Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.33691

Advanced Adaptive Early-Stopping YOLO for Traffic Sign Recognition

Chenjun Zhu

Faculty of Engineering, The University of Sydney, Sydney, Australia
15396992287@163.com

Abstract. The accuracy and efficiency of traffic sign recognition have to be very high to facilitate autonomous driving systems. AE-YOLO is a YOLOv8-based model that is combined with an adaptive early-stopping scheme to recognize traffic signs, which is proposed in this paper. The developed approach decreases redundant training at the cost of competitive recognition performance. Also, systematic hyperparameter optimization is performed to evaluate how model size, input size and stopping parameters affect the trade-off between accuracy and computational time. In order to test the robustness even further, one more set of road sign images acquired through online databases is added to the dataset and the same evaluation procedure is used. Experimental evidence shows that AE-YOLO attains a better trade-off between precision and speed with high level of robustness in various road sign settings. Final accuracy of AE-YOLO is 2.3 percentage points better than the one of the most well-known baselines (four factor groups). The code may be found at: https://github.com/chenjunzhu4-ctrl/Adaptive-Early-Stopping-YOLO-for-Road-Sign-Recognition

Keywords: Computer Vision, Traffic Sign Recognition, YOLO

1. Introduction

Traffic sign classification is a fundamental component of autonomous driving and advanced driver-assistance systems, as traffic signs provide essential semantic information such as warnings, prohibitions, and speed limits. This problem has been extensively studied from multiple perspectives. Study [1] introduced traffic sign recognition systems based on You Only Look Once (YOLO) traffic signals, while study [2] evaluated various YOLOv8 operating environments under time constraints. Study [3] improved classification performance using committee-based models, whereas other studies investigated lightweight designs and training optimisation [4, 5]. Other studies have focused on recognising colours, shapes, hand-designed features, and multi-level techniques [6-10].

Despite these advances, two main limitations persist in practice. First, efficiency remains low when using multi-stage pipelines, ensemble models, or compression-based training algorithms that frequently incur additional computation costs. Second, precision is likely to decline in difficult conditions of the environment like occlusion, motion blur, variations in illumination level, and small size targets where the traditional or poorly generalised models are ineffective.

DOI: 10.54254/2755-2721/2026.33691

Conclusively, there is still a need to have a brief and repeatable experimental design on the classification of traffic signs to further comprehend the trade-off between accuracy and efficiency.

The main contributions of this paper are as follows:

1. The paper suggests an innovative solution to the problem of identifying road signs, which is Adaptive Early-Stopping You Only Look Once (AE-YOLO) that combines an adaptive early-stopping process with the YOLOv8 model. The given approach minimizes unnecessary training epochs and provides better trade-offs between recognition accuracy and computational efficiency than the current baseline methods.

2. A systemized hyperparameter analysis is performed to examine the trade-off between the accuracy and time. The outcomes in four factor groups indicate that AE-YOLO always outperforms the baseline models with a difference of around 2.3 percentage points in final accuracy.

3. The research will conduct the robustness validation through the inclusion of extra images of road signs obtained online. With this larger dataset, AE-YOLO has an accuracy of about 0.97 and a time cost of about 110 seconds indicating that it is capable of performing well in more diverse settings of road signs.

2. Related work

2.1. Traditional methods

The initial traffic sign recognition algorithms were mainly based on the manual features including color, form, Histogram of Oriented Gradients (HOG) and transform distance, and combined with older style classifiers. Research [6, 7, 10] can follow this direction. These techniques build upon existing knowledge about traffic signs, i.e., consistent shapes and specific colors to make it easier to extract features and classify them. They are highly interpretable, easy to implement and have relatively low computation expenses, and therefore they may be useful baselines even under restricted conditions.

But their behavior can be often deteriorated by human-drawn attributes. Their sensitivity to changes in lighting, deformation, occlusion, blur and more complex backgrounds also restricts their ability to generalize in real-life contexts.

2.2. Deep learning methods

The latest studies have more and more shifted their attention to deep learning methodologies as a way of enhancing feature representation and recognition accuracy. The works [1, 2, 9] discussed the detection-classification systems and in particular, YOLO-based systems and real-time pipelines. The works [3, 5, 8] studied the classification models based on the CNN and the committee, whereas the work [4] studied a light-weight model via pruning and distillation.

