ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

Lightweight Instance Segmentation Algorithm Based on Knowledge Distillation

Qiangda Shan*, Hao Chen, Ziru Liu

School of Information and Artificial Intelligence, Wuhu Institute of Technology, Wuhu, 241000, China
**Corresponding author: 101148@whit.edu.cn*

Abstract: *Instance segmentation is a technique that performs instance-level and pixel-level segmentation of images simultaneously. this paper proposes an instance-aware segmentation network based on a cross-resolution knowledge distillation architecture. The attention mechanism is utilized to fuse high and low-resolution image features for training the Teacher network, which then guides the Student network - taking low-resolution images as input - in feature extraction, ensuring segmentation accuracy while effectively reducing the network's parameter size and computational load. This paper adopts a pruning method based on switchable normalization to prune the backbone of the network, significantly alleviating the computational pressure during the training of the Teacher network, and further enhancing the real-time performance of the Student network. Experimental results on the public COCO dataset and a self-made pedestrian instance segmentation dataset Person-pic show that the proposed instance segmentation network effectively improves real-time segmentation performance while maintaining segmentation accuracy, achieving a balance between precision and speed.*

Keywords: *Instance segmentation, knowledge distillation, attention mechanism, network Pruning, lightweight*

1. Introduction

Instance segmentation[1] technologyaims to predict the category labels of each instance based on image content, determine their specific locations in the image, and generate pixel-level instance masks. This technology has been widely applied in various fields such as autonomous driving, intelligent surveillance, medical assistance, and remote sensing image analysis.

However, existing instance segmentation models often suffer from large parameter sizes and high computational complexity, making them difficult to deploy on resource-constrained platforms such as mobile devices. To address these issues, this paper proposes an instance segmentation method that incorporates a knowledge distillation architecture. First, both high-resolution and low-resolution images are used as inputs and employ the CBAM[2] attention mechanism to guide feature fusion, thereby training a high-precision Teacher network model. Subsequently, the feature maps extracted from this Teacher network are used as guidance, and pruning of the backbone network is performed using switchable normalization[3], leading to the training of a more real-time Student network.This method not only effectively reduces the computational demands of the model but also significantly mitigates the decline in segmentation accuracy caused by model simplification.

2. Related Work

2.1 Instance Segmentation

With the remarkable feature extraction capability demonstrated by AlexNet[4] on image classification tasks, deep convolutional neural networks have begun to be applied across various tasks in machine vision, achieving desirable results. Recent studies on instance segmentation are mostly based on convolutional neural networks and can be categorized into two-stage and single-stage instance segmentation methods according to their overall model architecture.

In two-stage instance segmentation methods, SDS [5]one of the earliest CNN-based instance segmentation approaches, uses MCG[6]to generate candidate regions and employs two parallel CNNs for feature extraction; however, SDS only uses the top-level features from the CNN for mask

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

generation, lacking positional detail and thus yielding lower segmentation accuracy. Mask RCNN[7]introduced the ROI Align module to solve the pixel position offset caused by rounding operations and adopted a fully convolutional network branch for segmentation mask generation, significantly improving segmentation accuracy compared to earlier segmentation networks.

In single-stage instance segmentation methods, YOLACT[8]is a segmentation algorithm that achieves real-time performance by generating a series of mask prototypes through FCN and combining them with instance mask coefficients generated at different layers to produce final masks. The original authors subsequently improved this approach, proposing YOLACT++[9]which offers higher segmentation accuracy and faster speed.SOLO[10]directly divides images into grids, with each grid responsible for semantic classification and instance mask generation of objects whose centers fall within it, thereby breaking free from the constraints of bounding boxes. SOLOv2[11]further decomposes the mask generation branch into kernel generation and mask prototype branches on top of SOLO, using Matrix NMS for post-processing of segmentation masks to achieve better segmentation outcomes.

