Sensors and Materials, Vol. 36, No. 11 (2024) 4975–4989
MYU Tokyo

S & M 3844

Real-time Vehicle Recognition System Using You Only Look Once Model with Uniform Experimental Design

Chun-Hui Lin,[1] We-Ling Lin,[2] Cheng-Jian Lin,[1*] and Kang-Wei Lee[1]

[1]Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan
[2]Department of Intelligent Production Engineering, National Taichung University of Science and Technology, Taichung 404, Taiwan

(Received June 25, 2024; accepted October 30, 2024)

Keywords: intelligent transportation system, vehicle recognition, uniform experimental design, YOLO

As urban areas develop and technology advances, artificial intelligence technologies offer numerous applications to alleviate a pressing issue: traffic congestion. The importance of traffic management and safety monitoring underscores the crucial role of integrating vehicle recognition technology with the Internet of Things within intelligent transportation systems. In this study, You Only Look Once with Uniform Experimental Design (U-YOLOv4) is proposed to enhance the performance of vehicle recognition. The approach aims to optimize hyperparameters within YOLOv4, resulting in a high recognition rate in the model. Furthermore, two datasets were utilized: Vehicle from Beijing Institute of Technology (BIT) and Computational Intelligence Application Laboratory from National Chin-Yi University of Technology (CIA-NCUT). The experimental results revealed significant improvements when comparing U-YOLOv4 to YOLOv4. In the BIT-Vehicle dataset, U-YOLOv4 achieved a mean average precision of 97.84%, whereas in the CIA-NCUT dataset, it reached 89.19%, highlighting its superior performance over YOLOv4. The U-YOLOv4 model has overall demonstrated significant improvement in vehicle recognition, revealing its adaptability across different datasets and various scenarios. Its application is expected to play a crucial role in intelligent transportation systems, enhancing traffic management efficiency and road safety.

1. Introduction

With the global population continuing to grow and business activities flourishing, transportation and road traffic have become increasingly essential for daily operations. However, this also leads to a range of issues, including traffic congestion, heightened carbon dioxide emissions, increased risk of traffic accidents, and challenges in route planning and management. In response to these challenges, intelligent transportation systems (ITSs) utilize technologies such as the Internet of Things (IoT), big data analysis, and artificial intelligence to enhance transportation efficiency and public safety, and ultimately, to improve the convenience of travel.(1) Specifically, IoT technology facilitates connectivity among transportation facilities,

*Corresponding author: e-mail: cjlin@ncut.edu.tw
https://doi.org/10.18494/SAM5196

ISSN 0914-4935
https://myukk.org/

vehicles, and pedestrians, enabling instant data collection and sharing. Big data analysis extracts valuable information from massive data to support traffic management and decision-making. Artificial intelligence technology is applied in traffic monitoring, signal control, and intelligent traffic prediction, enhancing the intelligence of the transportation system. Among these technologies, the vehicle recognition system, a pivotal component of ITS, connects to the IoT to identify vehicles and transmit relevant information to the monitoring center or other traffic management platforms. Moreover, it provides traffic managers with real-time data access for adjustments and responses. Specifically, the vehicle recognition system not only analyzes traffic flow but also estimates driving delay times and suggests alternative routes, thereby reducing energy consumption and carbon dioxide emissions. Simultaneously, regarding urban security, the vehicle recognition system can pinpoint accident locations, identify illegal vehicles, track suspect vehicles, and provide real-time information to relevant units to maintain public safety.[2,3] These benefits collectively promote urban transportation efficiency and safety, mitigate the negative impact of transportation on the environment, and enhance the travel experience for individuals.

Accurately detecting and identifying vehicles in the constantly shifting traffic environment remain a significant challenge. The diversity and complexity of traffic scenes, including factors such as traffic volume, road directions, and the multitude of vehicle types, pose substantial hurdles for vehicle recognition technology. Additionally, adverse weather conditions, varying light sources, and object occlusions exacerbate the difficulty of vehicle identification, leading to diminished accuracy and stability.[4] Consequently, there is a clear need to enhance and refine vehicle recognition technology to meet the changing demands of road traffic. Conventional machine learning (ML) approaches mainly consist of two steps in target recognition:[1] feature extraction and classification. Specifically, features are typically extracted directly from image pixels using low-level feature extraction methods such as speeded-up robust features (SURF),[5,6] scale-invariant feature transformation (SIFT),[6,7] and histogram of oriented gradients (HOG).[8,9] Subsequently, classification is commonly performed using algorithms such as k-nearest neighbor (KNN)[10] and support vector machine (SVM).[9] However, the instability of conventional ML lies in the fact that feature extraction relies on manual design by experts, which can result in a time-consuming and computationally tedious process. Fortunately, with the development of deep learning, conventional approaches are gradually being replaced by convolutional neural networks (CNNs). CNNs perform image localization and classification by extracting features from images using multi-level convolution and pooling layers, and it has been widely used in object detection tasks.[11] In vehicle recognition tasks, CNN models can not only detect and classify different categories of vehicles through model training but also adapt to various scenes and environments.

