four aspects: precision, recall, AP, and mAP. The precision is a measure of metrics of quality and is the number of positive samples we predicted to be correctly predicted divided by the number of predicted positive samples. The Recall is the recall rate, which means that the number of correct predictions we correctly predict accounts for the number of all correct positive samples as depicted in Fig 5.

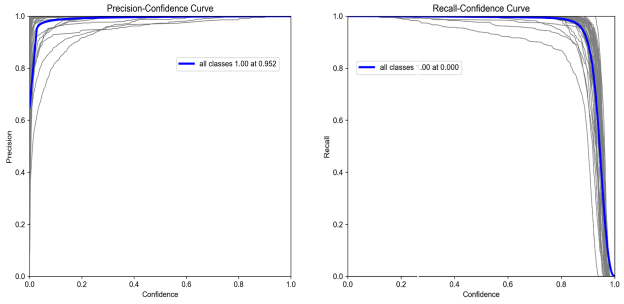

Fig. 5. Precision and recall of the model fluctuate during confidence change (**a**) Precision-Confidence Curve (**b**) Recall-Confidence Curve

However, these two evaluation indicators can only reflect the performance of the model to a certain extent, and cannot accurately represent the model. Therefore, we introduce AP and mAP. The PR-Curve value is the curve composed of precision and recall, and the AP is the area under the line of the curve composed of these two values [19].

$$AP = \int_0^1 P(R)dR, \tag{1}$$

$$mAP = \sum_{i=1}^{C} \frac{AP_i}{C}, \tag{2}$$

Following the application of YOLOv5 with more than 147000 iterations across 60 epochs. Our proposed model achieved a precision score of 99.73% and a recall score of 99.76% for the test dataset. The model consists of 157 layers, 7126096 parameters, 0 gradient, and 16.1 GFLOPS.

Here are a table and a figure (Fig 6) showing the detailed parameters of the trained model.

TABLE II
HYPER PARAMETERS USED IN THE MODEL TRAINING PROCESS.

Metric	Value
Precision	0.9973
Recall	0.9976
mAP@0.5	0.9948
mAP@0.5:0.95	0.9546
Box Loss	0.0020
Object Loss	0.0015

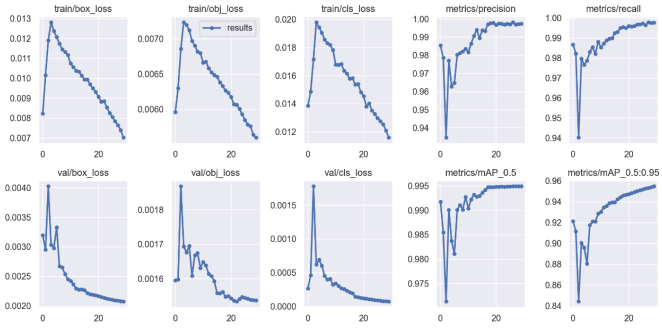

Fig. 6. Evaluation metrics of the model ranging from mAP@0.5 and mAP@0.5:0.95

A confusion matrix is a tool used to visualize the predictions of an N-gram classifier in N x N tables. It is normally used in supervised learning. The following Fig 7 shows the confusion matrix of the model.

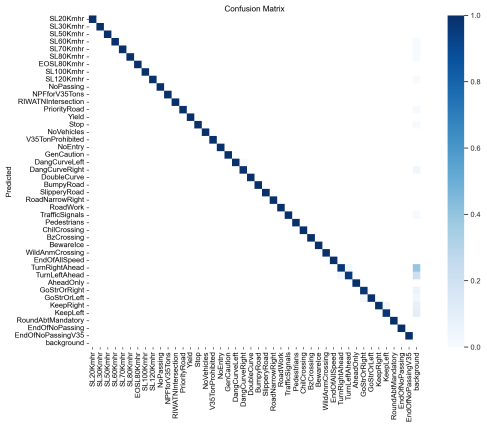

Fig. 7. Confusion matrix of our present model.

