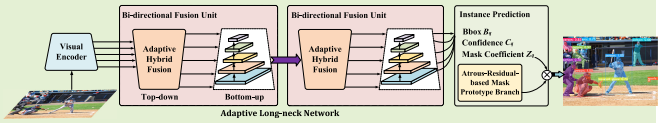
Adaptive Long-Neck Network With Atrous-Residual Structure for Instance Segmentation

Wenjie Geng, Zhiqiang Cao, *Senior Member, IEEE*, Peiyu Guan, Guangli Ren, Junzhi Yu, *Fellow, IEEE*, and Fengshui Jing

***Abstract*—Instance segmentation is an important yet challenging task in the computer vision field. Existing mainstream single-stage solution with parameterized mask representation has designed the neck models to fuse features of different layers; however, the performance of instance segmentation is still restricted to the layer-by-layer transmission scheme. In this article, an instance segmentation framework with an adaptive long-neck (ALN) network and atrous-residual structure is proposed. The long-neck network is composed of two bidirectional fusion units, which are cascaded to facilitate the information communication among features of different layers in top-down and bottom-up pathways. In particular, a new cross-layer transmission scheme is introduced in a top-down pathway to achieve a hybrid dense fusion of multiscale features and weights of different features are learned adaptively according to their respective contributions to promote the network convergence. Meanwhile, a bottom-up pathway further complements the features with more location clues. In this way, high-level semantic information and low-level location information are tightly integrated. Furthermore, an atrous-residual structure is added to the mask prototype branch of instance prediction to capture more contextual information. This contributes to the generation of high-quality masks. The experimental results indicate that the proposed method achieves effective segmentation and the outputted masks match the contours of objects.**



***Index Terms*— Adaptive long-neck (ALN) network, atrous-residual structure, instance segmentation.**

I. Introduction

INSTANCE segmentation aims to predict the object locations and pixel-wise instance masks in the image from a visual sensor. It is an important task in the computer vision field with wide applications in robot manipulation [1] and autonomous driving [2], [3].

Benefiting from the development of deep learning, the research of instance segmentation has made substantial progress. Existing deep learning-based solutions are generally categorized into two- and single-stage methods. The former follows the procedure of detect-then-segment, which first executes detection to generate a set of regions of interest (ROIs) and then utilizes the features from these regions to calculate the mask of each instance [4], [5], [6], [7], [8], [9]. The object detection result possibly affects the mask quality with a longer processing time. In contrast, the single-stage scheme directly obtains the pixel-wise instance masks without relying on detection. It is subdivided into two types: mask regression and parameterized mask representation. The first one directly regresses instance masks by a classification layer [10], [11], [12], where an object-related image patch instead of whole image is used during training. The second type predicts the mask-related parameters, which are then assembled for the final masks [13], [14], [15]. It takes the whole image from a visual sensor as input, which becomes gradually prevailing as more global context information can be captured. This is beneficial to remove the interference from background and other instances of the same class.

After the input image provided by a vision sensor is processed by a basic backbone, features form different layers are obtained. The features in higher layer are rich in semantics, while those in lower layer may provide the

Manuscript received 10 January 2023; accepted 3 February 2023. Date of publication 17 February 2023; date of current version 31 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62073322 and Grant 61633020, in part by the CIE-Tencent Robotics X Rhino-Bird Focused Research Program under Grant 2022-07, and in part by the Beijing Natural Science Foundation under Grant 2022MQ05. The associate editor coordinating the review of this article and approving it for publication was Prof. Xiaofeng Yuan. *(Wenjie Geng and Peiyu Guan contributed equally to this work.) (Corresponding author: Zhiqiang Cao.)*

Wenjie Geng, Zhiqiang Cao, Peiyu Guan, Guangli Ren, and Fengshui Jing are with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhiqiang.cao@ia.ac.cn).

Junzhi Yu is with the Department of Advanced Manufacturing and Robotics, BIC-ESAT, College of Engineering, Peking University, Beijing 100871, China.

Digital Object Identifier 10.1109/JSEN.2023.3244818

detailed location clues. Researchers have designed the neck models to fuse features of different layers. A typical neck model is feature pyramid network (FPN) [16]. Xie et al. [14] and Bolya et al. [15] introduced it to their single-stage instance segmentation methods. However, they mainly concern one-way top-down information propagation between two neighboring layers. This leads to that the detailed information from the lower layers cannot be effectively fed back to the high ones. To deal with this problem, Liu et al. [7] made a good attempt in two-stage instance segmentation method path aggregation network (PANet) by adding bottom-up propagation in the neck model. It is worth mentioning that existing instance segmentation methods have regarded the neck model as an effective means for performance improvement; however, the features in a layer are mainly passed to its neighboring layer [7], [14], [15]. Such a layer-by-layer transmission scheme inadequately fuses features from different scales. If more cross-layer connections are involved, features of different layers can be better fused, and an advanced neck model is expected.

Another noteworthy aspect is parameterized mask representation based on the fused features provided by the neck model. Compared to the octagon mask [13] and polar mask [14], the mask with a prototype branch and corresponding coefficients [15] is better, especially in efficiency. A problem of the mask prototype branch is that it is insufficient to capture contextual information, and thus, the local and global information from different receptive fields cannot be fully parsed. This shall increase the possibility of misclassification and even leads to mask leakage where the mask of an instance leaks to another one in the same class. In this case, contextual relationship in the domain of complex scene understanding [17], [18], [19] provides a promising improvement direction.

To solve the problem of insufficient feature fusion in existing neck models, we construct a novel ALN network with two bidirectional fusion units to facilitate the information communication among features of different layers in top-down and bottom-up pathways. In the top-down pathway, multiscale feature maps are fused by cross-layer connections, while the bottom-up one further complements the features with more location clues. Moreover, weights of different feature maps are learned in an adaptive way according to their respective contributions to promote the network convergence. As a result, the high-level semantic information and low-level location information are tightly integrated. For the mask prototype branch, atrous convolution with residual connection (ACRC) is introduced to capture contextual information. Therefore, instance masks with high quality are attained. The main contributions of this article are given as follows.

1) An ALN network with atrous-residual structure is proposed for instance segmentation, which achieves good accuracy and efficiency.
2) Two bidirectional fusion units are cascaded to create an ALN network, which achieves a tight fusion of features from different layers. Multiscale features are adequately blended by cross-layer connections along the top-down pathway; meanwhile, local location information is supplied to high levels in the bottom-up pathway. Besides, we refer to the object detection model EfficientDet in [20] and omit the connections between the topmost layer and other layers to simplify the network.
3) An atrous-residual-based mask prototype branch is constructed in the instance prediction to capture contextual content for clear mask.
4) The experiments on the MS COCO dataset, including the robustness test under different interferences, prove the effectiveness of the proposed instance segmentation method.

The remainder of this article is organized as follows. The related work is reviewed in Section II. Section III describes the proposed method in detail. The experiments are provided in Section IV, and Section V concludes this article.

