Journal of Advanced Marine Engineering and Technology, Vol. 47, No. 6, pp. 393~399, 2023
J. Advanced Marine Engineering and Technology (JAMET)
https://doi.org/10.5916/jamet.2023.47.6.393

ISSN 2234-7925 (Print)
ISSN 2765-4796 (Online)
Original Paper

Object location estimation system based on instance segmentation

Ju-Won Bae[1] · Dong-Hoan Seo[2] · Ju-Hyeon Seong[†]

(Received November 14, 2023 ; Revised December 7, 2023 ; Accepted December 18, 2023)

Abstract: Owing to the increasing demand for parcel delivery, automated systems for the reception or organization of parcels in warehouses have been widely researched. Application of existing automation systems has been challenging owing to the high construction costs incurred and limited lighting conditions required. To address these issues, this paper proposes a system that can determine the location and area of a measured object by using instance segmentation. The proposed system uses YOLACT, a lightweight image-segmentation algorithm optimized in real time, to determine the pixel-level area of the target in the image. The proposed system can estimate the location and area of an object in an image, calculate the pixel area, and accurately identify non-square objects. Furthermore, the proposed system can accurately estimate the position of an object even when the lighting changes by directly constructing and learning datasets collected in various lighting environments.

Keywords: Instance Segmentation, Computer vision, Deep Learning

1. Introduction

The increased consumption of delivery and courier services—particularly during COVID-19 when the time that people spent indoors was remarkably long—has had a significant impact on the logistics industry. With increased use of courier services owing to mobile shopping, which facilitates non-face-to-face transactions, many items are shipped to and from the warehouses of post offices and distribution companies. As the use of courier delivery has increased, many people have been able to make satisfactory transactions; however, accidents, such as items being damaged during delivery or being delivered to the wrong address, are also increasing. Consequently, the demand for manpower to handle courier work is also increasing. To address this issue, it is essential to perform verifications that automatically check the volume and destination of the goods in the logistics warehouse after receiving the delivery, during the process of classifying goods according to their size and destination.

To automate verification in situations wherein many objects move quickly, it is necessary to obtain accurate measurement performance and ensure low construction costs with minimum processing time to acquire object information. Existing systems have primarily measured the volume of objects by using multiple high-performance cameras and multiple LiDARs and have measured object information by using image processing algorithms such as edge detection **[1]-[3]**. However, these methods incur high construction costs because they require the installation and use of several equipment or expensive sensors **[4]-[6]**. Additionally, conventional image processing algorithms are vulnerable to changes in lighting and therefore require limited lighting conditions. Further, when multiple objects are photographed simultaneously, more complex information processing is required to separate them and obtain information for each object. This requires long processing time. Therefore, research efforts must focus on systems that can achieve low construction costs and high accuracy even in non-fixed environments.

Deep learning-based computer vision systems play a major role in areas that require the measurement of objects, such as inspecting product defects in smart factories **[7]-[9]**, providing information in medical images **[10]-[12]**, and estimating sample volumes in the civil engineering domain **[13]**. Recently, light-

† Corresponding Author (ORCID: http://orcid.org/0000-0002-8198-0439): Professor, Division of Maritime AI and Cyber Security & Interdisciplinary Major of Maritime AI Convergence, Korea Maritime & Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: jhseong@kmou.ac.kr, Tel: 051-410-5031

1 Ph. D. Candidate, Department of Electrical & Electronical Engineering & Interdisciplinary Major of Maritime AI Convergence, Korea Maritime & Ocean University, E-mail: wjb0406@g.kmou.ac.kr, Tel: 051-410-4822

2 Professor, Division of Electronics & Electrical Information Engineering & Interdisciplinary Major of Maritime AI Convergence, Korea Maritime & Ocean University, E-mail: dhseo@kmou.ac.kr, Tel: 051-410-4412

weight deep-learning algorithms **[14]-[16]** have been investigated in this regard, and methods for obtaining information quickly and accurately have been studied. To accurately obtain product information, a deep learning-based computer vision system can achieve fast processing time and accurate measurement performance. Additionally, because information concerning an object can be obtained using only one camera rather than multiple cameras, construction costs can be reduced.

