Multi-level Feature Reweighting and Fusion for Instance Segmentation

Xuan-Thuy Vo, Tien-Dat Tran, Duy-Linh Nguyen and Kang-Hyun Jo
Department of Electrical, Electronic and Computer Engineering,
University of Ulsan
Ulsan (44610), South Korea
Email: {xthuy, tdat}@islab.ulsan.ac.kr; ndlinh301@mail.ulsan.ac.kr; acejo@ulsan.ac.kr

***Abstract*—Accurate instance segmentation requires high-resolution features for performing a dense pixel-wise prediction task. However, using high-resolution feature maps results in highly expensive model complexity and ineffective receptive fields. To overcome the problems of high-resolution features, conventional methods explore multi-level feature fusion that exchanges the information between low-level features at earlier layers and high-level features at top layers. Both low and high information is extracted by the hierarchical backbone network where high-level features contain more semantic cues and low-level features encompass more specific patterns. Thus, adopting these features to the training segmentation model is necessary, and designing a more efficient multi-level feature fusion is crucial. Existing methods balance such information by using top-down and bottom-up pathway connections with more inefficient convolution layers to produce richer multi-scale features. In this work, we contribute two folds: (1) a simple but effective multi-level feature reweighting layer is proposed to strengthen deep high-level features based on channel reweighting generated from multiple features of the backbone, and (2) an efficient fusion block is proposed to process low-resolution features in a depth-to-spatial manner and combine enhanced multi-level features together. These designs enable the segmentation models to predict instance kernels for mask generation on high-level feature maps. To verify the effectiveness of the proposed method, we conduct experiments on the challenging benchmark dataset MS-COCO. Surprisingly, our simple network outperforms the baseline in both accuracy and inference speed. More specifically, we achieve 35.4% AP^{mask} at 19.5 FPS on a GPU device, becoming a state-of-the-art instance segmentation method.**

***Index Terms*—Instance segmentation, multi-level features, multi-scale fusion, cross-scale reweighting**

I. Introduction

In recent years, Convolutional Neural Networks (CNNs) and vision Transformers [1] served as the backbone networks or feature extractors have achieved remarkable improvements in solving down-stream tasks such as image classification [2], human pose estimation [3], object detection [4], and image segmentation [5]. In the common paradigm, the feature extractors are pretrained on the classification dataset ImageNet [2] and the downstream tasks use pretrained models as initialization weights and fine-tune these weights for other specific tasks. The backbone networks extract informative features from visual data in a hierarchical processing [6]. In the shallower layers of the feature extractors, low-level features are cast with more descriptive information such as edges, corners, color conjunctions, and textures. Deeper layers of the backbone learn global semantic information such as whole objects with diverse poses, contextual scenes, and structural parts of objects.

However, downstream tasks require dense pixel-wise predictions at high-level feature maps. This requirement creates two bottlenecks: (1) high computational cost since the model grows quadratically in the increase of input resolution, and (2) making the receptive fields of the models ineffective because the models demand more stacked convolution layers, and heavy atrous layers to form long-range dependencies in input data. The key solution for this problem is to design a multi-scale feature interaction that fuses the information among multiple features with different dimensions. The multi-scale feature fusion brings two merits: feature-level balance and solved scale variations. Firstly, in the feature-level balance aspect, downstream tasks perform consistent predictions on the balanced features. Secondly, in solving scale variation problems, the models can predict multiple objects on multi-scale features. Naturally, the scales of objects in the real-world datasets vary in the enormous range, for example, too small objects and too large objects. Therefore, downstream tasks assign each object's scale range to each feature. For instance, high-level features with small scales contain strong information for identifying large objects, and low-level features at shallow layers encode the information about small objects.

In the existing literature, there are many introduced methods to solve the aforementioned problems. FPN [7] constructs the feature pyramid networks for the object detection task, proposing top-down pathway connection to aggregate multi-scale features. Inspired by this intuitive idea, recent methods introduce bottom-up pathway connections [8], balanced feature pyramid networks [9], stair-step FPN [10], and generalized FPN [11].

