Instance Segmentation of Road Marking Signs Using YOLO Models

Rung-Ching Chen, Wei-Kai Chao, William Eric Manongga *, and Chayanon Sub-r-pa

Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan
Email: crching@cyut.edu.tw (R.-C.C.); s11214616@cyut.edu.tw (W.-K.C.); s11014907@cyut.edu.tw (W.E.M.); t5220317@gm.cyut.edu.tw (C.S.)
*Corresponding author

Abstract—**Recently, Taiwan has witnessed a significant rise in the number of vehicles, including cars and motorcycles, leading to increased traffic accidents. In many instances, unclear or improperly marked road markings have led drivers to misjudge driving directions, resulting in accidents and penalties. Addressing the challenge, our study focuses on developing a system for detecting road markings, which can help build an Advanced Driving Assistant System (ADAS) and reduce the number of accidents caused by drivers' negligence of road marking signs. We employed and compared the performance of YOLOv5n-seg and YOLOv8n-seg, two versions of You Only Look Once (YOLO) version for instance segmentation. We also compiled and proposed our dataset for instance segmentation of Taiwan road marking signs. Our research shows that YOLOv8n-seg performs better than YOLOv5n-seg in segmenting Taiwan road marking signs. YOLOv8n-seg also converges faster during training, leading to shorter training time than YOLOv5n-seg.**

Keywords—**road marking sign, Advanced Driving Assistant System (ADAS), Instance segmentation, You Only Look Once (YOLO)**

I. INTRODUCTION

Taiwan has witnessed a significant rise in vehicles, including cars and motorcycles in recent years, leading to increased traffic accidents. The trend has brought traffic safety into the spotlight, becoming a frequently discussed issue. From January to September 2023, data from the Taiwan Road Safety Information Inquiry Network indicated that common causes of car accidents nationwide include various factors such as inattention to the road, inadequate driving distance, traffic signal violations, improper lane changes, and road marking violations. In many instances, unclear or improperly marked road markings have led drivers to misjudge driving directions, resulting in accidents and penalties.

To address the challenge, our study focuses on developing a system for detecting road markings. As a preliminary step, we compile a comprehensive road marking dataset specific to Taiwan. Our dataset is distinctive from previous datasets in Taiwan, which predominantly targeted object detection. In contrast, our dataset emphasizes instance segmentation, which is categorized according to our defined classifications.

We have meticulously curated the Taiwan road marking dataset, emphasizing instance segmentation. Our paper will detail the dataset, its structure, and the methodology behind its creation. For the training, YOLOv5 and YOLOv8 segmentation models are employed. We aim to ascertain the more effective model through experimental analysis and identify reasons for underperformance in specific categories.

The primary contribution of our research lies in creating the Taiwan road marking segmentation dataset, which labels 14 common Taiwan road marking signs for instance segmentation. The annotations are provided in the You Only Look Once (YOLO) format. We also used YOLO models to perform instance segmentation in real-time. Furthermore, we conduct a comparative analysis of the experimental outcomes using YOLOv5 and YOLOv8 segmentation models. The comparison not only elucidates the efficacy of these models but also lays the groundwork for the future development of a road marking detection system tailored to Taiwan's unique traffic conditions.

The remaining part of this paper is organized as follows: Section II provides an overview of the related works; Section III describes and explains the materials and methodology of our research; Section IV presents the experiment results, results analysis, and discussions of the results; and Section V provides the conclusion.

II. LITERATURE REVIEW

A. Instance Segmentation

Instance segmentation [1] is the simultaneous detection, segmentation, and classification of object instances in pictures, combining object detection [2] (identifying and locating the bounding boxes of objects) and semantic segmentation [3] (classifying each object into a category). Outputs a set of detected objects and instances, assigning each object and instance the same class label and providing greater detail than bounding boxes. It can also identify multiple instances of the same category, such as separating all instances of a right-turn arrows in an image rather than just labeling all right-turn arrows. Popular instance segmentation datasets include Microsoft Common Objects in Context (MSCOCO) [4], Cityscapes [5], and Mapillary

Manuscript received February 15, 2024; revised March 5, 2024; accepted April 12, 2024; published October 15, 2024.

Vistas [6]; they all provide labels and ground truth for images. These datasets are often used with advanced methods for training, such as various versions of Mask Region-based Convolutional Neural Network (R-CNN) [7], You Only Look At CoefficienTs (YOLACT) [8], and You Only Look Once (YOLO) [9].

