Journal of Advanced Zoology

ISSN: 0253-7214

Volume 45 Issue S2 Year 2024 Page 123-130

Cloud-Enhanced Traffic Sign Classification With Wandb

Dasari Elsie Gnana Shiney [1*],Tathineni Sai Harshitha[2],Parasa Naga Sai Chaithanya[3],Nethala Sandhya[4],Shaik Sameer[5]

*[1 *] Assistant Professor, Department of IT, Seshadri Rao Gudlavalleru Engineering College(SRGEC),AP,shineycharles@gmail.com*

[2,3,4,5] Department of IT, Seshadri Rao Gudlavalleru Engineering College (SRGEC),AP,saiharshithatathineni@gmail.com[2],chaithanyaparasa@gmail.com3, sandhyanethala1315@gmail.com[4], sameershaik1909@gmail.com[5]

***Corresponding Author:** Dasari Elsie Gnana Shiney*

**Assistant Professor, Department of IT, Seshadri Rao Gudlavalleru Engineering College(SRGEC),AP,shineycharles@gmail.com*

Abstract

In the contemporary landscape of transportation, ensuring precise traffic sign prediction is paramount for both road safety and effective traffic management. This project introduces an innovative cloud-based approach that amalgamates YOLOv8, a state-of-the-art object detection model, with the Weights & Biases (wandb) platform to refine and optimize traffic sign detection within a cloud environment. The primary goal of this project is to formulate a cloud-centric traffic sign recognition system powered by YOLOv8, capitalizing on wandb's capabilities to streamline model development, training processes, and performance monitoring. YOLOv8's real-time object detection prowess makes it an ideal solution for swiftly identifying and localizing traffic signs amidst intricate visual scenarios. The project relies on cloud-based infrastructure as its foundation, delivering scalable computing resources crucial for intricate deep learning models such as YOLOv8. By harnessing the cloud, the project mitigates local computational constraints, thereby expediting model training and evaluation. The incorporation of wandb into this cloud environment facilitates dynamic visualization of vital training metrics, loss functions, and detection accuracy, furnishing real-time insights into the model's behavior. The collaborative and iterative aspects are pivotal to the project's success. Wandb's collaborative functionalities facilitate seamless teamwork among researchers and developers dispersed across different geographical locations. Through this integration, the project endeavors to contribute to safer roads and more intelligent traffic management, exemplifying the transformative potential of cloud-centric machine learning.

Keywords: YOLOv8, wandb

I. INTRODUCTION

With the rise of driver assistance systems, driverless cars, and traffic safety as a whole, traffic sign identification has become an important area in computer vision. When it comes to recognizing and detecting traffic signals, Convolutional Neural Networks (CNNs) have been very effective. However, the efficacy of these models is intricately tied to the availability of extensive labeled datasets for training purposes. While resources like the German Traffic Signals Dataset (GTSRB), the Chinese Traffic Sign Database (TSRD), and the Tsinghua Tencent 100K (TT100K) are available, gathering diverse, high-quality images of traffic signs from a wide range of countries is still a challenging and time-consuming task.
Image synthesis presents a solution to this challenge, allowing researchers to generate diverse and cost-effective training data by using standard templates. For Traffic Sign Detection (TSD), speed and accuracy in detecting small objects, especially at high speeds, are critical. CNN has shown effectiveness in this regard. Yolo, known for its efficiency and speed, has evolved with the introduction of Yolo V4 in 2020. Notably, Yolo V4 addresses the need for multiple GPUs during training, providing a fast and precise object detector even with a single GPU.
This paper delves into a comprehensive analysis of CNN models, focusing specifically on Yolo V7 and Yolo V8 for object identification. The study refines these models using a dataset created for Taiwan prohibitory signs, encompassing classes such as no entry, no stopping, no parking, and speed limit. Notably, the research aims to fill a gap in the literature by evaluating a substantial number of object detectors tailored for traffic sign recognition, considering critical aspects like mAP, IoU, and detection time.
In the broader landscape, there is a pronounced emphasis on the escalating significance of intelligent technology, specifically in the swift detection and recognition of traffic signs, a critical aspect within the domain of smart driving. The precise identification of traffic signs stands as a fundamental building block for the thorough analysis of environmental perception and decision-making processes within intelligent driving systems. Despite its indisputable significance, accurately detecting traffic signals in complex road settings is difficult and requires careful use of computer resources in addition to high accuracy.

