

Pedestrian Detection System Based on Deep Learning Algorithm

Won-hyuk Choi¹, Woo-Jin Jung^{*2}

¹Department of Aeronautical System Engineering, Hanseo University, Taean 32158, Korea

²Department of Avionics, Hanseo University, Taean 32158, Korea

Cite this paper as: Won-hyuk Choi, Woo-Jin Jung, (2025) Pedestrian Detection System Based on Deep Learning Algorithm. *Journal of Neonatal Surgery*, 14 (4), 291-298.

ABSTRACT

In South Korea, the incidence of pedestrian traffic accidents is higher than the Organisation for Economic Co-operation and Development (OECD) average. In response, recent legal regulations have been strengthened to prevent accidents in school zones, with a greater focus on pedestrian safety. Consequently, the necessity for real-time pedestrian detection systems is becoming increasingly apparent. This study proposes the implementation of a deep learning-based pedestrian detection system, which would enable drivers to accurately detect and make informed decisions regarding pedestrians, vehicles, and crosswalks. The study employs a monocular camera and an image segmentation algorithm to compare the architectures of R-CNN and YOLOv8. Subsequently, the YOLOv8-seg model, which incorporates a Segmentation Branch structure for instance segmentation, was trained and tested on a variety of models. Subsequently, the system's functionality was validated through real-time streaming within a vehicle.

Keywords: Pedestrian, Yolo, Driver, Deep Learning, Image Segmentation.

1. INTRODUCTION

In South Korea, approximately 35,000 pedestrian-related accidents occur annually, resulting in an average of 35,000 injuries. The most recent data indicate that approximately 900 individuals in South Korea have perished as a result of pedestrian-related incidents, representing a rate of approximately 1.7 per 100,000 individuals. In comparison to the average for Organization for Economic Co-operation and Development (OECD) countries, this figure is markedly elevated. Moreover, the number of pedestrian accidents involving children in school zones has averaged more than 500 incidents over the past five years, with more than 500 injuries sustained in each instance. These figures indicate the necessity for additional measures to enhance pedestrian safety.

On December 24, 2019, the National Assembly enacted a bill entitled the Child Protection Zone Manslaughter Act. This legislation represents an amendment to the Road Traffic Act and the Special Law, also referred to as the Minshik Law. The Child Protection Zone Death Penalty Act, which was enacted on March 25, 2020, is designed to enhance safety and prevent accidents in child protection zones and crosswalks. Moreover, the Road Traffic Act, also designated as the Right Turn Law, has been in force since the conclusion of January 2023. In accordance with the Enforcement Rules of the Right Turn Law, drivers are obliged to halt at designated stop lines, crosswalks, and intersections prior to executing a right turn when the traffic signal is red. A growing number of potential strategies for enhancing pedestrian safety in pedestrian crossing areas are being considered, including the installation of auxiliary traffic signals, such as right turn signals.

The objective of this research is to propose a driver-based pedestrian safety algorithm that can readily discern the status of pedestrian signals, crosswalks, and sidewalk areas from the driver's perspective, thereby assisting pedestrians in maintaining their safety. In this study, we utilized a monocular camera and one of the object detection algorithms, namely image segmentation. Representative deep learning models that provide segmentation include FCN [1], SegNet [2], U-Net [3], Mask R-CNN [4] and YOLO [5]. Each of these models possesses distinctive features and strengths, and their performance may vary contingent on the specific application. To detect pedestrians, crosswalks, and cars from the perspective of the driver, it is necessary to utilize the vehicle's cameras and implement real-time image processing to apply the algorithm to the moving vehicle's cameras. For this reason, the YOLO model was selected for use in this research project. Among the models under consideration, the YOLOv8 model was selected based on its provision of instance segmentation. To facilitate object recognition and learning, 80 pre-trained object detection models were employed as part of the COCO-seg (Common Object in Context-Segmentation) dataset of YOLOv8. Subsequently, the performance of each model was compared and analyzed.

