Sensors and Materials, Vol. 36, No. 4 (2024) 1605–1625
MYU Tokyo

S & M 3621

Vehicle Detection on Express Roads Using YOLOv7 with Taguchi Parameter Optimization Method

Mei-Kuei Chen,[1] Chun-Lung Chang,[2] Cheng-Jian Lin,[3*] and Wen-Jong Chen[1]

[1]Department of Industrial Education and Technology, National Changhua University of Education,
Changhua County 50007, Taiwan
[2]Department of Artificial Intelligence and Computer Engineering, National Chin-Yi University of Technology,
Taichung City 41170, Taiwan
[3]Department of Computer Science & Information Engineering, National Chin-Yi University of Technology,
Taichung City 41170, Taiwan

(Received October 20, 2023; accepted March 26, 2024)

Keywords: vehicle detection, YOLOv7, Taguchi method, evaluation metrics, hyperparameters

In road traffic management, high-speed vehicle detection is often affected by factors such as vehicle speed, weather, camera angle, and image resolution, making vehicle detection on express roads very challenging, Therefore, we propose a Taguchi-based You Only Look Once (YOLOv7) model, called T-YOLOv7, for vehicle detection on high-speed roads. The Taguchi method is used to optimize the combination of hyperparameters of YOLOv7. Experimental results show the precision rate, recall rate, and F1-score of the proposed T-YOLOv7 to be 82.2, 86.3, and 84.2%, respectively. Compared with the original YOLOv7, T-YOLOv7 has improved precision, recall, and F1-score by 12.1, 17.9, and 15.0 percentage points, respectively. Compared with YOLOv4, the improvements in precision, recall, and F1-score using T-YOLOv7 are 4.0, 2.4, and 3.3 percentage points, respectively. The experimental results also show that the proposed T-YOLOv7 is effective in adjusting hyperparameters through the Taguchi method and can be applied to real-time vehicle detection in real environments.

1. Introduction

The rapid advancement of technology has greatly contributed to the development and improvement of intelligent transportation systems. To enhance the efficiency and safety of traffic management, vehicle detection technology is used for the real-time monitoring of road environments. Real-time vehicle detection on expressways is essential for improving road safety, especially during adverse weather conditions and at night when vehicles may be speeding. However, real-time vehicle detection remains a major challenge in any traffic monitoring system owing to factors such as varying vehicle speed, size, shape, and color, as well as the direction of surveillance cameras and weather conditions.(1)

Machine learning technologies have been widely used in object recognition, face and facial expression recognition, and defect detection. Specifically, convolutional neural networks (CNNs)

*Corresponding author: e-mail: cjlin@ncut.edu.tw

https://doi.org/10.18494/SAM4718

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

have demonstrated excellent performance in object detection and recognition tasks, making them an integral part of deep learning and artificial intelligence.(2) Object detection methods based on CNNs can be broadly categorized into two-stage detection methods such as Fast Region-based CNN (R-CNN)(3,4) and Faster R-CNN(5,6) and single-stage detectors such as Single Shot MultiBox Detector (SSD), RetinaNet, and the You Only Look Once (YOLO) series. To train a YOLO model, machine learning techniques are used to train the hyperparameters of the model. While two-stage detectors often achieve higher precision, they require more parameters and have lower execution speed,(7) making them unsuitable for real-time vehicle detection on edge CPUs or embedded devices. On the other hand, single-stage detectors are easier to train, more computationally efficient, and better suited for edge computing. SSD(8) is an effective multitarget detection method, and RetinaNet(9) and the YOLO series(10–13) have also achieved impressive results in target detection algorithms. Among them, YOLOv7 stands out for its high precision and high processing speed.(14)

Properly defining hyperparameters is crucial to avoiding unnecessary time consumption during model training, as training speed and final model performance can significantly vary depending on the hyperparameter adjustments. Therefore, various research efforts have been focused on hyperparameter optimization strategies. Grid search is the simplest and most basic method for hyperparameter tuning, where the performance of all parameter combinations is evaluated to select the best set of hyperparameters. Although this method is comprehensive and straightforward, as the number of parameters increases slightly, the search time grows exponentially, making it time-consuming.(15) Other simple search strategies include random search or more complex approaches such as Bayesian optimization.(16) Reparameterization is a commonly used technique in deep learning, where reparameterizing a model can improve its performance without increasing complexity.(17) It involves fine-tuning the existing model weights, adjusting hyperparameters, or changing the model structure to achieve better performance in new tasks or environments.(18) However, not all proposed reparameterization modules can be perfectly applied to different architectures.(19)

In recent years, researchers have widely adopted the Taguchi method in the field of deep learning. The integration of the Taguchi method with deep learning provides researchers with a novel approach to optimizing the parameter settings of deep learning models more effectively, thus improving their performance and efficiency. Yu *et al.*(20) designed hyperparameter combinations to determine the optimal configuration of neural networks for radio network spectrum prediction. Demir *et al.* confirmed that the Taguchi method(21) not only reduces process time and cost but also provides valuable data optimization and improves model efficiency and robustness. Zhao *et al.*(22) developed an optimized deep learning model architecture using the Taguchi method. This architecture was applied to real traffic flow data collected from highways in Birmingham, UK. Because of the aforementioned advantages of the Taguchi method, in this study, we use it to design experiments and optimize the training hyperparameters of the YOLOv7 model. By applying the optimal parameter combinations determined through Taguchi experiments, the proposed Taguchi-based YOLOv7 (T-YOLOv7) model is intended to overcome the limitations and improve the detection precision of YOLOv7, thereby enhancing its performance and robustness. In the experiment, the Beijing Institute of

Technology (BIT)-vehicle public dataset was utilized for vehicle classification.(23) Additionally, a large amount of traffic video footage (as shown in Fig. 1) was collected from Taiwan's Highway 74 to create the T-74 vehicle dataset, enabling the application of T-YOLOv7 vehicle detection technology in real-world scenarios to address critical issues concerning highway traffic. Therefore, the main contributions of this study are as follows.

(1) The proposed T-YOLOv7 is a combination of YOLOv7 and the Taguchi method to cope with the problems of parameter selection and high precision.

(2) The proposed T-YOLOv7 slightly outperforms YOLOv4 and considerably outperforms YOLOv7 in precision, recall, and F1-score measures. These results demonstrate that the proposed T-YOLOv7 model with Taguchi optimization showed performance superior to the original YOLOv7 and YOLOv4 in object detection tasks.

The rest of this paper is organized as follows. We present the research methodology in Sect. 2, including the experimental strategy and the overall framework for optimizing the YOLOv7 model based on the Taguchi method. In Sect. 3, we introduce the BIT-vehicle and T-74 vehicle datasets and compare the performance of the T-YOLOv7 with the original YOLOv7 and YOLOv4 models. Finally, Sect. 4 is the conclusion.

2. Materials and Methods

A network with strong representation ability can capture complex patterns, enabling the accurate recognition of objects in image recognition. However, such a network's power comes from its deep structure, which can result in heavy parameterization, large memory usage, and slow execution.(24) In this study, we propose the use of T-YOLOv7 with Taguchi parameter optimization for vehicle detection on expressways to improve classification precision. Figure 2 shows an overview of the execution strategy adopted in this study. We first utilize the Taguchi method for parameter design to obtain the optimal parameters for optimizing YOLOv7. Then, we conduct experiments using T-YOLOv7 with Taguchi parameter optimization on two traffic datasets: the BIT-vehicle dataset and T-74 vehicle dataset.

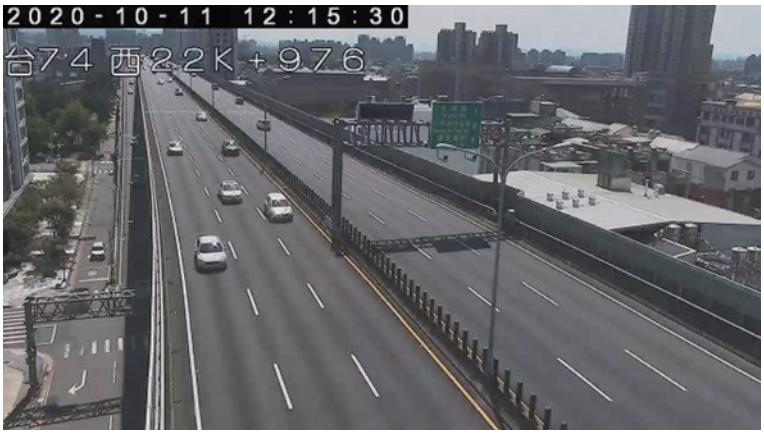


Fig. 1. (Color online) Video footage of T-74 vehicle dataset.