2.2 Knowledge Distillation

Knowledge distillation[12]is a model compression technique in deep learning aimed at transferring knowledge from a large, complex model (teacher model) to a smaller, lightweight model (student model) to enhance the performance of the latter. Knowledge distillation can be divided into soft label distillation, relation-based distillation, feature-based distillation, etc. HintonHinton first proposed soft label distillation in 2015, where the class probability distribution (soft labels) output by the teacher model is used, followed by the student model learning these through minimizing the KL divergence between its own output and the teacher's. Relation-based distillation transfers global knowledge of the teacher model by comparing or modeling sample relationships. Feature-based distillation utilizes intermediate layer features from the teacher model (e.g., hidden layer outputs) to guide the student model, aligning intermediate layer features of both models (using L2 distance or attention mechanisms) to force the student to mimic the feature representation capabilities of the teacher.

2.3 Neural Network Pruning

Neural network pruning was inspired by synaptic pruning in the human brain and involves removing redundant neurons and weight parameters based on their importance, aiming to reduce model complexity while maintaining performance. Depending on the granularity of operations, pruning can be classified into structured and unstructured pruning, as shown in Figure 1. Structural pruning removes neurons, operating at a coarser granularity, offering more significant optimizations for inference speed and storage space, but potentially causing a greater impact on accuracy. Unstructured pruning focuses on weight parameters, providing finer granularity but resulting in sparse weight matrices that require hardware and low-level computational library support for handling.

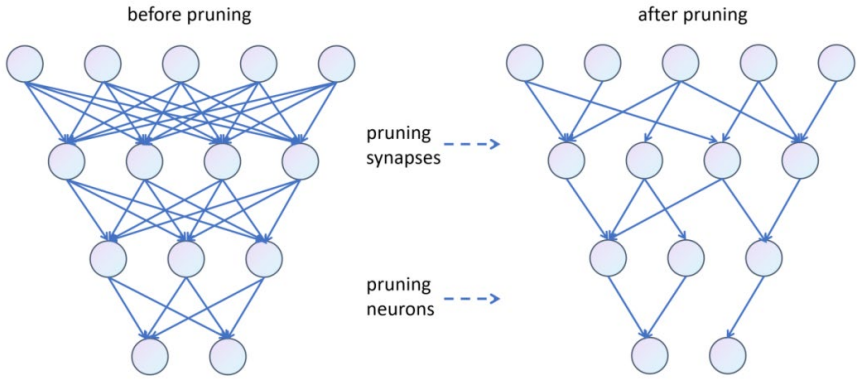


Figure 1 Neural NetWork Pruning

3. Model Design

3.1 Network Architecture

This paper introduces the MSAD[13] framework, originally proposed for object detection tasks, and

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

integrates it with a segmentation framework based on CondInst[14]to design an instance-aware segmentation architecture. This integration results in a lightweight segmentation network. The overall network structure is illustrated in Figure 2.