These approaches are often found to be more efficient than the traditional ones, since they are able to discover more discriminative features through learning them directly on data. Nevertheless, there are a number of constraints. Both the computational complexity and training time can grow with multi-stage pipelines, ensemble models, or large-scale architectures. However, compressed models can also be less robust in unfavorable conditions like shadow, blur, or small targets. Therefore, the trade-off between accuracy and efficiency is an important problem in deep learning-based traffic sign recognition.

DOI: 10.54254/2755-2721/2026.33691

2.3. Limitations

In general, there are still two significant limitations to current methods. One of them is inefficiency. Pipelines of detection and classification normally need extra localisation and recognition processes, ensemble models comprise several networks working at once, and compressor-based algorithms commonly need extra pruning, distillation or retraining. Therefore, such methods cost more to train, take longer to make inferences, or have greater complexity to implement, and thus cannot be used in live traffic instances.

The second restriction is poor precision in adverse weather. Traditional methods are based on hand-crafted features that have a small representational capacity and frequently fail to distinguish between visually similar traffic sign categories. A few light or loosely generalized deep models could be unable to discern targets under difficult conditions. Consequently, their recognition rate may deteriorate greatly as road scenes are more varied and intricate.

3. Methodology

3.1. Overview

The proposed AE-YOLO model involves three consecutive stages. Step 1 involves collecting, annotating, preprocessing, and splitting the road sign dataset. Step 2 involves feeding the processed images into the AE-YOLO detection pipeline to localise and classify road signs. Step 3 applies an adaptive early-stopping algorithm during training to eliminate unnecessary epochs and retain the most effective model. The general pipeline of the proposed approach is illustrated in Figure 1.

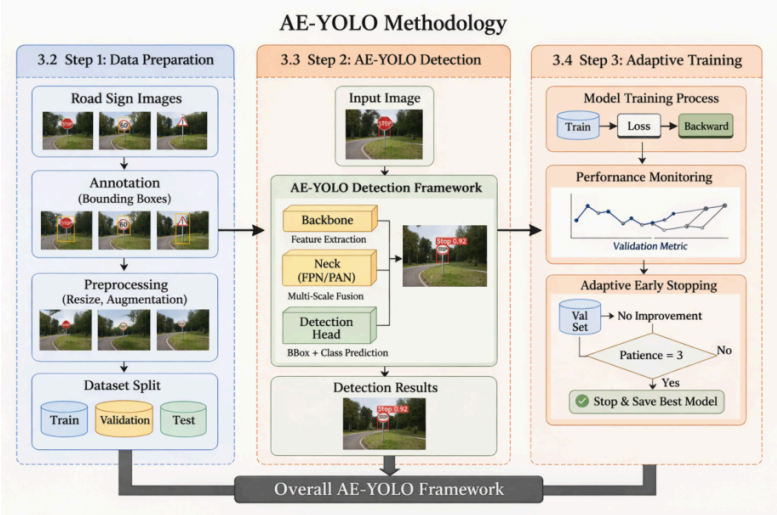

Figure 1. Overall methodology of the proposed AE-YOLO framework

3.2. Step 1: data preparation and preprocessing

The research involves obtaining road sign images at the first stage by capturing the images of actual traffic conditions through photography, marking the images with bounding boxes, and labeling them in the form of a YOLO label. The paper will keep both the target sign and the surrounding road environment so that AE-YOLO can be more suitable to real-life driving conditions [11-13].

The subsequent article scales and enlarges the images prior to training. The next step after preprocessing is to divide the data into training set, validation set, and test set. The training set is

DOI: 10.54254/2755-2721/2026.33691

used to optimise, the validation set is used to monitor and the test set is used to evaluate finally.

3.3. Step 2: AE-YOLO detection

After that, this article apply the pre-processed images of road signs to the AE-YOLO detection pipeline to localize and label road signs. At this stage AE-YOLO accepts processed images as input and outputs results of detections with a united detection architecture [12, 14, 15]. At this step, it is necessary to specify what positions road signs have in an image and what category they are in.

