AFPSNet: Multi-Class Part Parsing based on Scaled Attention and Feature Fusion

Njuod Alsudays Jing Wu Yu-Kun Lai Ze Ji
Cardiff University, UK
{alsudaysn, wuj11, laiy4, jiz1}@cardiff.ac.uk

Abstract

Multi-class part parsing is a dense prediction task that seeks to simultaneously detect multiple objects and the semantic parts within these objects in the scene. This problem is important in providing detailed object understanding, but is challenging due to the existence of both class-level and part-level ambiguities. In this paper, we propose to integrate an attention refinement module and a feature fusion module to tackle the part-level ambiguity. The attention refinement module aims to enhance the feature representations by focusing on important features. The feature fusion module aims to improve the fusion operation for different scales of features. We also propose an object-to-part training strategy to tackle the class-level ambiguity, which improves the localization of parts by exploiting prior knowledge of objects. The experimental results demonstrated the effectiveness of the proposed modules and the training strategy, and showed that our proposed method achieved state-of-the-art performance on the benchmark datasets.

(a) Input (b) CSR (c) Ours (d) GT

Figure 1. Challenges of multi-class part parsing. (a) Input images with multiple objects. (b) Part-level ambiguity and inaccurate boundary localization present in the current state-of-the-art method [26]. (c) Results from our method. (d) Ground truth masks.

1. Introduction

Multi-class part parsing can be considered as a special case of semantic segmentation, decomposing objects into semantic components. This vision task aims to understand a scene at multiple levels of abstraction, by simultaneously detecting multiple semantic classes in the scene and accurately parsing the parts within each class. High-quality prediction of parts would be of great use for many tasks, such as object detection [1, 5], fine-grained action detection [30], pose estimation [7, 35], and image classification [25, 14]. Despite the importance of this problem in understanding the details of objects, it is not sufficiently explored at present.

Existing methods in the field of object part parsing mainly address single-class settings, such as human bodies [34, 17, 8, 38], vehicles [8, 24, 18] and animals [28, 29, 10]. Multi-class part parsing has only been considered in recent works [37, 19, 26]. It is more challenging, as it needs to tackle both class-level and part-level ambiguities. In particular, one challenge is that sometimes, parts in different objects can have similar appearance making it difficult to distinguish between them. For example, as shown in Fig. 1 (b), even the state-of-the-art method [26] mistakenly detects the back of the car as a bus, and the horse body as a cow body. Another challenge is that, unlike single object part parsing, the cluttered appearance of several objects in the scene often leads to occlusions, which results in inaccurate boundary detection. An example is the segmentation of the airplane as shown in Fig. 1.

To tackle these challenges, we propose AFPSNet with an object-to-part training strategy. AFPSNet is based on the popular DeepLab v3+ network architecture [4], but improves it in two ways. Firstly, it integrates the attention refinement module (ARM) and the feature fusion module

(FFM) from [36] into DeepLab v3+ instead of the simple concatenation of features. The attention refinement module refines the feature maps at each scale to focus on features with importance and improve the representation efficiency, while the feature fusion module enables the exploitation of both global and local information for prediction of the context and boundaries of parts. Secondly, we propose to fuse features in a cascaded way, i.e., instead of fusing features at different scales all at once, we fuse two features at similar scales at one time, and gradually progress to fuse all features. It aims to tackle the issue pointed out in [4] that a direct concatenation of all these features at once may cause confusion and thus disregard some required features. Moreover, inspired by the observation that humans tend to locate objects first before looking into details, we propose a two-stage training strategy to train the model on object labels first before proceeding to part labels. This object-to-part training strategy aims to improve the localization of parts by exploiting prior knowledge.

The contributions of this work are as follows:

- We integrate attention refinement and feature fusion into DeepLab v3+ and propose a cascaded way of feature fusion to enhance the features for boundary prediction and in turn to improve parts prediction.
- We propose an object-to-part training strategy, which improves the localization of parts by exploiting prior knowledge from object-level segmentation.
- The proposed method achieves the state-of-the-art performance on the benchmark part parsing datasets.

2. Related Work

2.1. Part Parsing

Single-object part parsing. Recently, existing research has shown effective performance in accurate segmentation of parts of one specific category. Strategies for handling object part parsing could be divided into two categories. The first category is based on coarse-to-fine, also known as top-down, strategies. Hariharan et al. [11] proposed three different architectures to sequentially perform object detection, object segmentation and part segmentation. Nevertheless, their approach suffers from the difficulty of the training process and the error propagation from object masks throughout the pipeline. Xia et al. [32] proposed a two-stage process consisting of three same structural networks to integrate the global feature with the detected local features. Chen et al. [3] introduced a scale attention model to learn pixel-wise weights for a specific category and fuse the parsing results from three fixed scales. The second category focuses on structure-based methods [16, 33, 21, 9] to model part relations. Liang et al. [16] proposed a self-supervised structure-sensitive learning approach to simultaneous estimation of human pose and part parsing. Some research shows that pose estimation tasks can be useful for part parsing tasks [33, 21, 9]. Xia et al. [33] proposed a framework employing the two tasks to improve the segmentation results by supervised pose estimation. Nie et al. [21] presented a mutual learning model to improve the part segmentation results by adapting the pose estimation task. Fang et al. [9] proposed a pose-guided model, which exploits similarity among humans to transfer the part parsing results between different persons with similar poses.