In recent years, convolutional neural networks have gained prominence in traffic sign detection tasks, with scholars proposing various frameworks to enhance accuracy and speed. There have been suggestions for YOLOv3 optimizations such as pruning the network, branching the scale prediction, and improving the loss function. YOLOv5, introduced in 2020, offers a one-stage target detection algorithm known for its recognition accuracy and speed. However, challenges persist in balancing detection accuracy and speed, especially for small targets like traffic signs.
The introduction of deep learning, a subset of artificial intelligence, has revolutionized the approach to learning and decision-making. Deep learning involves the emulation of human learning processes to automate prediction-based analyses without the need for explicit programming. The evolution of deep learning entails the training of neural networks on labeled datasets, allowing the network to learn intricate features and correlations, ultimately enhancing its ability to recognize objects and make predictions autonomously.

II. LITERATURE SURVEY

The project's literature study delved into previous studies on locking mechanisms, approaches for guaranteeing data consistency in shared database settings, and concurrent data access.

Sun, Y., Ge, P., & Liu, D. (2019) introduces a traffic sign recognition system (TSRS) employing deep learning, image preprocessing, Hough Transform, and CNN. The proposed method achieves over 98.2% accuracy in identifying circular symbols in German datasets.

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016) Addressing challenges in simultaneous traffic-sign detection and classification, the paper introduces the Tsinghua-Tencent 100K benchmark and proposes a robust CNN demonstrating superior performance under varying conditions.

Bhatt, N., Laldas, P., & Lobo, V. B. (2022) This study presents a real-time Traffic Sign Detection and Recognition (TSDR) system that employs convolutional neural networks (CNNs) and deep learning. When tested on hybrid datasets that include both German and Indian traffic sign datasets, the suggested model performs well.

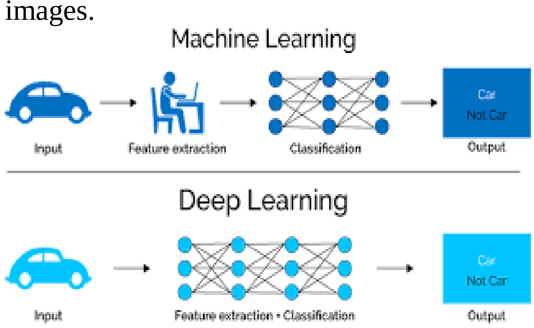
Islam, M. T. (2019), The paper proposes a system for detecting and classifying 28 globally recognized traffic signs using two separate CNNs and image augmentation for creating accurate training and validation datasets.

Macheev, E. M., Devyatkin, A. V., & Muzalevsky, A. R. Important for developing UAVs, this research discusses how to use convolutional neural networks for traffic sign detection and identification.

Oza, R. M., Geisen, A., & Wang, T. (2021): This paper focuses on using neural networks for identifying traffic sign patterns, employing image processing methods and Neural Networks stages to classify complex traffic sign patterns accurately.
Kapoor, A., Nehra, N., & Deshwal, D. (2021). The research employs Convolutional Neural Networks (CNN) for traffic sign recognition, comparing various CNN architectures and achieving an accuracy of 93.58%.
These papers collectively contribute to advancements in traffic sign detection and recognition systems, emphasizing the role of convolutional neural networks and deep learning techniques.

III. RELATED WORKS

3.1 EVOLUTION OF DEEP LEARNING
To grasp the evolution of deep learning, let's consider a scenario with a set of images, each depicting objects falling into four distinct categories. The objective is for the algorithm to identify the category of an object within any given image. A labeled dataset is created to train the network by associating categories with the images.



The network initiates the process of recognizing unique features, correlating them with specific categories. Successive layers build upon the information gathered from preceding layers, creating a hierarchical learning structure. As data traverses through the layers, the complexity of the learned details increases. Importantly, the network autonomously learns from acquired data, with no user influence on the specific details assimilated by the algorithm.

3.2 STRUCTURE OF NEURAL NETWORKS
In deep learning applications, the fundamental process remains consistent. The algorithm acquires data, subjects it to nonlinear transformations, learns through these transformations, and generates a model as the output. This iterative process unfolds through multiple layers, undergoing numerous trials until a reliable and accurate output is achieved.