The AE-YOLO model has three major elements, namely the backbone, the neck and the detection head. The backbone accepts an input image and produces hierarchical visual features. Such features are descriptive of the visual look of road signs and the environment around them on various levels, including shape, contour, surface or contextual clues. Most often, traffic signs only take up a tiny area of the picture so that this step plays a critical role in preserving useful visual information and reducing the loss of discriminative information.

The neck combines features of various levels and intervals. The multi-scale fusion process allows AE-YOLO to operate on road signs located at different distances and scales. When it comes to actual driving conditions some of the traffic signs are near the camera and take up a significant amount of space, whereas others are far away and look like tiny targets [11, 16]. The combination of fine-grained details of local areas with more general higher-level information allows AE-YOLO to improve its ability to identify signs with differences in scale, viewpoints, and complicated backgrounds.

The last step that detection head performs is outputting the end prediction results, such as bounding box coordinates, object confidence scores, and class labels of every identified road sign. Consequently, AE-YOLO does not only classify the sign category but also identifies the position of a particular sign in the picture. It is especially significant with regard to traffic sign recognition since most of the signs have similar appearances, are relatively small, and are vulnerable to environmental factors like variations in lighting, blurriness, or obscuration.

One of the main aspects of this stage is that AE-YOLO is performing localization and classification concurrently. Contrary to seeing recognition as a distinct classification task, this technique considers all the traffic conditions and provides location and class estimates at once. The design is more applicable to realistic settings in a driving context, where the signs have to be first located and then identified.

This finishes the central recognition stage of AE-YOLO. The processed image is fed into the model, visual features are extracted and combined and the final detection results are produced. Therefore, the second step is the main detection and recognition phase of AE-YOLO and it can be considered as the basis of the training optimization presented in the following step.

3.4. Step 3: adaptive early-stopping training

In Step 3, AE-YOLO is trained with a dynamic early-stopping strategy. The aim at this step is to provide more efficient training without jeopardizing detection. Rather than a certain number of epochs, the model keeps track of validation performance and terminates training each time it does not improve significantly. This removes redundant computation of epochs.

The training process involves optimising model parameters using forward propagation, loss calculation, and backpropagation using the training set during the training procedure. This procedure allows the model to acquire visual evidence of road signs based on the input data. Following every single training epoch, the model is tested on the validation set and the validation score is recorded.

DOI: 10.54254/2755-2721/2026.33691

The fresh validation score is compared with the best validation score obtained until now. Using this validation-oriented mechanism, AE-YOLO does not depend upon a definite number of epochs but changes the length of training depending upon the trend of performances [17].

Let Mt denote the validation score at epoch t, M_{best} the best validation score recorded so far, δ the minimum improvement threshold, and P is a user-defined parameter(used for control the epoch size). The best score is updated as:

$$M_{best} = max(M_1, M_2, \ldots, M_t) \tag{1}$$

If the current improvement satisfies:

$$M_{\text{best}} - M_t < \delta \tag{2}$$

for P consecutive epochs, then the training is stopped early, and the best checkpoint is retained.

Such an approach allows AE-YOLO to end training once additional epochs add just marginal gains. In reality, more advanced epochs can take up much computation time with little value. The adaptive early-stopping strategy is used to address this issue, which indicates the moment at which the model has already performed steadily on the validation set. As a result, there is no extra training time and the best-performing model is saved instead of the last epoch.

One of the other advantages associated with this approach is enhanced training stability [18-20]. A long training procedure can use up too much computing power and not achieve any significant performance advantages. The ability to terminate at an appropriate location enables AE-YOLO to maintain the optimisation process effective and make the training procedure more experiment-friendly in the real world, when detection rate and time cost are important considerations. It is especially applicable to the detection of road signs tasks that typically require repetitions and comparisons between the parameters.

AE-YOLO achieves the optimalization step of the methodology at this point. Step 3 controls learning and ending of training after Step 2 identifies objects in the image through the main recognition pipeline. Step 3 is not a new model or inference pathway but an improvement of the training behavior of AE-YOLO as part of the entire structure. This ensures that AE-YOLO will be effective in the training process maintaining the fundamental detection ability that was introduced in Step 2.

