CAFENet: Class-Agnostic Few-Shot Edge Detection Network (Supplementary Material)

BMVC 2021 Submission # 684

1 Additional Experimental Setup

In this section, we provide detailed information about experimental setup. We implement our framework using the Pytorch library and adopt Scikit-learn library to construct the precision-recall curve and compute average precision (AP). For the encoder, ResNet-34 pretrained on ImageNet is adopted. All parameters except the encoder parameters are learned from scratch. The entire network is trained using the Adam optimizer with weight decay regularization. In both experiments on FSE-1000 and SBD-5^i, we use a learning rate of 10^{-4} and an $l2$ weight decay rate of 10^{-2}. For FSE-1000 experiments, the model is trained with 40,000 episodes and the learning rate is decayed by 0.1 after training 38,000 episodes. For SBD-5^i experiments, 30,000 episodes are used for training, and the learning rate is decayed by 0.1 after training 28,000 episodes. Higher shot training is employed in 1-shot experiments for both datasets. In every experiment of our paper, single NVIDIA GeForce GTX 1080ti GPU is used for computation.

We adopt ImageNet pretrained ResNet-34 with 64-128-256-512 channels for each residual block from [link] as the encoder. To construct skip architecture, we employ the bottleneck block of ResNet as post-processing blocks $S^{(0)} \sim S^{(4)}$. Each bottleneck block consists of two 1x1 convolutional layers and one 3x3 convolutional layer with an expansion rate of 2. Dropout with a ratio of 0.25 is applied to the end of each bottleneck block. For the ASPP Module in front of $S^{(3)}$, we adopt dilation rate of 1,4,7,11. Hyperparameters $\lambda_1, \lambda_2, \lambda_3$ in L_{final} are set to 0.1, 1, 1, respectively. The learnable temperature parameter τ is initialized to 10. For the learning rate, we apply learning rate ten times smaller for the pretrained encoder network and twenty times larger learning rate for the decoder network. For the decoder side, each decoder block is composed of three consecutive 3x3 convolutional layers, and a dropout with the ratio of 0.25 is again located at the end of each layer. During meta-training of CAFENet, we set the number of query samples in training episodes to be 5 for FSE-1000 and 5 for SBD-5^i. During training, we adopt data augmentation with random rotation by multiples of 90 degrees for both SBD-5^i and FSE-1000. We additionally apply zero-padding on data of SBD-5^i to become 512×512, both in training and evaluation. No such zero-padding is performed on images in FSE-1000 since they are already square-shaped. In evaluation, we employ the *average_precision_score* function of Scikit-learn library to measure the Average Precision (AP) score. We compute the AP score for each image and use the average score as a measure of overall performance. For the Maximum F-measure (MF) score, we measure

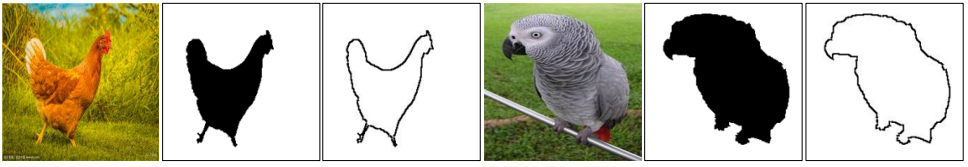

Figure 1: Result of edge label generator.

true positives (TP), false positives (FP) and false negatives (FN) at every 0.01 threshold intervals for each image, and accumulate the values for all images in 1000 test episodes. The MF score is computed using the accumulated TP, FP, and FN values.

2 Label Generator

2.1 Edge Label Generator

Algorithm 1 generates the edge labels from the segmentation labels. The edge label generator finds the regions where the pixel value of a segmentation label drastically changes, and determines the pixels in the regions as the boundary. Note that the pixels at the border of the image are also determined as the boundary.

Algorithm 1 Edge Label Generation

Input: Segmentation label M of an image
Output: Edge label y of an image.