The proposed method can be compared with other methods working on the same dataset. In [20] Dewi, Christine et al. achieved 84.9% accuracy using YOLOv3 on GTSRB and 89.33% using YOLOv4. Kankaria, Romit Vinod achieved 91.12% accuracy at 30 fps, giving solid results. Khnissi achieved 95.44% accuracy using the upgraded compact YOLO-V4. Jayant Mishra and Sachin Goyal built the model on GTSRB using YOLOv5 and they achieved 97.7% accuracy. Qin Zongbing also tried on GTSRB with YOLOv5 and ended up with 90.7, 97.7, and 94.5% when the image was split by 200 and 400-pixel sizes for the dividing line, respectively. As of now (2022), the model has achieved excellent performance in fast recognition algorithms, and the model performance is comparable to two-stage algorithm studies on this dataset.

In fact, YOLOv7 already exists in the research field, which is a newer version of the YOLO object detection algorithm than YOLOv5. YOLOv7 was released after YOLOv5 and generally offers improved performance and accuracy over its predecessor. However, YOLOv5 is still often preferred due to its superior performance and efficiency in this case. Gunasekara et al. tried to use YOLOv7 on GTSRB, but they only achieved 92.11% in the end.

YOLOv7 and YOLOv5 are both recognized as object identification algorithms that have gained popularity and are extensively employed across diverse applications. However, YOLOv7 and YOLOv5 differ in their implementation details and network architectures, which can affect their performance and efficiency. We have made some attempts on YOLOv7, but the results are not very good and finally chose to use YOLOv5.

One potential factor is the implementation of Extended-ELAN in YOLOv7. In the context of large-scale ELAN (Elastic Local Area Network), it may be observed that the Internet attains a condition of equilibrium irrespective of the path length gradient direction and the total number of blocks involved. Nevertheless, in the event that the computation blocks are continuously piled, this equilibrium could potentially be disrupted, resulting in a decrease in the consumption rate of parameters. Within the domain of system architecture, it is seen that E-ELAN solely impacts the system architecture within the computation block, while leaving the system architecture of the transition layer unaltered. Fig 8 shows the evaluation metrics of our model.

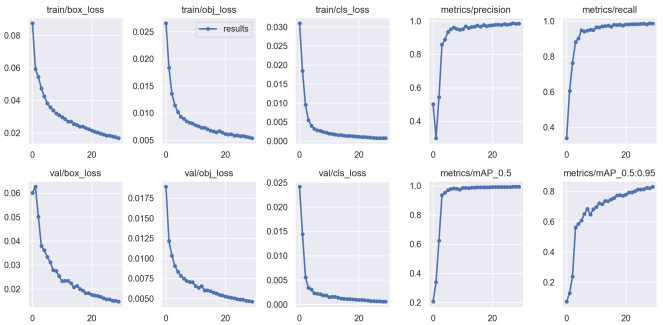

Fig. 8. Evaluation metrics of our model.

The dataset chosen for model-2 is CCTSDB which was proposed in 2017 IEEE Access by Jianming Zhang. We finally realized 98.4% precision and 98.6% in the end for the

proposed YOLO model. The evaluation metrics of the model can be seen in Figure 6.

All the experiments are performed in a Pytorch 1.8.0, 2080Ti, I9-9900K, CUDA 10, 32GB RAM machine.

V. Conclusion

This study proposes a traffic sign recognition algorithm based on the fine-tuning YOLOv5 model. It also shows the potential of deep learning and how it can be applied to the area of traffic sign recognition. Through a multi-scale feature detection method and a small model volume, it can ensure a high detection accuracy while still having a fast detection speed, which basically meets the needs of the industry [21]. The synthetic image is integrated with the original image through the implementation of a specific transformation on the original dataset. This process aims to augment the dataset and boost the efficacy of the deep learning model. A mosaic augmentation technique is also applied which combines multiple training images at specific scales into one.

Future research endeavors will focus on enhancing the robustness of the model. One potential avenue involves employing the Generative Adversarial Network (GAN) technique to enhance the quality of challenging images that are difficult to discern. Subsequently, these improved images may be utilized to train the model, hence augmenting its accuracy. We can also improve the model by using Deep autoencoders which can help us detect traffic signs while leaving any other objects with only traffic signs.

References

[1] F. J. Ansari and S. Agarwal, "Fast road sign detection and recognition using colour-based thresholding," in *International Conference on Computer Vision and Image Processing*. Springer, Conference Proceedings, pp. 318–331.