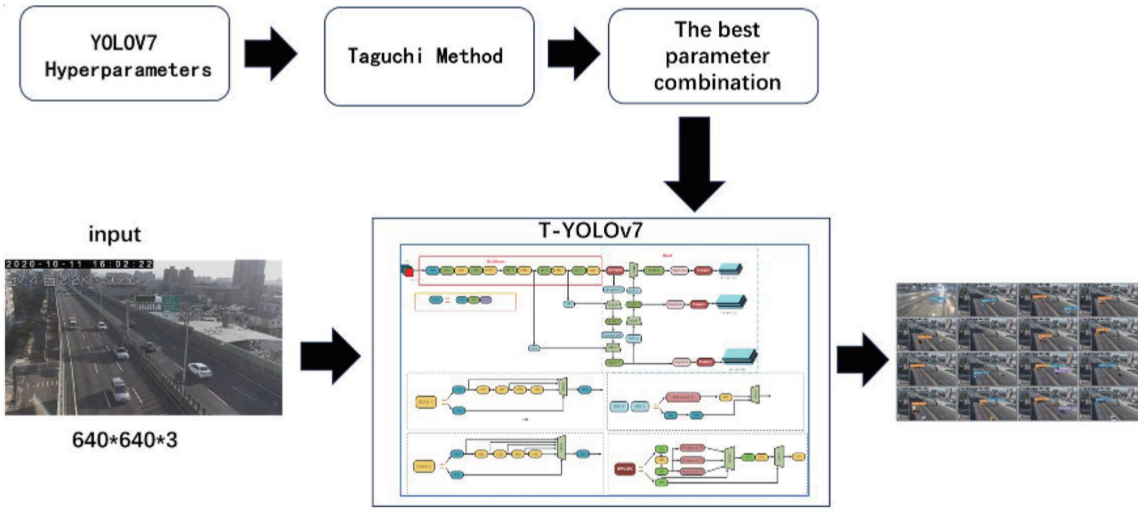


Fig. 2. (Color online) Overall architecture of the proposed vehicle detection system.

2.1 YOLO

YOLO is a target detection method in deep learning proposed by Redmon and Farhadi.(10) This method optimizes two-stage detection using a single network in an end-to-end manner. By predicting bounding boxes and class probabilities for the entire image, we can treat the problem of detecting target objects is treated as a regression problem of spatially separated bounding boxes and associated class probabilities,(25) which enhances object detection and recognition in augmented reality. The YOLOv3 network architecture has been updated to utilize multiscale prediction. It automatically extracts features through specific combinations such as Darknetconv2D + BatchNormalization + LeakyReLU activation function, upsampling, concatenation, and convolution, achieving a runtime of only 22 ms at a resolution of 320 × 320.(13) The model can process images in real time at a very high speed. The YOLOv4 architecture mainly consists of the backbone, neck, and head, as shown in Fig. 3. It does not use Feature Pyramid Networks (FPN), unlike YOLOv3. Instead, YOLOv4 combines new features such as Weighted Residual Connections (WRC), Cross Stage Partial (CSP) connections, Cross mini-Batch Normalization (CmBN), Self-adversarial Training (SAT), Mish activation, mosaic data augmentation, CmBN, DropBlock regularization, and Complete-Intersection over Union (CIoU) loss.(14) The Spatial Pyramid Pooling (SPP) module is used to increase the receptive field and extract the most important contextual features in SPDarknet53. The path Aggregation Network (PANet) is selected as the method for aggregating parameters from different detectors. This not only does not reduce the network's runtime speed but also applies to large- and small-scale datasets, addressing most of the issues in model detection.

Gong *et al.*(26) proposed a lightweight version of YOLOv4, incorporating depthwise separable convolutions instead of the original backbone network, to reduce computational complexity and model size, thereby improving real-time traffic detection. Lin *et al.*(27) conducted research on YOLOv4 and improved it by clustering labels using the K-means method to enhance localization

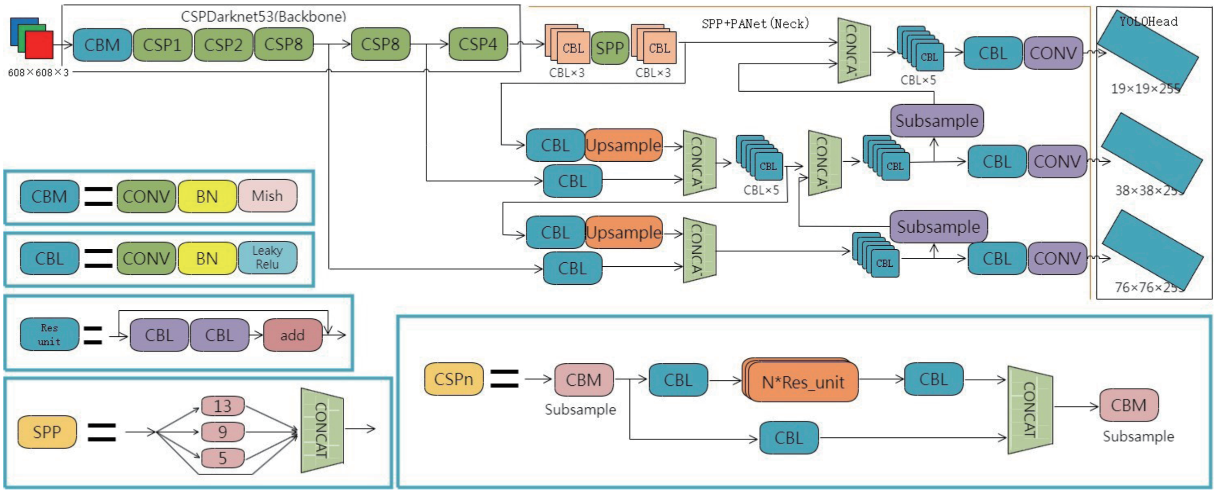


Fig. 3. (Color online) YOLOv4 network architecture.

precision while reducing the parameter count, making it suitable for the detection of small objects such as traffic signs.

YOLOv7 not only addresses the replacement of reparameterized modules with original modules and the allocation strategy for dynamic label handling in different output layers but also reduces the number of parameters by 40% and the amount of computation by 50% through the use of extension and compound scaling methods. Its performance has made it the most advanced detector for real-time object detection.(4) The YOLOv7 network architecture is shown in Fig. 4.

2.2 Taguchi method

The Taguchi method was used in this study to determine the optimal combination of hyperparameters for the optimized model. Figure 5 shows the Taguchi method process. The following optimization steps are involved in this method.

- Determine the control factors and levels.
- Design an orthogonal array.
- Conduct experiments.
- Analyze the experimental results.
- Validate the experimental results.

In this experiment, effective influencing factors were selected for hyperparameter design in YOLOv7. The eight selected factors were the initial learning rate (lr0), momentum (mo), weight_decay parameter (wt), warmup_epochs (w_epc), weights for the bounding box loss (box), weights for classification loss (cls), objectness loss (obj), and iou training threshold (iou_t). Two levels were set for each factor, as shown in Table 1. For the above numbers of factors and levels, each factor has its own characteristics.

The Taguchi method is used to design the hyperparameter tuning structure in order to reduce the uncertainty caused by certain assumptions and the time consumed by trial-and-error methods. With these selected number of factors and levels, conducting a full-factor experiment