In this study, we devised a position-estimation system that can determine the location and area of a measured object by using deep learning-based computer vision. The proposed system uses only a single two-dimensional (2D) camera and applies an image-segmentation algorithm to determine the pixel-level area of the target in the image. Further, it uses YOLACT [17] as the image-segmentation algorithm. This algorithm is an instance-segmentation model that recognizes each instance. Because the location and pixel area of an object can be accurately determined in an image, even objects that are not square-shaped can be precisely identified. In addition, they exhibit remarkable resilience to noise such as a person's hands and feet, which are simultaneously photographed while measuring an object.

The remainder of this paper is organized as follows. Section 2 details related research and image segmentation, and Section 3 describes the flow and configuration of the proposed system. Section 4 presents the verification conducted in various environments to evaluate the performance of the proposed system. Finally, Section 5 summarizes the conclusions and future research.

2. Related Works

2.1 Image Segmentation

Image segmentation is a technology that identifies the desired object in an image pixel by pixel and assigns class information to all pixels in the image. Unlike classification and detection (for finding the location), the entire area occupied by the target in the image is displayed. Thus, this method has applicability in various fields such as medical image processing **[18]-[20]** and autonomous driving **[21]-[23]**. Image segmentation is classified into semantic segmentation and instance segmentation. Semantic segmentation distinguishes only the classes of all objects found in the image, whereas instance segmentation can distinguish instances among object classes. Instance segmentation is a more difficult method than semantic segmentation because instances must be distinguished; however, it is advantageous in that it can access information concerning each object because it can separate and identify multiple objects in a given scene.

Image segmentation has been investigated using various approaches since the development of deep learning. In particular, high processing speed and accurate performance have been demonstrated by combining upper- and lower-level features with a feature pyramid network (FPN)-based structure **[24]**. Another study **[25]** proposed a method of dividing an image into patches of the same size and using them as input to a Transformer for segmentation. Further, a method to locally connect image feature information to the best extent possible has been proposed **[26]** by processing features and combining patches initially divided into 16×16 pixels into 8×8 and 4×4 pixels from the next input. However, because Transformer-based methods entail numerous calculations, they require a long processing time depending on the performance of the processing device.

3. Proposed System

3.1 Overall System

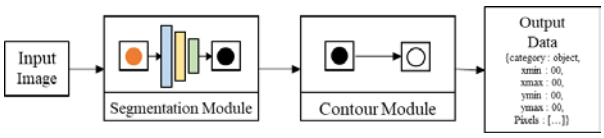

Figure 1: Structure of proposed system

This study devised a position-estimation system that determines the location and area of an object to be measured in real time using a single camera. A schematic of the operation of the proposed system is shown in **Figure 1**.

First, the system determines the pixel area of the object by using a segmentation module for the input image. YOLACT, which outputs a matrix that distinguishes objects and noise of the same size as the image, is used as the segmentation model. The contour module extracts only the object information from the output matrix and detects edges by imaging the matrix with the highest accuracy. The location of the object is then obtained by outputting the coordinate information of the category, pixel areas (xmin, ymin), and (xmax, ymax) from the detected edge. The following subsections describe the detailed methods of the proposed system and construction of the dataset used in the proposed system.

3.2 Segmentation Module

The segmentation module determines the pixel-level area of an object using an instance-segmentation model. Existing

instance-segmentation models require significant computation and high hardware performance owing to their complex structures. Therefore, we used YOLACT, which is a lightweight model. YOLACT is a real-time instance-segmentation method that enables fast processing. The structure of YOLACT is shown in **Figure 2**.

YOLACT, used in the segmentation module, extracts image features using Darknet-53 **[27]** as the backbone and obtains features at each stage with an FPN structure. In contrast to other instance-segmentation methods, YOLACT simultaneously performs classification to recognize feature localization objects, which is a preprocessing step for extracting pixel-level areas. It determines the approximate part of the pixel area in ProtoNet and calculates the mask coefficient using the prediction head. Subsequently, the output of the ProtoNet and the mask coefficients are combined to obtain the segmentation result of the final pixel area. The formula for obtaining these results is expressed as follows:

$$M = \sigma(PC^T) \quad (1)$$

where P is the prototype mask and is of the form (h, w, k). Here, h and w denote the horizontal and vertical dimensions of the image, respectively; k is the output of the mask coefficient; and C is (n × k) for n output instances. The size is multiplied by a single matrix and a sigmoid function. The loss function is calculated using a single-shot detector (SSD) and binary cross-entropy. By utilizing these methods, YOLACT can achieve fast processing speed and accurate segmentation results.

