Urban Street Scene Instance Segmentation: An Integrated Hybrid Network Merging Top-Down and Bottom-Up Strategies

Ruifa Zhou, Ji Zhao*

***Abstract*—There are two standard methods in instance segmentation: top-down and bottom-up. The top-down approach performs object detection to generate candidate proposals and then performs pixel-level segmentation for each proposal. It is accurate and flexible, capable of handling objects of different sizes and shapes. However, it is computationally complex and relies on object detection accuracy. The bottom-up approach first performs pixel-level clustering or segmentation and then combines candidate instances to obtain the final segmentation result. It can handle overlapping cases and has lower computational complexity, but it may need to localize accurately, and segment instances, and the segmentation granularity is coarser. In this paper, the Urban Street Scene Instance Segmentation (UISNet) algorithm is proposed. Firstly, the feature extraction network is the foundation of UISNet, which uses EfficientNet as the backbone network. Secondly, MPAFPN is the feature pyramid network part of UISNet, used for multi-scale feature fusion. By using EfficientNet and MPAFPN as the backbone network and bottleneck layers, the accuracy of UISNet is improved by 4% compared to ResNet and FPN. In the inference phase, this paper introduces an innovative dual-branch design that combines top-down and bottom-up strategies. One branch is the bounding box aggregation branch, which generates high-dimensional information such as the shape and orientation of bounding boxes based on the FCOS Head. The other branch is the mask decoding branch, which creates mask prediction results. These two branches are fused using the Mask FCN Header to obtain the final instance segmentation result. With this dual-branch design, the model can effectively utilize the information from both top-down and bottom-up approaches, thereby improving the accuracy and robustness of instance segmentation. Through experimental comparisons, the proposed network model in this paper achieves the best performance in terms of accuracy compared to other instance segmentation networks, with an accuracy of 36.28%. Moreover, the proposed model performs better in urban street scenes, enhancing object detection and segmentation and offering more reliable and efficient solutions for applications such as autonomous driving and intelligent transportation.**

***Index Terms*—Instance segmentation; EfficientNet; MPAFPN; FCOS**

I. Introduction

IN instance segmentation, standard methods include top-down and bottom-up approaches. The top-down approach first performs object detection[1] to generate candidate proposals and then performs pixel-level segmentation for each proposal. It is accurate and flexible, capable of handling objects of different sizes and shapes. However, it is computationally complex and relies on object detection accuracy. The bottom-up approach first performs pixel-level clustering or segmentation and then combines candidate instances to obtain the final segmentation result. It can handle overlapping instances and has low computational complexity, but it may need to localize accurately, and segment instances and the segmentation granularity is coarse. In this paper, we propose the UISNet algorithm. Firstly, the feature extraction network is the foundation of UISNet, and it adopts EfficientNet as the backbone network. EfficientNet is an efficient convolutional neural network structure with low parameter count and computational complexity while effectively extracting high-level semantic features from images. Through multiple convolution and pooling layers in the feature extraction network, EfficientNet gradually reduces the size of feature maps and increases the number of channels, thereby extracting feature representations with rich semantic information.

Secondly, MPAFPN is the feature pyramid network part of UISNet, used for multi-scale feature fusion. It consists of multiple parallel feature pyramid networks, each responsible for extracting features at different scales. These feature pyramid networks, through parallel operations, can simultaneously process feature maps at different scales and effectively capture information about targets at different scales. This multi-path design enables UISNet to have good receptive fields at different scales, improving the accuracy and robustness of the model, for instance, segmentation tasks.

During the prediction stage, top-down and bottom-up strategies are employed to improve instance segmentation performance. The boundary aggregation and mask decoding branches play crucial roles in the algorithm. The boundary aggregation branch utilizes the Fully Convolutional One-Stage Object Detection head to generate high-dimensional information related to the bounding boxes, such as shape and pose. This branch extracts and aggregates fine-grained features for each bounding box, capturing object details and contextual information. By introducing this branch, the instance segmentation model can better understand the features of the bounding boxes, thereby improving the accuracy and precision of object detection. On the other hand, the mask decoding branch is responsible for generating mask prediction results. This branch uses a mask Fully Convolutional Network (FCN) head to decode the feature map and create masks for each instance. The mask decoding branch achieves pixel-level segmentation[2] for each instance, resulting in

Manuscript received Nov 17, 2023; revised Mar 27, 2024. The research work was supported by Liaoning Natural Science Foundation (2020-MS-281), Research Project of Liaoning Provincial Department of Education (LJKZZ2022043).

Ruifa Zhou is a Postgraduate of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China (e-mail: apodx199733@163.com).