Multi-class part parsing. Although effective methods have been proposed to tackle single-object part parsing, multi-class part parsing has been explored only recently [37, 19, 26]. Zhao et al. [37] proposed a joint parsing framework called BSANet, which is composed of boundary and semantic awareness modules, to enhance part localization and promote the expression of class-relevant feature channels. Predicted part boundaries are also passed into an attention mechanism to promote features near boundaries at the decoding stage. Michieli et al. [19] proposed a GMNet framework consisting of three subnetworks and a graph matching module to improve parts segmentation and localization. Object-level segmentation maps are passed into a semantic embedding network to serve as guidance for part parsing within the object at the decoding stage. Part-level segmentation maps are then enhanced using a graph matching technique that preserves the relative spatial relationships between ground truth and predicted parts. Tan et al. [26] proposed a framework, known as CSR, which employs the confident semantic ranking loss function to model the relationships between pixels. BSANet and CSR use predicted boundaries and object labels during the training to guide the prediction task and refine the segmentation results. GMNet and CSR incorporate additional loss functions to exploit the relationships of pixels. These methods improve the segmentation results. However, they require dealing with the object-level and part-level labels at the same time, and thus require loading of a larger number of label masks at one time. This prevents them from using large batch size and thus results in a longer training time. We thus introduce the object-to-part training strategy to both improve the part localization and relieve the memory demand to allow a larger batch size and speed up training.

2.2. Attention and Feature Fusion

Attention Mechanism. The attention mechanism gives a model the ability to concentrate on the most relevant features as needed rather than attempting to process a whole scene at once. Attention modules have demonstrated their usefulness across many visual tasks, including image processing, object tracking and video understanding [3, 27, 12, 31, 36]. In these works, attention processing is incorporated to improve the performance of their networks. Various attention mechanisms are applied in com-

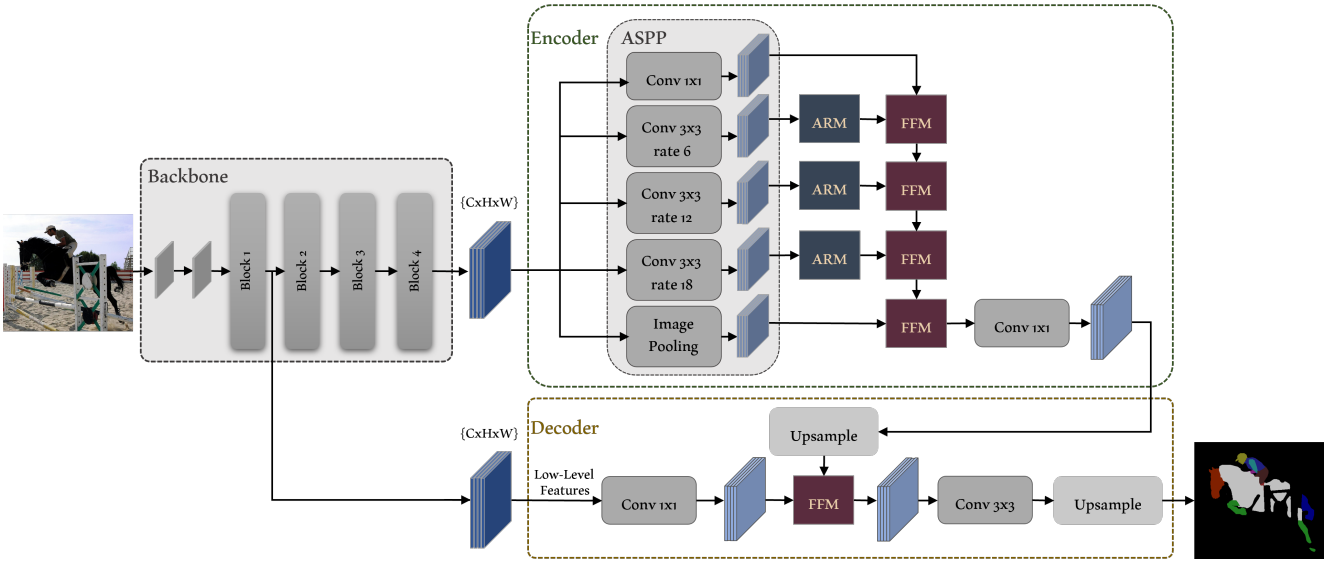

Figure 2. An overview of the proposed AFPSNet approach. The architecture is the DeepLab V3+ network with integrated attention refinement modules (ARM) and feature fusion modules (FFM).

puter vision, including spatial attention and channel attention. Hu et al. [12] exploited the channel-wise relationships by introducing the Squeeze-and-Excitation module. Woo et al. [31] combined the spatial and channel attention in a compact module called CBAM block. Yu et al. [36] introduced a lightweight module to perform channel-wise attention, which is adopted in our work.