B. Road Marking Sign Segmentation

Road marking segmentation is paramount in Advanced Driver Assistance Systems (ADAS), which are pivotal for the future of road transportation. Zhang *et al*. [10] have delved into the various challenges encountered in this domain, such as fluctuating lighting conditions and obstructions. Their work provides a comprehensive overview of the evolution from conventional methodologies to more sophisticated deep learning techniques in the realm of lane marking detection.

In recent research, Mamun *et al*. [11] proposed Encode-Decode Instant Segmentation Network (EDIS-Net) which is based on the E-net [12] architecture with combined cross-entropy and discriminative losses, to segment lane markings. Liu *et al*. [13] introduce ASA-BiSeNet, which can do real-time road lane semantic segmentation in low-light driving scenes. Hsieh *et al*. [14] focus on segmenting road speed limit marking using mask R-CNN. In their research, Nguyen *et al*. [15] applied the combination of mask R-CNN and Otsu's algorithm to segment and detect deterioration of road line markings. Due to the lack of datasets and the class limitations of existing datasets, Wu *et al*. [16] introduced the RMD (Road Marking Dataset), combined with their Multiscale Attention-Based Dilated Convolutional Neural Network (MSA-DCNN) to segment the road sign markings. This research either focuses on the segmentation of road lane marking only [11, 13, 15] or limited road marking categories [14], while [16] focuses more on the segmentation accuracy rather than real-time segmentation.

C. YOLO for Instance Segmentation

You Only Look Once (YOLO) [17] is a popular object detection algorithm and framework that enables computers to detect and classify objects efficiently. Redmon *et al*. [17] initially presented YOLO in 2016, and it has since undergone several updates.

The YOLOv5 [18] model, developed by the Ultralytics team in 2020, has established itself as a notable entity in deep learning, primarily for its single-stage object detection proficiency. Originated by Glenn Jocher in June 2020, YOLOv5 represents a significant evolutionary stride in computer vision. Many research studies in object detection are done based on YOLOv5 [19, 20].

Architecturally, YOLOv5 is divided into the backbone, the neck, and the head. The backbone, employing Cross Stage Partial Networks (CSP) [21], is pivotal in feature extraction from the input image, capturing vital characteristics and details. The neck's primary role is to develop a feature pyramid network, enhancing the model's ability to effectively identify objects across varying scales. The feature pyramid is instrumental in recognizing the same object at assorted sizes and proportions. The head component is responsible for the final regression prediction, ascertaining the specific details and classes of the objects within the image.

In the rapidly evolving field of computer vision, YOLOv8 [22] stands out as a pivotal advancement within the esteemed YOLO series. The new version introduces a transformative shift in the real-time object detection and segmentation approach. YOLOv8 adopted the anchor-free detection method [9], which marks a departure from the conventional anchor box system, focusing instead on the direct prediction of object centers. The change not only streamlines the detection process, making it more efficient, particularly in accelerating the Non-Maximum Suppression (NMS), and addresses historical inefficiencies associated with anchor boxes. Further enhancing YOLOv8's capabilities significantly improves its convolutional layers [9]. The model has seen the replacement of the traditional 6x6 convolution with a more efficient 3x3 variant and strategic modifications in the bottleneck structure, leading to an overall boost in performance.

In September 2022, YOLOv5 introduces an additional feature: the ability to do instance segmentation. Both YOLOv5 and YOLOv8 for instance segmentation adopts the same method, by adding the ProtoNet, combined with the YOLO detection head, which is extended by the addition of an additional head to output the mask coefficient to produce instance segmentation masks. ProtoNet is a small, fully connected neural network that is similar to those used for semantic segmentation that produces prototype masks.

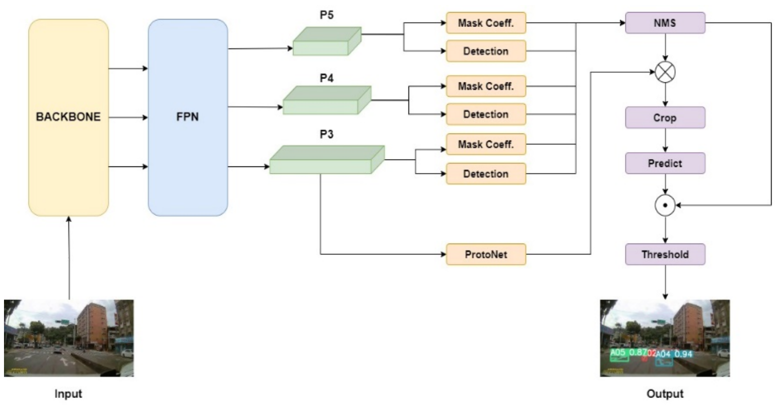


Fig. 1. General Architecture of YOLO for instance segmentation.