Ji Zhao* is a Professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China (corresponding author to provide phone: +086-139-9808-6167; e-mail: 319973500069@ustl.edu.cn).

more accurate instance segmentation results. Finally, the outputs of these two branches are fused to obtain the final instance segmentation result. The fusion process typically involves aligning and matching the bounding box information with the mask information to ensure consistency between the segmentation result and the position and shape of the target bounding boxes. Through this dual-branch design, the model can fully leverage the top-down and bottom-up information, thereby improving the accuracy and robustness of instance segmentation.

This combined top-down and bottom-up dual-branch approach has significantly improved instance segmentation tasks. It can effectively utilize the global context and fine-grained information of objects, handle instances of different sizes and shapes, and effectively address the segmentation of overlapping cases.

II. Related work

A. Top-down Instance Segmentation

Top-down instance segmentation is a commonly used method in instance segmentation, where it first performs object detection to generate candidate proposals and then performs pixel-level segmentation on each proposal. The top-down approach utilizes the results of object detection to guide the instance segmentation process and can handle objects of different sizes and shapes. Two famous classical algorithms for top-down instance segmentation are Mask R-CNN[3] and FCN-8s[4].

Mask R-CNN is a region-based convolutional neural network method that performs pixel-level segmentation on top of object detection. It extends the Faster R-CNN model by adding a segmentation branch to generate precise masks for each candidate object.

FCN-8s is a fully convolutional network used for semantic segmentation and instance segmentation. It replaces the fully connected layers of a convolutional neural network with convolutional layers, enabling pixel-level classification or segmentation of each pixel in the input image.

The advantage of top-down instance segmentation methods is that they can utilize object detection results to guide the segmentation process, reducing the computational burden of pixel-level segmentation on the entire image. They can handle objects of different sizes and shapes and provide higher instance segmentation accuracy through the accuracy and robustness of object detection. However, top-down methods may be sensitive to object detection accuracy and the quality of candidate proposals. If object detection produces errors or misses, it may lead to inaccurate segmentation results. Additionally, top-down methods typically require high computational resources and time, especially when dealing with large-scale data or complex scenes.

B. Bottom-up Instance Segmentation

Bottom-up instance segmentation is a pixel-level segmentation method that performs pixel-level clustering or segmentation on an image and then combines the candidate instances to obtain the final segmentation result. In contrast to top-down methods, bottom-up methods focus more on aggregating and combining local information without generating candidate proposals. Two famous classical algorithms for bottom-up instance segmentation are HED[5] (Holistically-Nested Edge Detection) and SLIC[6] (Simple Linear Iterative Clustering).

HED is a bottom-up method based on edge detection. It generates pixel-level segmentation results by merging edge maps at multiple scales. It utilizes deep convolutional neural networks to learn edge features.

SLIC is a superpixel-based clustering algorithm used for bottom-up segmentation. It divides an image into compact superpixel blocks and then merges or separates them based on the similarity between superpixels, resulting in the final segmentation result.

The advantage of bottom-up instance segmentation methods is their ability to handle overlapping instances and adapt well to unknown classes or irregular-shaped objects. However, bottom-up methods often have higher computational complexity and may need help dealing with large-scale data. Additionally, due to the lack of global contextual information, bottom-up methods may not achieve the same level of localization and segmentation accuracy as top-down methods.

C. Fully Convolutional One-Stage Object Detection

Fully Convolutional One-Stage Object Detection (FCOS)[7] is a single-stage object detection algorithm based on fully convolutional networks proposed by Facebook AI Research. The core idea of FCOS is to predict both the object category and location information for each position using fully convolutional networks, enabling detection to be completed with a single forward pass during training and inference.

Specifically, FCOS predicts the center coordinates, width and height information, and class probabilities for each position on the feature map. It divides the feature map into equally sized grids, where each grid predicts the offset from its position to the nearby object center and the object's width, height, and classification probability. These predictions are used to generate object bounding boxes and class labels.

Compared to traditional methods, FCOS has several advantages: Its fully convolutional structure can handle inputs of arbitrary sizes, avoiding cropping and scaling operations and enabling better detection of objects at multiple scales. FCOS only requires a single forward pass for inference, making it faster and more suitable for real-time applications. FCOS's training mechanism balances samples of various sizes, avoiding excessive focus on large objects and improving detection accuracy, especially in complex scenes.

Due to these advantages, FCOS has been widely applied in instance segmentation algorithms such as BlendMask[8] and CondInst[9], achieving excellent results. FCOS provides strong support for tasks like instance segmentation and has been widely adopted.

III. Methods

A. Overview of the UISNet Algorithm

UISNet is an object detection model, as shown in Fig.1. It has three main components: the backbone network, the Multi-Path Aggregation Feature Pyramid Network (MPAFPN), and the prediction head.