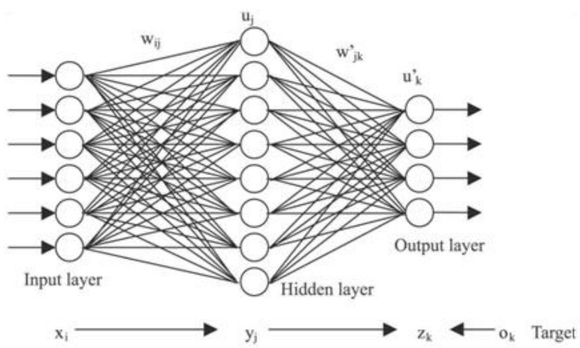


Fig – Neural Network

3.3 DIFFERENCE BETWEEN DEEP AND MACHINE LEARNING

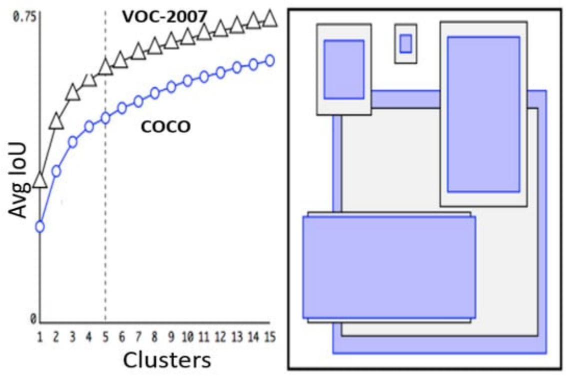
A significant distinction between deep learning (DL) and machine learning (ML) lies in the level of specificity required from the user. ML algorithms demand a high degree of user specificity, as the computer relies on explicit instructions to discern search parameters. Achieving high accuracy in ML necessitates meticulous manual inputs, relying entirely on the user's precision.

DL, on the other hand, excels in illustrating feature sets autonomously, eliminating the need for manual control or input while maintaining high accuracy levels. This not only expedites the process but also enhances reliability and precision, making DL a more efficient and self-sufficient approach.

IV. METHODOLOGY

4.1 YOLO-v2/9000

In 2016, Joseph Redmon introduced YOLO-v2/9000 as an improvement over the original YOLO architecture, aiming to address observed inefficiencies while preserving its remarkable speed. A key enhancement was the integration of batch normalization into the internal architecture, intended to improve model convergence and accelerate the training process. This addition played a crucial role in eliminating the need for additional regularization techniques, such as dropout. The result was a notable 2% improvement in mean average precision (mAP) when compared to the performance of the original YOLO model.



4.2 YOLO-v3

While earlier architectures like VGG underscored the correlation between deeper networks and enhanced accuracy, YOLO-v2 faced challenges with the downsampling process causing the loss of fine-grained features, especially for smaller objects. YOLO-v3 introduced a hybrid architecture, amalgamating components from YOLO-v2, Darknet-53, and ResNet. This innovative approach maintained fine-grained features by enabling gradient flow from shallow to deeper layers. Boasting 106 convolutional layers, including 53 from Darknet-53 and 53 for the detection head, YOLO-v3 also pioneered multi-scale detection to improve small object detection capabilities.

4.3 YOLO-v7

YOLO-v7, a successor to YOLO-v6, prioritized architectural enhancements and Trainable Bag-of-Freebies (BoF). The backbone of YOLO-v7 featured the Extended Efficient Layer Aggregation Network (E-ELAN), drawing inspiration from strides in network efficiency. Taking into account factors such as memory access cost, input/output channel ratio, and gradient path, YOLO-v7 aimed to enhance both accuracy and speed.

4.4 YOLO-v8

Introduced by Ultralytics in January 2023, YOLO-v8 stands as the latest iteration in the YOLO lineage. While a comprehensive paper release is pending, initial comparisons showcase its superiority over predecessors, including YOLO-v5. Figure 8 illustrates that YOLO-v8 variants, trained on 640 image resolution, surpass YOLO-v5 and YOLO-v6 in throughput with comparable parameter counts. This suggests a focus on hardware-efficient, architectural reforms, hinting at YOLO-v8's potential emphasis on high-inference speed for deployment on resource-constrained edge devices. Ultralytics' presentation of YOLO-v8 alongside YOLO-v5, renowned for real-time performance, implies a dedication to efficient edge device utilization.