4. Experiment

4.1. Experimental setup

Equally, like in experiment [2] all experiments were performed according to one training and assessment program. All comparison conditions used the same dataset division, implementation framework and reporting criteria. In order to apply AE-YOLO, the YOLOv8 framework was used and all the experiments were executed on the same software and hardware environment so that the results could be compared fairly.

4.1.1. Dataset

The German Traffic Sign Recognition Benchmark (GTSRB) was introduced in the IJCNN 2011 competition and has been widely used as a benchmark in traffic sign recognition research [12]. GTSRB is a multi-class single-image classification dataset with more than 50,000 images in 43

DOI: 10.54254/2755-2721/2026.33691

classes of traffic signs. It has large variations in scale, lighting conditions, view point, blurry, and partially obstructed scenes, which makes it useful to test real-world road situations.

In order to allow training with validation, the dataset was separated into training, validation, and test subsets. The training subset was used to optimize model parameters, convergence was observed through the validation subset, adaptive early stopping was performed depending on validation performance, and lastly the test subset was evaluated.

4.1.2. Baseline models

To assess the performance of AE-YOLO, three representative baseline models were chosen following previous works.

The first baseline was a committee based neural network model, which is an ensemble style high accuracy classifier [3].

The second baseline was a lightweight deep model of study [4], which was based on model compression and lightweight design.

The third baseline that was considered is the Adam-optimised CNN classifier that has been introduced in study [5] as a typical deep learning model but with a better training strategy depending on the type of optimisers.

Altogether the given baselines encompass the ensemble learning, lightweight design and traditional methods of deep classification.

4.1.3. Experimental factors

A single-variable approach was adopted, where only one factor was varied at a time while all other parameters were kept constant. Four factors were investigated:

4. Filters - the number of convolutional filters in the feature extraction layers. More filters improve representation capacity but increase computational cost.

5. Kernel Size - controls the receptive field size of the convolution operation. Smaller kernels focus on local details, while larger kernels capture broader spatial information from the image.

6. Dropout Rate - controls regularisation strength. Increased dropout rate has been demonstrated to decrease overfitting, but may also harm the process of learning features when applied excessively.

7. Pool Size - determines down-sampling in pooling layers. A bigger pool size will be more aggressive in shrinking the feature maps, which might reduce the amount of computation but also destroy important information.

These settings were used to evaluate the impact of each factor on both model accuracy and training time.

4.1.4. Evaluation metrics

The two evaluation metrics are accuracy and timing. Accuracy is a measure of model performance and time is a measure of computational efficiency, including total training time and inference related runtime. These metrics enable the analysis of the trade-off between recognition performance and computational cost.

DOI: 10.54254/2755-2721/2026.33691

4.2. Experimental results

4.2.1. Effect of filters

The effect of filters on model accuracy and training time was explored initially. The ultimate accuracy of various filter settings is depicted in Figure 2, whereas the training time is illustrated in Figure 3.

As shown in Figure 2, AE-YOLO attained accuracies of 0.967, 0.975, 0.972, 0.980, and 0.974 with filter values {16, 32, 64, 128, 256} respectively. The optimal result was at 128 filters. The three baseline models were following the same pattern and they all had a highest accuracy at 128 filters. In any setting, AE-YOLO performed better than the baselines.

The increase in training time with the number of filters was observed in all models as illustrated in Figure 3. AE-YOLO trained time grew to 97s to 170s. All three baseline models exhibited the same trend.

On average, it can be stated that there is a positive correlation between the number of filters, accuracy, and computational cost. The best trade-off between accuracy and training time was reached when employing 128 filters.