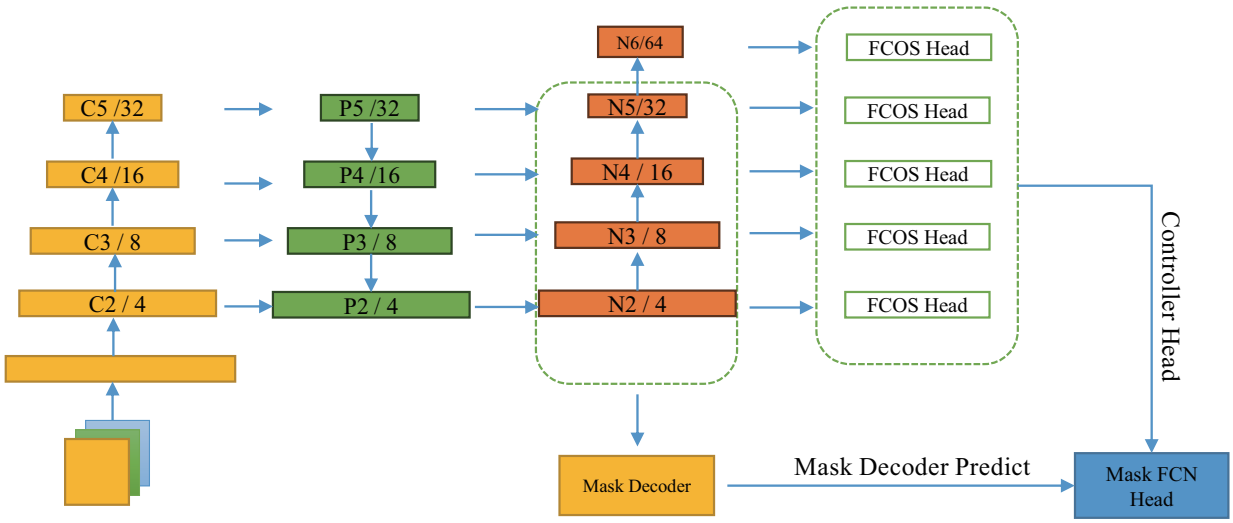

Fig. 1. UISNet Network Architecture

The Backbone network utilizes EfficientNet[10] to extract high-level features from the input image.

In the Neck section, UISNet employs MPAFPN, a multi-scale feature fusion module. It consists of multiple parallel feature pyramid networks that effectively extract features at different scales, thereby improving the detection performance of the model.

In the Head part, the model is divided into the bounding box aggregation branch and the mask decoding branch. The bounding box aggregation branch utilizes the FCOS Head to generate high-dimensional information for the bounding boxes, such as shape and pose. The mask decoding branch generates mask prediction results. Finally, the mask FCN Head combines the high-dimensional information from the bounding box aggregation branch with the mask prediction results.

B. Backbone

The backbone network of UISNet is composed of EfficientNet, an efficient and accurate convolutional neural network architecture proposed by the Google Research team in 2019. This architecture balances the relationship between network depth, width, and resolution using a compound scaling method to achieve better performance and efficiency. In traditional convolutional neural networks, the network's depth, width, and input resolution are usually manually set, which can lead to poor performance or overfitting under resource constraints. EfficientNet automatically adjusts these three critical parameters using a compound scaling method, allowing the network to perform well under different resource constraints. The compound scaling method of EfficientNet consists of three main components: width scaling, depth scaling, and resolution scaling. Width scaling controls the model's width by adjusting the number of channels, depth scaling increases the depth of the model by adding repeated network blocks, and resolution scaling adjusts the resolution of the input image. Through experimental comparisons, using EfficientNet as the backbone network achieves higher accuracy than ResNet, with similar parameters and computational complexity.

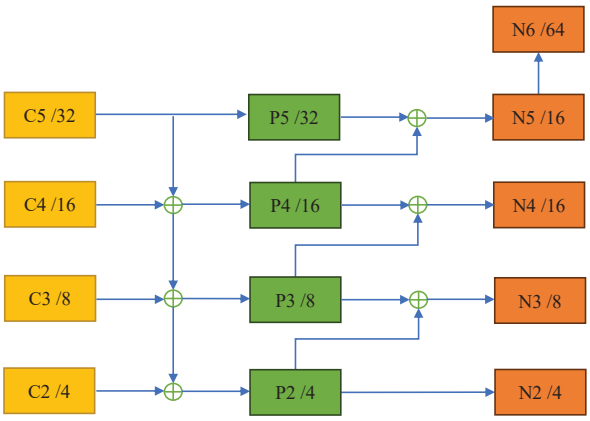

Fig. 2. MPAFPN Network

C. Multiple Path Aggregation Feature Pyramid Network

As shown in Fig.2, this paper presents a multi-path feature pyramid network(MPAFPN) as the Neck structure. MPAFPN draws inspiration from the PANet[11]] architecture and incorporates the following improvements.

The backbone network EfficientNet extracts the C2-C5 feature maps and then upsamples and downsamples them to obtain the P2-P5 and N2-N6 feature maps, respectively. Firstly, the P2-P5 feature maps are used for a top-down pathway to pass semantic information. Then, the N2-N6 feature maps are used for a bottom-up pathway to pass localization information.