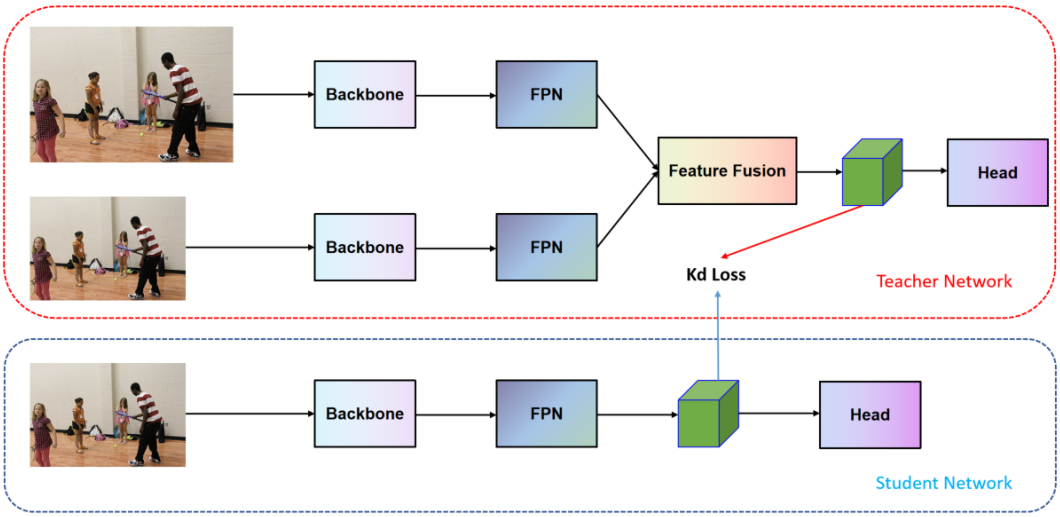


Figure 2 Network Structure

In this architecture, we first independently train a Teacher network that achieves high segmentation accuracy. The Teacher network uses two different image resolutions as input data. The large resolution input images have dimensions (H, W), while the small resolution input images are scaled down to (H/2, W/2). After feature extraction through the Feature Pyramid Network (FPN) module, features are fused using a Feature Fusion module guided by an attention mechanism. Finally, the fused features are fed into an instance segmentation head (inspired by CondInst) for mask generation.

During the training of the Teacher network, images of different resolutions are separately input into identical but parameter-non-shared backbone networks for feature extraction. When using ResNet50[15]as the backbone, the training process generates twice the number of parameters and computational load compared to a single ResNet50. To improve training efficiency and further lighten the network structure, a pruning method based on switchable normalization is applied to prune the ResNet50, resulting in a pruned ResNet50 with a pruning rate of 25% being used as the backbone network in our model.

After completing the training of the Teacher network, the feature maps generated by the Teacher network guide the Student network in feature extraction. This guidance process is realized through Kd Loss.

3.2 Channel Pruning Based on switchable normalization

To improve the training and inference speed of the model, especially focusing on the backbone network which is the most computationally intensive part of the entire network, a channel pruning operation[16] is applied to the backbone during the training of the Teacher network.where sub-networks with varying numbers of channels are trained within a larger network. Different channel widths are set, and as illustrated in Figure 3, the weight parameters are divided into four equal parts along the channel dimension. The orange sections represent activated parameters, whereas the grey sections indicate inactive parameters.

By dividing the channels into four segments, four differently sized sub-networks with widths of [0.25, 0.5, 0.75, 1.0] are formed. Before each gradient descent step, backpropagation is performed separately on all four sub-networks. As a result, the convolution parameters in the top 25% are shared among all four sub-networks and undergo four rounds of backpropagation. This iterative process gradually increases the importance of channels at the front, enabling incremental compression of less critical channels.This approach not only optimizes the backbone network for better efficiency but also ensures that the model maintains high segmentation accuracy despite the reduction in computational complexity.