3.3 Contour Module

The contour module specifies the coordinate and area information for the detected pixel area. The flowchart of the contour module algorithm is shown in **Figure 3**.

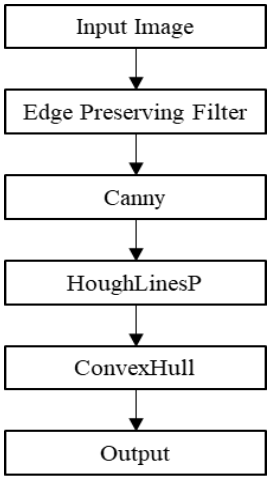

Figure 3: Flow of contour module

The contour module receives the pixel area image, which is the result of the segmentation module as input, emphasizes edges using OpenCV's EdgepreservingFilter **[28]** for edge detection, and determines the edges using Canny **[29]**. Points for the border area are then created using the Hough transform and objects are created using the ConvexHull method. At each point of the object, the x-axis minimum xmin, y-axis minimum ymin, x-axis maximum xmax, and y-axis maximum ymax are determined and output as information regarding the position, including the pixel area of the input image. Because it is not accurate to specify the coordinates of atypical objects such as vinyl or circles, which do not have a shape that can easily specify coordinates, such as a rectangular parallelepiped, the information of xmin, ymin, xmax, ymax, and the pixel area coordinates can be obtained as the output. This information is used following conversion to the center coordinates where it is located or in the form of a bounding box.

3.4 Dataset Structure

To build the proposed object location tracking system, a dataset was constructed as shown in **Figure 4**. The left side of Figure 4 shows the collected raw data, and the right side shows the labeled image. Because of the influence of light sources and the

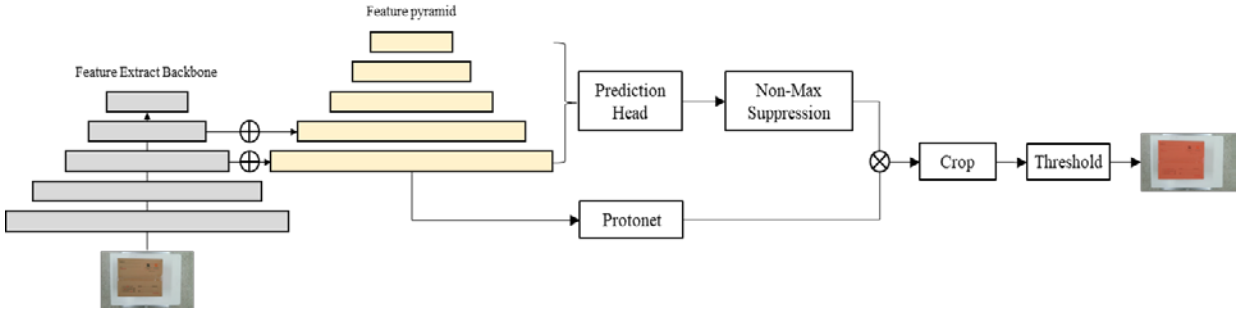

Figure 2: Flow of YOLACT

surrounding environment, we collected data from eight locations—including outdoor and indoor environments—and objects, including boxes of various colors, and parcels wrapped in plastic and collected 8,000 pieces of data related to the object at 4K resolution. During labeling, the target object to be measured and other noises, such as hands or feet, were labeled separately. This method allowed for the object location tracking system to distinguish between object and noise areas. Therefore, even when an object was obscured by noise, its area could be estimated in pixel units.

Figure 4: Dataset for segmentation

4. Results

4.1 Experimental Settings

The proposed system was implemented in Python 3.7, PyTorch 1.12.1, and Cuda 11.3 environments under the GPU specifications of NVIDIA RTX A6000. In Section 4.2.1, we analyze the results in an implementation environment. In Section 4.2.2, we analyze the actual environment. In the actual environment, the experiment was performed using edge-computing equipment, that is, NVIDIA Jetson Xavier NX. To facilitate the learning of YOLACT in the segmentation module, 6,000 of the 8,000 sheets were used. Of the remaining sheets, 1,000 sheets were used for validation and 1,000 sheets were used for verification. The images were randomly selected from the dataset described in Section 3.4. The test dataset includes 579 images from an indoor environment and 421 images from an outdoor environment, which were divided evenly into eight environments and eight objects. There were 677 objects classified as noise in the image, and 1,040 objects were classified as objects.