Due to the great success of attention modules, they have been used in part parsing. In [3], an attention model is proposed to fuse the feature maps from different image zooming scales. Zhao et al. [37] employed spatial attention to enhance the features near boundaries in the boundary detecting module. An additional attention module is used in the semantic selection module to emphasize the class-correlated features and suppress the irrelevant ones. Unlike BSANet, where spatial attention is used, in our work, we adopt the attention refinement module from [36] to apply channel-wise attention to select the most important feature maps and refine the features at different scales.

Feature Fusion. Feature fusion as a merging module for features of different levels plays an important role in segmentation performance. Recently, various designs of feature fusion modules have been proposed to selectively extract beneficial information from different levels of feature maps. Li et al. [15] proposed a Gated Fully Fusion module to selectively fuse features from multiple levels using a gating mechanism in a fully connected way. Nie et al. [20] introduced an add-multiply-add fusion block with learned weights, which first adds and multiplies the different levels of features separately and then adds these two output features together. Poudel et al. [23] performed simple addition of features by using bilinear upsampling and convolutions. Although employing this simple operation is not good enough to output a high-accuracy segmentation map, it reduces the computational cost. Yu et al. [36] introduced a feature fusion module that performs a weighted fusion to refine and fuse features from different levels. In our work, we adopted the feature fusion module in [36], where an embedded channel-wise attention operation is performed to refine the summation of the features.

3. Method

3.1. Overview

In this section, we describe our method for multi-class part parsing. We use DeepLab v3+ [4] as the backbone. Then, we use the attention refinement module and the feature fusion module from [36] and propose to integrate them into DeepLab v3+. The network architecture is shown in Fig. 2. The attention module is integrated into the different layers in the Atrous Spatial Pyramid Pooling (ASPP) unit in order to refine the required features at each scale. The feature fusion module is integrated in two ways. One is to fuse features with different scales from the ASPP. The other is to fuse the high-level features with the low-level ones from early blocks. With both, the feature fusion enables the utilization of both global and local information for prediction.

3.2. Attention Refinement Module

The attention refinement module (ARM) is from [36], and we integrate it into each of the three layers in the ASPP unit. The ASPP unit employs multiple parallel filters with

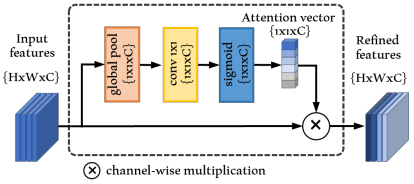

Figure 3. The attention refinement module (ARM) [36]. A refined feature map is obtained by channel-wise multiplication of the input feature map with the attention vector.

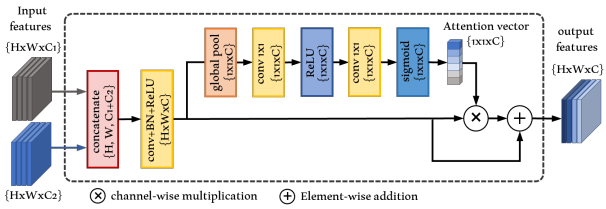

Figure 4. The feature fusion module (FFM) [36]. An output feature is refined by performing channel-wise attention and then employing element-wise addition to fuse the refined feature with the concatenated feature.

different atrous rates to adjust the fields of view of filters and thus obtain multi-scale features [4]. Integration of the ARM aims to selectively focus on the important features along the channel dimension for the following prediction tasks. Herein, we give a brief introduction about the ARM.

As Fig. 3 shows, the ARM first applies the global average pooling to capture the global context in each channel of the feature map, and then calculates an attention vector which indicates the importance of different channels. The input features are then refined by channel-wise multiplication with the attention vector.

Specifically, a feature map $F^{H\times W\times C}$ is passed to the ARM, where H, W, and C represent the height, width, and channel number of the feature map, respectively. The global average pooling is first applied to each channel, which reduces the input feature map to a $C \times 1 \times 1$ feature vector. This can be expressed as:

$$F^c_{avg} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} f^c(i,j) \qquad (1)$$

where $f^c(i,j)$ denotes the channel elements of the input feature map F at position (i,j). The feature vector then goes through a convolution with a sigmoid activation to compute the channel-wise attention vector. After ARM, a refined feature map is obtained by channel-wise multiplication of the input feature map with the attention vector.

3.3. Feature Fusion Module

Features from different scales capture details at different levels. The combination of different scaled features will be beneficial, as different aspects in part parsing (*e.g.*, object/part localization, boundary extraction, etc.) require attention to different detail levels. We therefore use the feature fusion module (FFM) from [36] and propose to integrate it into DeepLab v3+. Herein, we give a brief introduction about the FFM.

As shown in Fig. 4, two feature maps $X_1 \in R^{H\times W\times C}$ and $X_2 \in R^{H\times W\times C}$ are first concatenated and passed through batch normalization [13] in order to avoid the gradient vanish or explosion problems. Simple concatenation does not give emphasis on which scale of features to be used. Thus, a channel-wise attention as used in ARM was introduced. A global average pooling is applied to obtain the feature vector, which then goes through a convolution with a ReLU activation and a further convolution with a sigmoid activation to calculate the channel-wise attention V_{att}. Then, the concatenated feature is refined as:

$$X_{FFM} = (X_{concat} \otimes V_{att}) \oplus X_{concat}, \qquad (2)$$

where $\otimes$ denotes channel-wise multiplication, and $\oplus$ element-wise addition.