4.5 ANNOTATION AND LABELING

Drag and Select:

Indicated by a hand icon, this functionality enables the selection, editing, and dragging of individual annotations. A single click on an existing bounding box selects it, allowing adjustments to size using circular white handles on corners and sides. Alternatively, use the class editor to modify the box's label.

• Move a box by dragging it.
• Pan by dragging the background.
• Deselect all boxes by clicking on the background.

Bounding Box Annotation Tool:

Represented by a rectangular box icon, this tool facilitates the creation of new bounding-box annotations. In this mode, crosshairs guide where to commence drawing. Click and drag across an image to generate a new annotation, followed by using the Class Selector to assign its label.

Polygon Annotation Tool:

This tool, indicated by a polygon icon, allows the drawing of polygonal annotations. Similar to the bounding box tool, crosshairs guide the initial points of the polygon.
Click to set each vertex, and double-click to close the polygon. Choose the label using the Class Selector.

Smart Polygon:

A more advanced version of the polygon tool, the smart polygon tool automatically detects object boundaries, streamlining the annotation process.
Click to place points, and the tool intelligently connects them to form a polygon. Confirm the polygon, and assign a label with the Class Selector.

Label Assist:

The label assist feature enhances annotation efficiency by providing suggestions for labels based on previously used annotations.
Select an annotation, and label suggestions will be offered, speeding up the labeling process.

Zoom Tool:

Facilitated by a magnifying glass icon, the zoom tool allows for a closer inspection of image details.
Click and drag to create a zoom box. To reset the view, double-click within the image.
These annotation and labeling tools offer a comprehensive set of features, empowering users to efficiently and precisely annotate images for various applications.

4.6 MODEL TRAINING

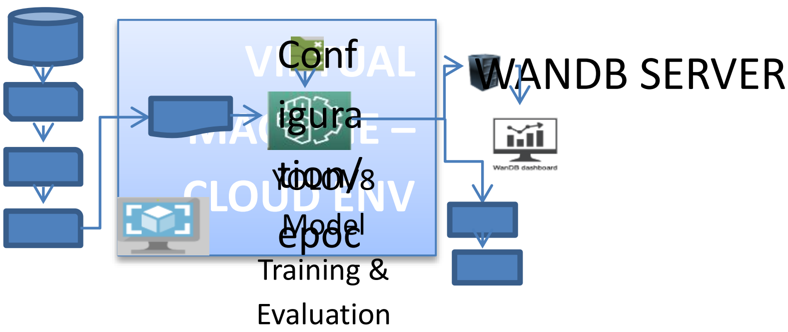


Fig – Architecture

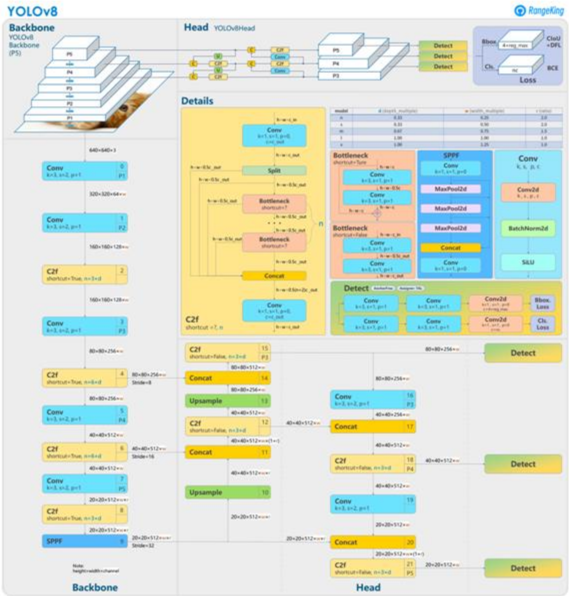
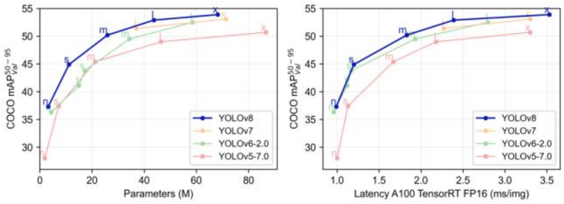