Among all CNN models, object detection models can be mainly divided into two types: two-stage and one-stage.[12] The former models usually generate candidate regions of interest (region proposals) first, then feed those candidate areas into CNN for further processing such as feature extraction and classification to identify the objects' locations and categories. Representative models, such as region-based CNN (R-CNN), [13] Fast R-CNN, and Faster R-CNN,[14] achieve high accuracy and are particularly suitable for complex scenes where accurate object recognition

is required. However, they are hindered by the time-consuming nature of the two-step process of generating and classifying region proposals. On the other hand, one-stage object detection models, such as the Single Shot MultiBox Detector (SSD)[15] and You Only Look Once (YOLO),[16] integrate region proposals and object classification into a single network, allowing model parameters to propagate from the input layer to the output layer. One-stage models offer the advantage of reducing computational time, making them particularly suitable for real-time applications. However, their accuracy tends to be lower than that of two-stage models. In the context of vehicle identification tasks, it is crucial to consider both computational time and accuracy to achieve effective real-time vehicle identification.

Notably, the series of YOLO models[17] have demonstrated a strong capability for feature extraction and rapid object recognition through multiple improvements and optimizations. Its excellent performance has been acknowledged in numerous practical applications. For instance, Liu *et al.* replaced the backbone network with the ConvNeXt-S network for the feature extraction part in YOLOv4.[18] Additionally, a global attention mechanism was integrated, and the loss function was modified with SIoU for real-time vehicle detection tasks.[18] Liu *et al.* also applied the GhostbottleNet algorithm to the feature extraction process within YOLOv5 to address the issue of incomplete feature extraction caused by the uneven distribution of image features in surface ship detection and recognition tasks.[19] Safonova *et al.* implemented YOLO models to detect infested trees using images acquired from high-resolution unmanned aerial vehicles.[20] However, the performance of these models was impeded by the types of detection and the complexity of the images, highlighting the importance of appropriately adjusting hyperparameters to improve the model's accuracy, thereby enhancing its performance and robustness.

The selection of hyperparameters can significantly impact the model performance by enhancing accuracy, improving model generalization, reducing overfitting, and even expediting the training process. Numerous researchers have proposed several methods, including full factorial design, grid search,[21] and random search.[22] However, these approaches typically demand a substantial number of experiments and impose high requirements on computing resources and time.[23] Considering constraints on time and cost, the Taguchi method and uniform experimental design have emerged as widely adopted alternatives. The Taguchi method is an effective experimental design approach that enables the discovery of optimal hyperparameter combinations in a limited number of experiments. Although the number of experiments in the Taguchi method typically exceeds the number of levels squared, it is still relatively smaller than those of full factorial experiments and grid searches, making it easy to implement. Nevertheless, the Taguchi method has limitations, as the settings for factors and levels cannot be too extensive and are determined by experts with specific experience and intuition regarding the experiments.[24] On the other hand, in uniform experimental design, factors and levels are determined on the basis of predistribution, ensuring balanced exploration across different factors and levels. This approach effectively reduces the number of experiments while maintaining the comprehensive exploration of the hyperparameter space. In other words, a uniform experimental design can discover better hyperparameter combinations in a limited number of experiments, effectively addressing multi-factor and multi-level parameter selection

problems.[25] Therefore, to enhance model accuracy while conserving computational time through fewer experiments, in this study, we employed a uniform experimental design to optimize parameters, effectively addressing the challenge of multi-factor and multi-level parameter selection. Moreover, building upon the concept of a rapid and accurate real-time vehicle recognition system, we proposed an improved YOLOv4 model with uniform experimental design (U-YOLOv4) to achieve high-precision vehicle detection and classification. The contributions of this study can be summarized as follows:

- The proposed U-YOLOv4 model achieved real-time vehicle detection and enhances recognition accuracy.
- The uniform experimental design method is utilized to determine optimal parameter combinations in multi-factor and multi-level parameter selection efficiently.
- A practical vehicle dataset, Computational Intelligence Application Laboratory from National Chin-Yi University of Technology (CIA-NCUT), was established from traffic videos in western Taiwan. Furthermore, U-YOLOv4 was implemented in actual scenes along a country road in Taiwan.

The remainder of this study is organized as follows. In Sect. 2, we introduce the proposed real-time vehicle recognition system using the U-YOLOv4 method. In Sect. 3, we present a comparison of the results between U-YOLOv4 and YOLOv4, and in Sect. 4, we draw conclusions and outline future prospects.

2. Real-time Vehicle Recognition System

The overall workflow of this study is illustrated in Fig. 1. To address the challenges present in vehicle identification, the uniform design experiment method was employed to optimize the model hyperparameters. Subsequently, through regression analysis, the optimal parameter combination was determined to enhance the performance of the YOLOv4 model. Additionally, real-time images captured from cameras installed on highways in western Taiwan were obtained, and the system was deployed on an AGX Xavier embedded platform.

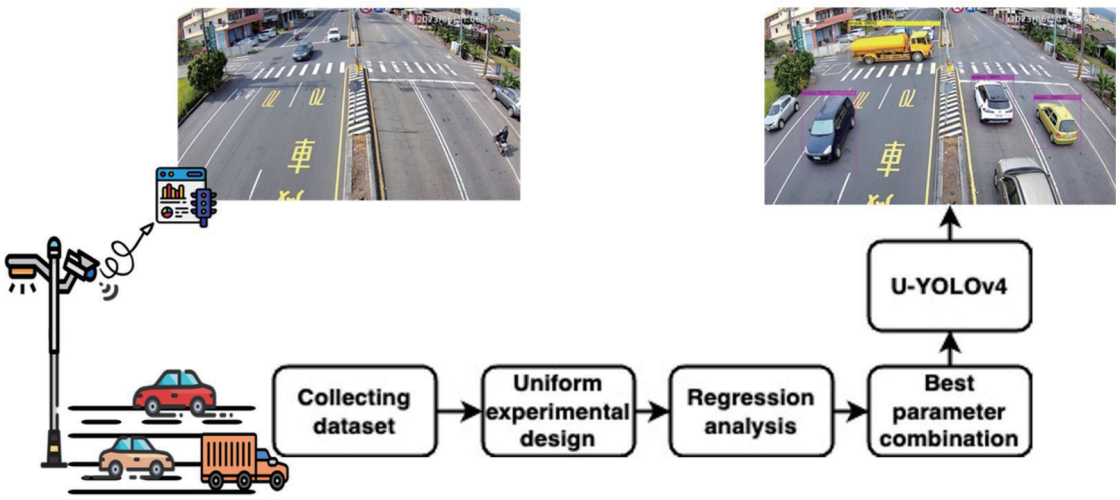


Fig. 1. (Color online) Overall workflow of real-time vehicle recognition system.