Experimental results demonstrate that compared to FPN[12], MPAFPN has better feature fusion, higher computational efficiency, improved cross-scale feature representation, and enhanced semantic information representation. As a result, MPAFPN can improve the performance of object detection and segmentation.

D. Predict Head

As shown in Fig.3, the prediction branch consists of three parts: the FCOS branch, the Mask Decoder branch, and the Mask FCN Head. Next, we will provide specific introductions for these three parts.

As shown in Fig.4, The FCOS Head Branch in the top-down part plays a crucial role in instance segmentation. It is responsible for object-level detection by processing the prediction maps N2-N6, which provide information about the object's class, position, and centerness. This branch typically comprises sub-branches for classification, regression, control, and centerness. The predicted maps N2-N6 generated by MPAFPN are fed into the FCOS Head in the specific process. In this process, N2-N6 shares the same parameters of the FCOS Head. As shown in Figure 3, the FCOS Head is a fully convolutional network consisting of four branches: Classification, Regression, Controller, and Centerness. The shape of the Classification branch is H×W×C, where C represents the number of classes (excluding the background). In instance segmentation tasks, this feature map is used for predicting the class of each pixel in the image. The Regression branch has the same shape as the Classification (H×W×C) and is responsible for predicting and adjusting the positions of the target bounding boxes. The Centerness branch has a shape of H×W×1 and is used to estimate the centrality of the targets. Its purpose is to suppress the predictions of pixels far from the target's center, thus eliminating low-quality predictions.

As shown in Fig.5, In the bottom-up part, the Mask Decoder takes N2-N5 as inputs, where N2 represents the lower-level feature map, and N3-N5 represents the higher-level feature maps. The higher-level feature maps are upsampled and fused with the lower-level feature map. The fused output is simultaneously resized to the original image size. This operation allows for preserving positional information from the lower-level features while incorporating the semantic information from the higher-level features.

Finally, the Mask FCN Head takes the Controller and Mask Predict Head as inputs. The Controller is used as weights and offsets to perform convolutional operations on the Mask Predict Head, resulting in the final prediction structure. Its primary purpose is to gradually upsample the low-level feature maps of the convolutional network to feature maps of the same size as the input image and generate pixel-level semantic segmentation predictions.

IV. Experiment

A. Datasets

This paper uses two datasets: the COCO[13] dataset and the Cityscapes[14] dataset.

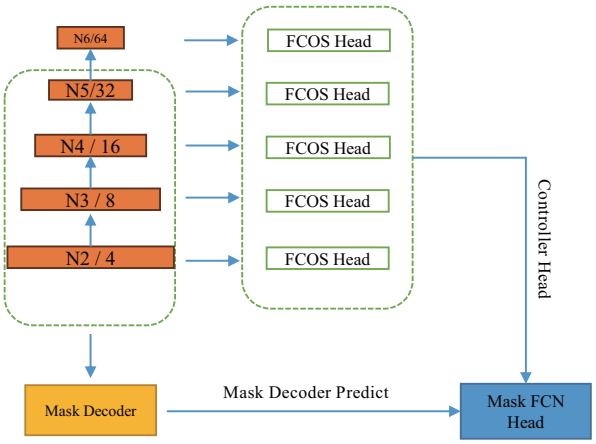

Fig. 3. Predict Head Network

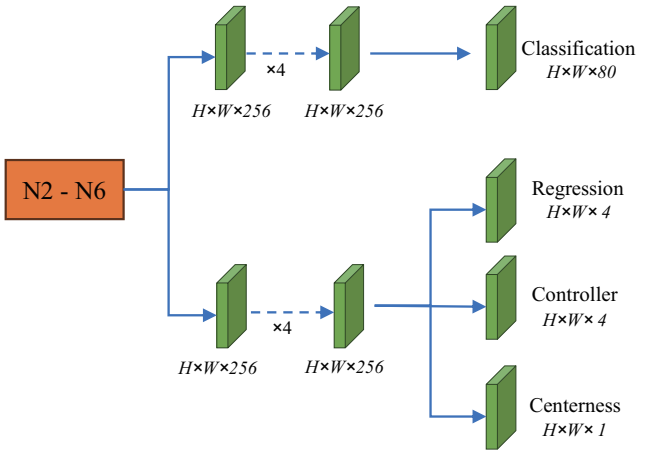

Fig. 4. FCOS Head Branch

The COCO dataset consists of 330,000 images with annotations for 200,000 instances across 80 object categories. In addition to object bounding box annotations, COCO provides rich information such as segmentation masks and human keypoint annotations. Due to these characteristics, the COCO dataset is widely used in visual tasks such as object detection, image segmentation, and human pose estimation. It has become a standard benchmark dataset for these tasks, and many state-of-the-art methods and models are developed and evaluated on COCO. We use the COCO dataset to compare our model with others and validate its effectiveness.