Fig. 1 shows the general architecture of YOLO for instance segmentation YOLOv5 and YOLOv8 for instance segmentation are both available in various model sizes, including: nano (n), small (s), medium (m), large (l), and extra-large (x). The multiple model scales can better accommodate training of customized datasets with a wide range of training requirements.

III. MATERIALS AND METHODS

A. Research Workflow

Fig. 2 depicts our entire research workflow. To conduct our experiment, we created our dataset. We collected images for the dataset from multiple sources. Two internet sources used for data collection are Mapillary and Google Maps. Mapillary is a community-based road mapping application that provides road images in street-level view, similar to Google Street View from Google Maps. The street images from Mapillary are carefully selected and downloaded using the provided API, while images from Google Street View are collected by taking screenshots of the selected images.

Additionally, some images are collected using a driving recorder camera. The recording is done by using a camera attached to a motorcycle helmet. The recorded video was converted into image frames before being labeled.

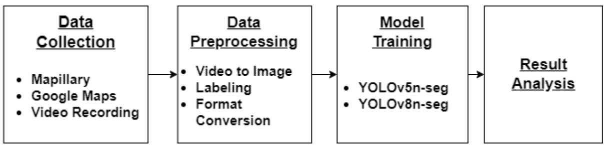


Fig. 2. Research workflow.

The labeling process was conducted manually using Labelme [23], an open-source image labeling tool. The images are labeled for instance segmentation, focusing on the road marking signs in Taiwan. When labelling the road marking signs, we label the road markings by the instance. We label each line as a separate instance for road line road marking, while road markings for pedestrian crossing are labeled as one instance, not per line. LabelMe generates an output in the JSON format for each labeled image. Since a YOLO format annotation is required, annotation format conversion was necessary. LabelMe2YOLO tool [24] converts the annotation file from the LabelMe JSON format to the YOLO txt format.

In the model training, the training set from our dataset is utilized to train YOLOv5n-seg and YOLOv8n-seg models (hereafter referred to as YOLOv5 and YOLOv8 for simplicity). The nano variant, which is the smallest variant of the YOLOv5 and YOLOv8 for segmentation, is selected in our research because these models are lightweight. This allows for deployment on devices with limited computation power, thus increasing their potential for use in computer vision applications. The models were then evaluated using the test set, and have the results analyzed.

B. Dataset

The dataset used in our research focuses on road marking signs in Taiwan. The images are street-level views of driving conditions on Taiwan's road. In the dataset, 14 classes are labeled, including seven types of road marking lines, five types of arrows, a pedestrian crossing, and a motorcycle sign. Samples of these classes are displayed in Fig. 3. These classes represent the common road marking signs encountered in the streets of Taiwan.

Fig. 3. Sample images of the road signs used in the research.

Our dataset contains 451 images, divided into the training and testing set with an 80:20 ratios. The details of the dataset are presented in Table I.

TABLE I. DATASET DETAILS

Class	Description	Number of instances
A01	Turn right	62
A02	Turn left	61
A03	Go straight	123
A04	Turn right or go straight	88
A05	Turn left or go straight	51
L01	Stop line	108
L02	Lane line	163
L03	Solid white line	252
L04	Barrier line	124
L07	Overtaking prohibited	158
L08	Red line	74
L09	Cross hatch	71
C01	Pedestrian crossing	171
O04	Motorcycle sign	60

Table I shows that the dataset has some data imbalances. The imbalance is primarily due to the road marking lines being commonly found on roads, while other types of signs are rarer. For instance, finding a road without a line lane or solid white line is almost impossible. In contrast, arrow road marking signs are typically found near the intersection, and motorcycle signs are commonly located near the traffic lights.

C. Evaluation Metrics

Precision, recall, and mAP@50 are employed to evaluate the model. These three key metrics are commonly used to evaluate object detection and segmentation models.