2.1 YOLOv4 model

YOLOv4,[26] an end-to-end designed model, as shown in Fig. 2, can be trained on a single graphics processing unit for fast and accurate performance.[16] Its architecture is mainly composed of three parts as shown in Fig. 2, namely, the backbone, neck, and head. The backbone utilizes CSPDarknet53 with the concept of the cross-stage partial network for achieving effective feature extraction. The neck combines a spatial pyramid pooling layer and path aggregation network components for integrating feature maps of different scales. Lastly, the head employs the original YOLOv3 architecture, which converts feature maps obtained from the backbone into prediction boxes and categories.[16,17] In this study, hyperparameters such as batch normalization (BN), the number of convolution kernels, and the activation function (AF) in the last three layers of the YOLOv4 head were chosen for uniform experimental design adjustment.

2.2 Uniform experimental design

The uniform experimental design method[23] replaces the orderliness and comparable characteristics of orthogonal experiments by evenly distributing parameters within a certain

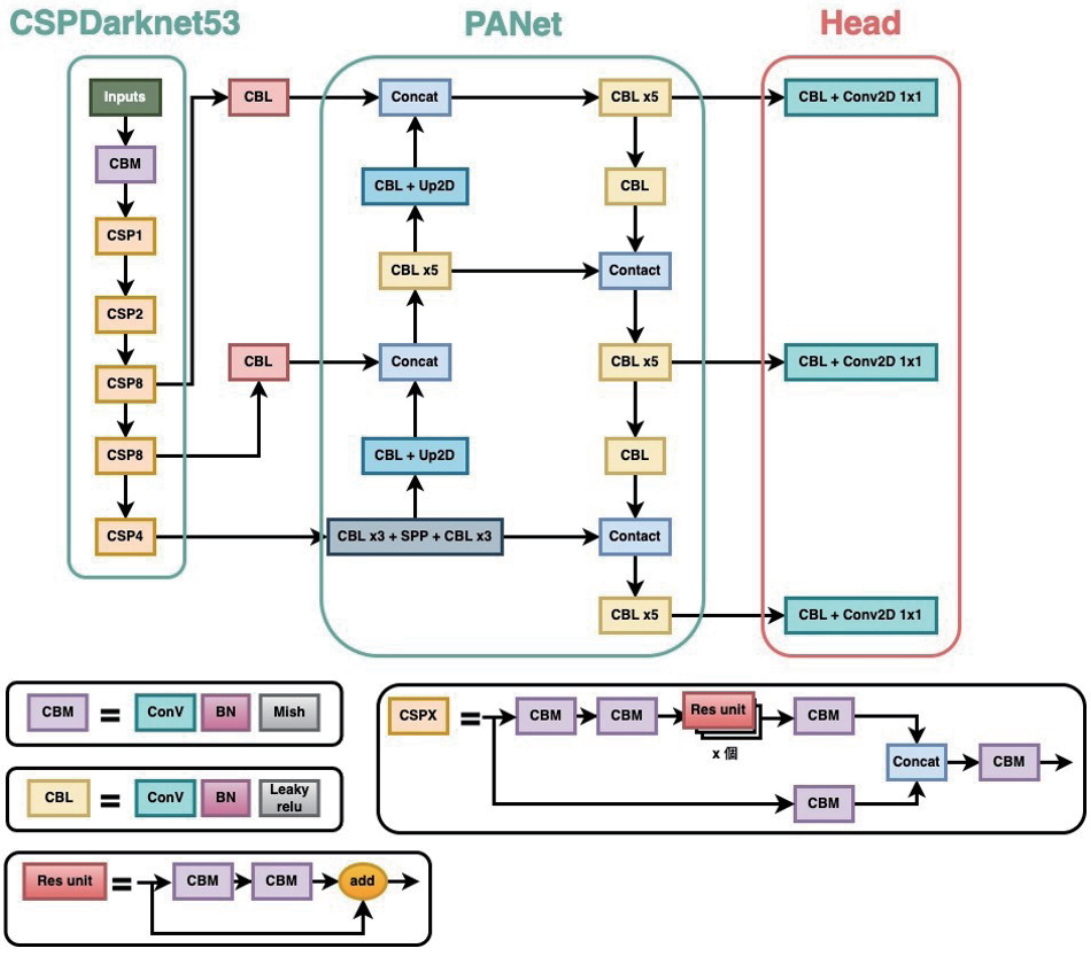


Fig. 2. (Color online) YOLOv4 architecture diagram.

range. This ensures that the parameters cover the entire design space and that each experimental point is representative.[25] This benefits the effective exploration within the design space, lowering computational costs and avoiding repeated experiments, especially when the design space is large or complex. In this study, seven factors, including three 2-level factors and four 3-level factors, were designed and are listed in Table 1. Moreover, the required number of experiments was determined according to the uniform layout and good lattice point sets, as revealed in Table 2. The uniform experimental design flowchart is also provided in Fig. 3.