2. SELECT OBJECT DETECTION ALGORITHM

The structural synthesis of CCPGTs will be performed based on the creative design methodology process [7-8]. Fig. 3 shows the flow chart for the approach. The process consists of six steps:

2.1 A comparative Analysis of Image Segmentation Deep Learning Models

Image segmentation is a computer vision task that categorizes images into pixels and assigns each pixel to a specific class. The objective is to distinguish the boundaries of objects or recognize backgrounds across an image, and it is employed in a number of fields, including autonomous driving, medical image analysis, and satellite processing.

The FCN model, the inaugural model to utilize a fully convolutional neural network, preserves location information by eliminating the fully connected layer and integrating a convolutional layer, upsampling, and skip connection, in contrast to conventional CNNs. Moreover, it enables the prediction of classes for each pixel within the image. The SegNet model employs a comparable architectural configuration to the FCN model, augmented with a network of decoders to enhance the preservation of intricate image details. The U-Net model, which was developed for the processing of biological images, incorporates a skip connection between the encoder and decoder with the objective of restoring details that may be lost during the encoding process. The Mask R-CNN model performs both object detection and instance segmentation and is based on the Faster R-CNN [6] model, with the additional step of generating a mask on a pixel-by-pixel basis.

The YOLO series employs a one-stage structure for object detection and instance segmentation, which enables the simultaneous detection, classification, and location prediction of objects. This structure, which is distinct from the two-stage structure typically employed in object detection, has been devised with the objective of maximizing computational efficiency. The backbone and head structure used in YOLO represents another example of the series' efforts to optimize processing..

2.2. Algorithm Selection

The objective of this study was to investigate the feasibility of object detection and real-time image processing. Among the various models, Mask R-CNN and YOLO are capable of utilizing instance segmentation as an object detection model. As driver-based pedestrian detection systems are designed to recognize pedestrians, cars, and crosswalks, as well as to determine the location and precise pixel-by-pixel boundaries of objects, instance segmentation is a necessary technique for clearly distinguishing each object.

Mask R-CNN is a model that is based on the Faster R-CNN architecture. Mask R-CNN incorporates a Segmentation Mask Branch structure subsequent to the RoIAlign step, thereby facilitating the extraction of precise features of the candidate region and the prediction of a pixel-by-pixel mask for each object. The RoIAlign process is employed for the precise extraction of features pertaining to the candidate regions. The previously utilized RoIPool serves to prevent distortion of the features due to alignment issues, while RoIAlign facilitates accurate mask segmentation. Figure 1 depicts the Segmentation Mask Branch structure, which is integrated subsequent to the RoIAlign step.