Precision is the proportion of true positive detections among all detections (true positives and false positives). The formula for precision is shown in Eq. (1).

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Recall is the proportion of true positive detections out of all actual positives (true positives and false negatives). The formula for precision is shown in Eq. (2).

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

True Positives (TP) are the cases where the model correctly predicts the positive class. False Positive (FP) occurs when the model incorrectly predicts the positive class (i.e., the model predicted an instance as positive, which is negative). False Negatives (FN) happen when the model fails to predict the positive class (i.e., the model predicted an instance as negative which is positive)

Mean average precision (mAP) averages the precision-recall curve across all classes and IoU (intersection over union) thresholds. IoU compares the ground truth and the model output, which are used for evaluation in the detection and segmentation model. The higher the IoU value, the better the detection result. Fig. 4 depicts the overlap area and the union area used for calculating IoU, in which the formula is shown in Eq. (3).

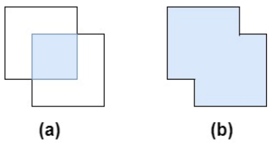


Fig. 4. Intersection over union (IoU) illustration. (a) Area of overlap; (b) Area of union.

$$IoU = \frac{Area\ of\ overlap}{Area\ of\ union} \quad (3)$$

mAP@50 is the mAP score at the 50% IoU threshold. It is a comprehensive metric combining aspects of precision and recall to provide an overall effectiveness of the object detection model. The formula for mAP is shown by Eq. (4).

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (4)$$

where n is the number of classes and AP_k is the AP of class k.

IV. Result and Discussion

A. Experimental Settings

The experiments were conducted on a PC with a 12th Gen Intel® Core i7-12700F CPU, NVIDIA GeForce RTX3060 GPU, and 32 GB memory. The models were trained for 500 epochs in each experiment, with the image size set to 640 pixels, and batch size set to 16. The augmentation parameter “fliplr” (flip left-right) was set to 0 to ensure that the model would not incorrectly recognize the left or right road signs. The experiment was run five times for each model, and the best result was taken.

B. Results

Table II presents the training results of both YOLOv5 and YOLOv8. The overall result indicates that YOLOv5 outperforms YOLOv8, shown by the higher precision, recall, and mAP50 scores. YOLOv5 achieved a precision score of 0.948, compared to YOLOv8’s 0.893. It also recorded a recall score of 0.927, while YOLOv8 scored 0.905. However, the mAP50 score of YOLOv5 is only slightly higher at 0.955, compared to YOLOv8’s 0.947.

TABLE II. Training Results

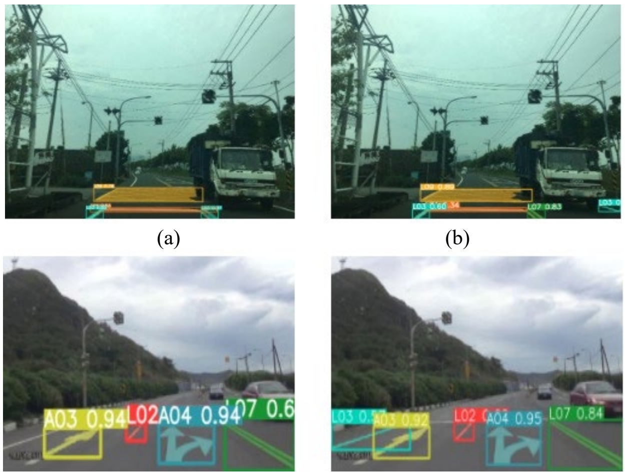
Class	YOLOv5			YOLOv8		
	P	**R**	**mAP50**	**P**	**R**	**mAP50**
A01	0.962	0.942	0.954	0.727	0.871	0.89
A02	0.906	0.902	0.942	0.793	0.752	0.912
A03	0.983	0.967	0.966	0.979	0.945	0.975
A04	0.984	0.977	0.986	0.923	0.891	0.977
A05	0.972	1	0.995	0.81	0.951	0.956
C01	0.901	0.872	0.918	0.899	0.854	0.915
L01	0.893	0.702	0.814	0.875	0.735	0.817
L02	0.931	0.852	0.937	0.891	0.891	0.93
L03	0.973	0.957	0.984	0.953	0.974	0.991
L04	0.992	1	0.995	0.901	0.969	0.99
L07	0.98	1	0.995	0.966	0.993	0.995
L08	0.964	0.963	0.982	0.912	1	0.986
L09	0.994	1	0.995	1	0.999	0.995
O04	0.832	0.851	0.899	0.869	0.851	0.922
all	0.948	0.927	0.955	0.893	0.905	0.947