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

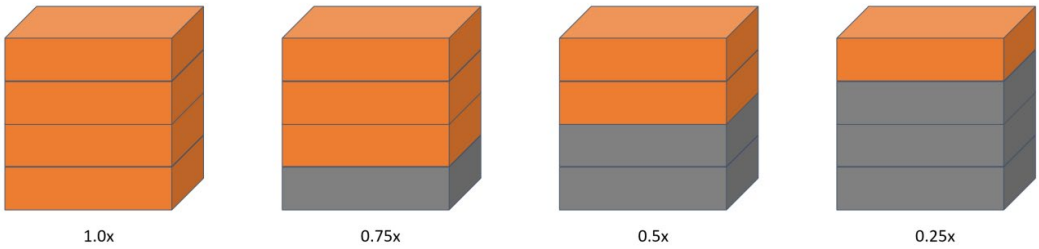
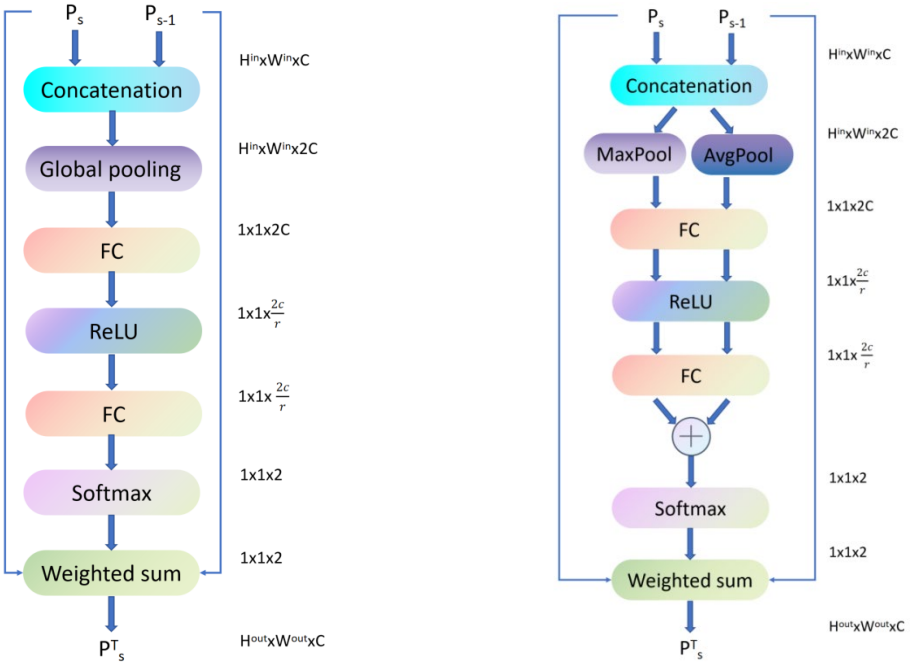


Figure 3 Channels divison

3.3 Feature Fusion Based on Attention Mechanism

As shown in Figure 1,the feature maps extracted from the upper branch of the Teacher network perform better for small object instance segmentation because high-resolution images preserve fine visual details more effectively. Conversely, the feature maps extracted from the lower branch of the Teacher network achieve better results for large object instance segmentation. when both branches have the same size of receptive fields, the lower branch captures more contextual information. Since the ratio between large and small objects in an image is not fixed, an attention mechanism is employed in the Feature Fusion module to guide the fusion of feature maps. In this paper, two types of feature fusion modules guided by attention mechanisms are designed.



(a)SE Fusion Module (b)CBAMFusion Module

Figure 4 Fusion Module

The first type, as shown in Figure 4(a), is a fusion module modified based on the SESE[17]modulewhich can adaptively adjust the importance of different feature maps according to the image content.

$$P_s^p = \frac{1}{H^{P_s} \times W^{P_s}} \sum_{i=1}^{H^{P_s} W^{P_s}} [P_s, P_{s-1}] (i, j) \quad (1)$$

The computational details are shown in Equation (1), where $P_s, P_{s-1}$are the feature maps generated from high and low resolution images through a feature pyramid, respectively. H^{P_s}, W^{P_s}represent the height and width of the feature map P_s, respectively. $[P_s, P_{s-1}]$indicates concatenation of the two feature maps along the channel dimension. $P_s^p$is input into a multi-layer perceptron (MLP) and normalized using the Softmax function to generate weights for high and low resolution feature fusion. Feature fusion is then performed using Equation(2), where $h_s\{0\}, h_s\{1\}$ represent the weights generated after passing through the Softmax layer.

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

$$P_s^T = h_s\{0\} \cdot P_s + h_s\{1\} \cdot P_{s-1} \tag{2}$$

The second method, shown in Figure 4(b), is modified based on the CBAM module, with an overall approach similar to the first method. The computational details are represented by Equations (3) and (4).

$$P = [P_s, P_{s-1}] \tag{3}$$

$$\begin{aligned} h_s(P) &= \sigma(\mathrm{MLP}(\mathrm{AvgPool}(P)) + \mathrm{MLP}(\mathrm{MaxPool}(P))) \\ &= \sigma\left(W_1\left(W_0\left(P_{avg}\right)\right) + W_1\left(W_0(P_{max})\right)\right) \end{aligned} \tag{4}$$

The notation in these equations maintains consistency with that in the SE feature fusion module. MLP denotes the fully connected layer in Figure 4(b), σ represents the Softmax function, W_0 and W_1 are the weight parameters of the fully connected layer, which are shared by the feature maps obtained from average pooling and max pooling.