Table 1
Experimental factors and levels in uniform experimental design.

Factor		Level 1	Level 2	Level 3
CBL 1	BN	0	1	—
	Filters	128	256	512
	AF	Mish	ReLU	Leaky ReLU
CBL 2	BN	0	1	—
	AF	Mish	ReLU	Leaky ReLU
CBL 3	BN	0	1	—
	AF	Mish	ReLU	Leaky ReLU

Table 2
List of uniform layout.

	1	2	3	4	5	6	7
1	8	6	14	1	11	13	4
2	10	10	11	4	14	10	15
3	4	15	1	5	9	7	5
4	9	7	2	13	8	1	14
5	1	11	4	10	12	12	10
6	13	12	8	7	13	3	1
7	14	9	5	2	1	6	9
8	12	14	9	14	6	14	7
9	2	2	10	3	7	2	8
10	7	13	15	9	4	4	12
11	3	8	12	12	2	9	2
12	5	5	7	6	3	15	13
13	15	3	13	11	10	8	11
14	6	4	6	15	15	5	6
15	11	1	3	8	5	11	3

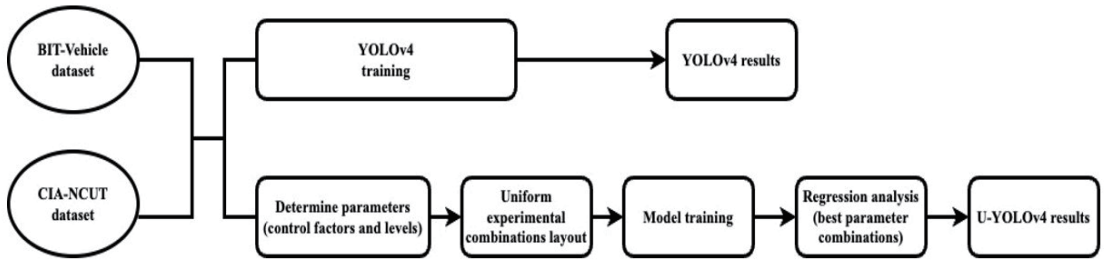


Fig. 3. Flowchart of uniform experimental design.

2.3 Multiple regression analysis

Multiple regression is a statistical analysis method that explores the relationship between one dependent variable and one or more independent variables. Unlike simple linear regression, which involves only one independent variable and one dependent variable, multiple regression considers the effect of multiple independent variables. This allows for a more comprehensive analysis of the factors affecting the dependent variable.[24] In this study, quadratic multiple regression was performed after completing the 15 experiments described in the previous section to identify the optimal parameter combination. The formula for quadratic multiple regression is

$$\mathcal{E} = Y - \left[\beta_0 + \sum_{i=1}^{n} \beta_{1i} X_i + \sum_{i=1}^{n} \beta_{2i} X_i^2 + \sum_{i=1}^{n-1} \sum_{m=i+1}^{n} \beta_{3i} X_i X_m \right], \tag{1}$$

where ε, Y, and β_0 represent the residual term, dependent variable, and intercept term, respectively. β_{1i} signifies the coefficient of the linear term, indicating the effect of the ith independent variable on the dependent variable; β_{2i} denotes the coefficient of the quadratic term, representing the effect of the square of the ith independent variable on the dependent variable; β_{3i} indicates the coefficient of the interaction term, representing the impact of the product of the ith independent variable and other independent variables on the dependent variable. Additionally, X_i and X_m denote the values of independent variables, while n represents the total number of independent variables.

2.4 Evaluation metric

Four evaluation metrics—mean average precision (*mAP*), precision, recall, and *F*1-*score*—are described to assess the performance of the U-YOLOv4 and conventional YOLOv4 models. The formulas for these metrics are listed as follows:

$$mAP = \frac{\sum_{k=1}^{k=n} AP_k}{n}, \tag{2}$$

$$Precision = \frac{TP}{TP + FP}, \tag{3}$$

$$Recall = \frac{TP}{TP + FN}, \tag{4}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}, \tag{5}$$

where *TP*, *TN*, *FP*, and *FN* respectively stand for true positives, true negatives, false positives, and false negatives, respectively. Moreover, n represents the number of categories and AP_k denotes the average precision in the kth category.

3. Results and Discussion

In this section, we aim to verify the effectiveness of U-YOLOv4 using two datasets. Moreover, the optimal parameter combinations and the vehicle recognition performance, along with further discussions, are provided. Furthermore, the model was also evaluated under various lighting scenarios and tested for real-time application on highways.

3.1 Results of U-YOLOv4 using Beijing Institute of Technology (BIT)-Vehicle dataset

3.1.1 BIT-Vehicle dataset

The BIT-Vehicle dataset comprises a total of 9850 vehicle images, including six types of vehicle, namely, trucks, buses, sedans, microbuses, minivans, and SUVs, as illustrated in Fig. 4. These images were captured from various sections of the highway, encompassing different light conditions, with each image featuring one or two vehicles. For the experiments, the data were randomly split into an 80:20 ratio, with 7880 images used for training and 1970 images for testing. The detailed breakdown of each category within the BIT-Vehicle dataset is provided in Table 3.