The testing results for both YOLOv5 and YOLOv8 are displayed in Table III. Contrary to the training results, the testing results indicate that YOLOv8’s overall performance is superior to YOLOv5. Comparing the performance of YOLOv5 and YOLOv8 on the testing set, YOLOv8 gives higher precision and mAP50, while YOLOv5 got a higher recall score. YOLOv8 scored higher on 11 categories out of 14, with an average precision of 0.803 compared to YOLOv5’s 0.728. The most extreme difference can be found in class L03, where YOLOv5 only gets 0.412 precision compared to 0.807 on YOLOv8. However, YOLOv8 performs poorly on class L01 by only getting 0.441 precision. YOLOv5 outperforms YOLOv8 on the recall score, getting higher scores on 11 out of 14 categories, averaging 0.73 compared to YOLOv8’s 0.667. However, both YOLOv5 and YOLOv8 perform badly on classes L01, L08, and O04. Class L01 is the worst, with both models missing almost all of the detection in this class. On the mAP50, each model performs better in 7 categories. YOLOv8 outperforms YOLOv5 by scoring 0.764, while YOLOv5 scores 0.756. Both models score low mAP50 on class L01 and L08, with less than half of the detections being segmented with IoU higher than 50%.

TABLE III. Testing Results

Class	YOLOv5			YOLOv8		
	P	**R**	**mAP50**	**P**	**R**	**mAP50**
A01	0.945	0.9	0.986	0.808	0.844	0.907
A02	0.747	0.885	0.755	0.75	0.8	0.818
A03	0.745	0.844	0.894	0.853	0.781	0.891
A04	0.799	0.958	0.915	0.822	0.792	0.9
A05	0.678	0.9	0.944	0.785	0.9	0.887
C01	0.519	0.8	0.704	0.676	0.8	0.733
L01	0.544	0.08	0.219	0.441	0.0353	0.247
L02	0.75	0.943	0.877	0.701	0.657	0.781
L03	0.412	0.668	0.63	0.807	0.597	0.703
L04	0.827	0.846	0.91	0.859	0.808	0.889
L07	0.759	0.792	0.82	0.837	0.75	0.922
L08	0.856	0.211	0.338	1	0.206	0.421
L09	0.939	0.917	0.984	1	0.981	0.995
O04	0.675	0.481	0.604	0.898	0.385	0.601
all	0.728	0.73	0.756	0.803	0.667	0.764

Fig. 5 displays sample outputs from YOLOv5 and YOLOv8 when evaluated on the test set. Fig. 5(a) and (c) show outputs from YOLOv5, while Fig. 5(b) and (d) are outputs from YOLOv8. These images show that both models can produce similar quality of road marking sign segmentation. However, a notable difference is that the output from YOLOv8 can detect and segment an additional instance of the L03 class, as seen in the right corner of Fig. 5(b) and the left corner of Fig. 5(d). These results align with the data shown in Table III, where YOLOv8 excels in the L03 class.



(a) (b)

(c) (d)

Fig. 5. Sample evaluation output. (a), (c) YOLOv5; (b), (d) YOLOv8.

C. Discussion

Based on the testing results, it is evident that YOLOv8 yields better overall results compared to YOLOv5. However, YOLOv8 has a lower recall score than YOLOv5, indicating that YOLOv8 missed more detection than YOLOv5. Fig. 6 illustrates an example of segmentation done by both models, where YOLOv8 fails to detect and segment the class A05. Another pattern observed in the evaluation results is that YOLOv8 tends to miss road marking signs, while YOLOv5 incorrectly classifies them. The issue is evident in Fig. 7 (a), where YOLOv5 segments the red line, but labels it as class A03 (solid white line). In contrast, YOLOv8 completely misses the road marking sign, as seen in Fig. 7(b). These examples explain why YOLOv5 has a lower precision score, while YOLOv8 has a lower recall score.

(a) (b)

Fig. 6. YOLOv8 missed to segment the class A05. (a) YOLOV5. (b) YOLOv8.

(a) (b)

Fig. 7. Wrong class by YOLOv5 and miss by YOLOv8. (a) YOLOv5. (b) YOLOv8.