There are two granularity of feature fusion in AFPSNet. One is the fusion of features obtained from different ASPP pooling layers. The other is the fusion of features from different ResNet blocks. To fuse features from different ASPP pooling layers, a sequence of FFMs is integrated, as shown in Fig. 2. Due to the use of different dilation rates in different layers in ASPP, the detail levels of these output features vary significantly. Direct concatenation of all these feature maps may disregard some required features, as pointed out in [4]. Therefore, we propose to fuse features in a cascaded way by combining features of similar detail levels and progressing from fine to coarse scales gradually. That is, the first FFM combines the two finest output feature maps from the ASPP. The remaining FFMs then combine the fused feature from the previous FFM with the output feature from the coarser ASPP layers. To fuse features from different blocks, we also use an FFM instead of direct concatenation. A single FFM is used to fuse the feature map from the first block and the fused feature map after the ASPP unit.

3.4. Object-to-Part Training Strategy

Given the object labels and part labels, the state-of-the-art part parsing methods [37, 19, 26] train the two objectives altogether by employing parallel multi-head predictions. It requires more memory due to passing the labels for all the tasks at once, which as a result only allows the model to be trained on smaller batch sizes and requires longer training time. Therefore, we propose to train the two tasks se-

quentially. On one hand, it allows to increase the batch size and speed up the training. On the other hand, the object-level prediction will provide useful priors for part-level prediction as well. Hence, our model is firstly trained on the object-level annotations, and then fine-tuned for part-level prediction.

4. Experiments

4.1. Implementation Details

Dataset. The PASCAL-Part dataset [5], including PASCAL-Part-58, PASCAL-Part-108, and PASCAL-Person-Part, is the largest dataset for multi-class part parsing. It is used for training and evaluating the proposed method. Both the PASCAL-Part-58 and PASCAL-Part-108 contains 10103 images of different sizes, along with 58 (PASCAL-Part-58) or 108 (PASCAL-Part-108) part-level annotations of 20 semantic object classes, excludes the background class. We follow the original split as in [5], where 4998 images are used for training and 5105 images are used for testing. PASCAL-Person-Part contains 3533 images of multi-person on various scales. The persons are annotated with 7 body parts. We follow the split as in [3, 33, 37], where 1716 images are used for training and 1817 images are used for testing. Also, we follow the state-of-the-art and other well known part parsing methods [37, 19, 26] to use the same evaluation metrics: the mean Intersection over Union (mIoU); and the same evaluation strategy. We use PASCAL-Part-58 to demonstrate the effectiveness of integrating the two modules and the object-to-part training strategy, while all three are used in comparisons with other methods.

Training details. The backbone of the network is DeepLab v3+ [4] trained on the ImageNet dataset [6]. We reproduce the DeepLab v3+ model in PyTorch [22] and follow the same training schemes as in [4, 2]. In our experiments, input images are cropped to 513×513 in resolution and randomly left-right flipped and scaled with a factor ranging from 0.5 to 2.0. During training, the Stochastic gradient descent (SGD) optimizer with momentum 0.9 and weight decay regularization $1e{-}4$ is used. The learning rate is set as 0.01 for object-level training and 0.05 for part-level training. The atrous rates of the ASPP are set to $(6, 12, 18)$ as in prior works [4, 37, 19]. Also, the down-sampling stride is set to 16 in all our models. For PASCAL-Part-58, we train our model with batch size 16 for 65K iterations, for PASCAL-Part-108, we train the model with batch size 14 for 64K iterations, and for PASCAL-Person-Part, we train the model with batch size 16 for 60K iterations.

4.2. Ablation Study

We first carried out ablation studies to evaluate the effectiveness of integrating the ARM and the FFM modules, and the object-to-part training strategy.

Table 1. Detailed performance comparison of each component in our proposed AFPSNet approach. For computational efficiency, the experiments are conducted by performing a parts-only training strategy.

Method	ARM	FFM1s	FFM2	mIoU(%)
DeepLab v3+				57.16
DeepLab v3+	✓			56.72
DeepLab v3+	✓	✓		57.89
DeepLab v3+		✓	✓	57.97
AFPSNet	✓	✓	✓	**58.33**

Table 2. Performance comparison of parts-only training strategy, multi-task learning strategy and object-to-part training strategy on our baseline and AFPSNet.

Method	Training strategy	mIoU(%)
DeepLab v3+	parts-only	57.16
DeepLab v3+	multi-task	**58.86**
DeepLab v3+	object-to-part	58.80
AFPSNet	parts-only	58.33
AFPSNet	multi-task	59.34
AFPSNet	object-to-part	**60.61**

Integrating ARM and FFM modules. Table 1 shows the results of the baseline method (DeepLab v3+), and the baseline with the addition of the ARM and FFM modules. To save the training time, in this experiment, we trained these models using part labels only. For simplicity, here we use FFM1s to refer to the sequence of FFM modules fusing features from ASPP, and FFM2 the FFM module fusing features from different blocks. When FFM modules are not added, simple concatenation of features is used instead.