4.2 Result and Discussion

4.2.1 Segmentation Module Result

First, we compared the output results of the segmentation to evaluate whether the proposed system accurately performed pixel area detection. We used the mean average precision (mAP) to evaluate the performance of the object and noise categories in the dataset. The mAP represents the average proportion of correctly answered questions among the pixel areas. For the test, 1,000 test images that were not used for learning were used among the 8,000 images collected, as described in Section 4.1. The proposed system used YOLACT as the model for the segmentation module. Therefore, the proposed system was compared with Mask R-CNN **[31]** to verify the validity of YOLACT. Mask R-CNN is a representative instance-segmentation model. It was proposed at approximately the same time as YOLACT and is known to exhibit higher performance in public datasets. Therefore, when comparing YOLACT and Mask R-CNN, we aimed to verify the feasibility of using YOLACT as a segmentation module by comparing the mAP and inference time for object and noise detection. **Table 1** presents the mAP and inference time comparisons for each category of each model.

Table 1: Performance comparison of YOLACT and Mask R-CNN in test dataset

Category		Mask mAP (%)	Box mAP (%)	Inference Time (s)
YOLACT	Object	83.43	82.36	0.76
	Noise	79.28	78.68	
Mask R-CNN	Object	84.9	80.9	2.89
	Noise	80.32	80.2	

In **Table 1**, the mask mAP refers to the mAP of the pixel area and the box mAP refers to the mAP of the bounding box results for xmin, ymin, xmax, and ymax of the pixel area. The YOLACT model yielded an mAP of 29.8 in the MS COCO dataset **[30]**, and when learning the constructed dataset, the mAP of the object was 83.43% and 82.36% for mask and box, respectively. This indicates that it accurately estimated the object area. In addition, the noise area also exhibited a high estimation performance of 79.28% and 78.68% in the mask and box mAPs, respectively.

The mask and box mAPs of the Mask R-CNN were slightly higher than those of YOLACT. However, the box mAP for noise was lower than that of YOLACT, and the resulting output indicated a pixel area that was slightly larger than the predicted pixel area of YOLACT. Therefore, YOLACT's precision performance in the test was poor; however, the value of Mask R-CNN appeared to be slightly low for the object's Box mAP. YOLACT was three times faster in terms of inference time compared to Mask R-CNN. Thus, it was found to be more appropriate to use

YOLACT, which has a similar performance but realizes faster processing for the dataset of the proposed system.

The actual output results in **Figure 5** indicate that the object and noise areas were accurately distinguished.

In addition, vinyl-shaped objects, which are atypical targets, and the hands and feet, which are essentially noise, were accurately classified and displayed. Further, by appropriately dividing the noise and object areas, it is possible to create a bounding box that can accurately estimate the location of the object, even in the hidden part.

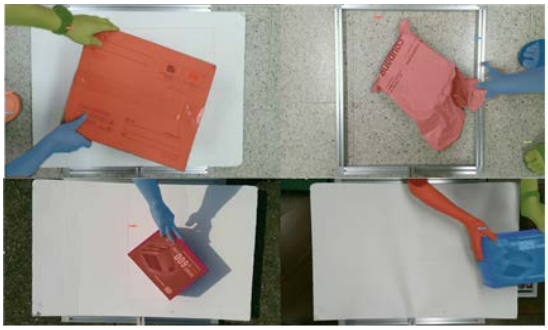

Figure 5: Result of segmentation model in test dataset

4.2.2 Results Analysis in Edge-Computing Environment

The proposed system sought to achieve low construction costs, fast processing speed, and high accuracy. Therefore, we compared the performance of the proposed system in terms of processing time and accuracy in an edge-computing environment rather than in a high-cost GPU environment. For the hardware environment of the edge-computing system, images were captured using a 4K camera sensor and resized to 500×500 pixels, which is appropriate for real-time image processing. Further, location estimation was performed. Three targets were tested in this study. The results are summarized in **Table 2**.

In **Table 2**, Case 1 refers to the same box as the post office box used for learning, while Cases 2 and 3 are boxes that were not used for learning. Additionally, tests were conducted under different lighting conditions in the indoor environment wherein the data were collected.