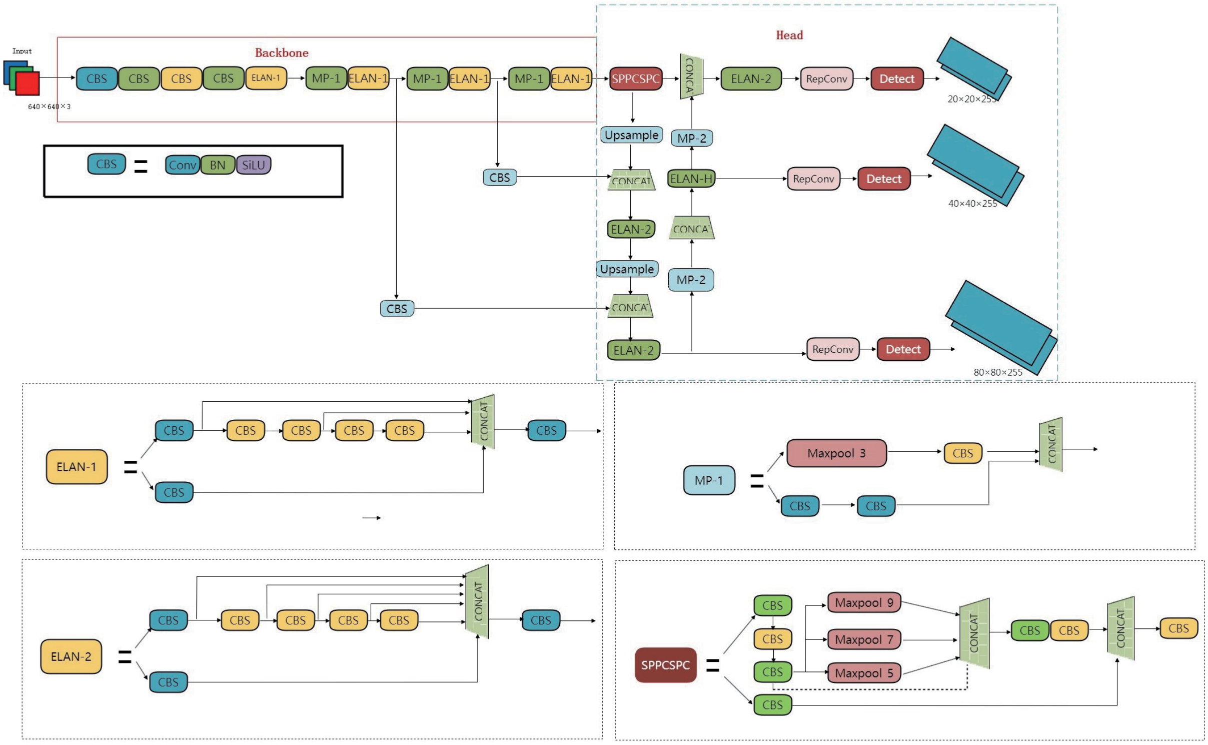


Fig. 4. (Color online) YOLOv7 network architecture.

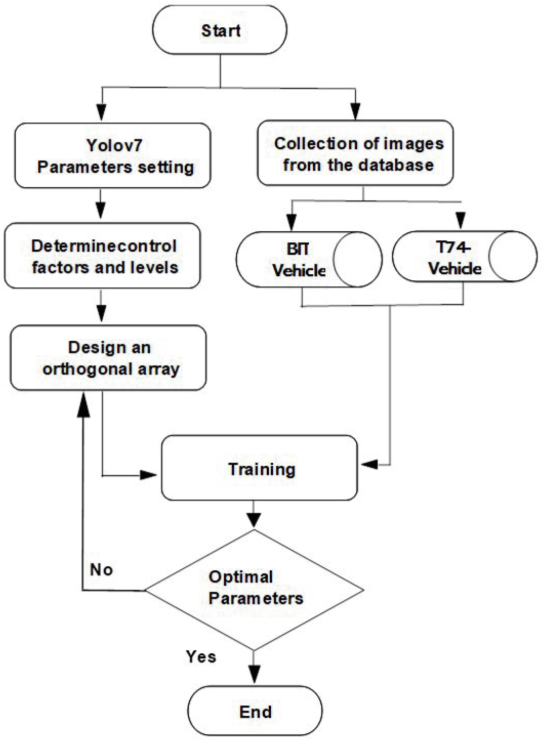

Start
Yolov7 Parameters setting
Collection of images from the database
Determinecontrol factors and levels
BIT Vehicle
T74-Vehicle
Design an orthogonal array
Training
Optimal Parameters
No
Yes
End

Fig. 5. Taguchi method process.

Table 1
Levels of influencing factors.

Factors	Level 1	Level 2
lr0	0.005	0.02
mo	0.8	0.99
wt_	0.0001	0.001
w_epc	2	4
box	0.01	0.1
cls	0.1	0.5
obj	0.5	0.9
iou_t	0.1	0.5

would require 256 trials (2^8) to find the optimal parameters. However, by applying the Taguchi method, results can be obtained with a smaller number of experiments. In our experiment, we used an L16 orthogonal array, which only required 16 trials.

2.3 Evaluation and validation metrics

To evaluate the output results of the model, the following metrics were adopted: mean average precision (mAP), precision, recall, and F1-score. These evaluation metrics can be calculated as follows.

$$\text{mAP} = \frac{1}{N}\sum_{k=1}^{N} AP_k \tag{1}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3}$$

$$\begin{aligned}\text{F1-score} &= \frac{2}{\dfrac{1}{Precision}+\dfrac{1}{Recall}} \\ &= \frac{2Precision \times Recall}{Precision + Recall} \\ &= \frac{2TP}{2TP+FP+FN}\end{aligned} \tag{4}$$

Here, N represents the number of classes and AP_k represents the average precision (AP) for the kth class. The model is evaluated using precision (P) and recall (R) for each class. TP, FP, TN, and FN represent true positive, false positive, true negative, and false negative, respectively. AP represents the average precision across multiple classes, mAP represents the average value of APs, and F1-score is the harmonic mean of precision and recall. F1-score is a critical evaluation metric in object detection as it serves as a criterion to assess the performance of the model.

3. Experimental Results

We design several experiments to evaluate the effectiveness of the proposed T-YOLOv7 model. Two datasets, the publicly available BIT-vehicle dataset and a custom T-74 vehicle dataset, are used. Then, the experimental setup of the T-YOLOv7 model is described. Experimental results show that the proposed T-YOLOv7 model outperforms the traditional YOLOv7 and is comparable to other similar models.

3.1 Datasets

Two datasets, the publicly available BIT-vehicle dataset and a custom T-74 vehicle dataset, were utilized. The BIT-vehicle dataset was curated by BIT and serves as a public dataset for vehicle detection. The dataset consists of 9850 vehicle images captured by two cameras at different times and locations on highways. It includes six vehicle types: sedan, SUV, bus, minibus, van, and truck. The distribution of vehicle types and quantities for each category are listed in Table 2. The dataset exhibits variations in lighting condition, scale, vehicle surface color, and viewpoint. It is a high-resolution image dataset. For this research, a total of 2400 images from the BIT-vehicle dataset were used in the experiment. Two hundred randomly selected images from each category were used for training, 100 images for validation, and 100 images for testing.

The T-74 vehicle traffic dataset consists of images captured on the Taiwan Route 74 expressway at different sections, times, locations, and angles. It takes into account various times, including morning, noon, afternoon, evening, and night; weather conditions including rainy weather; and lens flare. The dataset contains 149848 vehicle images categorized into five types: sedan, truck, scooter, bus, and Hlinkcar. The distribution of vehicle types and quantities for each category are listed and the numbers of images of each vehicle type used for training, validation, and testing are given in Table 3. These images were used to evaluate the performance and effectiveness of our T-YOLOv7 detection model.

Table 2
Number of vehicles in the training and testing phases of the BIT-vehicle dataset.

Vehicle type	SUV	Sedan	Microbus	Minivan	Truck	Bus
Quantity	400	400	400	400	400	400
Train	200	200	200	200	200	200
Validation	100	100	100	100	100	100
Test	100	100	100	100	100	100

Table 3
Number of vehicles in the training and testing phases of the T-74 vehicle datasets.

Vehicle type	Sedan	Truck	Scooter	Bus	Hlinkcar
Quantity	3400	1200	1200	1200	500
Train	200	200	200	200	200
Validation	200	200	200	200	200
Test	3000	800	800	400	100

3.2 Experimental setup

We implemented T-YOLOv7 on PyTorch 1.10.1 and trained and tested the model using NVIDIA GeForce RTX 3060. The model was trained for 300 epochs on datasets with an image size of 640 × 640. A batch size of 16 was used during testing. The experimental setup is shown in Table 4.

3.3 Impact of control factors on experimental results determined using Taguchi method

3.3.1 Determination of optimal parameters

In this study, the Taguchi method was used to determine the impact of each control factor on the experimental results, as evaluated on the basis of the signal-to-noise ratio (*S*/*N*). The Taguchi method categorizes the quality characteristics of experimental results into three types: smaller-the-better, larger-the-better, and nominal-the-best. In this study, the observed value was the precision of the deep CNN predictions, so the larger-the-better characteristic was chosen. To ensure the stability and improve the reliability of the experiments, three experimental runs were conducted for each parameter combination in the orthogonal array. The *S*/*N* was calculated on the basis of the three precision values obtained for each combination, as shown in Table 5. The formula for calculating *S*/*N* is as follows:

$$S / N = -10 \log \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{y_i^2} \right), \tag{5}$$

where n is the number of experiments and y_i represents the i-th experimental data.

On the basis of the larger-the-better characteristic, *S*/*N* was obtained for each factor and level combination from the 16 experimental results. Table 6 shows the *S*/*N* ratio, significance ranking, optimal level, and optimal parameter combination for each factor. Figure 6 shows the mean main effects of the *S*/*N* ratio, where a larger difference in *S*/*N* indicates a greater influence of the corresponding factor. The significance ranking represents the ranking of the impact of each factor. A larger difference in *S*/*N* indicates a greater influence of a particular factor. The final optimal parameter combination is as follows:
lr0 = 0.005, mo = 0.8, wt = 0.0001, W_epc = 2, box = 0.01, cls = 0.1, obj = 0.5, iou_t = 0.1.