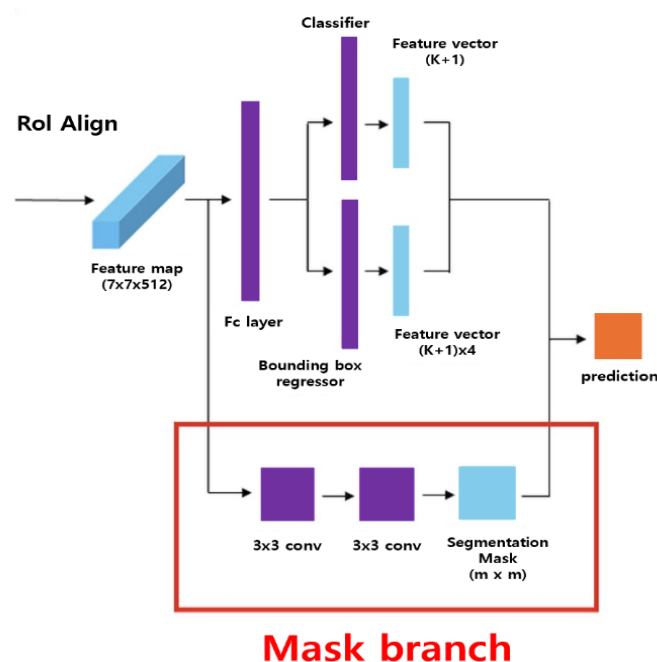


Fig. 1 Mask branch Module of Mask R-CNN

The YOLO algorithm employs a one-stage detection method, and the YOLOv8 model is equipped with instance segmentation capabilities. YOLO employs a backbone structure, and YOLOv8 utilizes CSPNet [7] for the purpose of feature extraction. This approach involves simplifying the backbone structure to enable accelerated computation. Furthermore, the Anchor-Free method enables the expeditious detection of objects irrespective of their shape or size. The anchor box is removed in order to enhance computational efficiency. In the neck structure, the FPN and PANet algorithms are employed to integrate feature maps of varying resolutions, thereby enabling the detection and segmentation of objects of disparate sizes with the integrated information.

The primary function of the YOLO head structure is to predict and detect the coordinates of bounding boxes and class labels. The YOLOv8-seg model incorporates a segmentation branch structure into the aforementioned structure. The implementation of the segmentation branch is in accordance with the YOLACT [8] principle. The YOLACT algorithm has been proposed as a means of efficiently performing instance segmentation, thereby facilitating the generation of instance segmentations using prototype masks and mask coefficients in a more straightforward and time-efficient manner. Figure 2 depicts the YOLACT network structure. The Segmentation Branch structure, which was introduced in YOLOv8-seg [9], receives a feature map from the Head and employs the FCN structure to generate a pixel-by-pixel mask. The Mask Head component is responsible for predicting a segmentation mask, which is capable of separating the regions of each object in the image on a pixel-by-pixel basis [10].

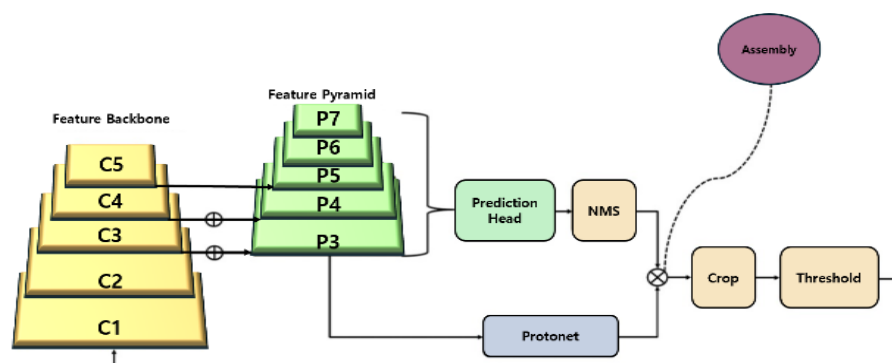


Fig. 2 Structure of YOLACT network

The YOLOv8 model is distinguished by its parallel processing of the detection and segmentation stages. The simultaneous prediction of the Bounding Box and Segmentation Mask allows for the computation of the Segmentation Branch to be performed without a significant increase in model complexity. This feature enables the implementation of Instance Segmentation while maintaining the real-time performance of the model. Conversely, Mask R-CNN is distinguished by its precise boundary segmentation and high segmentation accuracy through the use of the RoIAlign structure. However, it is limited by a slower processing speed compared to YOLOv8.

For the aforementioned reasons, this study employs the YOLOv8 model to conduct experiments [11]. The YOLOv8 Segmentation model provides five COCO-seg datasets comprising 80 pre-trained classes. The COCO-seg datasets, designated n, s, m, l, and x, exhibit elevated mAPs in the backward direction; however, they require a considerable time investment for training. Table 1 illustrates the mAP performance of the COCO-seg models. It can be observed that the mAP performance of all five models varies depending on the version. In this study, we evaluate the algorithms and assess their performance when training the models on a consistent custom dataset, with the same epoch, batch, and training configuration[12].