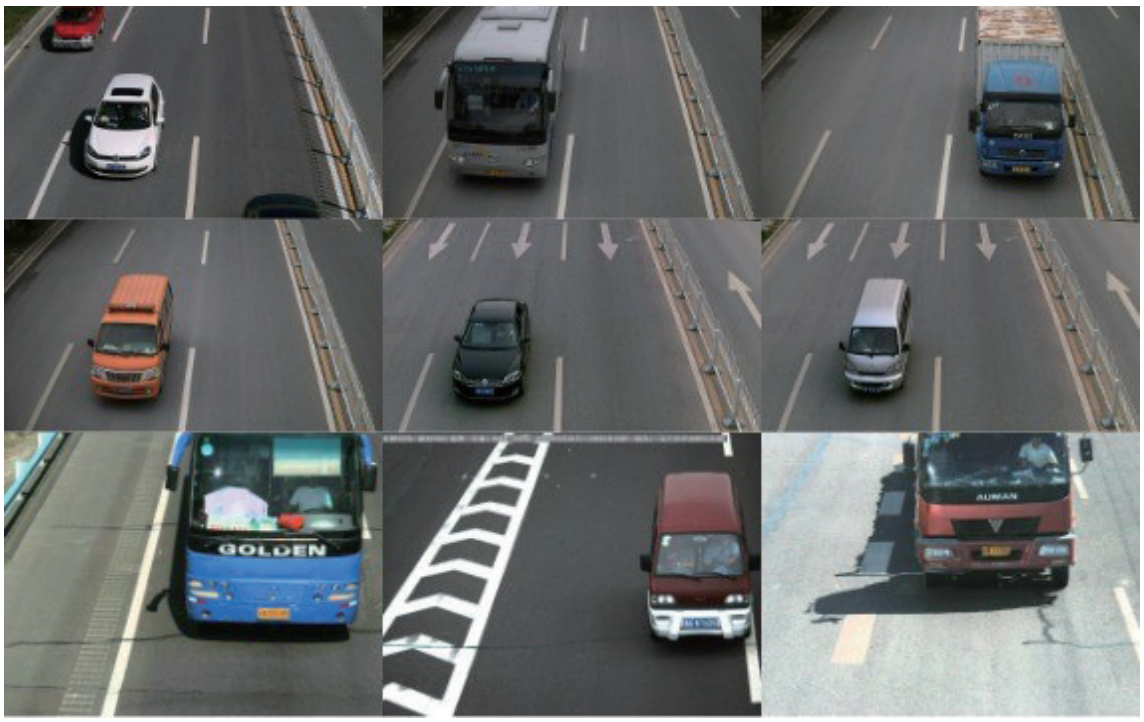


Fig. 4. (Color online) Images within BIT-Vehicle dataset.

Table 3
Number of vehicle categories within BIT-Vehicle dataset.

Category	Bus	Microbus	Minivan	Sedan	SUV	Truck
Training data	457	702	388	4751	1104	648
Testing data	101	181	88	1170	288	175
Total count	558	883	476	5922	1392	822

3.1.2 Uniform experimental design

In YOLOv4, the head plays a crucial role as it processes aggregated features and predicts bounding boxes, object scores, and classification scores. Therefore, optimizing pretraining parameter combinations can significantly enhance the model's performance. Herein, Table 4 lists a total of fifteen uniform design experiments. Following the completion of these experiments, all data underwent analysis using quadratic multiple regression to determine the optimal parameter combination. Finally, the results are presented in Table 5, with the optimal pretraining parameter combination displayed in Table 6.

3.1.3 Comparison of experimental results between U-YOLOv4 and YOLOv4 using BIT-Vehicle dataset

To assess the performance of U-YOLOv4, comparison experiments were conducted with the original YOLOv4 model. The performance was averaged from three experiments using the BIT-Vehicle dataset, as listed in Table 7. The U-YOLOv4 model achieved the same precision, recall, and *F*1-*score* as the original YOLOv4 model, at 94.5, 97.5, and 96%, respectively. Furthermore, U-YOLOv4 demonstrated a 0.58% increase in *mAP* compared with the original YOLOv4 model. Specifically, the U-YOLOv4 model exhibited better recognition in five categories: minibus, minivan, sedan, SUV, and truck. This indicates that U-YOLOv4 can more accurately locate targets in detection tasks and shows a slight improvement in overall detection accuracy.

3.2 Results of U-YOLOv4 using CIA-NCUT dataset

3.2.1 CIA-NCUT dataset

In this subsection, the CIA-NCUT dataset, collected by the team from National Chin-Yi University of Science and Technology, is introduced. This dataset is particularly suited to reflecting actual road conditions in Taiwan. Acquired through cameras installed on highway sections in western Taiwan, it was complemented by the development of the AGX Xavier embedded platform for video data collection. Moreover, since image recognition outcomes can be affected by weather conditions and lighting environments, the dataset comprises images captured under various conditions such as day, night, and rainy days as illustrated in Fig. 5. These images encompass both forward and backward perspectives from driving viewpoints. The CIA-NCUT dataset includes five vehicle categories: cars, buses, trucks, motorcycles, and

Table 4
List of fifteen uniform design experiments.

	CBL 1			CBL 2		CBL 3	
	BN	Filters	AF	BN	AF	BN	AF
1	1	512	Leaky	0	ReLu	1	ReLu
2	1	128	Leaky	1	ReLu	0	Leaky
3	0	512	Mish	0	Mish	1	Mish
4	1	128	Leaky	0	ReLu	1	Mish
5	0	256	Mish	1	Mish	0	ReLu
6	1	512	Leaky	0	Leaky	1	ReLu
7	1	512	Leaky	1	Leaky	0	Leaky
8	1	256	ReLu	1	Mish	0	ReLu
9	0	256	Mish	0	Leaky	0	Mish
10	0	128	ReLu	0	Leaky	0	Leaky
11	0	256	ReLu	1	ReLu	1	Mish
12	0	256	Mish	1	Mish	1	ReLu
13	1	512	Mish	0	Leaky	0	Mish
14	0	128	ReLu	0	Mish	1	Leaky
15	1	128	ReLu	1	ReLu	1	Leaky

Table 5
Results of uniform design experiments using BIT-Vehicle dataset.