Table 4
Experimental setup.

Windows	11
PyTorch	1.10.0
CPU	12th Gen Intel(R) Core(TM) i7-12700F 2.10 GHz
GPU	Nvidia GeForce RTX 3060
CUDA	11.4
cudnn	11.4

Table 5
Precision and *S*/*N* for different factor and level combinations.

Run#	Factor								Result				
	lr0	mo	wt	w_epc	box	cls	obj	iou_t	Y_1 (%)	Y_2 (%)	Y_3 (%)	Y_{ave} (%)	*S*/*N* (Y)
1	0.005	0.8	0.0001	2	0.01	0.1	0.5	0.1	89.0	89.6	88.6	89.1	−1.006
2	0.005	0.8	0.0001	4	0.01	0.5	0.9	0.5	88.0	87.6	88.0	87.9	−1.124
3	0.005	0.8	0.001	2	0.1	0.1	0.9	0.5	65.6	64.4	61.2	63.7	−3.924
4	0.005	0.8	0.001	4	0.1	0.5	0.5	0.1	81.2	80.4	81.0	80.9	−1.845
5	0.005	0.99	0.0001	2	0.1	0.5	0.5	0.5	68.2	64.0	65.6	65.9	−3.628
6	0.005	0.99	0.0001	4	0.1	0.1	0.9	0.1	68.8	72.6	72.6	71.3	−2.943
7	0.005	0.99	0.001	2	0.01	0.5	0.9	0.1	61.2	62.0	57.8	60.3	−4.401
8	0.005	0.99	0.001	4	0.01	0.1	0.5	0.5	87.4	88.6	88.2	88.1	−1.104
9	0.02	0.8	0.0001	2	0.1	0.5	0.9	0.1	69.8	67.4	67.2	68.1	−3.337
10	0.02	0.8	0.0001	4	0.1	0.1	0.5	0.5	70.0	71.0	65.6	68.9	−3.255
11	0.02	0.8	0.001	2	0.01	0.5	0.5	0.5	74.2	74.8	76.8	75.3	−2.471
12	0.02	0.8	0.001	4	0.01	0.1	0.9	0.1	84.0	86.4	86.4	85.6	−1.353
13	0.02	0.99	0.0001	2	0.01	0.1	0.9	0.5	80.6	83.2	80.6	81.5	−1.783
14	0.02	0.99	0.0001	4	0.01	0.5	0.5	0.1	60.0	60.0	60.6	60.2	−4.408
15	0.02	0.99	0.001	2	0.1	0.1	0.5	0.1	78.8	76.2	76.4	77.1	−2.258
16	0.02	0.99	0.001	4	0.1	0.5	0.9	0.5	8.8	12.4	7.6	9.6	−20.88

Table 6
Factor significance ranking and optimal parameters.

Factors	lr0	mo	wt	W_epc	box	cls	obj	iou_t
Level 1	−2.4969	−2.2894	−2.6855	−2.851	−2.2063	−2.2033	−2.4969	−2.6939
Level 2	−4.9681	−5.1756	−4.7795	−4.614	−5.2588	−5.2618	−4.9681	−4.7711
Difference	2.4712	2.8862	2.0940	1.763	3.0525	3.0585	2.4712	2.0772
Rank	4	3	6	8	2	1	4	7
Best level	1	1	1	1	1	1	1	1
Optimal parameter	0.005	0.8	0.0001	2	0.01	0.1	0.5	0.1

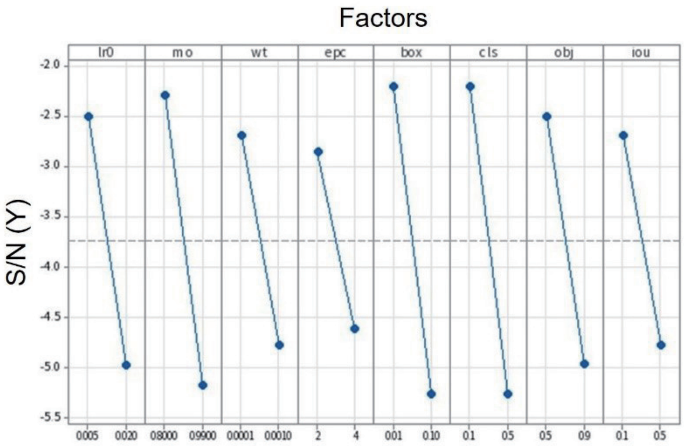


Fig. 6. (Color online) Main effects of S/N on factor levels.

3.3.2 Model training

In the experiments, the Taguchi method was utilized in the parameter design to determine the optimal parameter combination. The T-YOLOv7 model, optimized using the Taguchi method, and the original YOLOv7 were trained separately. As shown in Fig. 7, T-YOLOv7 began to converge after 100 epochs. Figures 8 and 9 depict the precision–recall (PR) curves for the

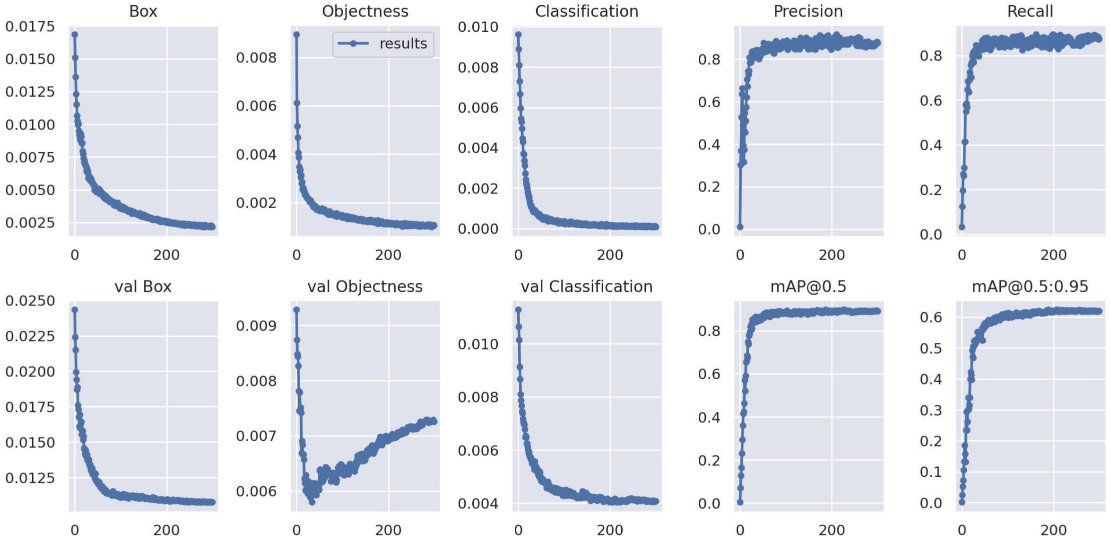


Fig. 7. (Color online) Training progress of T-YOLOv7 model.

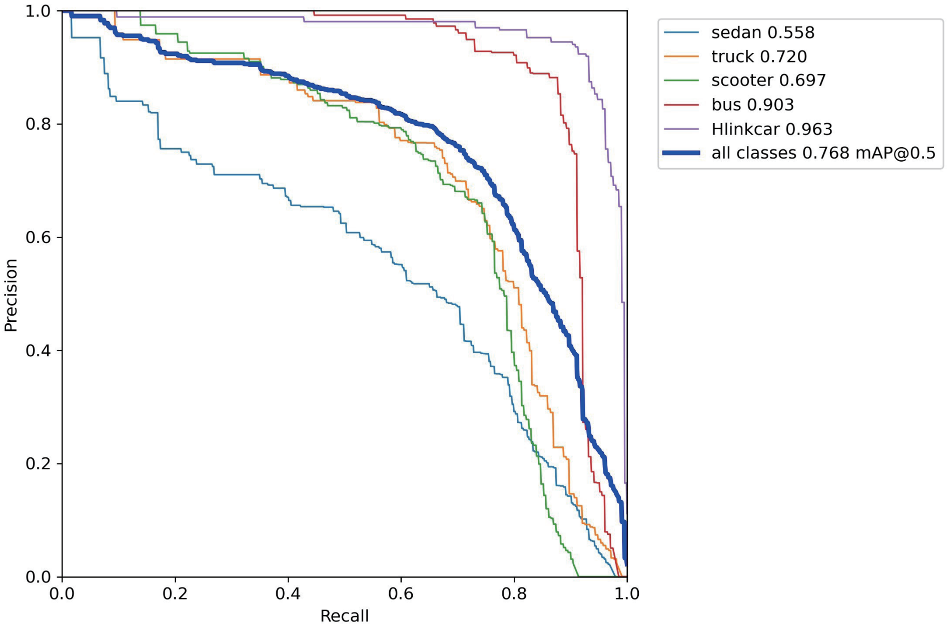


Fig. 8. (Color online) PR curves for training of original YOLOv7 model.