From Table 1, the baseline method achieved an mIoU of 57.16%. However, adding the ARM alone dropped the value to 56.72%. As pointed out in [4], direct concatenation of features at very different detail levels may cause some useful features to be discarded. Adding ARM, while enhancing the features, exaggerated the problem in this case, which explained the performance drop. However, when enhancing the features combined with feature fusion using FFM1s, the mIoU was increased to 57.89%, demonstrated the necessity of combining ARM with FFM1s and the effectiveness of the combination. The further addition of FFM2 achieved the best performance with 58.33% mIoU, further demonstrated the effectiveness of fusing low-level features. While the performance of our model without ARM drops from 58.53% to 57.97%, which demonstrates the effectiveness of ARM for selective focus on desired features.

Fig. 5 gives qualitative results, showing how the segmen-

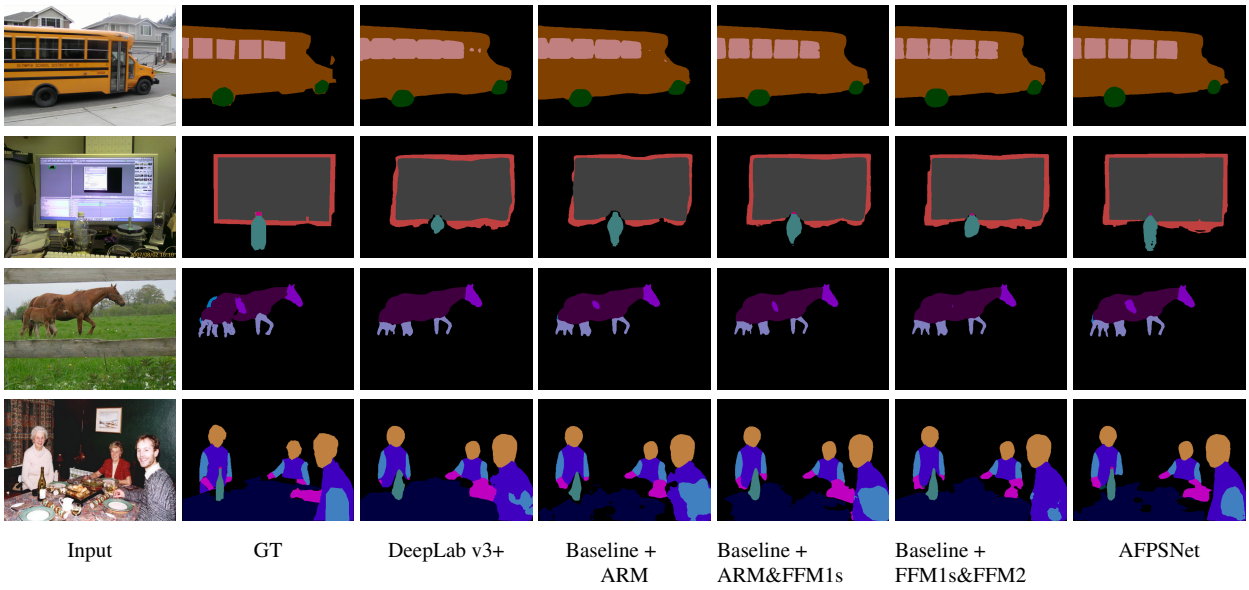

Figure 5. Qualitative comparisons of segmentation results of the baseline, AFPSNet and each component integrated with the baseline on PASCAL-Part-58 dataset.

tation improves with the addition of ARM and FFM modules. From the figure, we can see that the baseline method struggled to detect small parts or to give clear boundaries. For example, the lid of the bottle was not detected in the second and the fourth examples in Fig. 5, the head of the small horse in the third example was missed, and the bus windows in the first example were merged together missing gaps in between. Adding the ARM module (the fourth column in Fig. 5) improves the detection of some small parts, such as the body of the bottle and the head of the small horse. However, deterioration of boundary prediction was also observed, *e.g.*, the boundary of the monitor in the second image, showing the pros and cons of adding ARM alone. However, when adding the ARM and FFM1s together (the fifth column in Fig. 5), noticeable improvements were observed. For example, the gaps between windows in the first example and the lid of bottle in the second were detected. Moreover, the boundaries were more accurately predicted, as can be seen in the second example. Then, the last column in Fig. 5 shows further improvements on small parts segmentation (*e.g.*, lid of bottle in both examples, the head of the small horse) and boundary prediction (*e.g.*, gaps between bus windows, the boundary of the monitor). While removing ARM from our model (the sixth column in Fig. 5), *i.e.*, the body of the bottle and the head of the small horse were not detected, which shows the necessity of ARM in our model.

Table 1 and Fig. 5 demonstrate the effectiveness of integrating the ARM and FFM modules. We then evaluate the effectiveness of the object-to-part training strategy.

Object-to-part training strategy. We conducted several experiments by training AFPSNet and DeepLab v3+ using 1) a part-only training strategy, 2) a multi-task learning strategy (as used in existing multi-class part parsing methods), and 3) the proposed object-to-part training strategy. The results are shown in Table 2.