3.4 Loss Function

The overall loss function of this network can be expressed by Equation (5), where L_{kd} is the knowledge distillation loss function in Figure1and this loss is only included in the total loss when training the Student network. L_{fcos} is the original loss of the FCOS detection framework.

$$L_{overall} = L_{fcos} + \lambda L_{mask} + L_{kd} \tag{5}$$

In formula (5), λ represents the loss balancing factor, which is set to 1 in this paper. L_{mask}denotes the mask generation loss function, and its specific calculation details can be expressed by formula (6). Here, $M_{x,y}$represents the predicted mask, $M^*_{x,y}$represents the annotated mask, L_{dice} represents Dice Loss[18], $f_{\{c^*_{x,y}>0\}}$indicates the discrimination function for positive and negative samples.

$$L_{mask} = \frac{1}{N_{pos}} \sum_{x,y} f_{\{c^*_{x,y}>0\}} L_{dice}\left(M_{x,y}, M^*_{x,y}\right) \tag{6}$$

L_{kd}can be represented by Equation(7), where T and S represent the Teacher network and Student network, respectively. $P_s$denotes the feature map generated by the sth level of the feature pyramid. m is the stride for cross-level alignment, and$\tau$is a hyperparameter used to balance the instance segmentation loss and distillation loss, which is set to 3.0 in this paper.

$$L_{kd} = \tau \cdot \sum_s |P_s^T - P_{s-m}^S| \tag{7}$$

4. Experimental Results

4.1 Experimental Setup

This paper adopts a warmup training strategy, with the number of warmup iterations set to 500. The batch size is set to 4, and a gradient accumulation strategy is employed for double gradient accumulation. The initial learning rate is set to 0.01, which decays to one-tenth of its original value at 120,000 and 160,000 gradient descent steps, respectively. The entire training process involves 190,000 gradient descent steps. For optimization, a momentum-based SGD method is used, with the Momentum parameter set to 0.9.

4.2 Experimental Design and Result Analysis

To verify the effectiveness of the proposed pruning method and determine the optimal pruning rate, theoretical and experimental analyses were combined. To validate the effectiveness of the cross-resolution knowledge distillation architecture in overall instance-aware segmentation, relevant ablation experiments were designed. Additionally, the best feature fusion approach within the Feature Fusion module was explored. To demonstrate the superiority of the proposed method in this research field, performance comparison experiments were conducted with some classic instance segmentation models.

4.2.1 Backbone Pruning Experiments

Principal Component Analysis (PCA) was applied to reduce the dimensions of four parts of weights

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

for visualization analysis, as shown in Figure 5. red dots represent the top 25% of weights, black dots represent weights from 25% to 50%, blue dots represent weights from 50% to 75%, and green dots represent weights from 75% to 100%.