	Precision (%)	*Recall* (%)	*F*1-*score* (%)	*mAP* (%)
1	95	97	96	96.79
2	94	97	96	97.43
3	95	98	96	97.15
4	95	97	96	97.47
5	94	97	96	96.46
6	95	97	96	96.71
7	94	98	96	97.44
8	94	98	96	97.64
9	95	97	96	97.40
10	95	97	96	97.40
11	95	97	96	95.63
12	94	97	96	97.53
13	94	98	96	97.67
14	94	97	96	97.36
15	94	97	96	96.96

Table 6
Optimal pretraining parameter combination using BIT-Vehicle dataset.

CBL 1		CBL 2		CBL 3	
BN	1	BN	0	BN	1
Filters	128	AF	Leaky	AF	ReLU
AF	Leaky				

bicycles, totaling 76597 images. After image labeling, the data were randomly divided into a training dataset of 61278 images and a testing dataset of 15319 images, maintaining an 80:20 ratio. Given that each image contains multiple vehicles, the total number is presented in Table 8.

To ensure consistency and comparability across experiments, the uniform design experiments adopted the same process as described in Sect. 3.1.2 for the BIT-Vehicle Dataset. Subsequently, a

Table 7
Comparison of experimental results between U-YOLOv4 and YOLOv4 using BIT-Vehicle dataset.

Method	YOLOv4 (%)	U-YOLOv4 (%)
Precision	94.50	94.50
Recall	97.50	97.50
*F*1-*score*	96	96
mAP	97.26	**97.84**
Bus	100	100
Minibus	98.35	**98.64**
Minivan	92.70	**94.44**
Sedan	98.90	**98.92**
SUV	96.55	**97.32**
Truck	96.74	**97.73**

Fig. 5. (Color online) Images within CIA-NCUT dataset.

Table 8
Number of vehicle categories within CIA-NCUT dataset.

	Car	Bus	Truck	Motorcycle	Bicycle
Training data	57701	4204	9561	28293	4506
Testing data	14392	968	2480	6981	1103
Total count	72093	5172	12041	35274	5609

quadratic multiple regression analysis was conducted to select the optimal combination of pretraining parameters. The results of all fifteen uniform design experiments are presented in Table 9, with the optimal pretrained parameter combination revealed in Table 10.

Table 9
Results of uniform design experiments using CIA-NCUT dataset.

	Precision (%)	*Recall* (%)	*F*1-*score* (%)	*mAP* (%)
1	71	91	80	88.91
2	71	90	80	88.86
3	72	90	80	88.52
4	72	89	80	88.42
5	72	89	80	88.45
6	71	90	80	88.68
7	71	90	79	88.88
8	72	91	80	88.82
9	72	90	80	88.57
10	70	91	79	88.69
11	71	90	79	88.45
12	72	90	80	88.50
13	73	89	80	88.53
14	71	90	79	88.50
15	71	90	80	88.46

Table 10
Optimal pretraining parameter combination using CIA-NCUT dataset.

CBL 1		CBL 2		CBL 3	
BN	0	BN	1	BN	0
Filters	256	AF	Leaky	AF	Mish
AF	Leaky				

3.2.2 Comparison of experimental results between U-YOLOv4 and YOLOv4 using CIA-NCUT dataset

Table 11 presents a comparison of experimental results between U-YOLOv4 and YOLOv4. Despite a slight decline in *precision* and *F*1-*score* for U-YOLOv4, there was an approximate 1 to 2% improvement in *mAP* and *recall* rates. This trend highlights U-YOLOv4's proficiency in accurately identifying objects in target detection tasks, with heightened sensitivity. Specifically, with increasing *recall*, the model becomes more effective at capturing true positive samples, albeit potentially resulting in more false positives and thus reducing *precision* and *F*1-*score*. Importantly, U-YOLOv4 demonstrated superior recognition ability in each category compared with the original YOLOv4 model.

3.3 Implementation of U-YOLOv4 on highways in Taiwan

To understand the performance of the actual implementation of U-YOLOv4 in real-time vehicle recognition, a section of a provincial highway in Taiwan was selected. The workflow, as shown in Fig. 6, includes cameras installed on pillars, transmitting real-time video back to the monitoring center via the internet.

Figure 7 shows vehicle recognition images generated by U-YOLOv4 and YOLOv4. It is evident that YOLOv4 struggles to recognize vehicles appearing far away from the camera (highlighted in the red circle). Conversely, U-YOLOv4 demonstrates a higher recognition rate in

Table 11
Comparison of experimental results between U-YOLOv4 and YOLOv4 using CIA-NCUT dataset.