We observed that training DeepLab v3+ using part labels only achieved 57.16% in mIoU. While incorporating object-level segmentation, either parallel as a multi-task learning or sequentially as object-to-part, improved the mIoU by 1.7% and 1.64% respectively. Similar improvements were observed when training AFPSNet using the three strategies, where using multi-task learning achieved 1.01% increase in mIoU and using object-to-part training achieved 2.28%. The results on both networks demonstrate the effectiveness of object-to-part training strategy to improve the performance over part-only training strategy. However, a difference is also observed. Training AFPSNet using the object-to-part training strategy achieved noticeable improvements than using the other two strategies. While training DeepLab v3+ using the object-to-part training strategy achieved similar, or even slightly worse, performance compared to using multi-task learning strategy. This indicates that the effectiveness of object-to-part training may depend on the network architecture. With the ARM and various FFM modules added, AFPSNet is architecturally more complex than DeepLab v3+. More complex architectures may benefit more from the more stable sequential training. However, to verify this, more investigations are needed, which will be a direction for future work.

Table 3. Segmentation performance of mIoU on PASCAL-Part-58 benchmark. mIoU: per-part-class mIoU. Avg.: average per-object-class mIoU.

Method	backg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	potted	sheep	sofa	train	tv	mIoU	Avg.
DeepLab v3+ [4]	94.5	46.4	65.2	53.6	63.7	51.5	67.1	51.6	62.6	38.5	52.6	45.2	58.6	66.5	72.5	56.5	55.4	52.1	46.0	80.2	61.0	57.60	59.1
BSANet [37]	91.6	50.0	65.7	54.8	60.2	49.2	70.1	53.5	63.8	36.5	52.8	43.7	58.3	66.0	71.6	58.4	55.0	49.6	43.1	82.2	61.4	58.2	58.9
GMNet [19]	92.7	46.7	66.4	52.0	**70.0**	55.7	71.1	52.2	63.2	**51.4**	54.8	**51.3**	59.6	64.4	**73.9**	56.2	56.2	53.6	**56.1**	**85.0**	**65.6**	59.0	61.8
CSR [26]	91.9	**52.0**	64.9	**56.0**	61.7	56.9	**72.0**	**56.9**	64.0	36.3	59.2	45.1	**62.3**	68.6	72.9	55.2	56.9	53.6	43.5	79.8	63.5	60.7	60.6
AFPSNet	**94.8**	50.9	**68.1**	55.7	64.0	**57.7**	**72.0**	55.7	**65.1**	39.8	**60.7**	44.6	61.9	**70.4**	72.8	**61.4**	**58.3**	**57.0**	46.4	81.6	63.1	**61.3**	**62.0**

Table 4. Segmentation performance of mIoU on PASCAL-Part-108 benchmark. mIoU: per-part-class mIoU. Avg.: average per-object-class mIoU.

Method	backg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	potted	sheep	sofa	train	tv	mIoU	Avg.
DeepLab v3 [2]	90.9	41.9	44.5	35.3	53.7	47.0	34.1	42.3	49.2	35.4	39.8	33.0	48.2	48.8	23.2	50.4	43.6	35.4	39.2	20.7	60.8	41.3	43.7
DeepLab v3+ [4]	94.5	48.8	45.4	41.6	59.5	49.5	36.5	45.3	51.3	37.3	50.9	44.1	52.0	54.5	23.9	55.8	54.0	42.6	47.4	**23.3**	69.7	46.5	48.9
BSANet [37]	91.6	45.3	40.9	41.0	61.4	48.9	32.2	43.3	50.7	34.1	39.4	45.9	52.1	50.0	23.1	52.4	50.6	37.8	44.5	20.7	66.3	42.9	46.3
GMNet [19]	92.7	48.0	46.2	39.3	**69.2**	**56.0**	37.0	45.3	52.6	**49.1**	50.6	**60.6**	52.0	51.5	24.8	52.6	56.0	40.1	**53.9**	21.6	70.7	45.8	50.5
AFPSNet	**94.9**	**50.4**	**52.0**	**43.8**	61.1	52.1	**41.1**	**48.9**	**54.0**	38.0	**54.5**	43.0	**55.0**	**57.7**	**25.4**	**58.5**	**57.2**	**44.5**	47.2	23.1	**73.1**	**49.2**	**51.2**

Table 5. Segmentation performance of mIoU on Pascal-Person-Part benchmark. mIoU: per-part-class mIoU.

Method	backg	head	torso	u-arms	l-arms	u-legs	l-legs	mIoU
DeepLab v3 [2]	94.79	84.06	66.69	54.26	52.80	48.08	43.59	63.50
DeepLab v3+ [4]	97.12	87.00	70.91	59.69	59.54	52.96	49.42	68.09
BSANet-101 [37]	95.62	86.49	70.20	59.31	58.72	51.91	49.32	67.37
BSANet-152 [37]	95.79	86.98	71.35	61.36	60.26	53.28	49.95	68.43
AFPSNet	**97.28**	**87.60**	**72.68**	**62.07**	**61.48**	**54.59**	**51.22**	**69.56**