Cityscapes is a large-scale dataset for understanding urban street scenes. It contains 50,000 high-resolution street-level images from 50 different cities. The pictures of Cityscapes are clear and detailed, capturing rich urban elements. Cityscapes provides fine-grained semantic segmentation annotations for all 50,000 images, with each pixel labeled with a semantic category. The dataset includes eight categories and 20 subcategories; the annotations are precise and detailed. Using this dataset aims to evaluate the real-world practicality of our model in realistic street scenes.

B. Loss Function

The loss function is calculated by Equation (1):

$$Loss = L_{cate} + \lambda L_{mask} \tag{1}$$

Whereas L_{cate} is the Focal Loss[15] for semantic classification, and L_{mask} is the Dice Loss[16] for mask prediction.

Focal Loss(FL) is a loss function used to address class imbalance issues, and the formula is shown as Equation (2):

$$FL(p_t) = -\alpha(l - p_t)^{\gamma} \log(p_t) \tag{2}$$

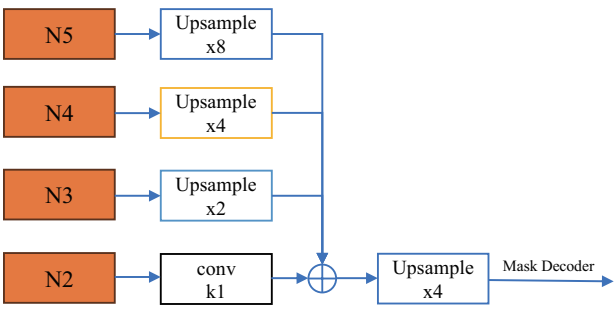

Fig. 5. Mask Decoder Branch

Where p_t represents the predicted probability of sample t belonging to the positive class, α represents the class weight for sample t, and γ is an adjustable parameter to adjust the weight of easily classified samples.

Dice Loss(DL) is a loss function used in segmentation tasks, and the formula is shown as Equation (3):

$$DL(p, y) = 1 - \frac{2\sum_{i=1}^{n} p_i y_i + \epsilon}{\sum_{i=1}^{n} p_i^2 + \sum_{i=1}^{n} y_i^2 + \epsilon} \tag{3}$$

Where p_i represents the value of the i_{th} pixel in the predicted results, y_i represents the value of the i_{th} pixel in the ground truth results, n represents the total number of pixels, and ϵ is a small constant to avoid division by zero errors.

C. Evaluation Metrics

Average Precision(AP) is the area under the precision-recall curve. The formula is shown as Equation (4):

$$AP = \sum_{n} (R_n - R_{n-1}) \times P_n \tag{4}$$

Where n represents the index of points on the precision-recall curve, R_n denotes the recall of the nth point, R_{n-1} represents the recall of the $(n-1)_{th}$ point, and P_n denotes the precision of the n_{th}point.

There are multiple classes in the COCO and Cityscapes dataset, and the mean Average Precision (mAP) is calculated by taking the average of the AP values for all classes. The formula for mAP is shown in Equation (5).

$$mAP = \frac{1}{C} \sum_{j}^{C} (AP_j) \tag{5}$$

N represents the number of target categories, and AP_j denotes the Average Precision of the i_{th} category.

For segmentation tasks, the evaluation metric is the mAP at different IoU thresholds. mAP is calculated as $mAP^{(IoU=0.5:0.05:0.95)}$, where IoU=0.5:0.05:0.95 refers to IoU values ranging from 0.5 to 0.95 with an interval of 0.05. The mAP is calculated for each IoU threshold, and the average is taken to obtain the final mAP value.

Additionally, specific mAP values are calculated at different IoU thresholds. $mAP^{(IoU=0.5)}$ (mAP_{50}) represents the mAP value at IoU=0.5, and $mAP^{(IoU=0.75)}$(mAP_{75})represents the mAP value at IoU=0.75.

Furthermore, mAP^{small} (mAP_S) represents the mAP value for region sizes smaller than 322 pixels, mAP^{medium} (mAP_M) represents the mAP value for region sizes between 322 and 962 pixels, and mAP^{large}(mAP_L) represents the mAP value for region sizes larger than 962 pixels.

D. Experimental Settings

The experimental hardware setup in this study consisted of an Intel(R) Xeon(R) Bronze 3104 CPU @ 1.70GHz processor, 128GB of memory, and two NVIDIA GeForce GTX TITAN XP graphics processors. The operating system used was Ubuntu 22.04. The experiments used the mmdetection 2.28 deep learning framework based on PyTorch 1.3.