Instance segmentation is fundamental but challenging research of the downstream tasks, which has been widely used in many real-world applications such as autonomous vehicles, surveillance systems [12], medical diagnosis and treatments, vision robotics, etc. The multi-scale feature fusion is the central part of instance segmentation models that require strong semantic and discriminative features to segment the presence of objects at the pixel level in the images or videos. Some instance segmentation methods [5], [9], [13], [14] use top-down pathway connections in FPN and bottom-up pathway connections in PANet [8] for performing multi-scale feature fusion. In this paper, we present a new multi-scale feature fusion for instance segmentation that reweights high-level feature maps based on the information of multi-scale features at the earlier and deeper layers. The proposed channel reweightings are dynamically learned conditioned on the feature maps, and calculated across the channel axis via efficient 1D convolution layers. These channel reweightings convey the information of low- and high-level features. Therefore, it brings strong information of all the features from the hierarchical backbone and is sufficient for interacting information among multi-level features. Moreover, our multi-level feature reweighting block is lightweight compared with atrous convolution or vanilla convolution because this block increases linearly in increases of the number of channels.

Finally, the refined multi-level features are fused together to create one final feature for mask predictions. The final feature is generated through depth-to-spatial block and summation. Depth-to-spatial block transforms values inside each feature from depth to spatial dimension. Thus, this operation increases the resolution of the feature map without any extra parameters, viewed as upsampling operation. Summation operation is used to take the average over multi-level features to one feature. The averaged feature contains strong information for instance kernel and mask classification. The experimental results are evaluated on the testing set of the benchmark MS-COCO. Without bells and whistles, the proposed method surpasses previous methods in both Average Precision (AP) and efficient model complexity. It demonstrates the effectiveness of the proposed method.

II. Literature Review

In this section, we briefly review existing instance segmentation methods and multi-level feature fusion in downstream tasks related to our method.

A. *Instance segmentation*

In the literature, instance segmentation methods are grouped into two kinds: top-down methods and bottom-up methods. Top-down methods heavily rely on the bounding box predictions of detection task, while bottom-up methods learn affinity embedding of same instances and different instances and require extra post-processing to distinguish object instances.

Being top-down methods, Mask R-CNN [5] follows the ”segment-by-detect” paradigm that attaches one classification branch into the R-CNN network of the Faster R-CNN [15] to segment the object instances inside each predicted bounding boxes. HTC [16] jointly learns instance segmentation and object detection tasks through a cascade network. YOLACT [14] predicts mask coefficients along with bounding box localization and prototype masks, and then final masks are generated by a linear combination of coefficients and prototype masks. YolactEdge [17] improves the efficiency of the YOLACT by using TensorRT optimization and similarity learning of temporal information.

Bottom-up methods directly generate instance masks without the need of bounding box annotation, based on instance categories and semantic categories to group each pixel into a predefined number of the objects in one image. Both instance and semantic categories are viewed as classification issues. PolarMask [18] formulates instance segmentation task into contour localization problem according to the Polar representation. SOLO [13] firstly divides the input image into cell grids, and then each grid cell is assigned to each object instance. Secondly, to generate masks, each grid cell is classified into the semantic category and associated instance masks. SOLOv2 [19] improves the efficiency of the SOLO network by using dynamic convolution, which produces final masks by learning similarities between instance kernels and mask features. Inspired by the interesting idea of the SOLO, K-Net [20] proposes a unified segmentation network that can predict semantic, instance, and panoptic segmentation into one cascade network. In the K-Net method, the features of instance kernels are strengthened through adaptive kernel updates using linear transformations and kernel interactions using the Transformer block. In this paper, motivated by the new design of the classified instance kernels, we adopt the classification network for learning instance kernels and associated masks as the baseline network. Unlike existing methods, this paper focuses on multi-level feature reweighting and fusion instead of the improvements of the segmentation head.