Fig – Yolo v8 Architecture

V. RESULTS AND DISCUSSIONS

5.1 EFFICIENCY AND ACCURACY



We may utilize the sample data that the Ultralytics writers have given us to compare the new release with previous iterations of YOLO. The plot above illustrates how YOLOv8 performs better during training in terms of mean Average Precision, size, and latency than YOLOv7, YOLOv6-2.0, and YOLOv5-7.0. The statistical comparison tables for the various sized YOLOv8 models are available on their own Github sites. The preceding table shows that when the size of the parameters, speed, and FLOPs rise, the mAP also increases. The highest mAP value of 50.7 was attained by the largest YOLOv5 model, YOLOv5x. The mAP gain of 2.2 units signifies a noteworthy enhancement in capabilities.

5.2 OUTPUT:

e	train/	train/	train/	metrics/p	metrics,	metric:	metrics/	val/b	val/c	val/d	lr	lr	lr/pg2
0	1.0569	4.6234	1.0922	0.033275	0.53383	0.06006	0.052226	0.81002	3.7863	0.85324	0.000114	0.000114	0.000114
1	1.01	3.6875	1.0017	0.19177	0.25775	0.09201	0.07365	0.79259	3.1911	0.8512	0.000224	0.000224	0.000224
2	1.0282	3.0241	1.0106	0.24596	0.39274	0.14687	0.11437	0.84658	2.7173	0.86328	0.00033	0.00033	0.00033
3	0.96583	2.6208	1.0007	0.23882	0.37044	0.18395	0.14912	0.74649	2.3769	0.85077	0.000325	0.000325	0.000325
4	0.9337	2.3847	0.98482	0.23729	0.41702	0.19601	0.15401	0.75692	2.2441	0.85768	0.000325	0.000325	0.000325
5	0.88696	2.2283	0.98109	0.24302	0.47305	0.24835	0.20084	0.71811	2.0455	0.8446	0.000318	0.000318	0.000318

Epoch: Indicating the training epoch or iteration number, this column signifies one complete traversal of the entire training dataset.
train/box_loss: This metric denotes the loss related to bounding box predictions during the training phase. It gauges the model's effectiveness in predicting object locations, particularly in identifying potential SIGN regions.
train/cls_loss: Focused on the classification component, this loss metric evaluates how well the model classifies identified objects or regions, potentially discerning between different classes like "sign."
train/dfl_loss: This loss may be linked to the model's dense feature learning or another specific architectural aspect.
val/box_loss: These values signify the validation losses for bounding box predictions, classification, and potentially other specific aspects.
lr/pg0, lr/pg1, lr/pg2: These columns may denote the learning rates for distinct parameter groups in the optimization process, commonly used in deep learning with techniques like learning rate schedules.
In summary, this table serves as a record detailing the model's performance during both training and validation across different epochs. It aids in evaluating whether the model is improving, overfitting, or converging to a

suitable solution for the traffic sign detection task. The provided losses and metrics offer insights into various facets of model performance, including object localization and classification accuracy.

VI. CONCLUSION

In conclusion, this project centered on the implementation of a traffic sign detection system utilizing YOLOv8, a state-of-the-art object detection model. The primary aim was to create an efficient and accurate solution for identifying and localizing traffic signs within intricate visual scenarios. YOLOv8's real-time object detection capabilities played a pivotal role in achieving the project's objectives. Its swift detection and classification of traffic signs contribute to enhanced road safety and streamlined traffic management. The model's robust performance in handling complex visual environments makes it well-suited for practical applications. The integration of YOLOv8 in traffic sign detection underscores the significance of leveraging advanced deep learning models to augment transportation systems. This project demonstrated the potential of YOLOv8 in providing reliable, real-time solutions for addressing challenges associated with traffic sign recognition. In future endeavors, further optimizations and fine-tuning of the YOLOv8 model can be explored to enhance its performance in specific traffic scenarios. Additionally, integration with real-world traffic systems and deployment in diverse environments would be essential steps to validate the practical effectiveness of the developed solution.