Table 2: Performance for Edge-Computing System

Case	Inference Time (s)	Mask mAP (%)	Box mAP (%)
Case 1	0.826	98.28	98.01
Case 2	0.831	96.5	93.31
Case 3	0.841	97.37	92.83

The mask and box mAPs showed high values (over 95%). Additionally, the processing time outputs quickly within 1 s. However, given that there were errors at the boundary, research on improving the segmentation performance in fine areas is necessary to achieve a more accurate performance.

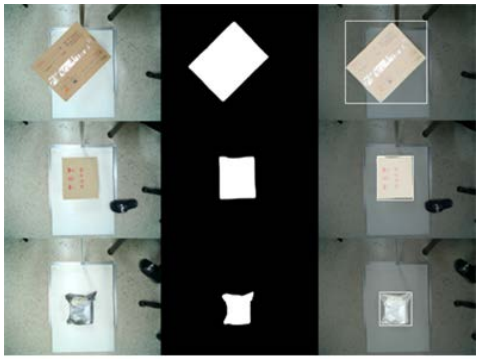

Figure 6: Result of segmentation model in test dataset

Figure 6 shows the actual system results for Cases 1, 2, and 3 (from the top). The left image is the input image, the middle image is the resulting image of the segmentation module, and the right image shows the pixel output area and the bounding box obtained by the contour module combined with the original image. The output accurately represented the bounding box area of the target, thus distinguishing the location of the object.

5. Conclusion

This study developed an object location estimation system that can determine the location and area of a measured object by using instance segmentation. The proposed system quickly determines the location of objects and can help automate a logistics warehouse or delivery reception. Further, it can be built at a low cost because it uses only a single 2D camera. In addition, by accurately distinguishing between objects and noise, the hidden parts and atypical objects can be identified. Tests in real environments confirmed that the location of the object was accurately determined, and even in an edge-computing environment, the processing speed was less than 1 s. However, in situations wherein objects move quickly, real-time processing may require at least 10 images per second. Furthermore, the accuracy apparently needs to be improved in fine areas such as the boundary between the object and background. To address this issue, in future studies, we plan to investigate segmentation models that can accurately infer fine areas.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2023-00254240, Development of non-face-to-face delivery of postal logistics and error verification system for parcel receipt.). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1A2C1014024).

Author Contributions

Conceptualization, J. W. Bae and D. H. Seo; Methodology, J. W. Bae and D. H. Seo; Software, J. W. Bae; Data Curation, J. W. Bae; Writing—Original Draft Preparation, J. W. Bae; Writing—Review & Editing, D. H. Seo and J. H. Seong; Supervision, J. H. Seong;

References

[1] R. J. Muthukrishnan and M. Radha, “Edge detection techniques for image segmentation,” International Journal of Computer Science & Information Technology, vol. 3, no. 6, p. 259, 2011.

[2] M. Basu, “Gaussian-based edge-detection methods-a survey,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 32, no. 3, pp. 252-260, 2002.

[3] O. Elharrouss, Y. Hmamouche, A. K. Idrissi, B. El Khamlichi, and A. El Fallah-Seghrouchni, “Refined edge detection with cascaded and high-resolution convolutional network,” Pattern Recognition, vol. 138, p. 109361, 2023.

[4] J. H. Kwon and M. C. Won, “Development of a TOF LADAR sensor and a study on 3D information acquisition using single axis driving device,” Journal of the Korea Institute of Military Science and Technology, vol. 20, no. 6, pp. 733-742, 2017.

[5] S. C. Byun, M. K. Choi and J. K. Kim, “Estimation of maximum volume in landfill site using airborne LiDAR measurement,” Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, vol. 28, no. 5, pp. 547-554, 2010.

[6] R. Jotje, M. J. Oh. H. K Kim, S. J. Oh, and S. B. Kim, “Real-time image segmentation and determination of 3D coordinates for fish surface area and volume measurement based on stereo vision,” Journal of Institute of Control, Robotics and Systems, vol. 24, no. 2, pp. 141-148, 2018 (in Korean).

[7] T. C. Hsu, Y. H. Tsai, and D. M. Chang, “The vision-based data reader in IoT system for smart factory,” Applied Sciences, vol. 12, no. 13, p. 6586, 2022.