II. Related Work

A. Two-Stage Instance Segmentation

The two-stage methods execute instance mask prediction depending on the detected object regions, where the regions can be represented in the form of bounding boxes. Dai et al. [5] proposed multitask network cascades (MNCs) for instance-aware semantic segmentation. It first extracts shared convolutional feature maps from the whole input image, which is followed by a region proposal network (RPN) to regress the bounding boxes of objects in the image. Then, the features from the bounding boxes in the feature maps are extracted to predict the pixel-level segmentation mask and category scores for each corresponding instance. A fully convolutional instance-aware semantic segmentation method (FCIS) is presented in [4], which produces the ROIs by RPN and predicts the position-sensitive inside/outside score maps at the same time. Then, the score maps within each ROI are assembled to generate instance masks and categories of objects. Mask region based convolutional neural network (R-CNN) [6] adds a new mask branch on Faster R-CNN to predict segmentation masks on each ROI generated from RPN. By combining the original branches of classification and bounding box regression, the positions and segmentation masks of objects are acquired. To promote the information propagation in Mask R-CNN, PANet [7] appends a bottom-up path augmentation after the FPN backbone of Mask R-CNN for better instance segmentation. Without RPN, the mask branch can be appended after the object detector for instance segmentation. RetinaMask [8] adds an instance mask prediction head after the RetinaNet detector [21], where the feature maps corresponding to bounding boxes detected by RetinaNet are sent to the mask prediction head to obtain the instance masks. Besides, an adaptive loss is introduced to improve robustness during training. CenterMask [9] adds a spatial attention-guided mask branch after fully convolutional one-stage (FCOS) detector [22] to focus on meaningful pixels and suppress uninformative ones in the image. This mask branch takes the features in the bounding boxes generated from FCOS as inputs and predicts the segmentation masks inside each detected box with the help of spatial attention module. A problem of the two-stage procedure is that it is usually time-consuming. Also, the mask quality is unsatisfactory.

B. Single-Stage Instance Segmentation

Earlier research directly regresses the pixel-wise instance mask based on an input patch [10], [11], where the procedure of mask prediction is similar to that of semantic segmentation [23], [24]. DeepMask (DM) [10] predicts the segmentation mask in the top branch and object score that describes how likely the input patch contains an object in the bottom branch. Since the positions of instances need to be distinguished, DM requires that the input patch is object-centered and it fully contains this object during training. SharpMask (SM) [11] further takes features from multiple layers into account on DM and the quality of mask is improved. Still, the training process is complicated due to the tedious training patches. Recently, researchers explore parameterized representations of instance masks [13], [14], [15]. ExtremeNet [13] represents an object with five extreme-point parameters, including topmost, leftmost, bottommost, rightmost, and center points. It adopts a keypoint estimation framework to directly predict these five parameters, with which an octagon mask is generated to segment the object instances in the image. Octagon masks can only provide a rough outline, which cannot accurately segment the pixel boundaries of an object. PolarMask [14] transforms the instance segmentation problem into instance contour prediction, which applies task-specific heads after the feature extraction to classify the pixel corresponding to mass center of instance and regress the dense distances of rays between mass center and contours. Nevertheless, with the increase of number of points located on the contour, the efficiency will be affected. Bolya et al. [15] proposed a real-time instance segmentation method YOLACT to calculate instance masks by combining the predicted parameters of mask prototypes and the corresponding coefficients in the detection branch. However, it adopts the layer-by-layer transmission mechanism, which cannot fully combine the high-level semantic information and the low-layer location information from different layers. This decreases the representation ability of features extracted from its backbone with ResNet [25] and the FPN neck model and thus affects the segmentation accuracy and the mask quality. In this article, a new ALN model is proposed and it employs cross-layer transmission with dense connections to effectively handle multiscale information.

III. Instance Segmentation With ALN Network and Atrous-Residual Structure

Fig. 1 shows the proposed instance segmentation network, which is composed of a visual encoder, an ALN network, and instance prediction. The visual decoder is used to provide multilevel feature maps with different scales. On this basis, the ALN network with bidirectional fusion units is responsible for feature fusion from high and low levels. This enhances the discrimination of features, which are fed into instance prediction heads for the detection results and instance masks.

The pipeline of the proposed method is given as follows. The input is first processed by the visual encoder and the multilevel feature maps denoted with P_{0i}, $i = 3, 4, \ldots, 7$ are acquired. They are sent to the first bidirectional fusion unit of ALN network. The first fusion unit employs dense cross-layer connections, and high-level semantic information and low-level location information are adaptively fused in a top-down way. The outputted feature maps are then processed by a bottom-up augmentation pathway to provide feature maps P_{1i}, $i = 3, 4, \ldots, 7$. The results of the first unit are further refined by the second fusion unit with the same structure, and we have feature maps P_{2i}, $i = 3, 4, \ldots, 7$ whose sizes are consistent with P_{1i}. Taking the input red, green, and blue (RGB) image with the size $550 \times 550 \times 3$ as an example, the sizes and channel numbers of feature maps are shown in Fig. 1. In the instance prediction heads, each outputted feature map of ALN network is operated with a convolution layer to predict the bounding boxes B of objects, confidence C that the detected objects belong to a class, and mask coefficients Z. Also, the mask prototype branch with an atrous-residual structure takes feature maps P_{23}, P_{24}, and P_{25} as inputs to capture different contextual content for better masks prediction.

A. Visual Encoder

The visual encoder is adopted to extract features of the input RGB image from a vision sensor. Compared to visual geometry group (VGG) [26] encoder in the instance segmentation method DM [10], ResNet [25] in the SM [11] shows an advantage in feature extraction as it builds deeper convolution layers and overcomes the gradient explosion and vanishing problems. Here, we directly employ the ResNet structure as the backbone in our visual encoder, which is initialized with the weights pretrained on the ImageNet dataset [27]. As shown in Fig. 1, the feature maps generated from each layer in ResNet are denoted as P_{01}, P_{02}, P_{03}, P_{04}, and P_{05}. In order to enrich higher level semantic information, the feature map P_{05} is successively downsampled to generate feature maps P_{06} and P_{07}. We regard P_{03}, P_{04}, P_{05}, P_{06}, and P_{07} as the outputs of the encoder.

B. ALN Network

This network aims to enhance the features from the visual encoder and it is composed of two bidirectional fusion units with the same architecture. These two units are cascaded to achieve the fusion of semantic information from high levels and location information from low levels. Inside each unit, the adaptive hybrid fusion (AHF) is implemented in a top-down manner, which is followed by a bottom-up pathway to further augment the details of features.

1) Adaptive Hybrid Fusion: The structure of AHF is shown in Fig. 2. It takes P_{r3}^{in}, P_{r4}^{in}, P_{r5}^{in}, P_{r6}^{in}, and P_{r7}^{in} as inputs, where these five feature maps gradually decrease in spatial size from low level to high level, and r represents the index of the bidirectional fusion unit. Each level receives the feature maps from the current layer and those from other layers. To achieve the fusion of current layer, a hybrid scheme is designed. For layers above and below current layer, we, respectively, choose the fused results and input feature maps. In this way, more semantic and location information is involved. As feature maps in different layers vary in dimensions, including size