In the data preprocessing stage, the following data preprocessing techniques were applied: RandomFlip[17], Normalize, Random Crop, and Simple Copy Paste[18]. During the training stage, the Adam optimizer was used with a learning rate of 0.0002, a batch size of 6, and a weight decay of 0.05. The model was trained for 12 epochs, incorporating the Exponential Moving Average (EMA)[19] training technique, and the training process was accelerated using the FP16 precision format.

E. Analysis of Experimental Results.

The experimental results, as shown in Table I, indicate that among the box-based methods, Mask R-CNN achieves the highest accuracy but has a slower speed. YOLACT[20], on the other hand, has a faster speed but slightly lower accuracy. YOLACT++[21] improves its accuracy while sacrificing some speed. In the box-free methods, when using the same backbone, the proposed algorithm in this paper achieves the same level of accuracy as YOLACT++ but with a faster speed. Furthermore, compared to SOLO[22] and PolarMask[23] , it outperforms them by 1.4% and 5.5% in terms of accuracy, respectively, and also maintains a slight speed advantage. This demonstrates that UISNet has better predictive performance in the inference stage. The performance is further enhanced when utilizing EfficientNet as the backbone and MPAFPN as the bottleneck layer. It surpasses SOLO and PolarMask by 4.5% and 8.6% in terms of accuracy, respectively, without sacrificing much speed.

As shown in Table II, this paper compares different types of backbone networks and bottleneck structures. In terms of the Backbone, ResNet50[25] and FPN were used as baselines, achieving an mAP of 33.1%. When replacing ResNet50 with Res2Net[26], the mAP slightly decreased to 30.76%, indicating that Res2Net has slightly lower expressive power than ResNet50. By using HRNet[27] as the backbone, the mAP improved to 34.61%, which is 1.5% higher than ResNet50, demonstrating that HRNet has more robust feature representation capabilities. Introducing the superior EfficientNet as the backbone further increased the mAP to 35.84%, surpassing HRNet by 1.2% and confirming the effectiveness of EfficientNet over the ResNet series and HRNet. Regarding the Neck, with EfficientNet as the backbone, the introduction of the MPAFPN structure raised the mAP from 35.84% to 36.28%, a 0.4% improvement over the original FPN neck. This validates that optimizing the neck structure can further enhance performance.

As shown in Table III, three different object detection algorithms, namely SOLO, YOLACT, and UISNet proposed in this paper, were evaluated on the Cityscapes dataset to assess their robustness and accuracy in real urban scenarios. The Cityscapes dataset encompasses complex scenes. The experimental results demonstrate that the UISNet algorithm achieves an mAP of 33.62%, significantly outperforming SOLO (29.96%) and YOLACT (27.12%). This indicates that UISNet exhibits notable advantages in accuracy and generalization compared to the previous two algorithms.

As shown in Fig.6, this paper compares different instance segmentation algorithms in urban street scenes; different colors distinguish other instances, such as red for pedestrians and blue for vehicles. The result images demonstrate that the