$y \leftarrow 0_{W \times H}$ ▷ Initialize y as zero matrix having same shape with M
for (h,w) in $(1,1)$,...,(H,W) **do** ▷ H/W is height/width of the image
  if $M_{h,w} = 1$ **then**
    for (a,b) = (-r,-r),...,(r,r) **do** ▷ radius r determines thickness of edge
      if $M_{h+a,w+b} = 0$ **then**
        $y_{h,w} \leftarrow 1$ ▷ $0/1$ means non-edge/edge pixel, respectively
        break
      else if $(h+a < 0)$ or $(h+a > H)$ or $(w+b < 0)$ or $(w+b > W)$ **then**
        $y_{h,w} \leftarrow 1$
        break
return y ▷ Return label annotation

2.2 Segmentation Label Generator

Algorithm 2 generates the segmentation label from the edge label. Before the label generation, the pixels are divided into several groups based on boundary labels. We employ the Breadth-First Search (BFS) algorithm and divide the pixels into groups $\{G^1, G^2, ..., G^n\}$. The segmentation label generator of Algorithm 2 classifies these groups into foreground and background. First, the algorithm sweeps each column and row to count the number of pixel value changes in edge label. If there are certain numbers of changes, the algorithm again

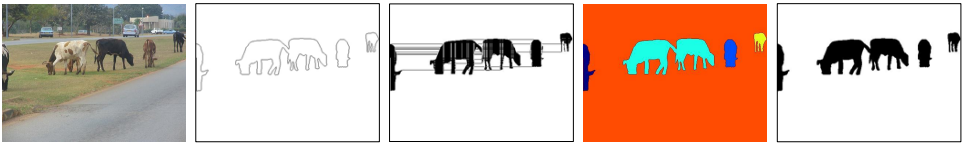

Figure 2: Result of segmentation label generator.

sweeps the column or row and record the location of foreground pixels and mark the foreground pixels in the column or row of matrix T. Based on pixel groups $\{G^1, G^2, ..., G^n\}$, the marking results in T are then divided into pixel value groups $\{T^1, T^2, ..., T^n\}$. The probability that each group G^i belongs to the foreground is calculated as the mean of pixel values T^i. The groups with probability above the threshold λ are determined as the foreground groups, and the pixels belonging to foreground groups are marked as foreground pixels. We set the threshold value λ at $20/255$.

3 Details on Datasets

3.1 FSE-1000

We build FSE-1000 using an existing few-shot segmentation dataset, FSS-1000. We extract the boundary labels from segmentation annotations using Algorithm 1. In the light of difficulty associated with few-shot setting, we set the radius value r in Algorithm 1 to 3 in FSE-1000 and extract edges of which thickness are around $2 \sim 3$ pixels on average. 1000 categories in FSE-1000 are split into 800 train classes and 200 test classes. For the detailed class configuration, the reader may refer to our attachment on class configuration. Figure 1 visualizes the result of our edge extraction algorithm.

3.2 SBD-5^i

SBD-5^i is constructed based on the existing semantic edge detection dataset (SBD). Due to the difficulty of few-shot setting and the noise of boundary annotations in original SBD, we utilize thicker edges as done in FSE-1000. To extract thicker edge, we generate the segmentation labels from the edge labels using Algorithm 2 instead of using existing segmentation labels of SBD. Figure 2 shows the process of generating the segmentation label from the edge label. From the generated segmentation labels, we extract edge labels using Algorithm 1 with a radius of 4, and thicknesses of the extracted edges lie between $3 \sim 4$ pixels on average. This process allows us to train the proposed CAFENet using only the edge labels.

While all images in FSE-1000 are of the same size, images in SBD-5^i have different sizes. However, constructing the training episode as a mini-batch requires images with the same size. Previous works on semantic edge detection typically apply random cropping to deal with this issue. For the few-shot setting, however, random cropping severely degrades informativeness of support set and consequently hinders learning. Alternatively, we utilize the training examples zero-padded to 512×512 to maintain the information of images as much as possible.

Algorithm 2 Segmentation Label Generation

Input: Edge label y of an image, pixel groups $G^1, G^2, ..., G^n$
Output: Segmentation label M of an image.