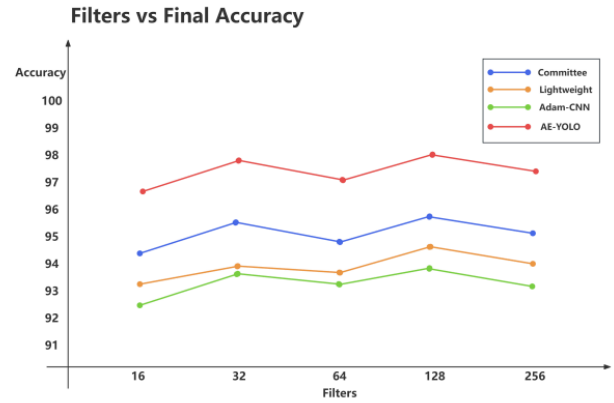

Figure 2. Final accuracy under different filter settings

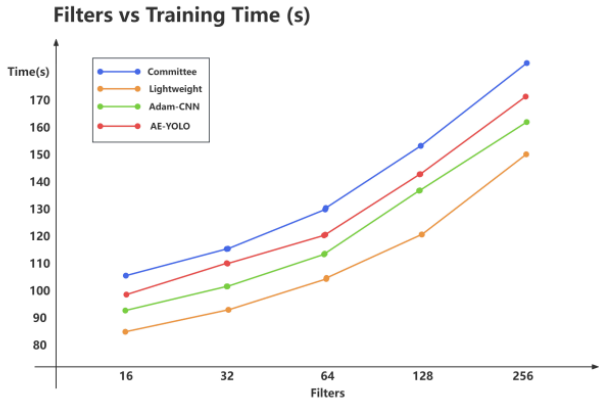

Figure 3. Training time under different different filter settings

4.2.2. Effect of kernel size

Accuracy and training time was afterwards considered regarding the effect of kernel size. The results are presented in Figure 4 and Figure 5.

It can be observed in Figure 4 that AE-YOLO obtained a score of 0.966, 0.970, 0.979, 0.962, and 0.954 on the accuracy of kernel sizes (3x3), (4x4), (5x5), (10x10), and (15x15) respectively where the highest was 5x5. Also, the three baseline models showed the same trend. In every setting, AE-YOLO always performed better than all baselines.

Figure 5: As the kernel size increased, training time also decreased. The training time of AE-YOLO was minimized by 114 seconds (3x3) and 95 seconds (15x15). All the baseline models showed a similar tendency.

The results indicate that an average kernel size boosts accuracy and excessive large kernels reduce the accuracy but are trained in less time. The optimal trade-off was observed with 5x5.

DOI: 10.54254/2755-2721/2026.33691

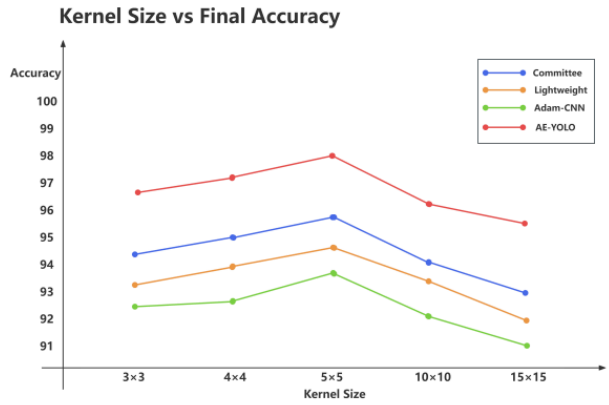

Figure 4. Final accuracy under different kernel-size settings

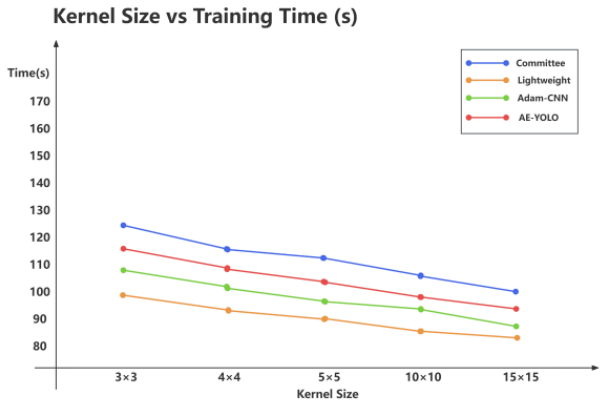

Figure 5. Training time under different kernel-size settings

4.2.3. Effect of dropout rate

Subsequently, the effect of the dropout rate was assessed. The results are presented in Figure 6 and Figure 7.