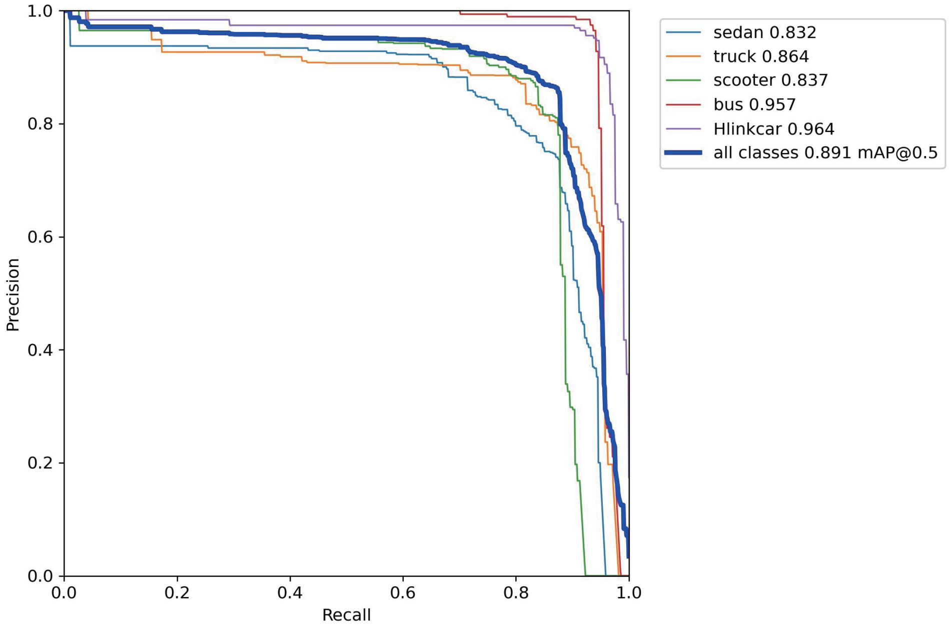


Fig. 9. (Color online) PR curves for training of T-YOLOv7 model.

original YOLOv7 and T-YOLOv7, respectively, demonstrating that T-YOLOv7, optimized with the Taguchi parameters, achieved a 12.3% improvement in precision compared with the original YOLOv7.

3.3.3 Results

The experimental results for the BIT-vehicle datasets are shown in Table 7. The overall precision, recall, F1-score, and mAP values are 96.7, 95.3, 96.0, and 98.3%, respectively. The experimental results for the T-74 vehicle datasets are shown in Table 8, with overall precision, recall, F1-score, and mAP values being 82.2, 86.3, 84.2, and 85.4%, respectively.

3.4 Comparison and analysis of experimental results

We evaluated T-YOLOv7 on Nvidia GeForce RTX 3060 GPU and compared the results with those of the original YOLOv7 and YOLOv4.

3.4.1 BIT-vehicle dataset

The experimental results are shown in Tables 9 and 10 and Figs. 10–12. As shown in Tables 9 and 10, T-YOLOv7 slightly outperforms YOLOv4 and considerably outperforms YOLOv7 in all four measures. These results demonstrate that the proposed T-YOLOv7 model with Taguchi optimization shows superior performance compared with the original YOLOv7 and YOLOv4 in object detection tasks.

Table 7
Experimental results of T-YOLOv7 on BIT-vehicle datasets.

	Class	Precision (%)	Recall (%)	F1-score (%)	mAP@.5 (%)
T-YOLOv7	All	96.7	95.3	96.0	98.3
	SUV	94.9	95.0	95.0	97.6
	Sedan	95.1	95.1	95.1	97.7
	Microbus	98.9	91.7	95.2	98.3
	Minivan	97.9	91.9	94.8	98.0
	Truck	94.2	98.0	96.1	98.6
	Bus	99.0	100.0	99.5	99.6

Table 8
Experimental results of T-YOLOv7 on T-74 vehicle datasets.

	Class	Precision (%)	Recall (%)	F1-score (%)	mAP (%)
T-YOLOv7	All	82.2	86.3	84.2	85.4
	sedan	91.7	74.6	82.3	88.7
	truck	78.2	85.1	81.5	86.8
	scooter	76.4	82.7	79.4	74.8
	bus	86.8	93.3	89.9	92.7
	Hlinkcar	78.1	95.9	86.1	83.9

Table 9
Experimental results for BIT-vehicle dataset with each model.

Model	Precision (%)	Recall (%)	F1-score (%)	mAP (%)
YOLOv4	94.7	93.5	94.1	97.9
YOLOv7	94.9	93.9	94.4	97.8
T-YOLOv7	96.7	95.3	96.0	98.3

Table 10
Results for various categories of BIT-vehicle dataset.

	Class	Precision (%)	Recall (%)	F1-score (%)	mAP (%)
YOLOv4	SUV	85.3	99.0	91.6	98.1
	Sedan	99.0	92.8	95.8	98.9
	Microbus	94.3	83.0	88.3	94.9
	Minivan	92.8	93.0	92.9	97.3
	Truck	96.9	94.3	95.6	98.7
	Bus	100.0	98.8	99.4	99.5
YOLOv7	SUV	97.8	88.6	93.0	97.7
	Sedan	93.2	98.0	95.5	98.9
	Microbus	89.9	91.0	90.4	95.6
	Minivan	91.9	93.0	92.4	97.4
	Truck	97.9	93.0	95.4	97.9
	Bus	98.9	100.0	99.4	99.6
T-YOLOv7	SUV	94.9	95.0	95.0	97.6
	Sedan	95.1	95.1	95.1	97.7
	Microbus	98.9	91.7	95.2	98.3
	Minivan	97.9	91.9	94.8	98.0
	Truck	94.2	98.0	96.1	98.6
	Bus	99.0	100.0	99.5	99.6

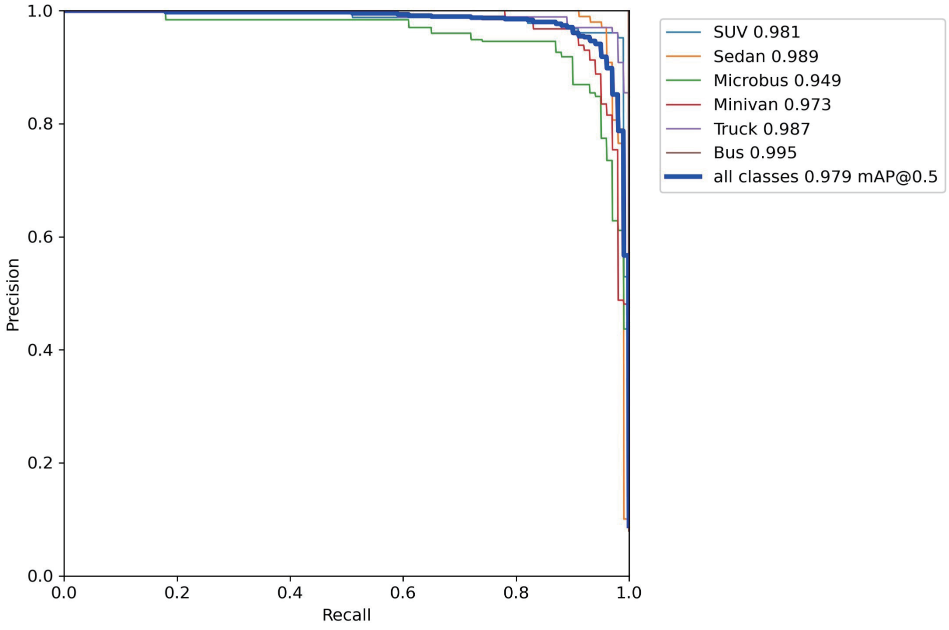


Fig. 10. (Color online) PR curves for YOLOv4 on BIT-vehicle dataset.