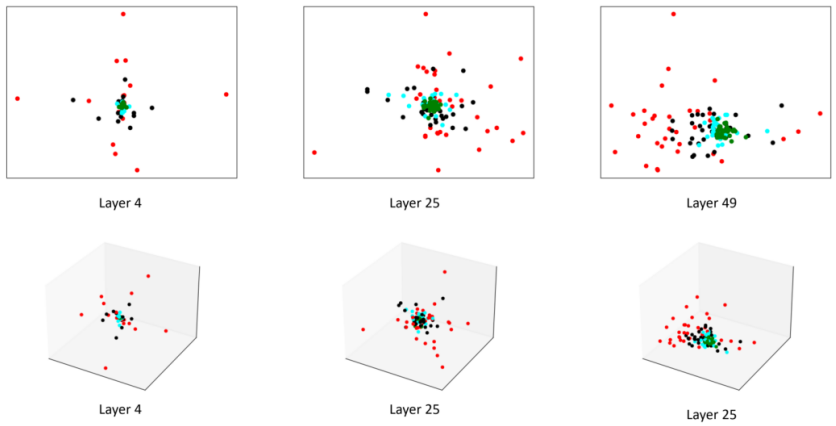


Figure 5 Weight distribution

Observations indicate significant overlap between the last 25% of weights and the first 75%, suggesting that the last 25% of channel weights are somewhat redundant. Removing these weights has little impact on the network's segmentation accuracy. Based on an optimal pruning rate of 0.25, experiments conducted on the COCO and Person-pic datasets (as shown in Table 1) confirm that removing one quarter of the original channels yields the best results, further validating the correctness of the optimal pruning rate determined through weight analysis.

Table 1 Pruning experiments

dataset	pruning rate	AP	AP50	AP75	APs	APm	APl
COCO	0.00	32.6	51.7	34.2	12.3	34.1	51.3
	0.25	31.6	50.2	33.2	12.1	32.9	49.9
	0.50	28.4	45.5	29.5	10.9	29.2	45.2
	0.75	23.3	40.1	24.6	8.5	24.0	39.3
Person-pic	0.00	65.8	89.3	74.5	-	-	-
	0.25	65.3	89.1	73.5	-	-	-
	0.50	64.5	89.2	72.1	-	-	-
	0.75	63.3	89.3	69.9	-	-	-

4.2.2 Cross-Resolution Knowledge Distillation Experiment

To verify the effectiveness of the cross-resolution knowledge distillation architecture, experiments were conducted on the COCO and Person-pic datasets using high-resolution and low-resolution images without embedding the knowledge distillation framework. The results were compared with those of the Teacher network and the Student network. The experimental results are shown in Table 2.

Here, HR and LR respectively represent networks that use high-resolution (original image resolution) and low-resolution (half of the high-resolution) images as input, without embedding the knowledge distillation framework. The LR network shows a significant drop in accuracy compared to the HR network; for example, in the COCO dataset, AP drops directly from 31.6 to 26.8. However, the Student network, which also uses low-resolution images as input but is guided by the Teacher network, greatly mitigates the decline in accuracy. Compared to the Teacher network's AP of 32.7, the Student network only experiences a decrease of 1.7.

Table 2 KD verification experiment

dataset	network type	AP	AP50	AP75	APs	APm	APl
COCO	HR	31.6	50.9	33.3	13.7	34.1	48.2
	LR	26.8	44.6	27.6	6.7	28.3	46.5
	Teacher	32.7	52.1	34.5	14.5	35.1	49.6
	Student	31.0	49.5	32.6	11.5	32.2	49.2
Person-pic	HR	66.5	88.8	73.1	-	-	-
	LR	62.6	87.3	70.1	-	-	-
	Teacher	67.6	89.9	74.6	-	-	-
	Student	64.8	88.7	73.0	-	-	-

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

4.2.3 Feature Fusion Experiment

Table 3 shows the results obtained by using different feature fusion methods in the Teacher network. Here, "Sum" indicates pixel-wise addition, "Conv" indicates fusion through simple convolution. "SEFF" and "CBFF" respectively represent the SE cross-layer feature fusion module and the CBAM cross-layer feature fusion module. The CBFF module generates attention maps by applying both max pooling and average pooling, taking into account both feature saliency and feature generality. It is the most effective among the four feature fusion methods.