[8] L. Malburg, M. P. Rieder, R. Seiger, P. Klein, and R. Bergmann, “Object detection for smart factory processes by machine learning,” Procedia Computer Science, vol. 184, pp. 581-588, 2021.

[9] B. J. Kim, “Image enhanced machine vision system for smart factory,” International Journal of Internet, Broadcasting and Communication, vol. 13, no. 2, pp. 7-13, 2021.

[10] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, “DRINet for medical image segmentation,” IEEE transactions on medical imaging, vol. 37, no. 11, pp. 2453-2462, 2018.

[11] Q. Xu, Z. Ma, N. HE. Na, and W. Duan, “DCSAU-Net: A deeper and more compact split-attention U-Net for medical image segmentation,” Computers in Biology and Medicine, vol, 154, p. 106626, 2023.

[12] T. Shen and H. Xu, “Medical image segmentation based on transformer and HarDNet structures,” IEEE Access, vol. 11, pp. 16621-16630, 2023.

[13] M. Kamari and Y. Ham, “Vision-based volumetric measurements via deep learning-based point cloud segmentation for material management in jobsites,” Automation in Construction, vol. 121, p. 103430, 2021.

[14] R. Mothkur and B. N. Veerappa, “Classification of lung cancer using lightweight deep neural networks,” Procedia Computer Science, vol. 218, pp. 1869-1877, 2023.

[15] H. Mokayed, T. Z. Quan, L. Alkhaled, and V. Sivakumar, “Real-time human detection and counting system using deep learning computer vision techniques,” Artificial Intelligence and Applications, vol. 1, no. 4, pp. 221-229, 2022.

[16] M. Arsalan, T. M. Khan, S. S. Naqvi, M. Nawaz, and I. Razzak, “Prompt deep light-weight vessel segmentation network (PLVS-Net),” IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 20, no. 2, pp. 1363-1371, 2023.

[17] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLACT: Real-time instance segmentation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9157-9165, 2019.

[18] M. T. El-Melegy, R. M. Kamel, M. Abou El-Ghar, N. S. Alghamdi, and A. El-Baz, “Kidney segmentation from dynamic contrast-enhanced magnetic resonance imaging integrating deep convolutional neural networks and level set methods,” Bioengineering, vol. 10, no. 7, p. 755, 2023.

[19] Y. Chen, L. Feng, C. Zheng, T. Zhou, L. Liu, P. Liu, and Y. Chen, “LDANet: Automatic lung parenchyma segmentation from CT images,” Computers in Biology and Medicine, vol. 155, p. 106659, 2023.

[20] Z. Zhu, X. He, G. Qi, Y. Li, B. Cong, and Y. Liu, “Brain tumor segmentation based on the fusion of deep semantics and edge information in multimodal MRI,” Information Fusion, vol. 91, pp. 376-387, 2023.

[21] B. De Brabandere, D. Neven, and L. Van Gool, “Semantic instance segmentation for autonomous driving,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 7-9, 2017.

[22] G. Rossolini, F. Nesti, G. D’Amico, S. Nair, A. Biondi, and G. Buttazzo, “On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving,” IEEE Transactions on Neural Networks and Learning Systems, pp. 1-15, 2023.

[23] C. H. Cheng, A. Knoll, and H. C. Liao, “Safety metrics for semantic segmentation in autonomous driving,” in Proceeding of 2021 IEEE International Conference on Artificial Intelligence Testing, pp. 57-64, 2021.

[24] Y. Wu, J. Jiang, Z. Huang, and Y. Tian, “FPANet: Feature pyramid aggregation network for real-time semantic segmentation,” Applied intelligence,” vol. 52, pp. 3319-3336, 2022.

[25] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, ... and L. Zhang, “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.6877-6886, 2021.

[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, ... and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9992-10002, 2021.

[27] J. Redmon, S. Divvala, and R. Girshick, “You only look once: Unified, real-time object detection,” in Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[28] E. S. Gastal and M. M. Oliveira, “Domain transform for edge-aware image and video processing,” International Conference on Computer Graphics and Interactive Techniques, no. 69, pp. 1-12, 2011.

[29] J. Canny, “A computational approach to edge detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, 1986.

[30] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, vol. 8693, pp. 740-755, 2014.

[31] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 2980-2988, 2017.