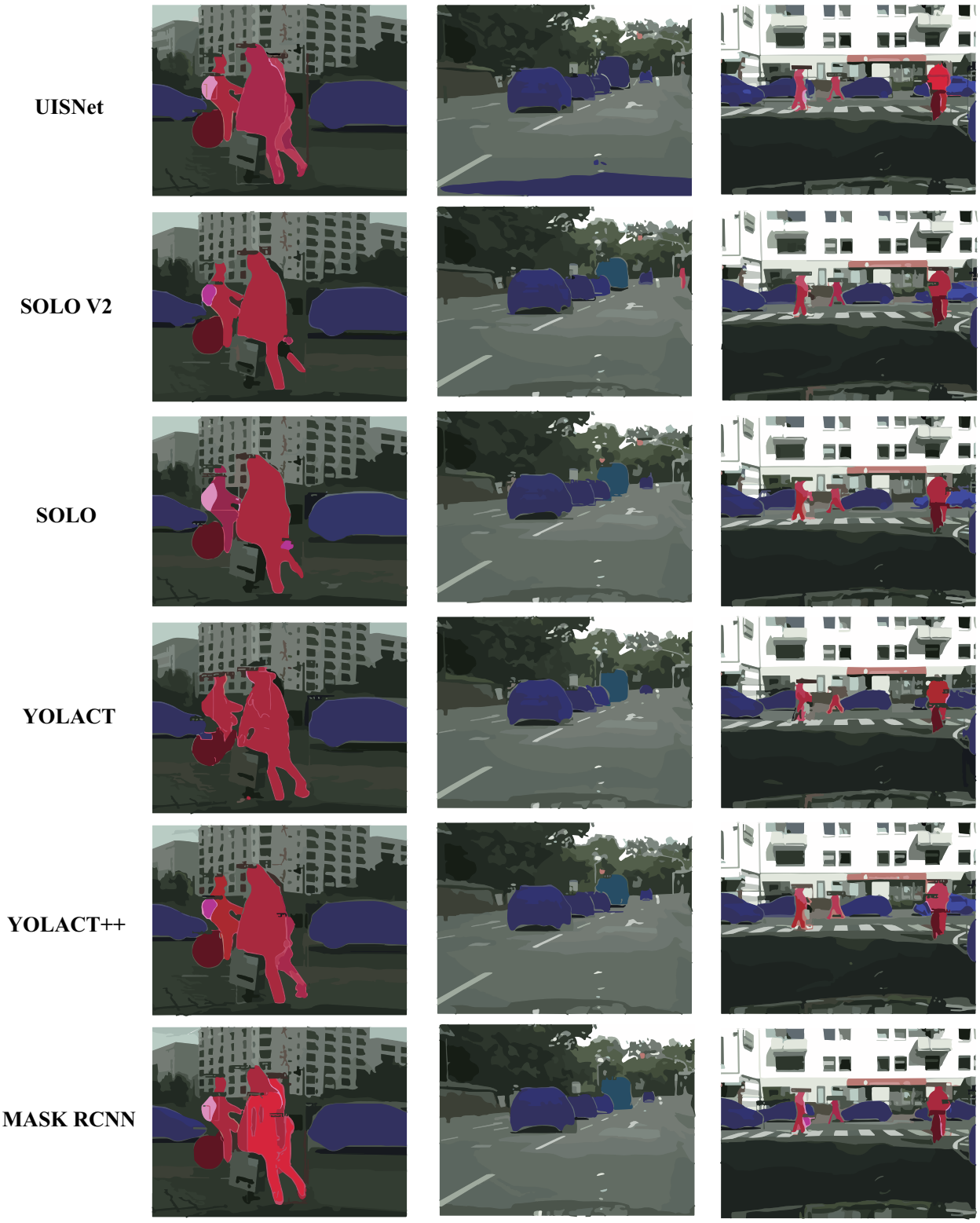

Fig. 6. Different Instance Segmentation Algorithm Result Images

YOLACT algorithm has the fastest recognition speed but the lowest recognition accuracy. In contrast, the proposed algorithm in this paper achieves significantly higher accuracy than YOLACT. The recognition accuracy of the YOLACT++ algorithm is also lower than that of the proposed algorithm in this paper. Although the Mask RCNN algorithm has good recognition accuracy, it has the slowest recognition speed. In contrast, the proposed algorithm in this paper has a much higher recognition speed than Mask RCNN. The SOLO V2[28] algorithm shows improvement compared to SOLO, but its recognition accuracy is not as high as the proposed algorithm in this paper.

V. Conclusion

We proposes an instance segmentation algorithm called UISNet. Firstly, the algorithm utilizes EfficientNet in the backbone to improve accuracy while enhancing the model's inference speed and efficiency. Secondly, the MPAFPN is employed in the neck module to enhance the model's feature fusion performance. Lastly, during the prediction stage, the

t

TABLE I
COMPARISON OF VARIOUS INSTANCE SEGMENTATION ALGORITHMS

Algorithm	Backbone Neck	mAP	mAP_{50}	mAP_{75}	mAP_S	mAP_M	mAP_L	FPS
box-based:								
MASK RCNN	Resnet50-FPN	33.01	52.43	35.06	17.35	35.93	45.33	3.6
YOLACT	Resnet50-FPN	26.99	44.43	27.75	8.81	28.01	42.72	**23.6**
YOLACT++	Resnet50-FPN	33.40	51.16	34.42	12.14	60.36	50.96	22.8
box-free:								
PolarMask	Resnet50-FPN	27.68	47.31	28.45	12.07	30.31	40.34	22
SOLO	Resnet50-FPN	31.77	51.19	33.28	11.65	34.64	48.59	21.2
UISNet	Resnet50-FPN	**33.10**	**52.39**	**35.10**	**12.85**	**36.24**	**51.46**	21.5
UISNet	EfficientNet-MAPFPN	**36.28**	**56.79**	**39.11**	**14.20**	**39.51**	**56.71**	19.8

TABLE II
COMPARISON BETWEEN BACKBONE NETWORKS AND FPN

BackBone	Neck	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_M	mAP_L
Resnet50	FPN	33.10	52.39	12.85	12.85	36.24	51.76
Res2net	FPN	30.76	47.57	11.69	11.69	32.1	46.88
HRNet	HRFPN[24]	34.61	53.96	13.6	13.6	37.67	53.87
EfficientNet-B4	FPN	35.84	55.62	13.9	13.9	38.67	55.37
EfficientNet-B4	MPAFPN	**36.28**	**56.79**	**14.2**	**14.2**	**39.51**	**56.71**