As illustrated in Figure 6, the accuracy results of AE-YOLO were 0.971, 0.976, 0.981, 0.977, and 0.979 when the dropout rate was set to {0.1,0.2,0.3,0.4,0.5} respectively and the highest accuracy was recorded at 0.3. The three baseline models had the same trend and peaked at 0.3. AE-YOLO was always better than all the baselines in the five contexts.

Training time was somewhat different between the dropout rates. In the case of AE-YOLO, training time varied between 102 seconds and 113 seconds. The same trends were seen with the three baseline models.

These findings mean that the best dropout percentage is minimal and it will not have much impact on time cost when a moderate value is used to obtain a good model generalization. In this research the best value was 0.3.

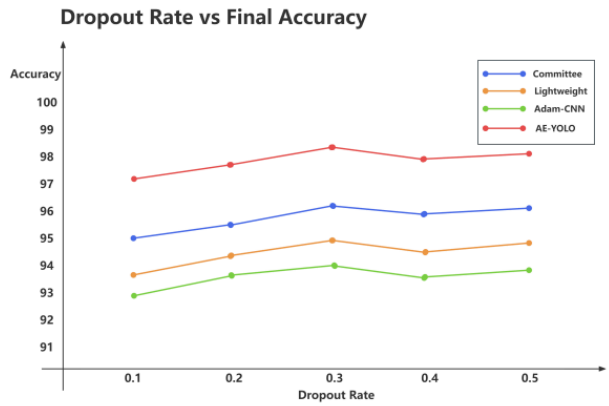

Figure 6. Final accuracy under different dropout-rate settings

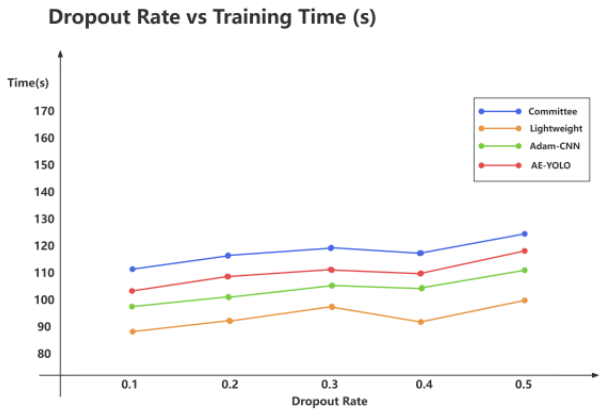

Figure 7. Training time under different dropout-rate settings

Owing to space constraints, the rest of the experimental findings are given in the technical report that follows [21].

DOI: 10.54254/2755-2721/2026.33691

4.2.4. Robustness validation results

Besides the control experiment, the robustness was measured based on the road signs found on the internet. AE-YOLO showed stable performance tendencies in terms of accuracy and time, meaning that the suggested solution is resistant to minor distribution changes. The consequence of this result is that AE-YOLO can be applied in a benchmark environment and possibly applied to a wider range of scenarios involving road signs.

An example of qualitative data can be seen in Figure 10 where AE-YOLO was able to identify and label all of the visible traffic signs, including a speed limit sign, with a high level of certainty. This example also shows that the proposed method can maintain the practical recognition ability even in more realistic urban-road settings.

Figure 8. Qualitative detection example of AE-YOLO on a real-world road scene outside the benchmark dataset

5. Conclusion

This article has presented in this paper AE-YOLO, a traffic sign recognition framework based on YOLOv8, with an adaptive early-stopping method. The results of experiments conducted within four different groups of parameters prove that AE-YOLO always enhances the trade-off between accuracy and efficiency and additionally offers more robustness under different kinds of traffic signs.

The future efforts will be directed at enhancing backbone feature extraction, increasing multi-scale feature fusion, and investigating lightweight model compression methods like pruning and architecture optimisation in order to enhance efficiency and generalisation still more.

References

[1] Flores-Calero, M., Astudillo, C. A., Guevara, D., Maza, J., Lita, B. S., Defaz, B., Ante, J. S., Zabala-Blanco, D., & Armingol Moreno, J. M. (2024). Traffic sign detection and recognition using YOLO object detection algorithm: A systematic review. Mathematics, 12(2), 297.