4.3. Comparisons with the state-of-the-art

We also compared our method with the baseline and the state-of-the-art multi-class part parsing methods [37, 19, 26]. We first compared the segmentation performance on the PASCAL-Part-58 benchmark. Two metrics were used to compare the performances of these methods quantitatively, *i.e.*, mIoU which computes the mean per-part IoU on the 58 part classes and Avg. which computes the average per-object mIoU on the 21 object classes (including background). As shown in Table 3, the baseline, DeepLab v3+, achieved 57.60% in per-part mIoU. While BSANet, which is the first work addressing part-based semantic segmentation, achieved 58.2%. GMNet improved the performance, achieving 59.0%. The current state-of-the-art method, CSR, achieved 60.7%. Compared to the above methods, the proposed AFPSNet achieved the highest per-part mIoU of 61.3%, outperforming the current state-of-the-art method. The same is observed for the average per-object-class mIoU as well. Closer examination of the class-level segmentation in Table 3 further shows that our model achieves the highest mIoU for 10 out of 21 categories (including background), more than other methods compared.

Fig. 6 illustrates qualitative comparison of the segmentation results from these methods. Our model shows overall better segmentation results with less missing parts and more accurate boundaries. For example, the segmentation of the cat head in the first row and the human leg in the second row are challenging for the other methods, while AFPSNet can successfully detect and segment them. Moreover, AFPSNet shows superior performance in detecting small object parts even in a very crowded scene. For example, in the third row, AFPSNet can better predict the boundaries of the chairs and successfully detect the legs behind the desks. And similar observations can be seen in the segmentation of the arm chair in the last row.

We further evaluate the performance of AFPSNet on PASCAL-Part-108 benchmark. Our method is compared with the baseline and 2 of the state-of-the-art multi-class part parsing methods [37, 19] with the reported performances, as shown in Table 4. Similar to PASCAL-Part-58 benchmark, our method achieved the highest per-part mIoU of 49.2%, outperforming the state-of-the-art method. As can be seen, AFPSNet achieves the highest mIoU for 15 out

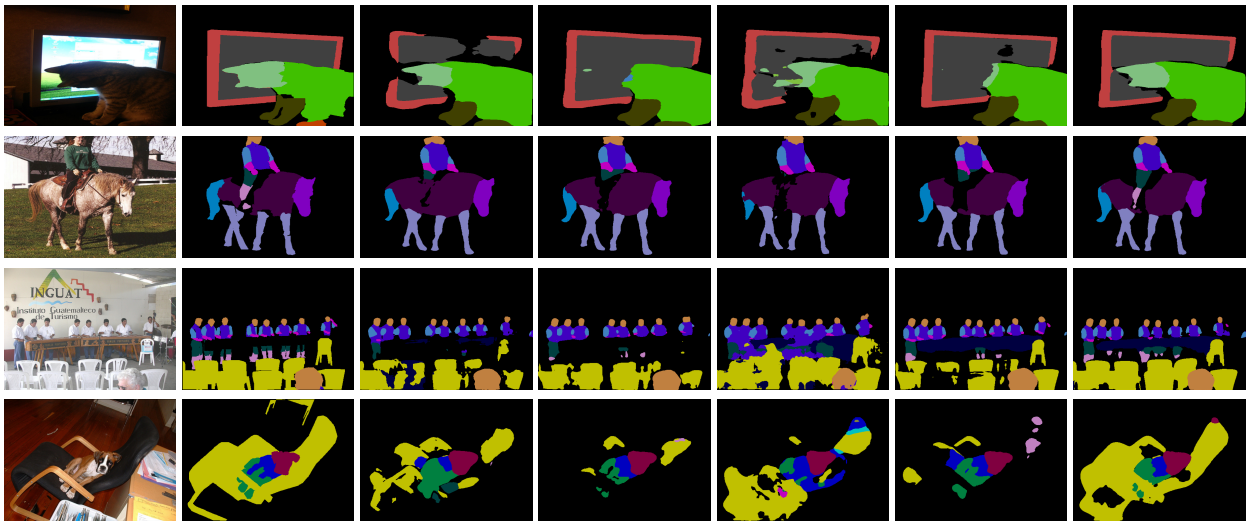

Figure 6. Segmentation results on PASCAL-Part-58 dataset. Our model generates notable results with more details of small object parts and better boundary localization compared to the-state-of-the-art models.

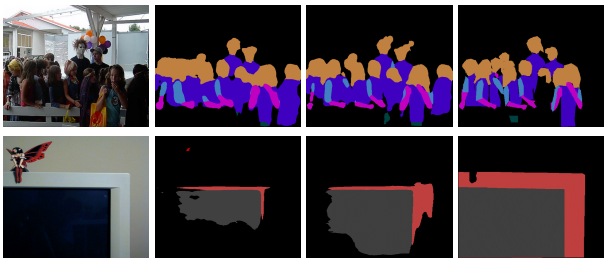

Figure 7. Two typical failure cases. Our model can be sensitive to similar shapes which occupy a small region in the scene. The incomplete appearance of the object can sometimes confuse our model.

of 21 categories, more than other methods compared.

Moreover, We compared the segmentation performance of our methods with the baseline and the reported performances of BSANet [37] on PASCAL-Person-Part benchmark. As shown in Table 5, AFPSNet achieves the highest mIoU results for all the parts, compared to other methods.