B. *Muli-level feature fusion*

FPN [7] is the first method that introduces multi-level feature fusion for the object detection task. FPN explores the top-down pathway connection from high-level features to low-level features, and thus, low-feature features contain the global features at the top layers. This procedure is computed through lateral connections using 1×1 convolution and upsampling layer using a simple nearest neighbor operation. Although FPN achieves promising results for downstream tasks, the top-down pathway connection is still straightforward since high-level features and low-level features are different after fusing. PANet [8] refines multi-scale features by using both top-down and bottom-up pathway connections and thus, low-level features useful for the localization task are propagated to top layers.

Instead of sequential connections, Libra R-CNN [9] proposes Balanced Feature Pyramid (BFP) network that takes an average over multi-level features to one feature and refines this fused feature by non-local operation. To construct the final feature pyramid, the BFP utilizes a residual connection between refined features and multi-scale features from the backbone. Stair-step FPN [10] uses 1×1 convolution layer between top-down and bottom-up pathways to enhance the fused feature along the channel dimension. Additionally, the information of features in top-down and bottom-up pathways is exchanged through the short-cut connection. GFPN [11] introduces heavy neck for detection task, exploring dense connection on the same feature level and queen-fusion computation on different feature levels. In dense connection, GFPN proposes log_2n-connection rule that removes some features at earlier layers. In the queen-fusion part, GFPN fuses possible features from different feature levels. Differently, this paper investigates the

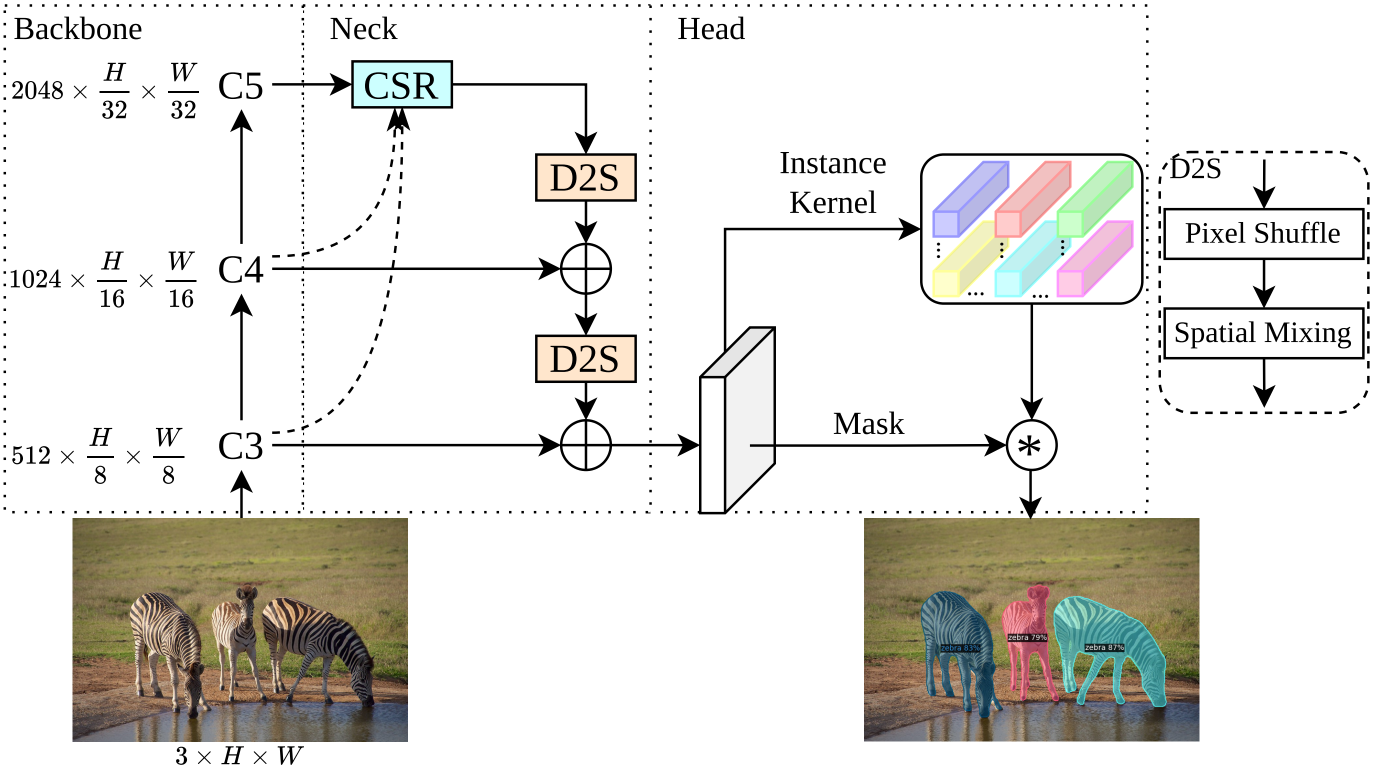
Backbone
Neck
Head
$2048 \times \frac{H}{32} \times \frac{W}{32}$
C5
CSR
D2S
Instance Kernel
D2S
Pixel Shuffle
Spatial Mixing
$1024 \times \frac{H}{16} \times \frac{W}{16}$
C4
D2S
Mask
$512 \times \frac{H}{8} \times \frac{W}{8}$
C3
zebra 83%
zebra 79%
zebra 87%
$3 \times H \times W$