$M, T \leftarrow 0_{W,H}$ ▷ Initialize M, T as zero matrix having same shape with y
for h = 1,...,H **do** ▷ H is height of the image
  $cnt, mode \leftarrow 0$
  for w = 1,...,W **do** ▷ W is width of the image
    if $y_{h,w} = mod(mode+1, 2)$ **then** ▷ Accumulate changes of pixel value
      $cnt \leftarrow cnt + 1$
      $mode \leftarrow mod(mode+1, 2)$
  if $mod(cnt, 4) = 0$ and $cnt \neq 0$ **then** ▷ Check if there are FG pixels in the row
    $cnt', mode' \leftarrow 0$
    for w' = 1,...,W **do** ▷ Find location of FG pixels in the row
      if $y_{h,w'} = mod(mode'+1, 2)$ **then**
        $cnt' \leftarrow cnt' + 1$
        $mode' \leftarrow mod(mode'+1, 2)$
      if $mod(cnt', 4) = 2$ **then**
        $T_{h,w'} \leftarrow 1$ ▷ Record location of FG pixels in the row
for w = 1,...,W **do** ▷ Repeat the same process for every column
  $cnt, mode \leftarrow 0$
  for h = 1,...,H **do**
    if $y_{h,w} = mod(mode+1, 2)$ **then**
      $cnt \leftarrow cnt + 1$
      $mode \leftarrow mod(mode+1, 2)$
  if $mod(cnt, 4) = 0$ and $cnt \neq 0$ **then**
    $cnt', mode' \leftarrow 0$
    for h' = 1,...,H **do**
      if $y_{h',w} = mod(mode'+1, 2)$ **then**
        $cnt' \leftarrow cnt' + 1$
        $mode' \leftarrow mod(mode'+1, 2)$
      if $mod(cnt', 4) = 2$ **then**
        $T_{h',w} \leftarrow 1$
for $i = 1, ..., n$ **do**
  $T^i \leftarrow T_{h,w|(h,w)\in G^i}$
  if $mean(T^i) \geq \lambda$ **then** ▷ Check the probability that G^i belongs to foreground
    $M_{h,w|(h,w)\in G^i} \leftarrow 1$ ▷ 1 means a foreground pixel
  else
    $M_{h,w|(h,w)\in G^i} \leftarrow 0$ ▷ 0 means a background pixel
return M ▷ Return segmentation annotation

Note that unlike Pascal-5^i, we do not consider the division of training and test samples of the original SBD dataset. As a result, the images in D_{train} might appear in D_{test} with different annotation from class in C_{test}.

4 Ablation Studies on Few-Shot Segmentation Baselines

In the main paper, we utilize the combination of a few-shot segmentator and a non-parametric edge detector as the baseline for few-shot edge detection. We use the few-shot segmentators of PANet and PMM, and the Sobel operator of OpenCV library as an edge detector. In the Sobel operator, the kernel size is an important factor since thickness of the extracted boundary is highly dependent on the kernel size. For fair comparison, we measure the performances of PANet + Sobel and PMM + Sobel with various kernel sizes from 1 to 9. Tables 1 and 2 show experimental results of PANet + Sobel and PMM + Sobel on SBD-5^i. Tables 3 and 4 show the results of PANet + Sobel and PMM + Sobel on FSE-1000. For the PANet + Sobel method, we choose the kernel size of 3 that shows the best performances in all cases in both dataset and report the results with a kernel size 3 in the main paper. For PMM + Sobel, on the other hand, the best performance of each case is obtained from different kernel sizes. For example, in SBD-5^i, PMM + Sobel shows the best MF score with a kernel size of 3 and shows the best AP score with kernel size of 1. Since it is impossible to select the optimal kernel size for the PMM + Sobel baseline, we report the best results with different kernel sizes for all cases instead of the results from a single optimal kernel size.

Metric	kernel size	SBD-5^0		SBD-5^1		SBD-5^2		SBD-5^3		Mean	
MF (ODS)	1	11.94	12.49	18.27	17.93	14.64	14.81	10.33	10.79	13.80	14.01
	3	**18.13**	**19.47**	**23.17**	**23.33**	**21.04**	**21.04**	**17.75**	**17.78**	**20.02**	**20.41**
	5	8.28	8.53	8.54	8.36	8.34	8.29	6.10	5.92	7.82	7.78
	7	4.32	3.99	4.37	4.06	4.66	4.43	2.82	2.86	4.16	3.95
	9	3.08	2.93	3.22	3.06	3.57	3.40	2.82	2.86	3.17	3.06
AP	1	11.35	11.63	14.83	14.11	12.33	11.75	9.33	9.20	11.96	11.67
	3	**11.56**	**11.52**	**14.78**	**14.10**	**12.40**	**11.84**	**9.46**	**9.29**	**12.05**	**11.69**
	5	5.38	5.29	5.54	5.00	5.78	5.63	4.56	4.44	5.32	5.09
	7	2.66	2.22	2.63	2.33	2.89	2.60	1.98	1.94	2.54	2.27
	9	1.62	1.51	1.70	1.56	1.78	1.74	1.44	1.47	1.64	1.57