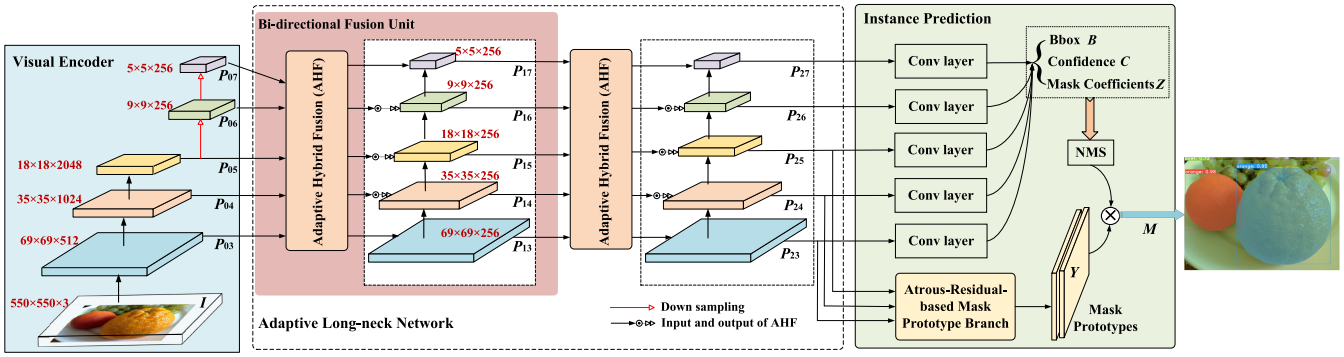


Fig. 1. Structure of the proposed instance segmentation network, which consists of a visual encoder, an ALN network, and instance prediction. The input is first processed by the visual encoder to output multilevel feature maps. These feature maps P_{0i}, $i = 3, 4, \ldots, 7$ are fused by the long-neck network with two cascaded bidirectional fusion units to output the fused feature maps P_{2i}. In each bidirectional fusion unit, the input feature maps are sent to the AHF, and by dense cross-layer connections, high-level semantic information and low-level location information are fused in a top-down way. In addition, a bottom-up augmentation pathway with accurate location signals in lower layers is also added to further enhance the feature hierarchy. Afterward, the fused feature maps are processed by instance prediction heads, which insert an atrous-residual structure into the mask prototype branch to capture different contextual contents. Finally, Bbox B, corresponding confidence C, and mask M are obtained. M is calculated by matrix multiplication between the mask prototypes Y and Z_f, where Z_f is the result after NMS is operated on the mask coefficient Z.

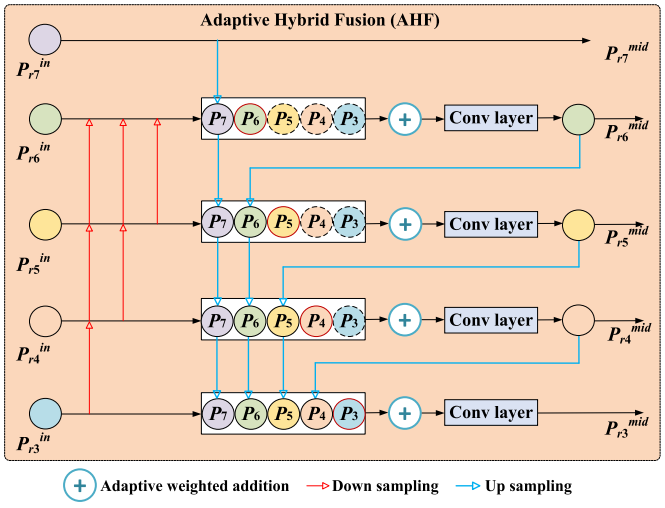


Fig. 2. Structure of the AHF.

and channel number, they have to be adjusted to adapt to the current fusion layer. The resultant feature maps P_3, P_4, P_5, P_6, and P_7 are then fused by weighted addition, where each weight is adaptively learned. Afterward, the corresponding convolutional layer is imposed and we get the fused results P_{r7}^{mid}, P_{r6}^{mid}, P_{r5}^{mid}, P_{r4}^{mid}, and P_{r3}^{mid} by a top-down way.

Taking the generation of P_{r5}^{mid} as an example, according to the aforementioned description, we choose the upper layer features P_{r6}^{mid} and P_{r7}^{mid}, the lower layer features P_{r3}^{in} and P_{r4}^{in}, and the current one P_{r5}^{in}, where the former four features should follow the dimension of P_{r5}^{in} by upsampling or downsampling when fusing. The resultant feature maps are operated by weighted addition and convolution layer to obtain P_{r5}^{mid}. Note that the hybrid fusion does not occur in the topmost layer, which means that the output P_{r7}^{mid} is the same as its input P_{r7}^{in}.

The output feature maps P_{rq}^{mid} can be formulated as follows:

$$P_{rq}^{\text{mid}} = \text{Conv}_3\left(\sum_{j=3}^{7} W_{rq}^{j} P_j\right), \quad q = 3, 4, 5, 6 \tag{1}$$

where $W_{rq}^{j} = (w_{rq}^{j}/(\delta + \sum_d w_{rq}^{d}))$ and w_{rq}^{j} is the learned weight, $j = 3, 4, \ldots, 7$. $\text{Conv}_3(\cdot)$ refers to a 3×3 convolution with batch normalization.

2) Bottom-Up Augmentation Pathway: This component receives the feature maps from P_{r3}^{mid} to P_{r7}^{mid} and fuses them gradually along a bottom-up direction, which is beneficial to supplement more local location information to high levels. Each layer receives the dimension-adjusted feature map from its lower layer and its corresponding AHF output for further fusion along a bottom-up direction. For the lowest layer, only its AHF output is involved. Still, this fusion employs adaptive weighting due to the fact that different features contribute differently to the fusion result. It is worth mentioning that for each of the middle three layers, the input of corresponding AHF layer is involved in enriching information.

Take P_{r4} for illustration. The input feature map P_{r4}^{in}, the output feature map P_{r4}^{mid} of corresponding AHF, and the downsampling feature map of P_{r3} are combined by adaptive weighting addition, and P_{r4} is then obtained. In particular, the generation of P_{r7} does not consider the input feature map of corresponding AHF and P_{r3} is in coincidence with the output feature map of AHF. Finally, the results P_{r3}, P_{r4}, P_{r5}, P_{r6}, and P_{r7} of a bidirectional fusion unit are obtained as follows:

$$P_{rq} = \sum_{b=0}^{\text{len}(T_{rq})-1} S_{rq}^{b} T_{rq}^{b} \tag{2}$$

$$T_{rq} = \begin{cases} \left\{P_{rq}^{\text{mid}}\right\}, & q = 3 \\ \left\{P_{r(q-1)}, P_{rq}^{\text{mid}}, P_{rq}^{\text{in}}\right\}, & 3 < q < 7 \\ \left\{P_{r(q-1)}, P_{rq}^{\text{mid}}\right\}, & q = 7 \end{cases} \tag{3}$$

where T_{rq} denotes the set of feature maps to be fused at the qth layer, $S_{rq}^{b} = (s_{rq}^{b}/(\delta + \sum_{e=0}^{\text{len}(T_{rq})-1} s_{rq}^{e}))$, and s_{rq}^{b} is the learned weight for the bth feature map T_{rq}^{b} in T_{rq}.

After two bidirectional fusion units are applied in sequence, we acquire the output feature maps P_{2i} of the long-neck network, $i = 3, 4, \ldots, 7$.