[2] Soylu, E., & Soylu, T. (2024). A performance comparison of YOLOv8 models for traffic sign detection in the Robotaxi-full scale autonomous vehicle competition. Multimedia Tools and Applications, 83(8), 25005-25035.

[3] Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2011, July). A committee of neural networks for traffic sign classification. In The 2011 international joint conference on neural networks (pp. 1918-1921). IEEE.

DOI: 10.54254/2755-2721/2026.33691

[4] Zhang, J., Wang, W., Lu, C., Wang, J., & Sangaiah, A. K. (2020). Lightweight deep network for traffic sign classification. Annals of Telecommunications, 75(7), 369-379.
[5] Mehta, S., Paunwala, C., & Vaidya, B. (2019, May). CNN based traffic sign classification using Adam optimizer. In 2019 international conference on intelligent computing and control systems (ICCS) (pp. 1293-1298). IEEE.
[6] Saadna, Y., & Behloul, A. (2017). An overview of traffic sign detection and classification methods. International journal of multimedia information retrieval, 6(3), 193-210.
[7] De La Escalera, A., Moreno, L. E., Salichs, M. A., & Armingol, J. M. (1997). Road traffic sign detection and classification. IEEE transactions on industrial electronics, 44(6), 848-859.
[8] Mishra, J., & Goyal, S. (2022). An effective automatic traffic sign classification and recognition deep convolutional networks. Multimedia tools and applications, 81(13), 18915-18934.
[9] Yang, Y., Luo, H., Xu, H., & Wu, F. (2015). Towards real-time traffic sign detection and classification. IEEE Transactions on Intelligent transportation systems, 17(7), 2022-2031.
[10] Zaklouta, F., Stanciulescu, B., & Hamdoun, O. (2011, July). Traffic sign classification using kd trees and random forests. In The 2011 international joint conference on neural networks (pp. 2151-2155). IEEE.
[11] Ji, B., Xu, J., Liu, Y., Fan, P., & Wang, M. (2024). Improved YOLOv8 for small traffic sign detection under complex environmental conditions. Franklin Open, 8, 100167.
[12] Mercaldo, F., Martinelli, F., & Santone, A. (2024). Real-time road sign localisation through object detection. Procedia Computer Science, 246, 30-37.
[13] Aldoski, Z. N., & Koren, C. (2025). Traffic sign detection and quality assessment using YOLOv8 in daytime and nighttime conditions. Sensors, 25(4), 1027.
[14] Yaseen, M. (2024). What is YOLO, M. Y. W. An in-depth exploration of the internal features of the next-generation object detector., 2024. DOI: https: //doi. org/10.48550/arXiv, 2408.
[15] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine learning and knowledge extraction, 5(4), 1680-1716.
[16] Huang, L., Tan, Y., Li, W., Shan, S., Liu, L., Shen, L., ... & Niu, Y. (2025). YOLO-PRO: Enhancing Instance-Specific Object Detection with Full-Channel Global Self-Attention. arXiv preprint arXiv: 2503.02348.
[17] Hussein, B. M., & Shareef, S. M. (2024). An empirical study on the correlation between early stopping patience and epochs in deep learning. In ITM Web of Conferences (Vol. 64, p. 01003). EDP Sciences.
[18] Kuhse, D., Teper, H., Buschjäger, S., Wang, C. Y., & Chen, J. J. (2025). You only look once at anytime (AnytimeYOLO): Analysis and optimization of early-exits for object-detection. arXiv preprint arXiv: 2503.17497.
[19] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks, 32, 323-332.
[20] Scholte, D., Zwemer, M. H., & Bondarev, E. (2025, September). Accelerating YOLO with EOBranch: An Early Exit Approach for Adaptive Object Detection. In International Conference on Image Analysis and Processing (pp. 442-455). Cham: Springer Nature Switzerland.
[21] Zhu, C. (2026). Advanced Adaptive Early-Stopping YOLO for Traffic Sign Recognition(Technical report).GitHub. [https: //github.com/chenjunzhu4-ctrl/Adaptive-Early-Stopping-YOLO-for-Road-Sign-Recognition/blob/main/Technical%20report.pdf]