Table 1: Experiment results of PANet+Sobel with various kernel size on SBD-5^i. Both 5-shot (right) and 1-shot (left) performances are considered. 1000 randomly sampled test episodes are used for evaluation. MF and AP scores are measured by %

Metric	kernel size	SBD-5^0		SBD-5^1		SBD-5^2		SBD-5^3		Mean	
MF (ODS)	1	30.27	30.42	28.46	28.56	25.00	25.58	24.57	24.36	27.08	27.23
	3	**31.18**	**31.73**	**29.23**	**29.99**	**29.38**	**29.91**	**25.65**	**26.03**	**28.86**	**29.42**
	5	20.29	20.64	16.09	16.44	15.53	16.86	12.82	13.57	16.18	16.88
	7	10.98	11.39	8.79	9.08	7.54	8.24	6.82	7.25	8.53	8.99
	9	6.27	6.44	5.58	5.81	4.79	5.02	4.23	4.45	5.22	5.43
AP	**1**	**22.77**	**23.26**	**20.21**	**20.76**	**19.85**	**20.38**	**17.56**	**17.94**	**20.10**	**20.59**
	3	22.14	22.74	20.45	20.48	19.17	20.09	17.86	17.81	19.91	20.28
	5	13.91	14.15	12.08	12.07	10.30	11.01	9.34	9.65	11.41	11.72
	7	7.17	7.43	6.18	6.38	4.66	5.08	4.57	4.83	5.65	5.93
	9	3.62	3.76	3.32	3.48	2.61	2.72	2.38	2.52	2.98	3.12

Table 2: Experiment results of PMM+Sobel with various kernel size on SBD-5^i. Both 5-shot (right) and 1-shot (left) performances are considered. 1000 randomly sampled test episodes are used for evaluation. MF and AP scores are measured by %

Metric	kernel size	1-shot	5-shot
MF (ODS)	1	38.41	39.56
	3	**38.68**	**39.83**
	5	18.58	19.66
	7	10.38	10.21
	9	6.67	6.37
AP	1	31.14	31.81
	3	**28.37**	**29.28**
	5	12.77	13.53
	7	6.72	6.55
	9	3.66	3.38

Table 3: Experiment results of PANet+Sobel with various kernel size on FSE-1000. 1000 randomly sampled test episodes are used for evaluation. MF and AP scores are measured by %

Metric	kernel size	1-shot	5-shot
MF (ODS)	1	31.04	35.25
	3	32.31	**36.53**
	5	**32.39**	31.88
	7	26.86	22.97
	9	12.74	11.19
AP	1	**27.82**	**33.45**
	3	26.67	31.84
	5	24.13	23.89
	7	18.91	15.37
	9	6.96	6.14

Table 4: Experiment results of PMM+Sobel with various kernel size on FSE-1000. 1000 randomly sampled test episodes are used for evaluation. MF and AP scores are measured by %

5 Feature Matching Method for Segmentation

In CAFENet, we introduce a novel multi-split matching regularization (MSMR). During meta-learning stage, MSMR divides the prototypes and query features into multiple sub-vectors, and generates multiple segmentation predictions from the sub-vectors. The segmentation losses are computed from the comparison between multiple prediction results and ground truth, and used for training of the model. However, in MSMR, only the segmentation mask from the original high-dimensional feature vectors is utilized for edge prediction and the multiple segmentation results from sub-vectors are not further utilized. The segmentation mask from the original high-dimensional vectors is concatenated to the query features and becomes input of the DAM and the edge detector in CAFENet. A natural question arises: Can we utilize the multiple segmentation prediction results in edge prediction? To answer this question, we provide additional experiment results with various feature matching methods to generate the segmentation mask used for edge prediction in Tables 5 and 6. The method **baseline** refers to the original method of CAFENet using only the segmentation prediction from the high-dimensional query feature and prototypes. On the other hand, the **average** and **weighted sum** methods utilize the segmentation predictions from low-dimensional sub-vectors with the prediction from high-dimensional vectors. The method **average** generates the final segmentation mask by averaging the segmentation pre-