Table. 1: COCO-seg Dataset

Model	mAPbox	mAPmask
v8n-seg	36.7	30.5
v8s-seg	44.6	36.8
v8m-seg	49.9	40.8
v8l-seg	52.3	42.6
v8x-seg	53.4	43.4

3. TRAINING ALGORITHMS

3.1. Configure Datasets

The dataset is comprised of a total of 1,725 images. The datasets utilized consisted of images of crosswalks, pedestrians, and cars, which were collected using a dashboard camera. The training, validation, and test datasets comprised 1,509, 144, and 72 images, respectively. For purposes of illustration, a representative training image is provided below. In the context of object detection, the three classes utilized for training were "car," "person," and "crosswalk." To guarantee the accurate delineation of object boundaries, we utilized Roboflow, a computer vision software that facilitates the polygon labeling method.



Fig. 3 Simulation Dataset

3.2. Training the Algorithm

Once the Pytorch and CUDA environments had been established within the virtual environment on the RTX 3060 ti single GPU, five versions of YOLOv8-seg models (n, s, m, l, and x) were trained with 50 epochs and 32 batch sizes on a Jupyter laptop. The training time for all models was approximately eight hours. The v8x-seg model, the longest of the five, required approximately seven hours to train, while the shortest, the v8n-seg model, required approximately 16 minutes. It can be observed that there is a slight discrepancy in the training speed between the v8n-seg and v8s-seg models.

Table. 2: Simulation Sertting

Category	Setting
GPU	RTX 3060 ti
DATA Set	Train: 4056, Validation: 397,Test: 185
Pytorch	2.0.1
Cuda	12.6
Batch	32
Epoch	50
learning Rate	0.01

3.3. Analyze Results

In examining the outcomes of this experiment, graphical representations were utilized to evaluate the mAP50, precision, recall, and loss values. The aforementioned values serve as pivotal indicators for the assessment and analysis of the performance of object detection and instance segmentation in deep neural networks. The mAP50 serves as an indicator of the model's accuracy in detecting and evaluating objects. The mAP calculates the average precision for different object classes and is employed to evaluate the accuracy and performance of the model. mAP can be utilized to ascertain the extent to which the model effectively detects and classifies diverse objects and to determine the degree of correspondence between the predicted and actual bounding boxes. As the term "precision" implies, this value signifies the proportion of accurately identified instances among all those that were classified.

It represents the proportion of objects identified by the trained model that were correctly identified. A high level of precision

indicates that the model is accurately detecting objects, with minimal false positives.

Precision allows for the evaluation of the model's ability to accurately identify the specified classes. The recall metric assesses the model's capacity to identify genuine objects, with a higher value indicating a reduced incidence of false negatives. The recall value can be analyzed to ascertain the model's ability to detect objects belonging to a specified class. The loss function indicates the discrepancy between the model's predicted outcomes and the actual values within the model's class. In order to gain a comprehensive understanding of the model's performance, it is essential to consider both the training loss and the validation loss when analysing the loss value. The training loss demonstrates the model's capacity to predict the training data, whereas the validation loss reflects the model's ability to generalize to the validation data. An examination of these loss values may reveal whether the model is exhibiting signs of overfitting or underfitting. A high validation loss value in conjunction with a low training loss value is indicative of overfitting.

Table 3 illustrates the values of mAP50, recall, and precision obtained after training the models 50 times. The final values are presented in tabular form. Upon examination of the results, it becomes evident that the models exhibit comparable values with minimal discrepancies. The v8l-seg model exhibits the highest mAP50 value, the v8s-seg model demonstrates the highest recall value, and the v8x-seg model displays the highest precision value.