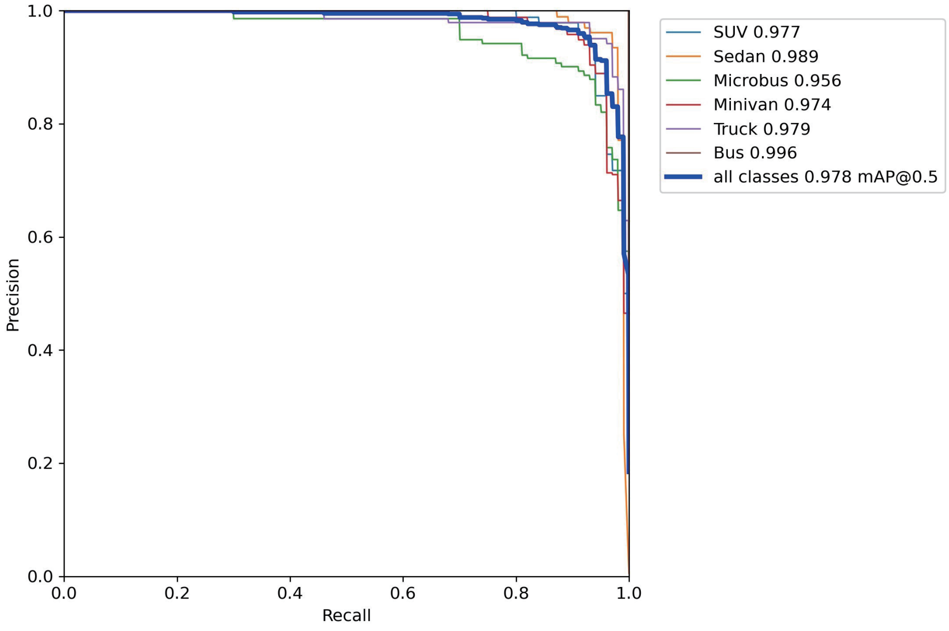


Fig. 11. (Color online) PR curves for original YOLOv7 on BIT-vehicle dataset.

3.4.2 T-74 vehicle dataset

The experimental results are shown in Tables 11 and 12, and Figs. 13–15 (PR curves). In Table 11, the T-YOLOv7 achieved a precision of 82.2%, whereas the original YOLOv7 and

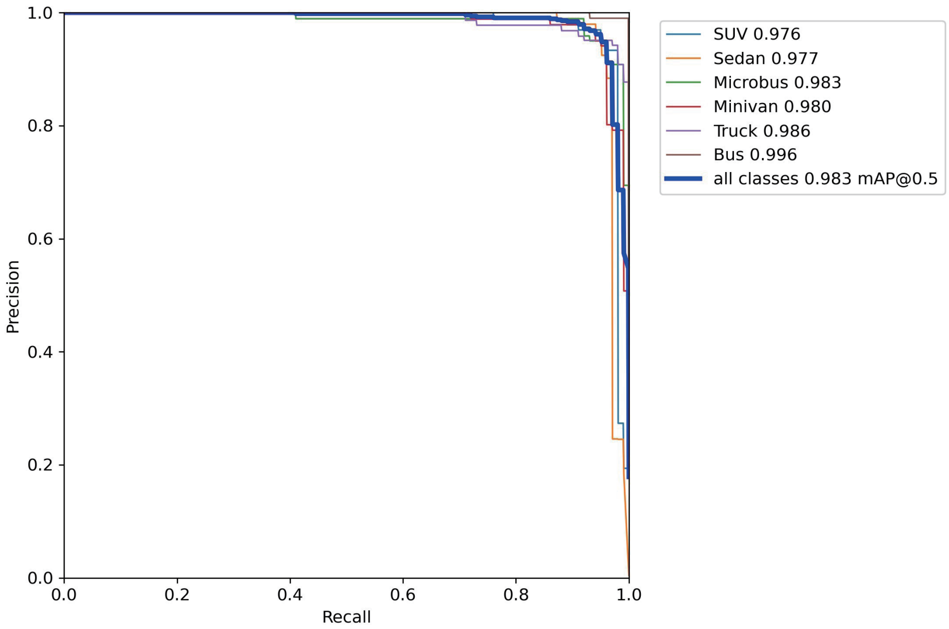


Fig. 12. (Color online) PR curves for T-YOLOv7 on BIT-vehicle dataset.

Table 11
Experimental results for T-74 vehicle dataset with each model.

Model	Precision (%)	Recall (%)	F1-score (%)	mAP (%)
YOLOv4	78.2	83.9	80.9	84.6
YOLOv7	70.1	68.4	69.2	74.9
T-YOLOv7	82.2	86.3	84.2	85.4

Table 12
Results for various categories of T-74 vehicle dataset.

	Class	Precision (%)	Recall (%)	F1-score (%)	mAP (%)
YOLOv4	sedan	89.5	70.6	78.9	87.2
	truck	74.2	81.8	77.8	85.3
	scooter	64.0	81.1	71.5	65.1
	bus	86.8	91.8	89.2	94.5
	Hlinkcar	76.8	94.2	84.6	90.7
YOLOv7	sedan	81.7	46.2	59.0	70.8
	truck	70.9	57.0	63.2	67.1
	scooter	60.6	60.5	60.5	58.6
	bus	71.4	84.1	77.2	85.9
	Hlinkcar	65.8	94.2	77.5	91.9
T-YOLOv7	sedan	91.7	74.6	82.3	89.1
	truck	78.2	85.1	81.5	86.8
	scooter	76.4	82.7	79.4	74.7
	bus	86.8	93.3	89.9	92.7
	Hlinkcar	78.1	95.9	86.1	83.9

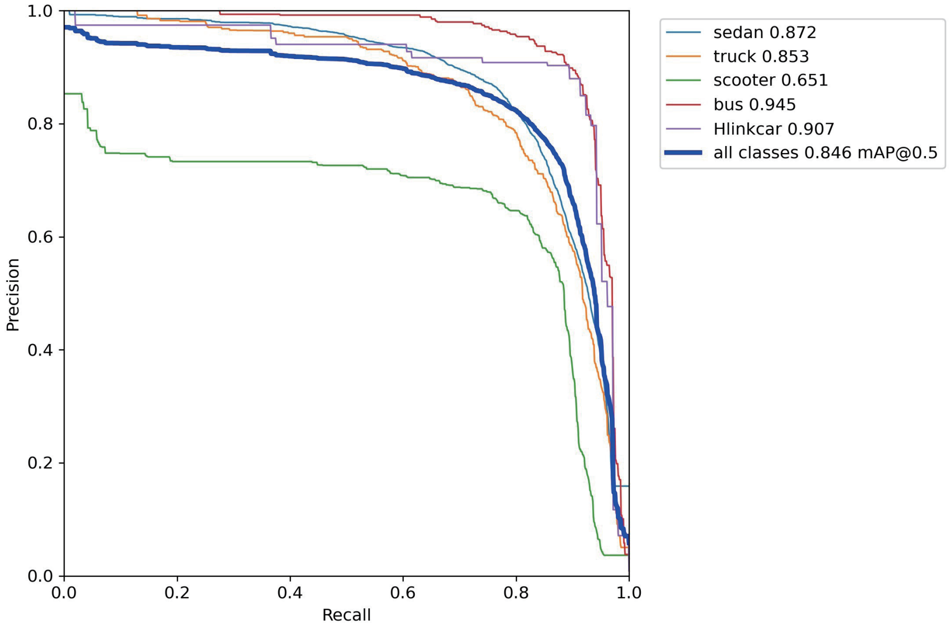


Fig. 13. (Color online) PR curves for YOLOv4 on T-74 vehicle datasets.

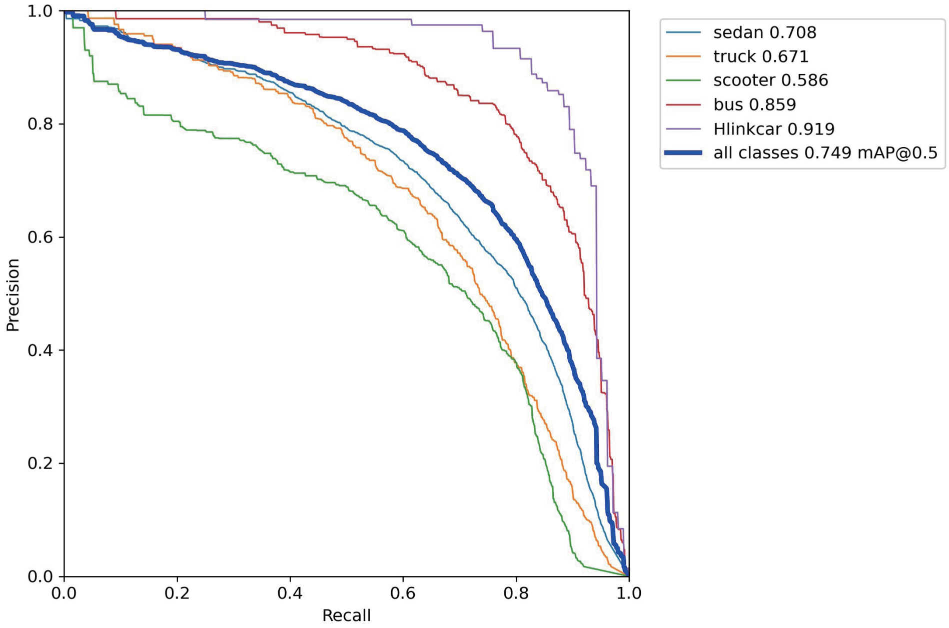


Fig. 14. (Color online) PR curves for original YOLOv7 on T-74 vehicle datasets.