Method	YOLOv4 (%)	U-YOLOv4 (%)
Precision	72	70
Recall	90	**92**
*F*1-*score*	80	79
mAP	88.31	**89.19**
Sedan	83.83	**84.07**
Passenger	93.68	**94.92**
Truck	94.56	**94.86**
Scooter	83.76	**84.48**
Bicycle	85.74	**87.64**

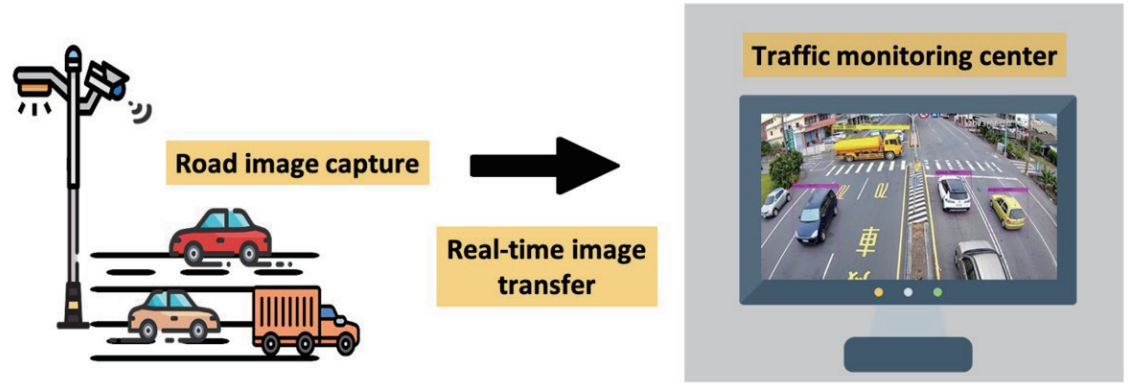
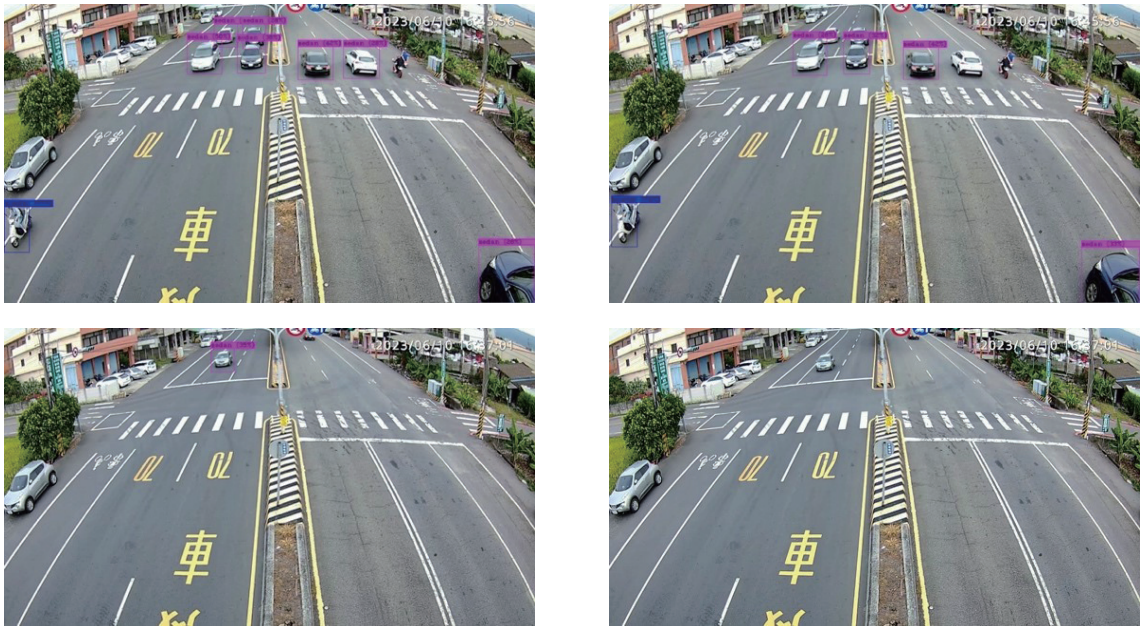


Fig. 6. (Color online) Workflow of actual implementation in real-time vehicle recognition.



(a) (b)

Fig. 7. (Color online) Vehicle recognition images generated by (a) U-YOLOv4 and (b) YOLOv4.

real-time image recognition and effectively identifies smaller vehicles at greater distances from the camera.

Overall, U-YOLOv4 improved *mAP* by 0.58% compared with the original YOLOv4 while maintaining consistent performance in *precision*, *recall*, and *F*1-*score* in the BIT-Vehicle dataset. On the other hand, the CIA-NCUT dataset, with more complex situations, showed that U-YOLOv4 improved by 0.88% to 2% in recall and *mAP*, but its performance in precision and F1-score declined slightly. This difference may be attributed to the diversity of the dataset, such as varying weather conditions, light sources, and vehicle directions, posing challenges to the model's generalization ability. Additionally, in real-time vehicle detection applications, U-YOLOv4 performs better than YOLOv4 in detecting smaller vehicles at greater distances. However, there is still room for improvement in the confidence of identifying those vehicles.

4. Conclusion

In this study, we employed a uniform experimental design method to optimize the YOLOv4 model for vehicle detection systems. By leveraging the characteristics of a uniform distribution, the parameter space was systematically explored, resulting in the successful development of U-YOLOv4 with improved performance. Additionally, the CIA-NCUT dataset was established by collecting traffic images from actual roads in Taiwan. Evaluation results on both the BIT-Vehicle and CIA-NCUT datasets demonstrated that U-YOLOv4 improved *mAP* by 0.58% to 2% compared with the original YOLOv4. Although there were slight decreases in some performance indicators, the overall performance was still enhanced. Moreover, U-YOLOv4 was applied to a highway section in Taiwan, confirming its effectiveness in real-world scenarios, particularly in identifying smaller vehicles at greater distances.