Fig. 1. The overall architecture of the proposed method includes three main components: backbone network, neck network, and segmentation head. The backbone network extracts the hierarchical features from the input image. The neck network exchanges the information among multi-level features of the backbone network, implemented by our proposed modules, Cross-Scale Reweighting (CSR) and Depth-to-Space (D2S). In this work, we only utilize three features {C3, C4, C5} from three stages of the backbone to produce the final refined feature. The segmentation head consists of two branches: instance kernel classification and mask branches. The instance masks are generated by performing similarity learning between predicted instance kernel and mask features through convolution operation denoted by $\circledast$.

importance of feature reweighting to highlight more semantic features.

III. Methodology

In this section, we develop the efficient neck network that effectively balances information between low-level and high-level features through our proposed modules: Cross-Scale Reweighting (CSR) and Depth-to-Space (D2S). The description of the CSR and D2S is illustrated in subsection III-A and subsection III-B, respectively.

The overall architecture of our proposed method is shown in Fig. 1. The backbone network is ResNet-50 [21] pre-trained on large-scale ImageNet [2] dataset as weight initialization during training, taking the input image $I \in \mathbb{R}^{3\times H\times W}$ where H, and W are height and width dimension of the image. The CSR module learns the important features from the hierarchical backbone. The D2S module upsamples the feature map to the higher dimension that is equal to the feature dimension at the earlier layer. The segmentation head takes the fused feature to predict instance masks via learned instance kernels and mask features.

A. Cross-Scale Reweighting (CSR)

The aim of our proposed method is to create multi-level reweighting revealing the global semantic information to strengthen high-level features and also low-level features from multiple features of the hierarchical backbone. Given the feature $\mathbf{C}_3 \in \mathbb{R}^{512\times\frac{H}{8}\times\frac{W}{8}}$, $\mathbf{C}_4 \in \mathbb{R}^{1024\times\frac{H}{16}\times\frac{W}{16}}$, and $\mathbf{C}_5 \in \mathbb{R}^{2048\times\frac{H}{32}\times\frac{W}{32}}$, we compute multi-level reweighting $\mathbf{R}$ as follows:

$$\mathbf{R} = \delta(\mathbf{W_1 W_2 H}), \tag{1}$$
$$\mathbf{H} = [\mathbf{F3}, \mathbf{F4}, \mathbf{F5}], \tag{2}$$

where $\mathbf{H} \in \mathbb{R}^{768}$ is the concatenated feature of the multi-level features and $[\mathbf{F3}, \mathbf{F4}, \mathbf{F5}]$ indicates concatenation of the three features along channel dimension. $\mathbf{W_1} \in \mathbb{R}^{768\times256}$, and $\mathbf{W_2} \in \mathbb{R}^{256\times256}$ are linear transformations implemented by lightweight 1D convolution operation, which learn the relationship between channels features from different scales. δ denotes the sigmoid activation function to output probability scores that form which channels are allowed to be highlighted during training.