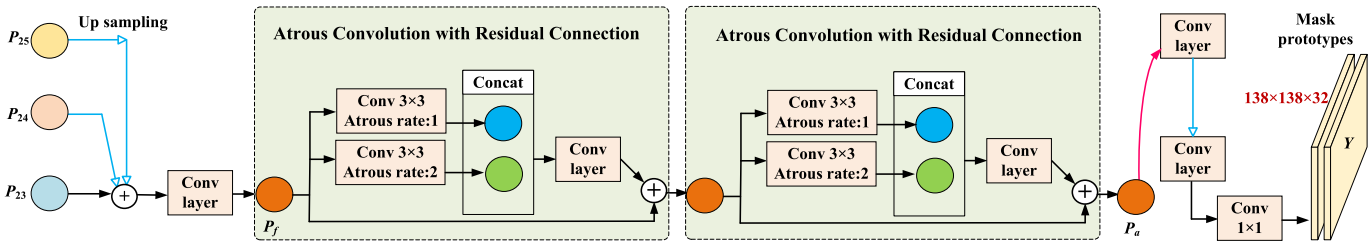


Fig. 3. Structure of the mask prototype branch. The feature maps P_{23}, P_{24}, and P_{25} from the ALN network are processed by summation and convolution to acquire the feature map P_f, which will go through two cascaded ACRC units to capture contextual information and output feature map P_a. After that, convolutions are imposed on P_a for the generation of mask prototypes.

C. Instance Prediction

The instance prediction endeavors to predict the pixel-wise instance mask of an image. Following the procedure of YOLACT, it consists of two branches: object detection and mask prototype branches. The former predicts the object detection results, including object bounding boxes B, confidence C related to object categories, and mask coefficients Z. The latter outputs the mask prototypes Y, which is combined with mask coefficients to generate the final segmentation mask for each instance.

1) Mask Prototype Branch: The structure of this branch is shown in Fig. 3. Different from YOLACT that only uses the bottom feature map as input, we choose the lower three feature maps P_{23}, P_{24}, and P_{25} as inputs. The deeper feature maps P_{26} and P_{27} are omitted as their information is severely diluted due to upsampling and thus have little impact on the mask prototypes Y. Specifically, the input feature maps P_{24} and P_{25} are first upsampled to the same size as P_{23}. After they are added, a convolution layer with 3×3 kernel is applied to capture local neighboring features and the feature map P_f is acquired. In order to perceive more contextual information, we introduce ACRC. Two ACRC units are cascaded to process P_f. In each ACRC unit, the outputs of two parallel atrous convolutions with atrous rates of 1 and 2 are concatenated in channel, which is followed by a convolution layer to facilitate the information associated with the channel reduction. Finally, a residual connection is exerted with the consideration of the input of the ACRC unit for the convergence of the mask prototype branch and preventing gradient explosion. We label the output of the second ACRC unit as the feature map P_a, which is further processed by convolutions to generate the final mask prototypes Y. With the assistance of ACRC module, the prototype generation branch improves its ability to distinguish the boundaries of objects and improve mask quality.

2) Instance Mask Generation: As shown in Fig. 1, the mask coefficient Z is processed by nonmaximum suppression (NMS) to screen reliable instances. The filtered mask coefficient Z_f is executed a matrix multiplication with the mask prototypes Y, which is followed by a sigmoid activation function and the instance mask M is generated

$$M = \text{sig}\left(YZ_f^T\right) \tag{4}$$

where $\text{sig}(\cdot)$ represents the sigmoid activation function. $Y \in \mathbb{R}^{h_m \times w_m \times n_m}$, $h_m = H/8$, $w_m = W/8$, and H and W denote the height and width of the input image, respectively. n_m is the number of mask prototypes. $Z_f \in \mathbb{R}^{n \times n_m}$, and n denotes the number of instances surviving NMS.

Algorithm 1 describes the proposed instance segmentation algorithm based on the ALN network with atrous-residual structure, where N_r is the number of bidirectional fusion units. $f_{\text{Encoder}}(\cdot)$ refers to the encoding operation on the input image. $f_{\text{AHF}}(\cdot)$ and $f_{\text{BU}}(\cdot)$ describe the processing of AHF and bottom-up augmentation, respectively. $f_{\text{Pre}}(\cdot)$ produces the detection result and mask coefficients. $f_{\text{NMS}}(\cdot)$ represents NMS and $f_{\text{ACR}}^{\text{Pro}}(\cdot)$ is used to generate mask prototypes of objects.

Algorithm 1 Instance Segmentation Based on Adaptive Long-Neck Network With Atrous-Residual Structure

Input: input image I.
Output: Bbox B, confidence C, and mask M.
$P_{0i} = f_{\text{Encoder}}(I)$, $i = 3, 4, \ldots, 7$;
for $r \in \{1, \ldots, N_r\}$ **do**
  $\{P_{\text{ri}}^{\text{mid}}\}_{i=3}^{7} = f_{\text{AHF}}(\{P_{(r-1)i}\}_{i=3}^{7})$;
  $\{P_{ri}\}_{i=3}^{7} = f_{\text{BU}}(\{P_{\text{ri}}^{\text{mid}}\}_{i=3}^{7})$;
end for
$(B, C, Z) = f_{\text{Pre}}(\{P_{N_r i}\}_{i=3}^{7})$;
$Z_f = f_{\text{NMS}}(Z)$;
$Y = f_{\text{ACR}}^{\text{Pro}}(P_{N_r 3}, P_{N_r 4}, P_{N_r 5})$;
$M = \text{sig}(YZ_f^T)$;
return B, C, M.

D. Loss Function

The loss function to train the model is composed of three parts: the location loss L_{loc} of B, the classification confidence loss L_{conf}, and the mask loss L_{mask}

$$L(b, c, l, g) = \alpha L_{\text{conf}} + \beta L_{\text{loc}} + \gamma L_{\text{mask}} \tag{5}$$

where b represents the selected valid boxes using the condition in [28]. α, β, and γ are hyperparameters, which are set to 1, 1.5, and 5.5, respectively. c, l, and g represent classification confidence, the positions of the predicted boxes, and the ground-truth boxes, respectively.

The calculation of L_{loc} is given as follows [28]:

$$L_{\text{loc}} = \sum_{it,jt \in \text{Pos}}^{N_{db}} \sum_{m \in \{cx,cy,w,h\}} v_{it,jt}^{p} \text{smooth}_{L1}\left(l_{it}^{m} - \hat{g}_{jt}^{m}\right) \tag{6}$$

TABLE I
COMPARISON OF DIFFERENT VARIANTS OF OUR ALNMASK METHOD ON THE COCO VAL2017 DATASET IN TERMS OF AP (%)

Method	FPN	Adaptive Long-neck	Original mask prototype branch	ACRC-based mask prototype branch with only the bottom feature map	ACRC-based mask prototype branch	aspect ratios		AP
						[1, 1/2, 2]	[1/4, 4/5, 2]	
ALNMask-I	√	–	√	–	–	√	–	24.9
ALNMask-II	√	–	–	√	–	√	–	25.7
ALNMask-III	√	–	–	–	√	√	–	25.9
ALNMask-IV	–	√	√	–	–	√	–	25.8
ALNMask-V	–	√	–	–	√	√	–	26.1
ALNMask	–	√	–	–	√	–	√	**26.6**

where N_{db} is the number of matched default boxes, l_{it}^{m} refers to the predicted box, and $\hat{g}_{jt}^{m}$ represents the intermediate deformation variables derived from the ground-truth box and default box. $v_{it,jt}^{p}$ is a Boolean variable to indicate whether the itth default box and the jtth ground-truth box are matchable. Pos denotes the positive samples. cx, cy, w, and h are x-coordinate and y-coordinate of the center, the width, and height of the box, respectively.

The classification confidence loss is given as follows [28]:

$$L_{\text{conf}} = -\sum_{it\in\text{Pos}}^{N_{db}} v_{it,jt}^{p}\log\left(\hat{c}_{it}^{p}\right) - \sum_{ig\in\text{Neg}}\log\left(\hat{c}_{ig}^{0}\right) \tag{7}$$

where $\hat{c}_{it}^{p} = (\exp(c_{it}^{p})/\sum_{p}\exp(c_{it}^{p}))$, Neg refers to negative samples, and $\hat{c}_{ig}^{0}$ denotes the confidence loss for a negative sample.