Table 3. Training Results

Model	mAP@50	Recall	Precision
v8n-seg	0.8059	0.7569	0.7875
v8s-seg	0.8092	0.7937	0.7707
v8m-seg	0.8051	0.7561	0.8034
v8l-seg	0.811	0.7883	0.8029
v8x-seg	0.8083	0.7516	0.8064

Figure 4 below illustrates the metrics of mAP50. Upon training the instance segmentation model with YOLOv8, it is possible to observe both the metric graphs for the box and the metric graphs for the mask. Moreover, the mask graph enables the visualization of the segmentation value. It can be observed that the v8s-seg model exhibits the lowest mAP value until epoch 20. Subsequently, it is evident that the v8n-seg model exhibits the lowest mAP value. From epoch 35 onwards, the v8l-seg model exhibits the highest mAP value and demonstrates consistent superiority in training performance.

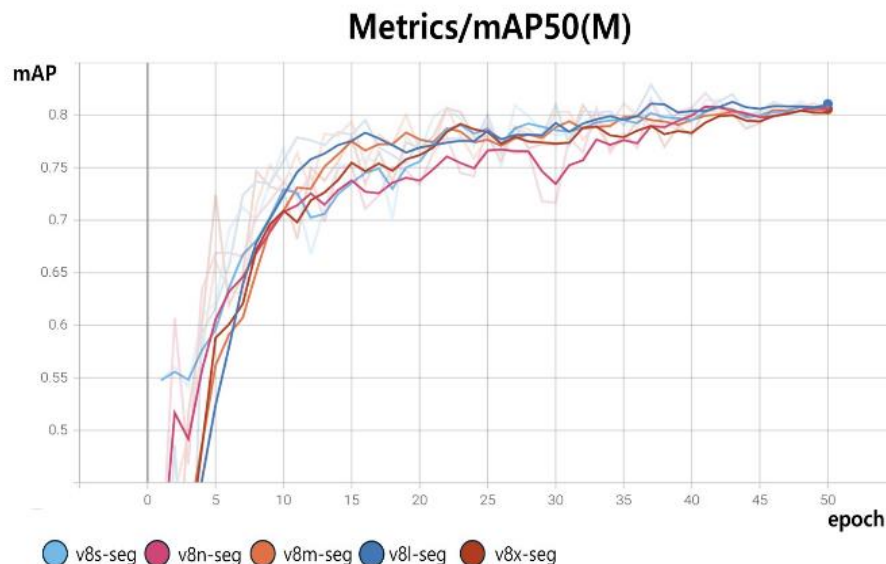


Fig. 4 mAP Results

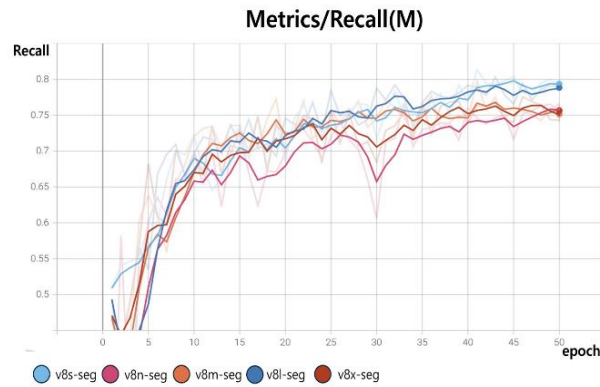


Fig. 5 Recall Results

Figure 5 depicts a graph illustrating the recall metric. As illustrated in the graph, the v8n-seg model demonstrates a persistent learning trajectory, exhibiting the lowest value throughout the process. In contrast, after epoch 30, both the v8s-seg and v8l-seg models exhibit a notable advancement in learning, as evidenced by their elevated recall values. Figures 6 and 7 illustrate the values of training loss and validation loss, respectively. As illustrated in both graphs, the v8n-seg model exhibited the most consistent training, demonstrating the highest values throughout. In contrast, the three graphs for v8m-seg, v8l-seg, and v8x-seg demonstrate that the values of v8m-seg, v8l-seg, and v8x-seg were consistently low.