The feature $\mathbf{F}_i \in \mathbb{R}^{256}$, where $i \in \{3, 4, 5\}$, is pooled from the spatial dimension to the channel dimension, computed by Global Average Pooling (GAP) and followed by 1D convolution layer $\mathbf{W}_0 \in \mathbb{R}^{B_i\times256}$ to reduce number of channels from $B_i \in \{512, 1024, 2048\}$ to 256 channels. These computation are performed as follows:

$$\mathbf{P}_i = \frac{1}{H_i \times W_i} \sum_{m}^{H_i} \sum_{n}^{W_i} \mathbf{C}_i(m, n), \tag{3}$$

$$\mathbf{F}_i = \mathbf{W}_0 \mathbf{P}_i, \tag{4}$$

The final output $\mathbf{O} \in \mathbb{R}^{256 \times \frac{H}{32} \times \frac{W}{32}}$ of the CSR module is produced by reweighting the high-level feature $\mathbf{C}_5$ with the attention activation $\mathbf{R}$ as follows:

$$\mathbf{O} = \mathbf{R} \odot \mathbf{C}_5 \tag{5}$$

where $\odot$ presents element-wise matrix multiplication. The detailed architecture of the proposed CSR module is shown in Fig. 2. Our module including several 1D convolutions is extremely lightweight and can be a plug-and-play tool to improve multi-scale feature fusion for downstream tasks.

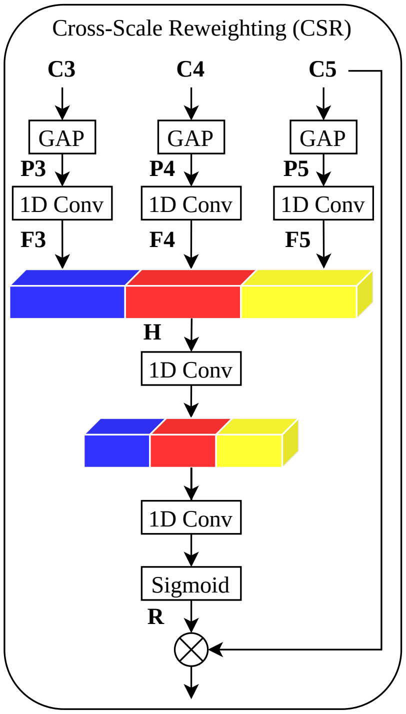

Fig. 2. The detailed architecture of the proposed Cross-Scale Reweighting (CSR) module. GAP indicates Global Average Pooling and 1D Conv means the 1D convolution layer.

B. Depth-to-Space (D2S)

Motivated by powerful upsampling operation in [22], this paper introduces an efficient Depth-to-Space (D2S) module that contains two efficient and effective operations: pixel shuffle and spatial mixing.

The pixel shuffle rearranges pixels in the features from depth dimension to spatial dimension. As a result, this operation upsamples the feature $\mathbf{C}_i$ with dimension $B_i \times H_i \times W_i$ to the feature with dimension $\frac{B_i}{r^2} \times rH_i \times rW_i$ where r is an upsampling factor, and B_i, H_i, W_i are the number of channels, height, and width of the feature $\mathbf{C}_i$ from the hierarchical backbone. During training, we set $r = 2$ for all implementations.

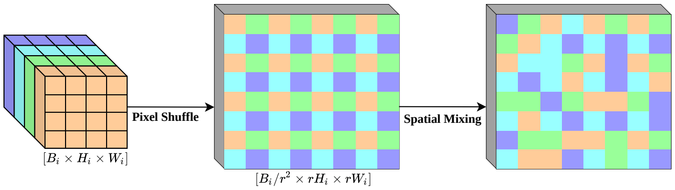
$[B_i \times H_i \times W_i]$
Pixel Shuffle
$[B_i/r^2 \times rH_i \times rW_i]$
Spatial Mixing

Fig. 3. The illustration of the CSR module. The pixel shuffle changes pixels at depth dimension to pixels at spatial dimension. Spatial mixing operation mixes spatial information in an arbitrary way, making the models learning diverse information.