For the mask loss L_{mask}, it is expressed as a binary cross-entropy loss between the predict mask M and the corresponding ground-truth mask M_g [15]

$$L_{\text{mask}} = E_{\text{BC}}\left(M, M_g\right) \tag{8}$$

where $E_{\text{BC}}(\cdot)$ denotes the pixel-wise binary cross entropy.

IV. EXPERIMENTS

A. Experimental Setup

In this section, extensive experiments are executed to validate the effectiveness of the proposed ALNMask on the public dataset MS COCO [29]. This dataset includes 80 object categories. The whole MS COCO dataset can be split into three parts: train2017, test-dev2017, and val2017 with 118k, 40k, and 5k images, respectively. Here, we train our model on the train2017 dataset and evaluate it on test-dev2017 and val2017 datasets. The evaluation criteria consider the average precision (AP) with the intersection over union (IOU) thresholds 50%, 75%, and 95% between the predicted mask and the ground truth, which are denoted as AP_{50}, AP_{75}, and AP_{95}, respectively. Also, the mean AP for all the objects is adopted. Moreover, to compare the instance segmentation results for different object sizes, all the objects in the dataset are divided into three categories: small, middle, and large. Accordingly, the mean APs are denoted as AP_S, AP_M, and AP_L.

During training, the stochastic gradient descent (SGD) optimization is used with the weight decay and momentum of 5×10^{-4} and 0.9, respectively. The initial learning rate is 0.8, and it is set to 0.5 after 200k iterations. Data augmentation is imposed on the training data by changing the brightness, contrast, and size of images. Our method runs on a platform with NVIDIA GTX1080 GPU with 8-GB memory and Intel Core i7-7770HQ CPU. In addition, the anchors with scales [32, 64, 128, 256, 512] are applied to the feature maps P_{23}–P_{27}, respectively. These scales can be combined with aspect ratios to generate different forms of anchors for better object detection, where aspect ratio refers to the ratio between the width and height of an anchor. The more accurate the setting of aspect ratios is, the better the network is trained. According to the size relationship of width and height of objects, three cases of anchors are concerned. When the object width is shorter than its height, its ratio is typically less than 1/2 and the aspect ratio is set to 1/4. For the case that the width and height of object are close, the aspect ratio is set to 4/5 as they are hard to be strictly equal. Besides, we set the third aspect ratio to 2 for the case where widths of some objects are larger than heights. In summary, three aspect ratios [1/4, 4/5, 2] of anchors are chosen.

B. Ablation Evaluation

To verify the performance of our proposed ALNMask method, its five variants are involved according to whether the FPN architecture [16], ALN network, original mask prototype branch [15], and ACRC-based mask prototype branch are considered. Besides, two parameter settings for aspect ratios are also involved: [1, 1/2, 2] [15] and [1/4, 4/5, 2]. Table I presents the comparison results of different variants on the val2017 dataset in terms of accuracy AP. Comparing ALNMask-I with ALNMask-II, one can see that the ACRC-based mask prototype branch with only the bottom feature map P_{23} is better than the original mask prototype branch in accuracy. Also, our ACRC-based mask prototype branch with three inputted feature maps P_{23}, P_{24}, and P_{25} further promotes the performance, which can be seen from the result of ALNMask-III. For ALNMask-IV with an ALN network, it achieves a higher performance than ALNMask-I, which indicates that our proposed ALN outperforms the FPN architecture. With the combination of ALN and ACRC-based mask prototype branch, the accuracy is further improved. Based on ALNMask-V, the aspect ratios are changed from [1, 1/2, 2] to [1/4, 4/5, 2] and the performance reaches the best (see the result of ALNMask).

C. Comparison With Existing Methods

In this section, the proposed method ALNMask is first compared with existing instance segmentation methods on

TABLE II
COMPARISON OF DIFFERENT METHODS ON THE COCO TEST-DEV2017 DATASET

Method	Backbone	Size	FPS	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
MNCS [5]	R-101	600 (shorter side)	0.73 (NVIDIA K40 GPU)	24.6	44.3	—	4.7	25.9	43.6
FCIS [4]	R-101-C5	600 (shorter side)	4.17 (NVIDIA K40 GPU)	29.2	49.5	—	7.1	31.3	50.0
Mask R-CNN [6]	R-101-FPN	[640, 800]	5.0 (NVIDIA Tesla M40 GPU)	35.7	58.0	37.8	15.5	38.1	52.4
DM+MPN [12]	VGG	800×800	—	25.0	45.4	24.5	7.2	28.8	39.0
SM+MPN [11]	VGG	800×800	—	25.1	45.8	24.8	7.4	29.2	39.1
PolarMask° [14]	R-101-FPN	[640, 800]	—	30.4	51.9	31.0	13.4	32.4	42.8
YOLACT [15]	R-101-FPN	550×550	24.3* (NVIDIA GTX1080 GPU)	29.8	48.5	31.2	9.9	31.3	47.7
ALNMask	R-101-ALN	550×550	17.2 (NVIDIA GTX1080 GPU)	30.9	49.7	32.4	10.0	32.4	50.3

* denotes the result on our NVIDIA GTX1080 GPU platform. ° means that data augmentation is not involved.
The results of MNCS are from [4].

the COCO test-dev2017 dataset. These methods include MNCS [5], FCIS [4], Mask R-CNN [6], DM with MultiPath network (DM + MPN) [12], SM with MPN (SM + MPN) [11], PolarMask [14], and YOLACT [15], where the first three methods belong to the two-stage solution and others are single-stage methods. Table II presents the accuracy and efficiency results of different methods on the COCO test-dev2017 dataset. Also, the backbones and image sizes of these methods are provided. In terms of AP, two-stage Mask R-CNN performs the best, and single-stage methods, including PolarMask, ALNMask, and YOLACT, attain good results. From the processing speed, YOLACT runs at a speed up to 24.3 frames/s and ALNMask runs at 17.2 frames/s. Overall, our method achieves the balance of accuracy and efficiency.

Notice that the AP gap reflects the difference of mask matching between the predicted mask and ground truth for two methods under the same IOU threshold. We observe from Table II that the AP gap between Mask R-CNN and our ALNMask at the 50% IOU threshold is 8.3%, while the AP gap is reduced to 5.4% at the 75% IOU threshold. This inspires us to further compare the AP gap at the 95% IOU threshold. The AP_{95} of ALNMask is 1.91%, while the values of Mask R-CNN and YOLACT are 1.3% and 1.6%, respectively. This brings in a negative AP gap at the 95% IOU threshold. Actually, AP_{95} is an important criterion to evaluate the mask quality of the predicted instances. The higher AP_{95} is, the closer the predicted mask and ground truth are. It is indicated that our ALNMask is helpful for good masks.