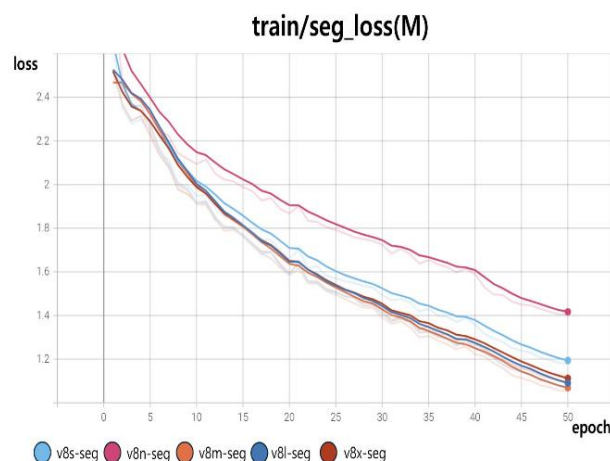


Fig. 6 Training / Loss Results

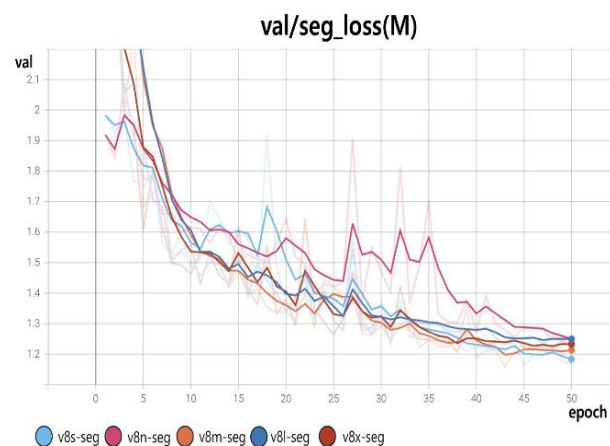


Fig. 7 Validation / Loss Results

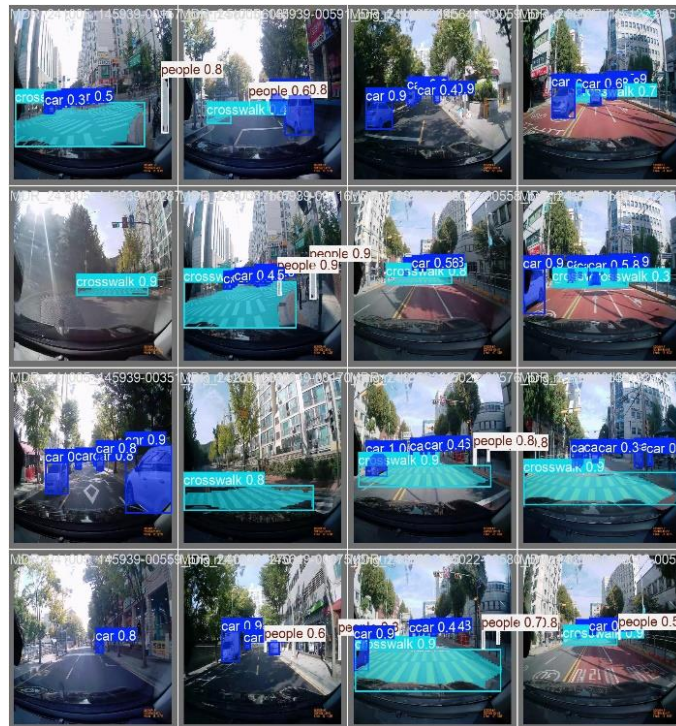


Fig. 8 YOLOv8x-seg Model TrainingResults



Fig. 9 YOLOv8x-seg Model Live streaming

Figure 9 shows an experiment with live streaming from a car using a webcam connected to a laptop.