Thereafter, spatial mixing exchanges the information of the rearranged feature, implemented by 3×3 depth-wise separable convolution since this operation computes the convolution along with spatial dimension. Therefore, the pixels in the rearranged feature are mixed to reason about richer semantic information. Both pixel shuffle and 3×3 depth-wise separable convolution are lightweight and can be plugged into existing dense prediction tasks. The illustration of the CSR module is shown in Fig. 3.

In this paper, the feature **C5** reweighted by the CSR module and the original feature **C4** are fused together by the D2S module. The fused feature and the low-level feature **C3** are summed to obtain the balanced semantic feature, which is critical for instance kernel predictions and mask feature learning. On each branch of the segmentation head, we stack four 3×3 convolution layers to produce instance kernel features and mask features. To generate the final masks, the 1×1 convolution layer is exploited to learn the affinity matrix between predicted instance kernels and mask features.

IV. Experiments and Results

A. Experiments

All the experiments are implemented by the Pytorch deep learning framework. The dataset used for training and evaluation is the benchmark MS-COCO [23]. This dataset includes 115k training images, 5k validation images, and 20k testing images with 80 classes. For ablation study and hyper-parameter selection, we conduct the experiment on the validation set. Since the annotation of the test set is not provided, we submit the experiential results to the evaluation server for fair comparisons. The metrics used for evaluation are Average Precision (AP), AP at different Intersection of Union (IoU) such as AP^{50} at IoU=0.5, AP^{75} at IoU=0.75, and AP with different object scales such as AP^{S} for small objects, AP^{M} for medium objects, and AP^{L} for large objects.

Followed by existing methods [5], [13], [16], [19], [20], all results are generated by using the toolbox mmdetection [24]. For detailed implementations, we use two GPU Tesla V100 to train the models for 12 epochs with a batch size of 4. The initial learning rate is set to 0.0001 and reduced by 10 at epoch 8 and epoch 11. The optimizer is the AdamW, and the weight decay is equal to 0.05 for all epochs. Following the settings

TABLE I
COMPARISON WITH STATE-OF-THE-ART INSTANCE SEGMENTATION METHODS

Method	Backbone	Learning schedule	AP^{mask}	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	#params	FPS
YOLACT [14]	ResNet-50	48	28.2	46.6	29.2	9.2	29.3	44.8	35.29	**43.5**
PolarMask [18]	ResNet-50	12	29.1	49.5	29.7	12.6	31.8	42.3	32.09	23.9
SOLO [13]	ResNet-50	12	33.1	53.5	35.0	12.2	36.1	50.8	36.08	12.7
K-Net [20]	ResNet-50	12	34.1	55.5	35.7	14.3	37	53.2	37.26	18.3
Mask R-CNN [5]	ResNet-50	12	34.7	55.7	**37.2**	**18.3**	37.4	47.2	44.17	17.5
SOLOv2 [19]	ResNet-50	12	34.8	55.2	36.8	13.6	37.9	53.5	33.89	17.7
Ours	ResNet-50	12	**35.4**	**56.8**	36.9	14.9	**38.3**	**53.8**	32.13	19.5