Metric	feature matching method	SBD-5^0		SBD-5^1		SBD-5^2		SBD-5^3		Mean	
MF (ODS)	average	32.99	38.57	38.86	42.17	33.48	37.68	29.96	**36.07**	33.82	38.62
	weighted sum	33.26	38.18	38.88	41.38	**35.16**	**39.76**	**32.36**	35.68	34.92	38.75
	baseline	**34.92**	**39.02**	**40.83**	**42.52**	34.75	38.41	32.16	35.54	**35.67**	**38.87**
AP	average	28.47	34.27	33.84	37.78	27.85	32.52	24.42	**30.24**	28.65	33.70
	weighted sum	28.96	34.17	34.49	38.31	**30.01**	**34.66**	**26.46**	30.02	29.98	34.29
	baseline	**31.29**	**35.41**	**35.95**	**38.92**	29.32	33.41	25.89	29.73	**30.61**	**34.37**

Table 5: Comparison of different feature mating methods on SBD-5^i. Both 5-shot (right) and 1-shot (left) performances are considered. 1000 randomly sampled test episodes are used for evaluation. MF and AP scores are measured by %

Metric	feature matching method	1-shot	5-shot
MF (ODS)	average	55.52	57.14
	weighted sum	57.13	57.86
	baseline	**57.88**	**59.52**
AP	average	56.42	58.53
	weighted sum	57.56	58.76
	baseline	**58.78**	**60.93**

Table 6: Comparison of different feature mating method on FSE-1000. 1000 randomly sampled test episodes are used for evaluation. MF and AP scores are measured by %

dictions from low-dimensional feature splits and original high-dimensional feature vectors. In the **weighted sum** method, the segmentation masks are combined using a weighted sum with learnable weights. Tables 5 and 6 show the experiment results with SBD-5^i and FSE-1000 datasets, respectively. The experiment results show that the **baseline** method works better than the other methods implying that the proposed multi-split matching method works best when it is used as a regularization method.

6 Comparison of MSMR and Dropout

Among the well-known regularization methods, MSMR is similar to dropout regularization in the sense that MSMR conducts learning using only part of the model. By doing so, we can encourage each part of the model to make the best use of its learning power and obtain the model with the better generalization capability. While dropout and MSMR share the same spirit of using the fraction of the learning capability of the model, they differ in how the model parameters are used during training. Dropout solves a single problem randomly disabling some of the model parameters and excluding them from training. On the other hand, proposed MSMR generates multiple small sub-problems by splitting a high dimensional vector into several low-dimensional sub-vectors, and solves the all sub-problems simultaneously. Thus, the entire parameters are trained at the same time without any excluded parameters in MSMR.

Metric	K	SBD-5^0		SBD-5^1		SBD-5^2		SBD-5^3		Mean	
MF (ODS)	1	34.29	38.67	39.67	41.75	33.31	35.82	31.22	33.91	34.62	37.54
	2	33.49	38.45	40.54	42.25	33.93	37.82	30.81	**35.66**	34.69	38.55
	4	**34.92**	**39.02**	40.83	**42.52**	**34.75**	**38.41**	**32.16**	35.54	**35.67**	**38.87**
	8	33.29	37.82	**41.24**	42.32	34.35	38.29	30.16	33.66	34.76	38.02
AP	1	30.43	34.58	34.89	37.54	27.62	29.94	25.31	27.95	29.56	32.50
	2	29.74	34.35	35.54	37.68	28.67	32.86	25.27	**29.94**	29.81	33.71
	4	**31.29**	**35.41**	35.95	**38.92**	**29.32**	**33.41**	**25.89**	29.73	**30.61**	**34.37**
	8	28.86	34.00	**36.48**	38.32	28.54	33.17	24.73	28.91	29.65	33.60

Table 7: Comparison of different numbers of vector splits K on SBD-5^i under 1-way 5-shot setting. Both 5-shot (right) and 1-shot (left) performances are considered