Another important point for discussion is the fact that both models perform poorly in certain classes, such as L01 (stop line), L08 (red line), and O04 (motorcycle sign). Class L01 has a very low recall score, indicating that both models frequently fail to detect it. It also has a low precision score, 0.544 for YOLOv5 and 0.441 for YOLOv8, suggesting that the models often misidentify other objects as class L01. The low precision and recall scores lead to low mAP50 scores. Class L08 is another class with low recall and mAP50 score. The difference with L01 is that the precision score is still high for both models, which means that while the models rarely misidentify an object as class L08, they miss most of the detection. The same is true for class O04. However, the precision score is lower than L08, while getting a higher mAP50 score.

Several problems are identified from the three classes that show poor detection performance. The first issue is the appearance of the road marking signs, particularly for classes L01 and L08. Both classes are similar to some other classes. For example, class L01 is identical to L02 and L03 but has fewer instances in the dataset. The data imbalance can cause difficulty for the model in identifying class L01 correctly. A similar issue occurs with class L08, which resembles class L03 but differs in color. Class O04 faces a different issue: the number of instances. Unlike other classes, O04 is not similar to other classes in the dataset. However, it only has 60 instances, which could lead to suboptimal learning by the model, as shown by the better recall and mAP50 score compared to L01 and L08. The model's performance might improve significantly with more instances to train class O04.

Section II B mentions that the current dataset has some data imbalance issues. While it cannot be avoided, knowing that some classes are more common than others, some classes still have a very limited number of instances. Increasing the number of instances to reduce the gap between classes may improve overall performance.

The experiment found that YOLOv8 is faster to train. Fig. 8 displays the training progress of YOLOv5 and YOLOv8, which shows that YOLOv8 trains and converges faster. YOLOv8 terminated training after 158 epochs, as there were no further improvements in the result for the last 50 epochs. Conversely, YOLOv5 converged slower and underwent more epochs of training. YOLOv5 only stopped training after reaching 358 epochs.

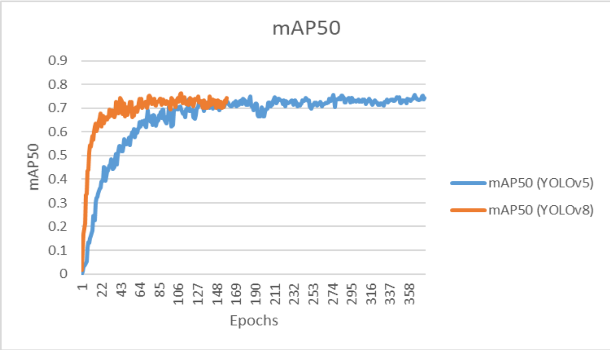


Fig. 8. YOLOv5 and YOLOv8 training progress in terms of mAP50.

Another exciting aspect found during the research is the fact that YOLOv5 gives better recall, while YOLOv8 outperforms in terms of precision. In the future, we will use the ensemble method to combine the results of both models, which might lead to a better overall performance in road marking sign segmentation.

V. Conclusion

In our research, we compared the performance of YOLOv5n-seg and YOLOv8n-seg in segmenting road marking signs in Taiwan. Additionally, we prepared and proposed our dataset for segmenting of Taiwan road marking signs.

From the research, we concluded that YOLOv8n-seg performs better than YOLOv5n-seg. YOLOv8n-seg achieved a 0.764 score in mAP50, which is higher than YOLOv5n-seg. Moreover, it took less time to train YOLOv8n-seg, as it can converge faster than YOLOv5n-seg. We also found that YOLOv5 gives a better recall score, while YOLOv8 provides higher precision.

Some issues were identified during the research that led to lower performance from the models. These include insufficient data for some classes and the problem caused by the similar appearance between some classes. In the future, we plan to continue improving the dataset by adding more images and classes. We will also strive to reduce the data imbalance between classes while further investigations are needed to address the issue of similar appearances between classes. To improve the segmentation result, we will investigate the effect of using ensemble methods to combine the results from both YOLO versions, combine the strength of each model, and achieve a better segmentation performance.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

W.K.C did the data collection, ran the experiment, and prepared the draft of the manuscript; W.E.M supervised the experiments, analyzed the data, and wrote the final version of the paper; R.C.C supervised and gave advice on the whole project, reviewed and edited the final version of the paper; C.S.P reviewed and edited the final version of the paper; all authors had approved the final version.