Fig. 4 visualizes the instance mask results of different methods, including ALNMask, YOLACT, and Mask R-CNN on the eight selected images of the test-dev2017 dataset. For each image, a local area is enlarged to illustrate the mask detail. Considering that Mask R-CNN mainly releases the office weights [30] with the backbone of R-50-FPN, ALNMask and YOLACT also adopt the same ResNet structure for a fair comparison. Generally, the flaws of masks may be divided into the following types: discontinuity, coarse edge, background blending, mask leakage, and incompleteness. Mask discontinuity means that the mask of an object is disconnected, such as the masks of cat and banana segmented by YOLACT in Fig. 4(a) and (h), respectively. The coarse edge corresponds to the low mask boundary of an object. This phenomenon occurs in the cow mask provided by YOLACT in Fig. 4(b), giraffe mask of Mask R-CNN in Fig. 4(d), and cup mask from YOLACT in Fig. 4(f). The third flaw is the background blending and in this case background appears on the mask of foreground object. Please see the boat masks of ALNmask, YOLACT, and Mask R-CNN in Fig. 4(c), and dining table mask provided by ALNmask in Fig. 4(f). Another defect is mask leakage, which denotes that the mask of an instance leaks to another one in the same class. It is found in the giraffe mask of YOLACT in Fig. 4(d). There is also another situation called mask incompleteness, where only local mask instead of the whole one of an object is outputted. The broccoli and banana masks given by Mask R-CNN in Fig. 4(g) and (h) are examples. On the whole, our ALNMask offers masks of objects with good boundary and matching with ground truth.

It is important to note that Mask R-CNN predicts bounding boxes whose number is usually more than objects number due to plentiful proposals provided by the detector, especially for large objects. This may be lead to generate several local masks for an object. On one hand, this processing improves the accuracy; however, on the other hand, the mask quality is reduced and one local mask of an object often overlaps with other local ones or the global one. The probability of false detection is also increased. Take the mask results in Fig. 4(a) as an example. For this image with two cats, Mask R-CNN outputs three cat masks, where the result of a leg is discontinuous. In this case, YOLACT acquires the correct masks with two cats, and there still exists the discontinuity of leg segmentation. By contrast, the proposed ALNMask obtains effective masks. It is also observed that the number of masks generated by Mask R-CNN is generally larger. Overall, the evaluation of an instance segmentation method needs to consider both the accuracy and the mask quality.

Table III provides the comparison of different methods on the COCO val2017 dataset, where Mask R-CNN, CenterMask [9], ExtremeNet [13], PolarMask [14], and YOLACT [15] are involved. The first two methods are two-stage methods and the rest are categorized to single-stage solution. To better compare ALNMask and YOLACT, ALNMask-RL, ALNMask-IS, YOLACT-RL, and YOLACT-IS are considered according to different backbones and image sizes. Also, ALNMask-L with large image size [800, 1333] is concerned. The results indicate that our method performs well. Compared with YOLACT, ALNMask with the same backbone and image size achieves better instance segmentation accuracy

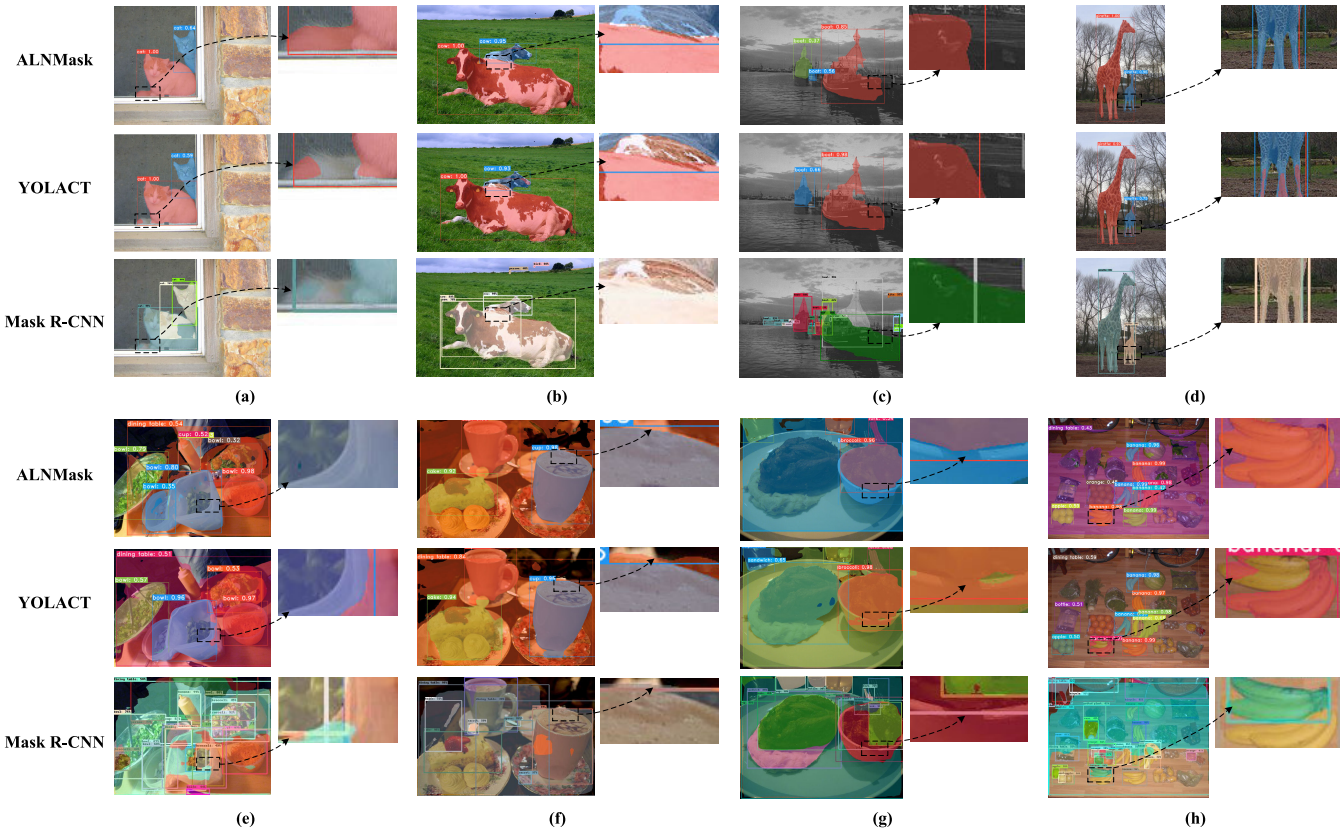


Fig. 4. Comparison of instance mask quality of different methods (R-50-FPN) for eight selected images from the COCO test-dev2017 dataset.

TABLE III
COMPARISON RESULTS OF DIFFERENT METHODS ON THE COCO VAL2017 DATASET

Method	Backbone	Size	FPS	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
Mask R-CNN* [6]	R-50-FPN	[640, 800]	8.56 (NVIDIA GTX1080 GPU)	34.0	55.3	36.0	15.1	37.0	50.0
CenterMask [9]	VoVNetV1-99 [31]	[800, 1333]	9.9 (Titan Xp GPU)	31.5	—	—	13.5	33.5	46.5
PolarMask [14]	R-101-FPN	[800, 1333]	—	30.4	51.1	31.2	13.5	33.5	43.9
ExtremeNet [13]	Hourglass [32]	511×511	—	18.9	44.5	13.7	10.4	20.4	28.3
YOLACT-RL* [15]	R-50-FPN	550×550	30.4 (NVIDIA GTX1080 GPU)	28.0	46.2	29.1	8.9	30.2	47.0
YOLACT* [15]	R-101-FPN	550×550	22.6 (NVIDIA GTX1080 GPU)	29.8	48.4	31.3	10.1	32.3	50.2
YOLACT-IS* [15]	R-101-FPN	700×700	16.1 (NVIDIA GTX1080 GPU)	30.9	49.9	32.5	12.0	34.3	48.9
ALNMask-RL	R-50-ALN	550×550	20.5 (NVIDIA GTX1080 GPU)	29.1	47.2	30.3	10.3	31.1	49.2
ALNMask	R-101-ALN	550×550	16.5 (NVIDIA GTX1080 GPU)	30.7	49.3	32.3	9.7	33.6	52.0
ALNMask-IS	R-101-ALN	700×700	12.3 (NVIDIA GTX1080 GPU)	32.0	51.3	33.6	13.1	35.0	50.4
ALNMask-L	R-101-ALN	[800, 1333]	9.1 (NVIDIA GTX1080 GPU)	33.4	52.5	35.5	17.2	35.6	48.5

* denotes that we run the corresponding method on our NVIDIA GTX1080 GPU platform to obtain results.

at the expense of processing time. In general, the proposed ALNMask method is considered as effective.