Metric	Method	SBD-5^0		SBD-5^1		SBD-5^2		SBD-5^3		Mean	
MF (ODS)	Random	**34.92**	**39.02**	**40.83**	**42.52**	**34.75**	**38.41**	**32.16**	**35.54**	**35.67**	**38.87**
	Deterministic	33.76	38.56	40.73	42.18	32.78	37.92	29.62	34.55	34.22	38.30
AP	Random	**31.29**	**35.41**	**35.95**	**38.92**	**29.32**	**33.41**	**25.89**	29.73	**30.61**	**34.37**
	Deterministic	30.14	34.74	35.82	37.98	27.55	32.97	24.47	**30.03**	29.50	33.93

Table 8: Comparison of Ramdom spliting scheme with deterministic spliting on SBD-5^i under 1-way 5-shot setting. Both 5-shot (right) and 1-shot (left) performances are considered

7 Additional Ablation Studies on MSMR

7.1 Number of vector splits

MSMR divides the high-dimensional feature into multiple low-dimensional splits. Table 7 shows the performance of proposed CAFENet with varying numbers of splits K. Comparing the K = 1 case with other cases, we can see that applying MSMR regularization consistently improves performance. We can see that K = 4 results in the best AP and MF performance. The performance gain is relatively marginal when we divide the embedding dimension into too small (K = 8) or too big (K = 2) a pieces.

7.2 Dimensional split via deterministic indices

In MSMR, high-dimensional feature vectors are randomly split into several feature sub-vectors. We also analyze the performance of the deterministic spliting scheme: it contrasts with MSMR, where the index to be used for each split is changed randomly every time. Table 8 shows that even when the same number of vector splits are considred, ramdomly sampling indices used for split indeed improves the performance over the deterministic counterpart. Deterministic spliting enforces a set of fixed projections from high-dimensional spaces to lower-dimensional spaces. We speculate that this kind of fixed projections discard a lot of information within each dimension and lead to the loss of expressive power.

8 Additional Qualitative Results

In Fig. 3, we illustrate the prediction results of the baseline, **Seg**, **Seg + DAM**, and **Seg + DAM + MSMR** methods. For fair comparison, all methods share the same support set. Fig.

3 clearly shows that the techniques proposed in CAFENet steadily improve the quality of prediction.

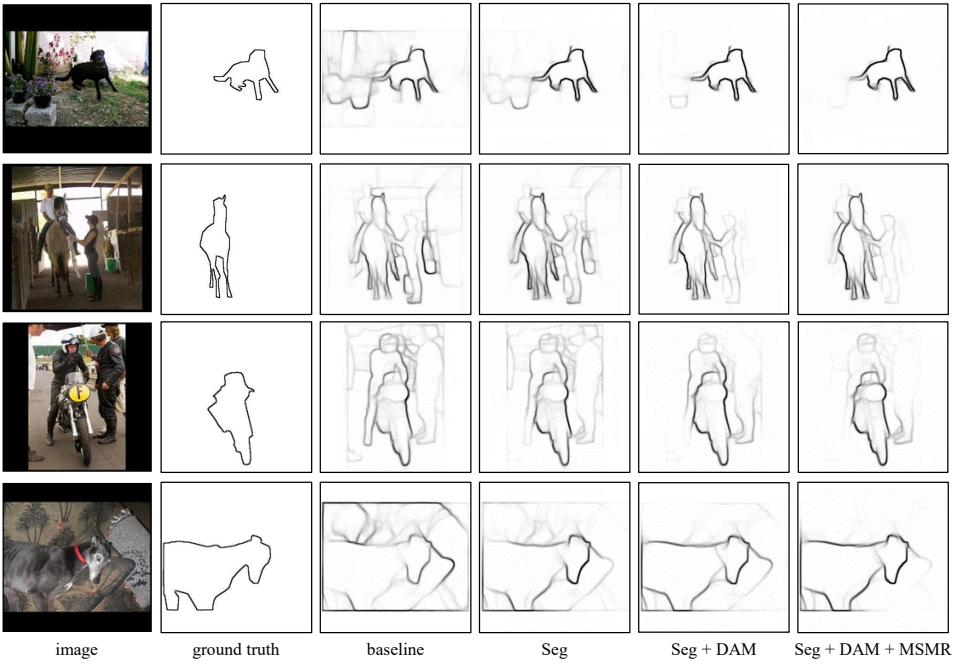

Figure 3: Qualitative comparison of ablation experiments