Funding

This paper is supported by the National Science and Technology Council, Taiwan. The Nos are NSTC-112-2221-E-324-003-MY3 and NSTC-112-2221-E-324-011-MY2.

References

[1] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: State of the art," *Int. J. Multimed. Inf. Retr.*, vol. 9, no. 3, pp. 171–189, 2020. doi: 10.1007/s13735-020-00195-x

[2] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11. pp. 3212–3232, Nov. 2019. doi: 10.1109/TNNLS.2018.2876865

[3] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1089–1106, 2019. doi: 10.1007/s10462-018-9641-3

[4] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for semantic urban scene understanding," in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2016, pp. 3213–3223. doi: 10.1109/CVPR.2016.350

[6] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. the IEEE International Conference on Computer Vision*, IEEE, Oct. 2017, pp. 5000–5009. doi: 10.1109/ICCV.2017.534

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020. doi: 10.1109/TPAMI.2018.2844175

[8] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++ better real-time instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1108–1121, Feb. 2022. doi: 10.1109/TPAMI.2020.3014297

[9] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023. doi: 10.3390/make5040083

[10] Y. Zhang, Z. Lu, X. Zhang, J. H. Xue, and Q. Liao, "Deep learning in lane marking detection: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 5976–5992, 2022. doi: 10.1109/TITS.2021.3070111

[11] A. Al Mamun, P. P. Em, M. J. Hossen, B. Jahan, and A. Tahabilder, "A deep learning approach for lane marking detection applying encode-decode instant segmentation network," *Heliyon*, vol. 9, no. 3, e14212, Mar. 2023. doi: 10.1016/j.heliyon.2023.e14212

[12] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," arXiv preprint, arXiv:1606.02147, Jun. 2016,

[13] Y. Liu, F. Yi, Y. Ma, and Y. Wang, "ASA-BiSeNet: Improved real-time approach for road lane semantic segmentation of low-light autonomous driving road scenes," *Appl. Opt.*, vol. 62, no. 19, 5224, Jul. 2023. doi: 10.1364/ao.486302

[14] Y. C. Hsieh, E. H. C. Lu, and J. M. Ciou, "Apply mask R-CNN to the detection of road speed limit markings," *J. Chinese Inst. Civ. Hydraul. Eng.*, vol. 34, no. 3, pp. 221–228, May 2022, doi: 10.6652/JoCICHE.202205_34(3).0005

[15] S. D. Nguyen, V. P. Tran, T. S. Tran, H. J. Lee, and J. M. Flores, "Automated segmentation and deterioration determination of road markings," *J. Transp. Eng. Part B Pavements*, vol. 149, no. 3, Sep. 2023. doi: 10.1061/jpeodx.pveng-1181

[16] J. Wu, W. Liu, and Y. Maruyama, "Automated road-marking segmentation via a multiscale attention-based dilated convolutional neural network using the road marking dataset," *Remote Sens.*, vol. 14, no. 18, 4508, Sep. 2022. doi: 10.3390/rs14184508

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91

[18] G. Jocher. (2020). YOLOv5 by Ultralytics (Version 7.0). [Online]. Available: https://github.com/ultralytics/yolov5/blob/master/CITATION.cff

[19] R. C. Chen, C. Dewi, Y. C. Zhuang, and J. K. Chen, "Contrast limited adaptive histogram equalization for recognizing road marking at night based on yolo models," *IEEE Access*, vol. 11, pp. 92926–92942, 2023. doi: 10.1109/ACCESS.2023.3309410

[20] C. Dewi, R. C. Chen, Y. C. Zhuang, and H. J. Christanto, "Yolov5 series algorithm for road marking sign identification," *Big Data Cogn. Comput.*, vol. 6, no. 4, 149, Dec. 2022. doi: 10.3390/bdcc6040149

[21] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1571–1580. doi: 10.1109/CVPRW50498.2020.00203

[22] Ultralytics YOLOv8 Documentation. [Online]. Available: https://github.com/open-mmlab/mmyolo/tree/main/contables/yolov8

[23] W. Kentaro. (2018). Labelme: Image Polygon Annotation with Python. [Online]. Available: https://github.com/thinh-huynh-re/labelme_v/blob/master/CITATION.cff

[24] Rooneysh. Labelme2YOLO. [Online]. Available: https://github.com/rooneysh/Labelme2YOLO