D. Qualitative Results

We choose 20 images on each of the COCO test-dev2017 and val2017 datasets, and the results of instance segmentation provided by ALNMask are presented in Figs. 5 and 6. For an image, the masks of objects are labeled different colors randomly. The objects of interest are related to animal (elephant, bird, and cow), traffic (traffic light, car, motorcycle, and bus), furniture (couch, chair, and dining table), household appliances (oven and remote), and person. In addition, the container (bowl, cup, and wine glass) in the table and the food (cake, sandwich, donut, and orange) are also concerned.

As shown in the second rows of Figs. 5 and 6, ALNMask successfully distinguishes different instances of the same category from the simple scene with two birds, overlapping scene with three buses to the complex scene such as six motorcycles. Even an object is split into several pieces or part of an object is observed, and ALNMask can also complete the segmentation. For example, four orange pieces are discovered in the first image of the last row of Fig. 6, and a person and his handheld cell phone are also found in the second image of the third row of Fig. 5. Besides, it is able to segment the objects within a scene from smaller size to larger size. In the last image of the second row of Fig. 5, both small person and large buses are segmented. The results show that ALNMask can achieve segmentation with good performance and the extracted masks can match the contours of objects.

Fig. 5. Visualization of predicted instance masks and bounding boxes of our ALNMask for images from the COCO test-dev2017 dataset.

Fig. 6. Visualization of predicted instance masks and bounding boxes of ALNMask for images from the COCO val2017 dataset.

E. Robustness Verification

Five interferences are imposed on an original image to verify the robustness of the proposed method, where YOLACT and Mask R-CNN are also involved for comparison. We conduct the experiments on an image from the COCO val2017 dataset, where there are multiple objects on a wooden tray placed in furniture. The instance segmentation results are shown in Fig. 7 and the methods include ALNMask-50 with R-50-ALN, ALNMask-101 with R-101-ALN, YOLACT-50 with R-50-FPN, YOLACT-101 with R-101-FPN, and Mask R-CNN. The first row of Fig. 7 gives the original image as well as the polluted images after Gaussian blur with kernel size of 3×3, brightness enhancement (30%), Gaussian noise with standard deviation of 0.008, salt-and-pepper noise with the

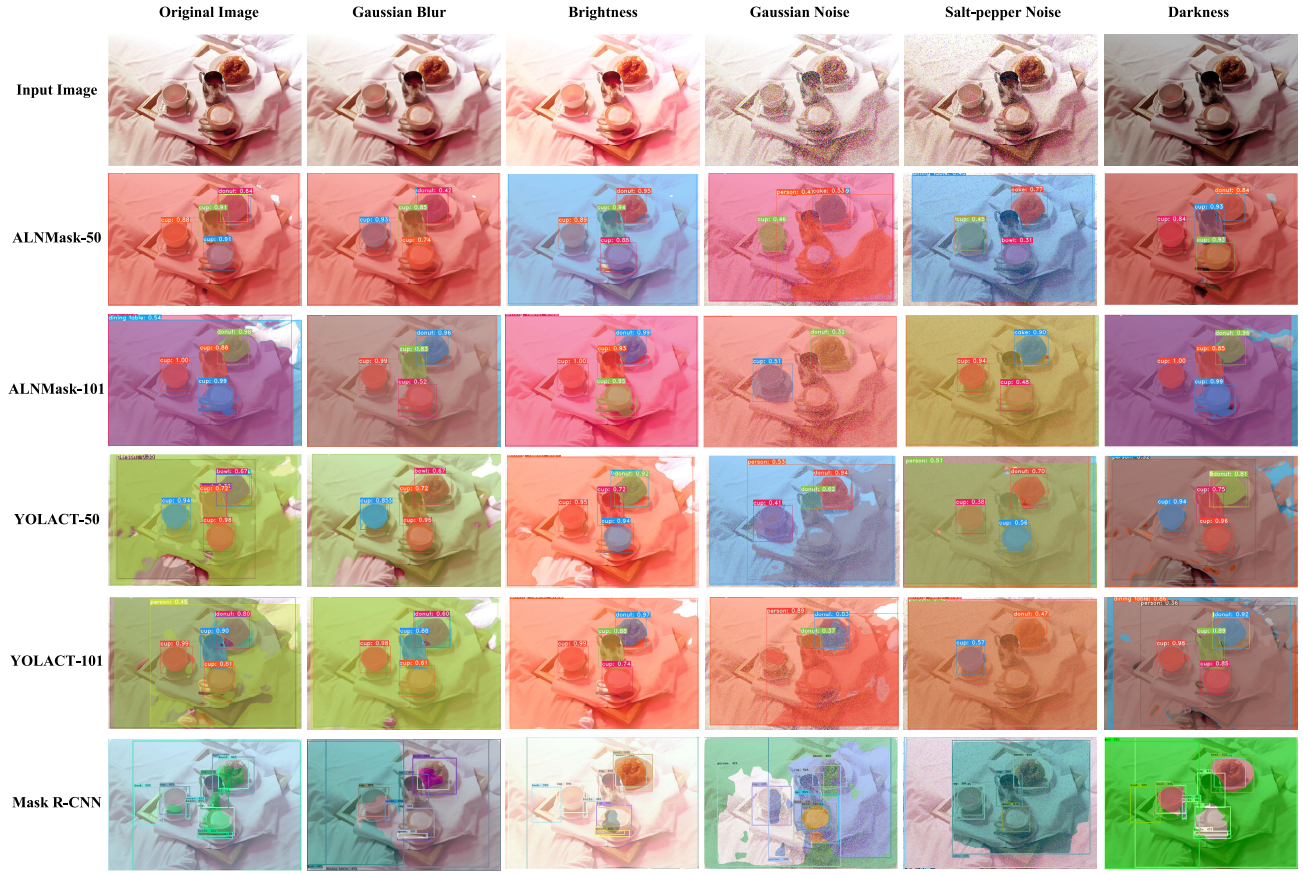
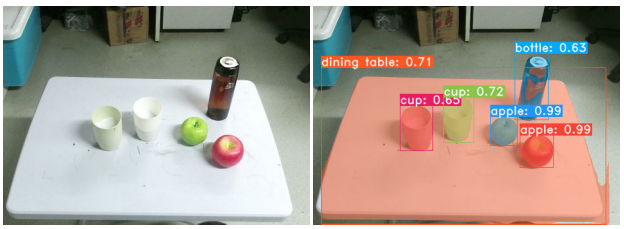


Fig. 7. Instance segmentation results of different methods under interferences.

ratio value of 0.08, and darkness (−30%) are exerted. From the whole results of Mask R-CNN, it has the advantage in capturing details, which is beneficial to mine small objects. For example, two spoons near the cups can be segmented (sometimes they are regarded as the knife class). However, the object masks provided by Mask R-CNN are incomplete. Besides, the corner of the wooden tray is segmented as the book class three times. Compared to two-stage Mask R-CNN, YOLACT and ALNMask are good in terms of mask quality. Next, we focus on comparison of these two methods under different interferences.