Table 3 Experiments of feature fusion

dataset	fusion method	AP	AP50	AP75	APs	APm	APl
COCO	Sum	31.7	51.1	33.4	13.7	34.3	48.4
	Conv	32.0	51.6	33.8	14.0	34.7	48.8
	SEFF	32.6	**52.1**	34.4	14.3	**35.2**	49.3
	CBFF	**32.7**	**52.1**	**34.5**	**14.5**	35.1	**49.6**
Person-pic	Sum	66.6	89.0	73.3			
	Conv	66.9	89.4	73.8			
	SEFF	67.4	**89.9**	74.3			
	CBFF	**67.6**	**89.9**	**74.6**			

4.2.4 Comparative Experiment

The performance comparison results between the instance segmentation network proposed in this paper and current state-of-the-art instance segmentation networks are shown in Table 4. The performance data for the YOLACT method is directly cited from the original paper, while all other data were obtained through experiments. This experiment was conducted using the publicly available COCO dataset.

Compared to conventional instance segmentation models, the real-time performance of our model has significantly improved; however, there is still a certain gap in processing speed compared to the real-time instance segmentation network YOLACT. The model proposed in this paper achieves a notable improvement in segmentation accuracy compared to the YOLACT model, striking a balance between processing speed and segmentation accuracy.

Table 4 Performance comparison experiments

Method	Backbone	FPS	AP	AP50	AP75	APs	APm	APl
BoxInst[19]	ResNet50	15	27.7	48.5	27.2	11.5	30.5	40.5
MEInst[20]	ResNet50	13	29.2	49.8	30.2	14.1	31.4	42.1
SOLO	ResNet50	14	29.3	48.7	30.2	11.0	31.2	44.2
SOLOv2	ResNet50	13	31.1	49.7	32.9	11.3	33.1	48.0
MaskRcnn	ResNet50	14	32.3	52.0	34.6	15.0	34.1	46.8
CondInst[21]	ResNet50	15	33.0	52.6	35.0	15.0	36.1	47.5
BlendMask[22]	ResNet50	15	33.1	52.3	35.2	14.8	35.8	47.4
YOLACT	ResNet50	42	28.2	46.6	29.2	9.2	29.3	44.8
outs	ResNet50_0.75x	32	31.0	49.5	32.6	11.5	32.2	49.2

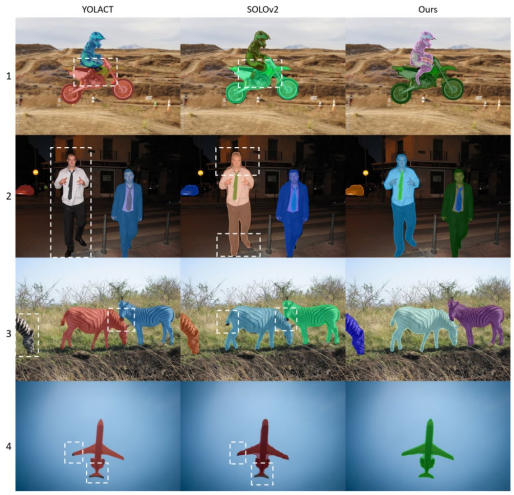


Figure 6 Segmentation quality visualization comparison

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

To further compare the segmentation quality with other methods, some images were selected from the COCO dataset and segmented using the YOLACT model, the SOLOv2 model, and the model proposed in this paper. The results are shown in Figure 6.

5. Conclusion

This paper embeds a cross-resolution knowledge distillation architecture into an instance segmentation model, suppressing the loss of segmentation accuracy caused by resolution reduction. The attention mechanism is introduced into feature fusion, enhancing the model's adaptability for segmentation. Pruning operations are applied to the backbone network, reducing the computational cost of the model. Compared with some real-time segmentation models, the segmentation speed of our model still has room for improvement, and different lightweight strategies can be further explored to optimize it in the future. In summary, this segmentation network achieves a balance between segmentation accuracy and processing speed.