4. CONCLUSIONS

This paper presents an algorithm capable of recognizing pedestrians using a driver base. We presented and analyzed an algorithm that is capable of not only recognizing pedestrians, but also identifying crosswalks and cars using deep learning techniques. The YOLOv8-seg model incorporates a segmentation branch structure into the algorithm, thereby enabling instance segmentation. Following an analysis and verification of the COCO-seg dataset provided by YOLOv8, it was determined that learning 50 times yielded comparable performance outcomes, irrespective of the existing models' performance. Furthermore, a comparison of the loss value between the S model, which exhibits superior performance compared to the N model, revealed that the loss value of the S model is higher. In the future, further experiments and analysis will be conducted on this aspect, with the aim of training models with a larger amount of data sets or learning in different configurations. Moreover, the objective is to integrate this technology into embedded devices, such as the Jetson Nano and Raspberry Pi, for practical applications through real-time detection

ACKNOWLEDGMENTS

This research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-004)

REFERENCES

- [1] Long, J., Shelhamer, E., and Darrell, T., “Fully Convolutional Networks for Semantic Segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, pp. 3431-3440, June 2015. DOI: 10.1109/CVPR.2015.7298965.
- [2] Badrinarayanan, V., Kendall, A., and Cipolla, R., “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), pp. 2481-2495, 2017. DOI: 10.1109/TPAMI.2017.1699961.
- [3] Ronneberger, O., Fischer, P., and Brox, T., “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, pp. 234-241, 2015. DOI: 10.1007/978-3-319-24574-4_28.
- [4] He, K., Gkioxari, G., Dollár, P., and Girshick, R., “Mask R-CNN,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 2961-2969, Oct. 2017. DOI: 10.1109/ICCV.2017.322.
- [5] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., “You Only Look Once: Unified, Real-Time Object Detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788, June 2016. DOI: 10.1109/CVPR.2016.91.-2969, Oct. 2017. DOI: 10.1109/ICCV.2017.322.
- [6] Girshick, R., “Fast R-CNN,” [Internet]. Available: <https://arxiv.org/abs/1504.08083>.
- [7] Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., and Yeh, I. H., “CSPNet: A New Backbone That Can Enhance Learning Capability of CNN,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, June 2020. DOI: 10.1109/CVPRW50498.2020.00359.
- [8] Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J., “YOLACT: Real-time Instance Segmentation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, pp. 9157-9166, Oct. 2019. DOI: 10.1109/ICCV.2019.00925.
- [9] Zhao, X., Liu, Y., Li, Z., and Wang, H., “Improved YOLOv8-Seg Based on Multiscale Feature Fusion and Deformable Convolution for Weed Precision Segmentation,” in Proceedings of the International Conference on Agricultural Robotics and Automation (ICARA), Beijing, China, pp. 112-118, July 2023. DOI: 10.1016/j.compag.2023.107453.
- [10] Chen, M., Li, J., Zhang, Y., and Wu, Q., “LAtt-Yolov8-seg: Video Real-time Instance Segmentation for Urban Street Scenes Based on Focused Linear Attention Mechanism,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Vancouver, BC, Canada, pp. 459-467, June 2024. DOI: 10.1109/CVPRW.2024.00459.
- [11] Liu, X., Wang, Y., Chen, R., and Zhang, L., “Improved YOLOv8-Seg Network for Instance Segmentation of Healthy and Diseased Tomato Plants in the Growth Stage,” in Proceedings of the International Conference on Agricultural and Environmental Informatics (ICAEI), Tokyo, Japan, pp. 102-110, August 2024. DOI: 10.1016/j.compag.2024.102345.
- [12] Smith, J., Lee, A., and Rodriguez, M., “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” in Journal of Computer Vision and Applications, vol. 45, no. 3, pp. 123-145, March 2024. DOI: 10.1016/j.jcva.2024.103456.
- [13]  Copyright© by the authors. Licensee TAETI, Taiwan. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC) license (<http://creativecommons.org/licenses/by/4.0/>).