We first check the segmentation results of image background. ALNMask-101 simultaneously outputs the bed and dining table classes four times and predicts dining table and bed classes in other two images. Accordingly, ALNMask-50 treats backgrounds of all six images as the dining table class. YOLACT series also regard background as dining table or bed class. Actually, the local surfaces of bed and dining table are similar, and in this case, it is acceptable to regard the background as either the dining table or bed class.

Furthermore, the four objects, including three cups and a donut on the tray are investigated. For ALNMask-101, these four objects are correctly segmented in four out of six images (see the first, second, third, and sixth columns). Among these four images, YOLACT-101 also outputs correct results. Meanwhile, ALNMask-101 has a



(a) (b)

Fig. 8. Experiment in an actual scene. (a) Original image. (b) Instance segmentation result.

higher average segmentation accuracy with good details than YOLACT-101. Correspondingly, ALNMask-50 achieves the correct segmentation of three cups and treats the donut as the donut and cake classes in three out of four times. For YOLACT-50, it gives correct results in the third and sixth images, and there exists a cup that is treated as the cup and bowl classes in the first and second images. In addition, for the image in the fifth column, YOLACT-50 acquires the best results with two cups and one donut, and our ALNMask-101 thinks the donut as the cake class. As for the image shown in the fourth column, ALNMask-101 behaves well with the results of one cup and one donut, and the YOLACT series consider a cup as the donut class.

Besides, there are plates under cup and donut, and YOLACT series partially mine them (bowl class). Moreover, affected

by the image and interferences, ALNMask-50 presents person class once among six images, YOLACT series output person class four times, whereas no person class is found in the results of ALNMask-101.

F. Experiment in an Actual Scene

We further testify the proposed ALNMask in an actual scene and the experimental result is shown in Fig. 8. Fig. 8(a) presents the original image, and there are five objects of interest placed on a table: two cups, two apples, and a bottle. From the instance segmentation result shown in Fig. 8(b), ALNMask segments different objects.

V. Conclusion

In this article, an instance segmentation method with an ALN network and an atrous-residual structure is proposed. The whole network is composed of three parts: a visual encoder, a long-neck network, and instance prediction. The first part is used to extract multilevel features from the vision image, and the second part implements the tight fusion of high- and low-level features in the top-down and bottom-up pathways with the help of cross-layer dense connection scheme. The third part is responsible for detection results and instance masks by an atrous-residual-based mask prototype branch. The experimental results demonstrate the effectiveness of the proposed method. It is observed that in crowded scenes, there sometimes exists unsegmented object because of shadowing from the foreground objects, such as an unnoticed sheep in the second image of the second row of Fig. 6. In the near future, we shall focus on mining more location information to improve the performance of instance segmentation network in crowded scenes.

References

[1] H. Sheng, S. Wei, X. Yu, and L. Tang, "Research on binocular visual system of robotic arm based on improved SURF algorithm," *IEEE Sensors J.*, vol. 20, no. 20, pp. 11849–11855, Oct. 2020.

[2] J. Yang, C. Wang, H. Wang, and Q. Li, "A RGB-D based real-time multiple object detection and ranging system for autonomous driving," *IEEE Sensors J.*, vol. 20, no. 20, pp. 11959–11966, Oct. 2020.

[3] Z. Guo, Y. Huang, H. Wei, C. Zhang, B. Zhao, and Z. Shao, "DALaneNet: A dual attention instance segmentation network for real-time lane detection," *IEEE Sensors J.*, vol. 21, no. 19, pp. 21730–21739, Oct. 2021.

[4] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4438–4446.

[5] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3150–3158.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," 2017, *arXiv:1703.06870*.

[7] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[8] C.-Y. Fu, M. Shvets, and A. C. Berg, "RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free," 2019, *arXiv:1901.03353*.

[9] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," 2019, *arXiv:1911.06667*.

[10] P. O. Pinheiro, R. Collobert, and P. Dollàr, "Learning to segment object candidates," 2015, *arXiv:1506.06204*.

[11] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollàr, "Learning to refine object segments," 2016, *arXiv:1603.08695*.

[12] S. Zagoruyko et al., "A MultiPath network for object detection," 2016, *arXiv:1604.02135*.

[13] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 850–859.

[14] E. Xie et al., "PolarMask: Single shot instance segmentation with polar representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12190–12199.

[15] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9156–9165.

[16] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[18] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," 2015, *arXiv:1506.04579*.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[20] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollàr, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[22] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," 2019, *arXiv:1904.01355*.

[23] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[24] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "BlitzNet: A real-time deep network for scene understanding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4174–4182.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[28] W. Liu et al., "SSD: Single shot MultiBox detector," 2015, *arXiv:1512.02325*.

[29] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*.

[30] Z. Wu. *Deep Learning for Image Processing*. Accessed: 2022. [Online]. Available: https://github.com/WZMIAOMIAO/deep-learning-for-image-processing/tree/master/pytorch_object_detection/mask_rcnn

[31] Y. Lee, J.-W. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and GPU-computation efficient backbone network for real-time object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 752–760.

[32] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016, *arXiv:1603.06937*.

Wenjie Geng received the B.S. and M.S. degrees from Harbin Engineering University, Harbin, China, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree in control theory and control engineering with the Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His current research interests include visual perception and robotic grasping.

Zhiqiang Cao (Senior Member, IEEE) received the B.E. degree in industrial automation and the M.E. degree in control theory and control engineering from the Shandong University of Technology, Jinan, China, in 1996 and 1999, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2002.

He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences. His current research interests include service robots and intelligent robots.

Peiyu Guan received the B.E. degree in electronic information science and technology from Jilin University, Changchun, China, in 2017, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2022.

She is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. Her current research interests include service robots and image processing.

Guangli Ren received the B.S. degree in intelligent science and technology from Dalian Maritime University, Dalian, China, in 2015, and the M.S. degree in technology of computer application from Capital Normal University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in control theory and control engineering with the Institute of Automation, Chinese Academy of Sciences, Beijing.

His current research interests include visual localization and robotic manipulation.

Junzhi Yu (Fellow, IEEE) received the B.E. degree in safety engineering and the M.E. degree in precision instruments and mechanology from the North University of China, Taiyuan, China, in 1998 and 2001, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2003.

From 2004 to 2006, he was a Postdoctoral Research Fellow with the Center for Systems and Control, Peking University, Beijing. In 2006, he joined the Institute of Automation, Chinese Academy of Sciences, as an Associate Professor, where he was a Full Professor in 2012. In 2018, he joined the College of Engineering, Peking University, as a Tenured Full Professor. His current research interests include intelligent robots, motion control, and intelligent mechatronic systems.

Fengshui Jing received the B.S. degree in mining engineering from the Huainan Mining Institute, Anhui, China, in 1991, the M.S. degree in safety technology and engineering from the Shandong Mining Institute, Taian, China, in 1994, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2002.

He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences. His current research interests include robotics, computer vision, and manufacturing systems.