Training AFPSNet using the object-to-part training strategy allow us to to increase the batch size to 16, while BSANet, GMNet and CSR using lower batch size, *i.e.*, 8, 10, 4 ,respectively. Enlarge the batch size contributed to speeding up the training process of our model by up to 60% compared to CSR, 33% compared to GMNet and 55% compared to BSANet.

4.4. Failure cases

However, AFPSNet still have limitation. As shown in the first row of Fig. 7, the balloons are mistakenly recognized as the head of a person, showing our model may be sensitive to small parts with similar shapes. A possible way for improvement is by further using the edge information in our method. In addition, our model may be confused if the appearance of the object is incomplete, such as the TV in the second row of Fig. 7. This may be a limitation of the current object-to-part training strategy, which will need further investigation.

5. Conclusion

In this paper, we proposed AFPSNet to address the multi-class part parsing problem, focusing on two main challenges, *i.e.*, part-level and class-level ambiguities. For the part-level ambiguity, we propose to use attention-refinement and feature-fusion to first detect more details of parts from finer scales and then to effectively fuse different scales of features. For the class-level ambiguity, we propose to use the object-to-part training strategy, which helps to exploit prior knowledge for more accurate part localization. Experiments demonstrated the effectiveness of our method, which achieves the state-of-the-art performance on multi-class part parsing on the benchmark PASCAL-Part datasets. In the future, we will consider exploiting the edge information and boundary measures to further improve the boundary localization and using spatial attention to enhance the parts segmentation. Moreover, further investigations of the object-to-part training strategy will also be carried out.

References

[1] Hossein Azizpour and Ivan Laptev. Object detection using strongly-supervised deformable part models. In *European Conference on Computer Vision*, pages 836–849. Springer, 2012.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[3] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[5] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Jian Dong, Qiang Chen, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Towards unified human parsing and pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 843–850, 2014.

[8] S Eslami and Christopher Williams. A generative model for parts-based object segmentation. *Advances in Neural Information Processing Systems*, 25, 2012.

[9] Hao-Shu Fang, Guansong Lu, Xiaolin Fang, Jianwen Xie, Yu-Wing Tai, and Cewu Lu. Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. *arXiv preprint arXiv:1805.04310*, 2018.

[10] Hussein Haggag, Ahmed Abobakr, Mohammed Hossny, and Saeid Nahavandi. Semantic body parts segmentation for quadrupedal animals. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 000855–000860. IEEE, 2016.

[11] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015.

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[14] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5546–5555, 2015.

[15] Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11418–11425, 2020.

[16] Xiaodan Liang, Ke Gong, Xiaohui Shen, and Liang Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):871–885, 2018.

[17] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2402–2414, 2015.

[18] Wenhao Lu, Xiaochen Lian, and Alan Yuille. Parsing semantic parts of cars using graphical models and segment appearance consistency. *arXiv preprint arXiv:1406.2375*, 2014.

[19] Umberto Michieli, Edoardo Borsato, Luca Rossi, and Pietro Zanuttigh. Gmnet: Graph matching network for large scale part semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 397–414. Springer, 2020.

[20] Dong Nie, Jia Xue, and Xiaofeng Ren. Bidirectional pyramid networks for semantic segmentation. In *Proceedings of the asian conference on computer vision*, 2020.

[21] Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 502–517, 2018.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[23] Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019.

[24] Yafei Song, Xiaowu Chen, Jia Li, and Qinping Zhao. Embedding 3d geometric features for rigid object part segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 580–588, 2017.

[25] Jian Sun and Jean Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3407, 2013.

[26] Xin Tan, Jiachen Xu, Zhou Ye, Jinkun Hao, and Lizhuang Ma. Confident semantic ranking loss for part parsing. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[27] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[28] Jianyu Wang and Alan L Yuille. Semantic part segmentation using compositional model combining shape and appear-

ance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1788–1797, 2015.

[29] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Joint object and part segmentation using deep learned potentials. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1573–1581, 2015.

[30] Yang Wang, Duan Tran, Zicheng Liao, and David Forsyth. Discriminative hierarchical part-based models for human parsing and action recognition. *Journal of Machine Learning Research*, 13(10), 2012.

[31] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[32] Fangting Xia, Peng Wang, Liang-Chieh Chen, and Alan L Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *European Conference on Computer Vision*, pages 648–663. Springer, 2016.

[33] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6769–6778, 2017.

[34] Kota Yamaguchi, M Hadi Kiapour, Luis E Ortiz, and Tamara L Berg. Parsing clothing in fashion photographs. In *2012 IEEE Conference on Computer vision and pattern recognition*, pages 3570–3577. IEEE, 2012.

[35] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392. IEEE, 2011.

[36] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[37] Yifan Zhao, Jia Li, Yu Zhang, and Yonghong Tian. Multi-class part parsing with joint boundary-semantic awareness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9177–9186, 2019.

[38] Long Leo Zhu, Yuanhao Chen, Chenxi Lin, and Alan Yuille. Max margin learning of hierarchical configural deformable templates (hcdts) for efficient object parsing and pose estimation. *International journal of computer vision*, 93(1):1–21, 2011.