Acknowledgement

General project: Research on Content Aware Lightweight Pedestrian Segmentation Method (wzyzr202521)

"Characteristics Specialty of Applying Artificial Intelligence Technology to Serve the Ten Emerging Industries" (Project No. 2023sdxx158)

"Typical Production Practice Project of School-Enterprise Cooperation between Wuhu Vocational and Technical College and Atech Automotive Electronics (Wuhu) Co., Ltd." (Project No. 2023dxscsj01)

References

[1] Garcia-Garcia, Alberto, Orts-Escolano, Sergio, Oprea, Sergiu, et al. A survey on deep learning techniques for image and video semantic segmentation. Applied Soft Computing, 70: 41--65, 2018.

[2] Woo, Sanghyun, Park, Jongchan, Lee, Joon-Young, et al. Cbam: Convolutional block attention module. //Proceedings of the European conference on computer vision (ECCV), 2018.

[3] Luo, Ping, Zhang, Ruimao, Ren, Jiamin, et al. Switchable Normalization for Learning-to-Normalize Deep Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(2): 712-728, 2021.

[4] Krizhevsky, Alex, Sutskever, Ilya, Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

[5] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, Jitendra Malik. Simultaneous detection and segmentation. //European conference on computer vision, 2014.

[6] Arbel, Pablo, Pont-Tuset, Jordi, Barron, Jonathan T, et al. Multiscale combinatorial grouping. //Proceedings of the IEEE conference on computer vision and pattern recognition, 2014.

[7] He, Kaiming, Gkioxari, Georgia, Dollar, Piotr, et al. Mask r-cnn. //Proceedings of the IEEE international conference on computer vision, 2017.

[8] Bolya, Daniel, Zhou, Chong, Xiao, Fanyi, et al. Yolact: Real-time instance segmentation. //Proceedings of the IEEE/CVF international conference on computer vision, 2019.

[9] Bolya, Daniel, Zhou, Chong, Xiao, Fanyi, et al. Yolact++: Better real-time instance segmentation. IEEE transactions on pattern analysis and machine intelligence, 2020.

[10] Wang, Xinlong, Kong, Tao, Shen, Chunhua, et al. Solo: Segmenting objects by locations. //European Conference on Computer Vision, 2020.

[11] Wang, Xinlong, Zhang, Rufeng, Kong, Tao, et al. Solov2: Dynamic and fast instance segmentation. Advances in Neural information processing systems, 33: 17721--17732, 2020.

[12] Geoffrey, Vinyals, Oriol, Dean, Jeff. Distilling the Knowledge in a Neural Network. Computer Science, 14(7): 38-39, 2015.

[13] Qi, Lu, Kuen, Jason, Gu, Jiuxiang, et al. Multi-scale aligned distillation for low-resolution detection. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

[14] Tian, Zhi, Shen, Chunhua, Chen, Hao. Conditional convolutions for instance segmentation. //European Conference on Computer Vision, 2020.

ISSN 2616-5775 Vol. 8, Issue 4: 40-48, DOI: 10.25236/AJCIS.2025.080405

[15] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, et al. Deep residual learning for image recognition. //Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
[16] Jiahui Yu, Linjie Yang, Ning Xu, et al. Slimmable Neural Networks. CoRR, abs/1812.08928, 2018.
[17] Jie Hu, Li Shen, Gang Sun. Squeeze-and-Excitation Networks. CoRR, abs/1709.01507, 2017.
[18] Milletari, Fausto, Navab, Nassir, Ahmadi, Seyed-Ahmad. V-net: Fully convolutional neural networks for volumetric medical image segmentation. //2016 fourth international conference on 3D vision (3DV), 2016.
[19] Tian, Zhi, Shen, Chunhua, Wang, Xinlong, et al. Boxinst: High-performance instance segmentation with box annotations. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
[20] Zhang, Rufeng, Tian, Zhi, Shen, Chunhua, et al. Mask encoding for single shot instance segmentation. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[21] Tian, Zhi, Shen, Chunhua, Chen, Hao. Conditional convolutions for instance segmentation. //European Conference on Computer Vision, 2020.
[22] Chen, Hao, Sun, Kunyang, Tian, Zhi, et al. Blendmask: Top-down meets bottom-up for instance segmentation. //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020.