REFERENCES

1. Bangquan X, Xiong WX (2019) Real-time embedded traffic sign recognition using efficient convolutional neural network. IEEE Access 7:53330–53346
2. Chen Q, Huang N, Zhou J, Tan Z (2018) An SSD algorithm based on vehicle counting method. Chinese Control Conference
3. Ellahyani A, Ansari M, Lahmyed R, Trémeau A (2018) Traffic sign recognition method for intelligent vehicles. J Opt Soc Am 35(11):1907–1914. https://doi.org/10.1364/JOSAA.35.001907
4. Garg P, Chowdhury DR, More VN (2019) Traffic sign recognition and classification using YOLOv2, Faster R-CNN and SSD. International Conference on Computing, Communication and Networking Technologies
5. Hao G, Yingkun Y, Yi Q (2019) General target detection method based on improved SSD. IEEE Joint International Information Technology and Artificial Intelligence Conference
6. He Z, Nan F, Li X, Lee SJ, Yang Y (2020) Traffic sign recognition by combining global and local features based on semi-supervised classification. IET Intel Transport Syst 14(5):323–330
7. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
8. Hu GX, Hu BL, Yang Z, Huang L, Li P (2021) Pavement crack detection method based on deep learning models. Wirel Commun Mob Comput 2021. https://doi.org/10.1155/2021/5573590
9. Huo A, Zhang W, Li Y (2020) Traffic sign recognition based on improved SSD model. International Conference on Computer Network, Electronic and Automation
10. Jin Y, Fu Y, Wang W, Guo J, Ren C, Xiang X (2020) Multi-feature fusion and enhancement single shot detector for traffic sign recognition. IEEE Access 8:38931–38940
11. Kuznetsova A, Maleva T, Soloviev V (2020) Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images. International Symposium on Neural Networks
12. Li S, Gu X, Xu X, Xu D, Zhang T, Liu Z, Dong Q (2021) Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm. Constr Build Mater 273:121949
13. Lian J, Yin Y, Li L, Wang Z, Zhou Y (2021) Small object detection in traffic scenes based on attention feature fusion. Sensors 21(9):3031
14. Lim K, Hong Y, Choi Y, Byun H (2017) Real-time traffic sign recognition based on a general-purpose GPU and deep-learning. PLoS One 12(3):e0173317
15. Liu X, Yan W (2021) Traffic-light sign recognition using capsule network. Multimed Tools Appl 80:15161–15171
16. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: Single shot multibox detector. European Conference on Computer Vision
17. Qin Z, Yan W (2021) Traffic-sign recognition using deep learning. International Symposium on Geometry and Vision (ISGV). Springer, Berlin, pp 13-25

18. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You Only Look Once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition
19. Shi X, Hu J, Lei X, Xu S (2021) Detection of flying birds in airport monitoring based on improved YOLOv5. International Conference on Intelligent Computing and Signal Processing
20. Sun W, Hongji D, Nie S, He X (2019) Traffic sign recognition method integrating multilayer features and kernel extreme learning machine classifier. Comput Mater Continua 60(1):147–161
21. Wang C (2018): Research and application of traffic sign detection and recognition based on deep learning. International Conference on Robots & Intelligent System
22. Wu Y, Qin X, Pan Y, Yuan C (2018) Convolution neural network based transfer learning for the classification of flowers. IEEE International Conference on Signal and Image Processing
23. Xiaoping Z, Jiahui J, Li W, Zhonghe H, Shida L (2021) People's fast-moving detection method on buses based on YOLOv5. Int J Sens Sensor Netw 9(1):30
24. Xing J, Yan W (2021) Traffic sign recognition using guided image filtering. International Symposium on Geometry and Vision (ISGV), Springer, Berlin, pp 85-99
25. Xu S, Niu D, Tao B, Li G (2018) Convolutional neural network based traffic sign recognition system. In International Conference on Systems and Informatics (ICSAI), pp 957-961
26. Yan B, Fan P, Lei X, Liu Z, Yang F (2021) A real-time apple targets detection method for picking robot based on improved YOLOv5. Remote Sensing 13(9):1619
27. Yao Y, Yang Y, Su X, Zhao Y, Feng A, Huang Y, Pu H (2019) Optimization of the bounding box regression process of SSD model. International Conference on Computer Engineering, Information Science & Application Technology
28. Yu G, Fan H, Zhou H, Wu T, Zhu H (2020) Vehicle target detection method based on improved SSD model. J Artif Intell 2(3):125
29. Zhang J, Hui L, Lu J, Zhu Y (2018)Attention-based neural network for traffic sign detection. International Conference on Pattern Recognition