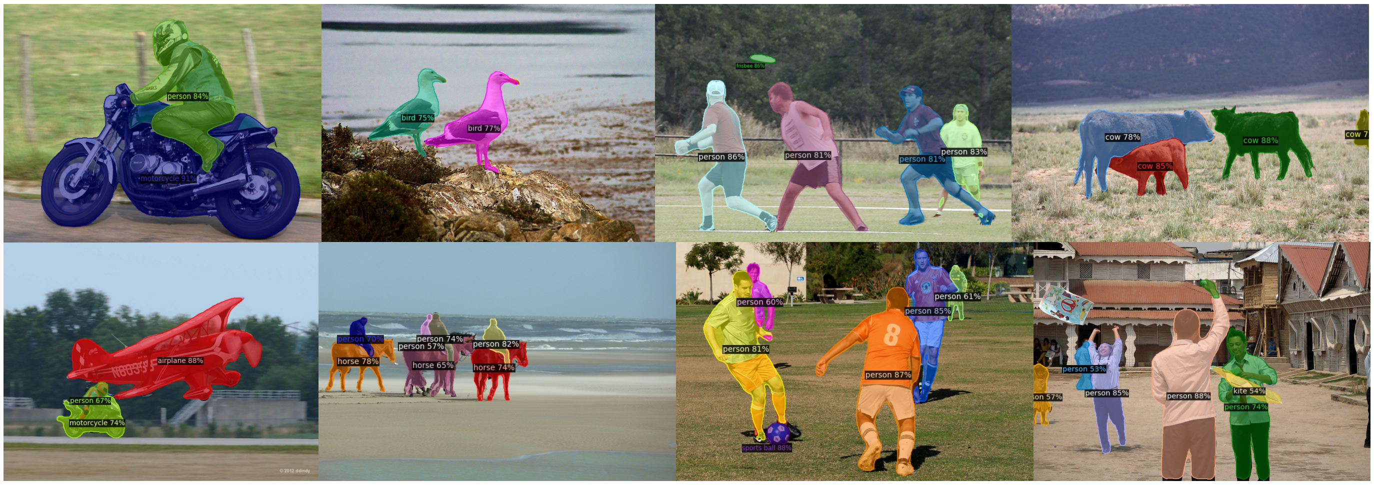

Fig. 4. The qualitative visualization of the proposed method on some cases. Each number denotes the classification score. During inference, we set the mask threshold as 0.5.

in [13], [19], [20], the loss functions are used during training, defined as follows:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{cls} + \beta\mathcal{L}_{dice} + \gamma\mathcal{L}_{focal}, \tag{6}$$

where the total loss $\mathcal{L}_{total}$ is a linear combination of three losses where α, β, γ are hard weighting factors [12]. During training, we set $\alpha = 1, \beta = 2, \gamma = 3$ for all experiments. $\mathcal{L}_{cls}$ is Cross-Entropy classification loss for learning instance kernels. $\mathcal{L}_{dice}, \mathcal{L}_{focal}$ are Dice loss and Focal loss [25] used for learning semantic masks.

B. Experimental results

1) Comparison with the state-of-the-art instance segmentation: Table I shows the comparison between our proposed method and state-of-the-art instance segmentation methods, where learning schedule means number of epochs, #params indicate number of the parameters, and FPS denotes frames per second. As a result, our method becomes the state-of-the-art method. More specifically, the proposed method achieving 35.4% AP at 19.5 FPS outperforms all the methods by a clear margin, including YOLACT [14] (28.2% AP at 43.5 FPS), PolarMask [18] (29.1% AP at 23.9 FPS), SOLO [13] (33.1% AP at 12.7 FPS), K-Net [20] (34.1% AP at 18.3 FPS), Mask R-CNN [5] (34.7% AP at 17.5 FPS), and SOLOv2 [19] (34.8% AP at 17.7 FPS). The experimental results demonstrate the effectiveness of the proposed method in both accuracy and inference speed.

TABLE II
THE EFFECT OF EACH COMPONENT

Baseline	CSR	D2S	AP	AP^{50}	AP^{75}	FPS
✓			34.1	55.5	35.7	18.3
✓	✓		34.9	56.4	36.7	19.5
✓		✓	34.8	56.3	36.4	19.5
✓	✓	✓	35.4	56.8	36.9	19.5

2) Ablation study: This subsection analyzes the importance of each component to the segmentation performance. Table II shows the experimental results of CSR and D2S modules. The baseline means the multi-level feature fusion uses FPN [7] without any modification, achieving 34.1% AP at 18.3 FPS. Adding Cross-scale Reweighting (CSR) to the top of the backbone boosts the segmentation performance by 0.8% AP while achieving higher speed than the baseline. It demonstrates the proposed CSR module is efficient and effective. When using Depth-to-Space (D2S) module as upsampling operation and spatial mixing, the model achieves 34.8% AP that bringing about 0.7% AP improvement. Finally, adding both the CSR and D2S module into multi-level feature fusion gain the performance by 1.3% AP at 19.5 FPS.