TABLE III
COMPARISON RESULTS OF OUR ALGORITHM ON THE CITYSCAPES DATASET

Algorithm	mAP	mAP_{50}	mAP_{75}	mAP_S	mAP_M	mAP_L
SOLO	29.96	49.89	NaN	2.4	26.14	57.85
YOLACT	27.12	42.46	NaN	1.9	20.62	50.81
UISNet	**33.62**	**54.79**	NaN	**3.6**	**28.82**	**62.64**

paper adopts a dual-branch design inspired by both top-down and bottom-up approaches. The two branches consist of a bounding box aggregation branch and a mask decoding branch. The bounding box aggregation branch generates high-dimensional information, such as the approximate shape and pose of the bounding boxes, based on the FCOS Head. The mask decoding branch generates mask prediction results. These two branches are fused using the Mask FCN Header. With these improvements, UISNet achieves an improvement of approximately 2%-5.0% over the baseline on the COCO dataset and a 4%-7% improvement on the Cityscapes dataset. This indicates that UISNet surpasses other models in terms of accuracy and speed and demonstrates remarkable generalization and robustness. UISNet has the potential to provide more reliable and efficient solutions for applications in autonomous driving, intelligent transportation, urban security, and other fields. In the future, further exploration of the model's performance and generalization capabilities in various scenarios, along with optimization and adjustments tailored to specific application contexts, will promote the development and application of instance segmentation.

REFERENCES

[1] S. Castillo, A. Bernal, and J. Rodríguez, "Object detection in digital documents based on machine learning algorithms." *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 688–699, 2023.

[2] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1713–1721, 2015.

[3] K.-M. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[4] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2016.

[5] S.-N. Xie and Z.-W. Tu, "Holistically-nested edge detection," *International Journal of Computer Vision,Kluwer Academic Publishers Norwell, MA, USA*, vol. 125, no. 1-3, pp. 3–18, 2017.

[6] K.-S. Kim, D.-N. Zhang, M.-C. Kang, and S.-J. Ko, "Improved simple linear iterative clustering superpixels," in *2013 IEEE International symposium on consumer electronics (ISCE)*, pp. 259–260, 2013.

[7] Z. Tian, C.-H. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9626–9635, 2019.

[8] H. Chen, K.-Y. Sun, Z. Tian, C.-H. Shen, Y.-M. Huang, and Y.-L. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8573–8581, 2020.

[9] Z. Tian, C.-H. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, Proceedings, Part I 16*, pp. 282–298, 2020.

[10] B. Koonce and B. Koonce, "Efficientnet," *Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization*, pp. 109–123, 2021.

[11] K.-X. Wang, J.-H. Liew, Y.-T. Zou, D.-Q. Zhou, and J.-S. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9196–9205, 2019.

[12] A. Kirillov, R. Girshick, K.-M. He, and P. Dollar, "Panoptic feature pyramid networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6392–6401, 2019.

[13] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, 2014.

[14] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K.-M. He, and P. Dollar, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

[16] R.-J. Zhao, B.-Y. Qian, X.-L. Zhang, Y. Li, R. Wei, Y. Liu, and Y.-G. Pan, "Rethinking dice loss for medical image segmentation," in *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 851–860, 2020.

[17] S. Takezaki and K. Kishida, "Data augmentation and the improvement of the performance of convolutional neural networks for heart sound classification," *IAENG Int. J. Comput. Sci*, vol. 49, pp. 1033–1040, 2022.

[18] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2918–2928, 2021.

[19] Y.-W. Li and X.-X. Zhang, "Object detection for uav images based on improved yolov6," *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 759–768, 2023.

[20] D. Bolya, C. Zhou, F.-Y. Xiao, and Y.-J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9157–9166, 2019.

[21] ——, "Yolact++ better real-time instance segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 1108–1121, 2022.

[22] X.-L. Wang, T. Kong, C.-H. Shen, Y.-N. Jiang, and L. Li, "Solo: Segmenting objects by locations," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, U.K., August 23–28, 2020, Proceedings, Part XVIII 16*, pp. 649–665, 2020.

[23] E. Xie, P. Sun, X.-G. Song, W.-H. Wang, X.-B. Liu, D. Liang, C.-H. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12 193–12 202, 2020.

[24] B. Huang, B.-Y. He, L.-N. Wu, and Z.-M. Guo, "High-resolution representations and multistage region-based network for ship detection and segmentation from optical remote sensing images," *Journal of Applied Remote Sensing*, vol. 16, no. 1, pp. 12 003–12 003, 2022.

[25] K.-M. He, X.-Y. Zhang, S.-Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[26] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 652–662, 2019.

[27] J.-D. Wang, K. Sun, T.-H. Cheng, B.-R. Jiang, C.-R. Deng, Y. Zhao, D. Liu, Y.-D. Mu, M.-K. Tan, X.-G. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[28] X.-L. Wang, R.-F. Zhang, T. Kong, L. Li, and C.-H. Shen, "Solov2: Dynamic and fast instance segmentation," *Advances in Neural information processing systems*, vol. 33, pp. 17 721–17 732, 2020.