YOLOv4 achieved accuracies of 70.1 and 78.2%, respectively. This indicates that T-YOLOv7 outperformed the original YOLOv7 and YOLOv4 by 12.1 and 4.0 percentage points in terms of precision, respectively. Regarding the recall results, T-YOLOv7 achieved a recall of 86.3%,

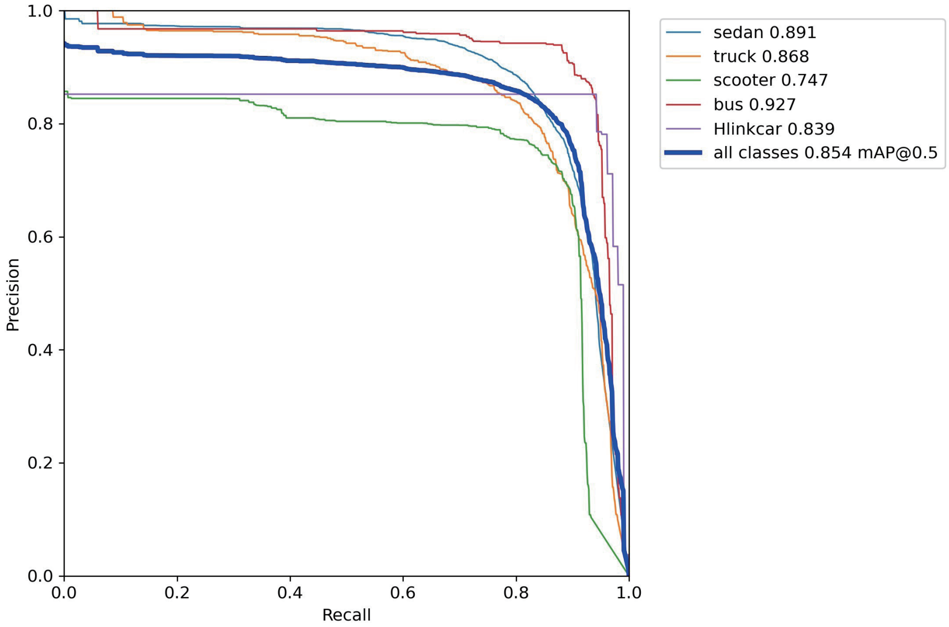
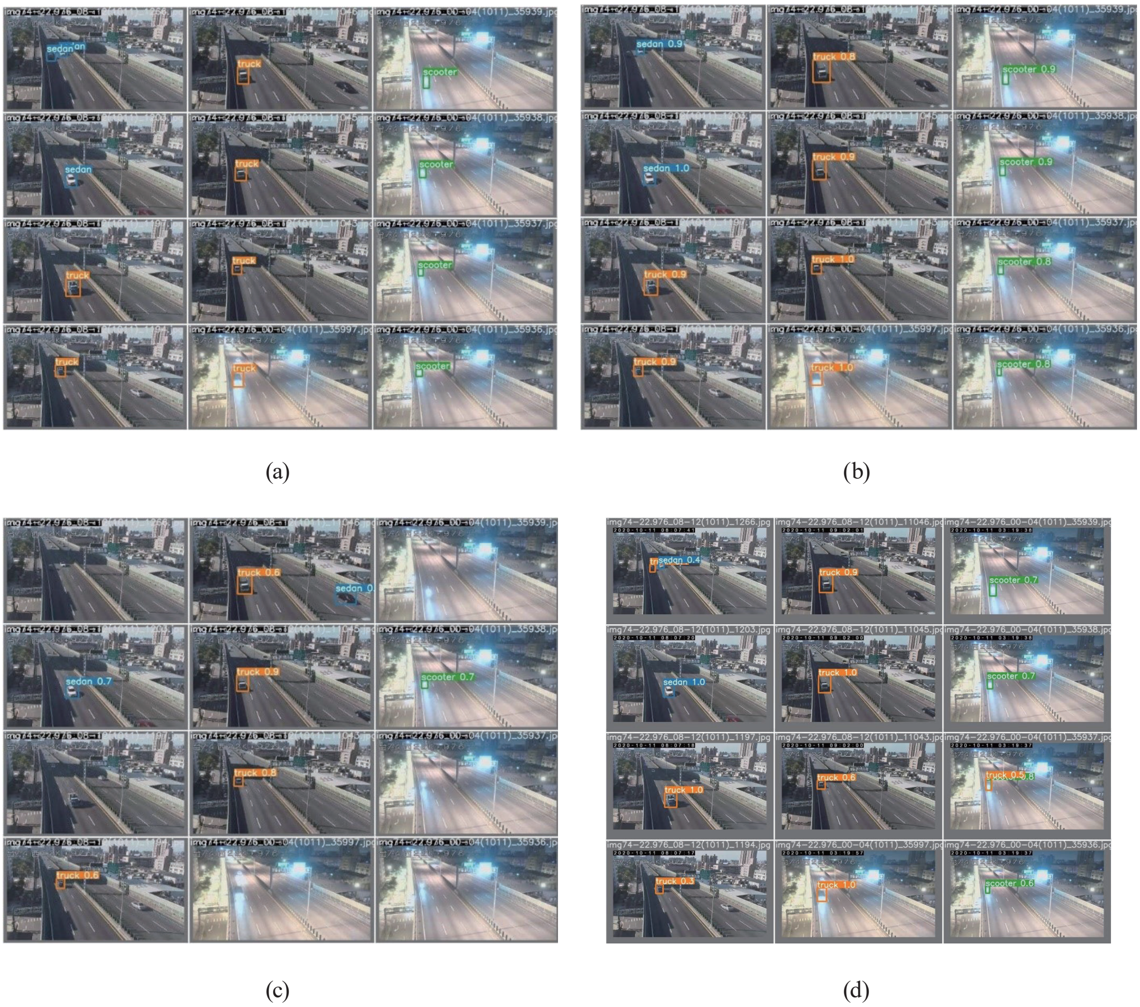


Fig. 15. (Color online) PR curves for T-YOLOv7 on T-74 vehicle datasets.

outperforming the original YOLOv7 and YOLOv4 by 17.9 and 2.4 percentage points, respectively. The F1-scores for T-YOLOv7, original YOLOv7, and YOLOv4 were 84.2, 69.2, and 80.9%, respectively, indicating that T-YOLOv7 had improvements of 15.0 and 3.3 percentage points over the original YOLOv7 and YOLOv4, respectively. In terms of mAP, T-YOLOv7 achieved a score of 85.4%, whereas the original YOLOv7 and YOLOv4 achieved scores of 74.9% and 84.6%, respectively. This indicates that T-YOLOv7 had improvements of 10.6 and 0.8 percentage points over the original YOLOv7 and YOLOv4, respectively. These results in Tables 11 and 12 demonstrate that the proposed T-YOLOv7 model with Taguchi optimization showed superior performance compared with the original YOLOv7 and YOLOv4 in object detection tasks.

The visualization of the detection results is shown in Fig. 16. Figure 16(a) shows the labeled ground truth. As seen in Fig. 16(c), the original YOLOv7 failed to correctly identify two frames of sedan and truck under shadow occlusion conditions, resulting in missing bounding boxes. In Fig. 16(d), YOLOv4 misclassified one frame of sedan as a truck under shadow occlusion and misidentified one frame of a small-pixel scooter as a truck in a nighttime road scene. Among all correctly recognized images, T-YOLOv7 consistently exhibited higher confidence scores for bounding boxes than did the original YOLOv7 and YOLOv4. As shown in Fig. 16(b), T-YOLOv7 with Taguchi parameter optimization has almost no missed detection or misjudgment, successfully recognizing T-74 highway vehicle categories.

Among the three sets of experimental results in the customized T-74 vehicle dataset, the performance for scooters was the worst in terms of vehicle recognition accuracy. This can be attributed to the cameras being positioned far away, resulting in a smaller field of view for



(a) (b)

(c) (d)

Fig. 16. (Color online) Detection results on T-74 vehicle dataset: (a) labeled ground truth, (b) T-YOLOv7, (c) original YOLOv7, and (d) YOLOv4.

scooters. The low-resolution images of scooters provided fewer prominent features, thereby adversely affecting their recognition accuracy across all sets. However, the T-YOLOv7 model, which was developed with optimized Taguchi parameters, exhibited excellent small-pixel recognition capabilities. As shown in Table 12, the scooter prediction accuracy increased by 12.4 and 15.8 percentage points compared with YOLOv4 and the original YOLOv7, respectively.