[2] X. Bangquan and W. X. Xiong, "Real-time embedded traffic sign recognition using efficient convolutional neural network," *IEEE Access*, vol. 7, pp. 53 330–53 346, 2019.

[3] F. Bi and J. Yang, "Target detection system design and fpga implementation based on yolo v2 algorithm," in *2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC)*. IEEE, Conference Proceedings, pp. 10–14.

[4] A. Mulyanto, R. I. Borman, P. Prasetyawan, W. Jatmiko, P. Mursanto, and A. Sinaga, "Indonesian traffic sign recognition for advanced driver assistent (adas) using yolov4," in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, Conference Proceedings, pp. 520–524.

[5] M. Çetinkaya and T. Acarman, "Traffic sign detection by image preprocessing and deep learning," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, Conference Proceedings, pp. 1165–1170.

[6] X. Changzhen, W. Cong, M. Weixin, and S. Yanmei, "A traffic sign detection algorithm based on deep convolutional neural network," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*. IEEE, Conference Proceedings, pp. 676–679.

[7] C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. D. Hartomo, "Yolo v4 for advanced traffic sign recognition with synthetic training data generated by various gan," *IEEE Access*, vol. 9, pp. 97 228–97 242, 2021.

[8] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Conference Proceedings, pp. 1625–1634.

[9] V. Goel and H. S. Paul, "Advanced driver assistant systems," EasyChair, Report 2516-2314, 2021.

[10] S. Goyal, "Traffic sign recognition detection using deeptrans learning."

[11] Y. Jin, Y. Fu, W. Wang, J. Guo, C. Ren, and X. Xiang, "Multi-feature fusion and enhancement single shot detector for traffic sign recognition," *IEEE Access*, vol. 8, pp. 38 931–38 940, 2020.

[12] K. Khnissi, C. B. Jabeur, and H. Seddik, "Implementation of a compact traffic signs recognition system using a new squeezed yolo," *International Journal of Intelligent Transportation Systems Research*, pp. 1–17, 2022.

[13] R. V. Kankaria, S. K. Jain, P. Bide, A. Kothari, and H. Agarwal, "Alert system for drivers based on traffic signs, lights and pedestrian detection," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, Conference Proceedings, pp. 1–5.

[14] E. Peng, F. Chen, and X. Song, "Traffic sign detection with convolutional neural networks," in *International conference on cognitive systems and signal processing*. Springer, Conference Proceedings, pp. 214–224.

[15] W. Li, D. Li, and S. Zeng, "Traffic sign recognition with a small convolutional neural network," in *IOP conference series: Materials science and engineering*, vol. 688. IOP Publishing, Conference Proceedings, p. 044034.

[16] A. A. Lima, M. Kabir, S. C. Das, M. Hasan, and M. Mridha, "Road sign detection using variants of yolo and r-cnn: An analysis from the perspective of bangladesh," in *Proceedings of the International Conference on Big Data, IoT, and Machine Learning*. Springer, Conference Proceedings, pp. 555–565.

[17] A. Liu, M. S. Khatun, H. Liu, and M. H. Miraz, "Lightweight blockchain of things (bcot) architecture for enhanced security: A literature review," in *2021 International Conference on Computing, Networking, Telecommunications Engineering Sciences Applications (CoNTESA)*. IEEE, Conference Proceedings, pp. 25–30.

[18] Z. Liu, J. Du, F. Tian, and J. Wen, "Mr-cnn: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, pp. 57 120–57 128, 2019.

[19] A. Unger, M. Gelautz, and F. Seitner, "A study on training data selection for object detection in nighttime traffic scenes," *Electronic Imaging*, vol. 2020, no. 16, pp. 203–1–203–6, 2020.

[20] C. Dewi, R.-C. Chen, and H. Yu, "Weight analysis for various prohibitory sign detection and recognition using deep learning," *Multimedia Tools and Applications*, vol. 79, no. 43, pp. 32 897–32 915, 2020.

[21] L. You, Y. Ke, H. Wang, W. You, B. Wu, and X. Song, "Small traffic sign detection and recognition in high-resolution images," in *International Conference on Cognitive Computing*. Springer, Conference Proceedings, pp. 37–53.