Table III shows the influence of the number of channels (#channels) in the CSR module on the segmentation performance. In this paper, we set channels in the CSR equal to 128, 245, 512, and 1024. As a result, using higher number of

TABLE III
ABLATION STUDIES ON THE NUMBER OF CHANNELS IN THE CSR MODULE

#channels	AP	AP^{50}	AP^{75}	AP^{S}	#params	FPS
128	34.9	54.7	36.2	14.8	31.63	20.4
256	35.4	56.8	36.9	14.9	32.13	19.5
512	36.1	57.2	37.1	15.1	33.34	17.6
1024	36.5	57.5	37.6	15.4	34.88	16.1

channels can bring significant improvements, but it increases number of parameters and reduces FPS scores. To achieve the better trade-off between average precision and the model complexity, this paper uses number of channels in the CSR module #channels = 256.

V. Conclusion

This paper presents an efficient multi-scale feature fusion for the instance segmentation task. The low-level features and high-level features from different scales are exchanged through the simple Cross-Scale Reweighting (CSR) module that forms large receptive fields and rich deep semantic features for dense pixel-wise prediction. The Depth-to-Space (D2S) module is proposed to upsample the high-level feature based on depth-to-space transforms and mix the changed feature to be more arbitrary and flexible. Finally, the gap between the local features and global features is alleviated by element-wise summing over the features from different scales. The experimental results verify the efficiencies of the proposed CSR and D2S module in both accuracy and inference latency. In the future, we will test the proposed CSR on other downstream tasks such as object detection, semantic segmentation, and multiple object tracking and on other large-scale datasets such as Pascal VOC and Cityscape to clarify the effectiveness of the proposed method and its generalization abilities.

Acknowledgment

This result was supported by “Region Innovation Strategy (RIS)” through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-003).

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[3] V.-T. Hoang and K.-H. Jo, “3-d human pose estimation using cascade of multiple neural networks,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2064–2072, 2018.

[4] X.-T. Vo and K.-H. Jo, “Accurate bounding box prediction for single-shot object detection,” *IEEE Transactions on Industrial Informatics*, 2021.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[6] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[8] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[9] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, “Libra r-cnn: Towards balanced learning for object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 821–830.

[10] X.-T. Vo, T.-D. Tran, D.-L. Nguyen, and K.-H. Jo, “Stair-step feature pyramid networks for object detection,” in *International Workshop on Frontiers of Computer Vision*. Springer, 2021, pp. 168–175.

[11] Z. Tan, J. Wang, X. Sun, M. Lin, H. Li *et al.*, “Giraffedet: A heavy-neck paradigm for object detection,” in *International Conference on Learning Representations*, 2021.

[12] X.-T. Vo, T.-D. Tran, D.-L. Nguyen, and K.-H. Jo, “Dynamic multi-loss weighting for multiple people tracking in video surveillance systems,” in *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*. IEEE, 2021, pp. 1–6.

[13] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “Solo: Segmenting objects by locations,” in *European Conference on Computer Vision*. Springer, 2020, pp. 649–665.

[14] 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*, 2019, pp. 9157–9166.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.

[16] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, “Hybrid task cascade for instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[17] H. Liu, R. A. R. Soto, F. Xiao, and Y. J. Lee, “Yolactedge: Real-time instance segmentation on the edge,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9579–9585.

[18] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. 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*, 2020, pp. 12 193–12 202.

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

[20] W. Zhang, J. Pang, K. Chen, and C. C. Loy, “K-net: Towards unified image segmentation,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[23] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[24] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, “Mmdetection: Open mmlab detection toolbox and benchmark,” *arXiv preprint arXiv:1906.07155*, 2019.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.