The recall rate for sedans was the lowest in all experimental sets, as they were often misclassified as trucks, resulting in a high proportion of FN in the sedan category and a decline in recall. A similar situation occurred for the truck category, where a significant proportion of sedans were mistakenly classified as trucks, resulting in an increase in FP in the truck category and a decrease in the prediction accuracy for trucks.

4. Conclusions

In this study, we developed a T-YOLOv7 model with Taguchi parameter optimization for vehicle detection on highways. Because of the time-consuming and challenging process of selecting parameters for the YOLOv7 network architecture, we applied the Taguchi method to optimize the hyperparameter combinations of YOLOv7. In the Taguchi method, a total of eight factors with two levels each were used. The experiments were mainly conducted on two datasets, the public BIT-vehicle dataset and our customized real-time traffic data T-74 vehicle dataset, to evaluate the performance of T-YOLOv7. The experimental results demonstrated that T-YOLOv7 achieved improvements in precision, recall, and F1-score compared with the conventional YOLOv7, with increases of 12.1, 17.9, and 15.0 percentage points, respectively. Compared with YOLOv4, T-YOLOv7 showed improvements of 4.0, 2.7, and 3.3 percentage points in the respective metrics. This proves that our proposed T-YOLOv7 model, optimized through the use of the Taguchi method for hyperparameter tuning, can achieve efficient real-time vehicle detection and has the potential for broad application in practical environments.

References

1 A. Mhalla, T. Chateau, S. Gazzah, and N. E. B. Amara: IEEE Trans. Intell. Transp. Syst. **20** (2019) 4006. https://doi.org/10.1109/TITS.2018.2876614
2 X. Mou, X. Chen, J. Guan, B. Chen, and Y. Dong: Proc. 2019 Int. Conf. Control, Automation and Information Sciences (ICCAIS) (IEEE, 2019) 1–5. https://doi.org/10.1109/ICCAIS46528.2019.9074588
3 K. Shi, H. Bao, and N. Ma: Proc. 2017 13th Int. Conf. Computational Intelligence and Security (CIS) (IEEE, 2017) 73–76. https://doi.org/10.1109/CIS.2017.00024
4 S. C. Hsu, C. L. Huang, and C. H. Chuang: Proc. 2018 Int. Workshop on Advanced Image Technology (IWAIT) (IEEE, 2018) 1–3. https://doi.org/10.1109/IWAIT.2018.8369767
5 F. Zhu, Y. Lu, N. Ying, and G. Giakos: Proc. 2017 IEEE Int. Conf. Imaging Systems and Techniques (IST) Conf. (IEEE, 2017) 1–4. https://doi.org/10.1109/IST.2017.8261505
6 W. Zhang, Y. Zheng, Q. Gao, and Z. Mi: IEEE Access **7** (2019) 100383. https://doi.org/10.1109/ACCESS.2019.2929432
7 C. J. Lin and J. Y. Jhang: IEEE Access **10** (2022) 14120. https://doi.org/10.1109/ACCESS.2022.3147866
8 Z. Li and F. Zhou: arXiv (2018) 1. https://doi.org/10.48550/arXiv.1712.00960
9 X. Wang, P. Cheng, X. Liu, and B. Uzochukwu: arXiv (2019) 1. https://doi.org/10.48550/arXiv.1808.05756
10 J. Redmon and A. Farhadi: arXiv (2018) 1. https://doi.org/10.48550/arXiv.1804.02767
11 A. Bochkovskiy, C. Y. Wang, and H.-Y. M. Liao: arXiv (2020) 1. https://doi.org/10.48550/arXiv.2004.10934
12 Q. Xu, R. Lin, H. Yue, H. Huang, Y. Yang, and Z. Yao: IEEE Access **8** (2020) 27574. https://doi.org/10.1109/ACCESS.2020.2966328
13 A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao: arXiv (2020) 1. https://doi.org/10.48550/arXiv.2004.10934
14 S. Liu, Y. Wang, Q. Yu, H. Liu, and Z. Peng: IEEE Access **10** (2022) 129116. https://doi.org/10.1109/ACCESS.2022.3228331
15 S. Lee and T. Kim: IEEE Access **11** (2023) 55046. https://doi.org/10.1109/ACCESS.2023.3281910
16 H. Alibrahim and S. A. Ludwig: Proc. 2021 IEEE Congr. on Evolutionary Computation (CEC) (IEEE, 2021) 1551. https://doi.org/10.1109/CEC45853.2021.9504761
17 Y. Wu, T. Zhou, X. Hu, L. Shi, and W. Yang: IEEE Geosci. Remote Sens. Lett. **19** (2022) 1. https://doi.org/10.1109/LGRS.2022.3225151
18 S. H. Fan, M. H. Lin, J. Y. Jiang, and Y. H. Kuo: IEEE Access **9** (2021) 133650. https://doi.org/10.1109/ACCESS.2021.3116064
19 C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao: arXiv (2022) 1. https://doi.org/10.48550/arXiv.2207.02696
20 L. Yu, J. Chen, G. Ding, Y. Tu, J. Yang, and J. Sun: IEEE Access **6** (2018) 45923. https://doi.org/10.1109/ACCESS.2018.2864222

21 U. Demir, G. Akgün, S. K. Md. E. Okay, A. M. Heydar, E. Akdogan, A. Yildirim, S. Yazi, B. Demirci, and E. Kaplanoglu: IEEE Access **11** (2023) 40107. https://doi.org/10.1109/ACCESS.2023.3268147

22 W. Zhao, Y. Gao, T. Ji, X. Wan, F. Ye, and G. Bai: IEEE Access **7** (2019) 114496. https://doi.org/10.1109/ACCESS.2019.2935504

23 Z. Dong, Y. Wu, M. Pei, and Y. Jia: IEEE. Trans. Intell. Transp. Syst. **16** (2015) 2247. https://doi.org/10.1109/TITS.2015.2402438

24 C. Wu, J. Li, R. Song, Y. Li, and Q. Du: IEEE. Trans. Geosci. Remote Sens. **61** (2023) 1. https://doi.org/10.1109/TGRS.2023.3264675

25 H. Seong, C. Chung, and D. H. Shim: IEEE Contr. Syst. Lett. **7** (2023) 1652. https://doi.org/10.1109/LCSYS.2023.3267041

26 Y. Gong, J. Peng, S. Jin, X. Li, Y. Tan, and Z. Jia: Proc. 2021 IEEE Int. Conf. Emergency Science and Information Technology (ICESIT) (IEEE, 2021) 333–336. https://doi.org/10.1109/ICESIT53460.2021.9696749

27 H. Lin, J. Zhou, and M. Chen: Proc. 2022 IEEE 10th Joint Int. Conf. Information Technology and Artificial Intelligence (ITAIC) (IEEE, 2022) 2156–2160. https://doi.org/10.1109/ITAIC54216.2022.9836923

About the Authors

Mei-Kuei Chen received her M.S. degree from the Institute of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C., in 2020. She is currently a Ph.D. student in the Department of Industrial Education and Technology, National Changhua University of Education, Taiwan. Her current research interests include machine learning, image recognition, intelligent control, optimal design, and evolutionary robotics. (d1131002@mail.ncue.edu.tw)

Chun-Lung Chang received his M.S. degree in power mechanical engineering from National Tsing-Hua University, Hsinchu, Taiwan, R.O.C., in 1992, and his Ph.D. degree in electrical and control engineering from National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in 2006. Currently, he is an associate professor in National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C. His current research interests are artificial intelligence, image processing, computer vision, and smart manufacturing.
(VincentChang0304@ncut.edu.tw)

Cheng-Jian Lin received his B.S. degree in electrical engineering from Ta Tung Institute of Technology, Taipei, Taiwan, R.O.C., in 1986, and his M.S. and Ph.D. degrees in electrical and control engineering from National Chiao Tung University, Taiwan, R.O.C., in 1991 and 1996, respectively. Currently, he is a chair professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C. His current research interests are machine learning, pattern recognition, intelligent control, image processing, intelligent manufacturing, and evolutionary robots. (cjlin@ncut.edu.tw)

Wen-Jong Chen received his B.S. degree in industrial education from National Changhua University of Education, Changhua, Taiwan, R.O.C., in 1987, M.S. degree from the Institute of Applied Mechanics, National Taiwan University, in 1992, and Ph.D. degree in mechanical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1997. He is currently a professor in the Department of Industrial Education and Technology, National Changhua University of Education, Taiwan. His current research interests include machine learning, smart manufacturing, smart control, optimal design and evolutionary computation. (wjong@cc.ncue.edu.tw)