Nonetheless, challenges persist in the real-time detection and identification application of U-YOLOv4, including insufficient confidence in certain situations, necessitating further improvement. In future research, U-YOLOv4 can be enhanced through refined feature extraction and model adjustments, thereby boosting the accuracy and confidence of identification and enhancing its application value and reliability.

References

1 Zakria, J. Deng, Y. Hao, M. S. Khokhar, R. Kumar, J. Cai, J. Kumar, and M. U. Aftab: Mathematics **9** (2021) 3162. https://doi.org/10.3390/math9243162
2 S. Kumar, R. Mallidi, and V. V. Vineela: Int. J. Adv. Comput. Res. **9** (2018) 738. https://doi.org/10.26483/ijarcs.v9i2.5870
3 C. J. Lin and J. Y. Jhang: IEEE Access **10** (2022) 14120. https://doi.org/10.1109/ACCESS.2022.3147866
4 C. J. Lin, S. Y. Jeng, and H. W. Lioa: Math. Probl. Eng. **2021** (2021) 1577614. https://doi.org/10.1155/2021/1577614
5 N. A. Arifin, B. Irawan, and C. Setianingsih: 2017 IEEE Asia Pacific Conf. Wireless and Mobile (APWiMob) (Bandung, Indonesia, 2017) 52. https://doi.org/10.1109/APWiMob.2017.8284004
6 Z. Malik and I. Siddiqi: Int. Conf. Frontiers Inf. Technol. (Islamabad, Pakistan, 2014) 330. https://doi.org/10.1109/FIT.2014.68
7 R. Widyastuti and C. K. Yang: 2018 IEEE 7th Global Conf. Consum. Electron. (GCCE) (Nara, Japan, 2018) 55. https://doi.org/10.1109/GCCE.2018.8574870
8 M. Shirpour, N. Khairdoost, M. A. Bauer, and S. S. Beauchemin: IEEE Trans. Intell. Veh. **8** (2023) 594. https://doi.org/10.1109/TIV.2021.3133849

9 X. Zhang, L. Zhang, and X. Lou: IEEE Trans. Circuits and Systems I: Regular Papers **69** (2022) 322. https://doi.org/10.1109/TCSI.2021.3098053
10 N. Laopracha, k. Sunat, and S. Chiewchanwattana: IEEE Access **7** (2019) 20894. https://doi.org/10.1109/ACCESS.2019.2893320
11 A. K. Shetty, I. Saha, R. M. Sanghvi, S. A. Save, and Y. J. Patel: 2021 6th Int. Conf. Convergence in Technology (I2CT) (Maharashtra, India, 2021) 1. https://doi.org/10.1109/I2CT51068.2021.9417895
12 P. Wang, X. Wang, Y. Liu, and J. Song: IEEE Access **12** (2024) 8198. https://doi.org/10.1109/ACCESS.2024.3351771
13 A. Arinaldi, J. A. Pradana, and A. A. Gurusinga: INNS Conf. Big Data **144** (2018) 259. https://doi.org/10.1016/j.procs.2018.10.527
14 Z. Jiang, L. Zhao, S. Li, and Y. Jia: arXiv (2020). https://doi.org/10.48550/arXiv.2011.04244
15 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg: European Conf. Comput. Vis. (ECCV) (Amsterdam, The Netherlands, 2016) 21. https://doi.org/10.1007/978-3-319-46448-0_2
16 A. Bochkovskiy, C. Y. Wang, and H. Y. Liao: arXiv (2020). https://doi.org/10.48550/arXiv.2004.10934
17 A. Nazir and M. Wani: 2023 10th Int. Conf. Comput. Sustain. Global Develop. (INDIACom) (New Delhi, India, 2023) 1088.
18 Y. Liu, Y. Sun, C. Zhang, F. Wang, and H. Zhao: 2023 8th Int. Conf. Intell. Inform. Biomed. Sci. (ICIIBMS) **8** (2023) 575. https://doi.org/10.1109/ICIIBMS60103.2023.10347712
19 T. Liu, B. Zou, Y. Zhao, and S. Yan: 2021 6th Int. Conf. Automat. Control Robot. Eng. (CACRE) (Dalian, China, 2021) 483. https://doi.org/10.1109/CACRE52464.2021.9501331
20 A. Safonova, Y. Hamad, A. Alekhina, and D. I. Kaplun: IEEE Access **10** (2022) 10384. https://doi.org/10.1109/ACCESS.2022.3144433
21 Pirjatullah, D. Kartini, D. Nugrahadi, Muliadi, and A. Farmadi: 2021 4th Int. Conf. Comput. Inform. Eng. (IC2IE) (Depok, Indonesia, 2021) 390. https://doi.org/10.1109/IC2IE53219.2021.9649207
22 J. Bergstra and Y. Bengi: J. Mach. Learn. Res. **13** (2012) 281. http://jmlr.org/papers/v13/bergstra12a.html
23 D. E. Puentes G., C. J. Barrios H. and P. O. A. Navaux: 2022 IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI) (Montevideo, Uruguay, 2022) 1. https://doi.org/10.1109/LA-CCI54402.2022.9981104
24 K. Fang, D. Lin, P. Winker, and Y. Zhang: U: Technometrics **42** (2000) 237. https://doi.org/10.2307/1271079
25 Q. Chen, D. Zhai, S. Li, and L. Zeng: 2011 IEEE Int. Symp. IT Med. Educ. **2** (2011) 165. https://doi.org/10.1109/ITiME.2011.6132081
26 J. Redmon, S. Divvala, R. B. Girshick, and A. Farhadi: 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) (Las Vegas, NV, USA, 2016) 779. https://doi.org/10.1109/CVPR.2016.91