SCIENCE CHINA
Information Sciences

March 2021, Vol. 64 130105:1–130105:12
https://doi.org/10.1007/s11432-020-3065-4

• RESEARCH PAPER •
Special Focus on Weakly Supervised Learning

Deep graph cut network for weakly-supervised semantic segmentation

Jiapei FENG, Xinggang WANG* & Wenyu LIU

School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430000, *China*

Received 15 March 2020/Revised 13 June 2020/Accepted 30 July 2020/Published online 7 February 2021

Abstract The scarcity of fully-annotated data becomes the biggest obstacle that prevents many deep learning approaches from widely applied. Weakly-supervised visual learning which can utilize inexact annotations is developed rapidly to remedy such a situation. In this paper, we study the weakly-supervised task achieving pixel-level semantic segmentation only with image-level labels as supervision. Different from other methods, our approach tries to transform the weakly-supervised visual learning problem into a semi-supervised visual learning problem and then utilizes semi-supervised learning methods to solve it. Utilizing this transformation, we can adopt effective semi-supervised methods to perform transductive learning with context information. In the semi-supervised learning module, we propose to use the graph cut algorithm to label more supervision from the activation seeds generated from a classification network. The generated labels can provide the segmentation model with effective supervision information; moreover, the graph cut module can benefit from features extracted by the segmentation model. Then, each of them updates and optimizes the other iteratively until convergence. Experiment results on PASCAL VOC and COCO benchmarks demonstrate the effectiveness of the proposed deep graph cut algorithm for weakly-supervised semantic segmentation.

Keywords semantic segmentation, weakly-supervised learning, semi-supervised learning, graph cut

Citation Feng J P, Wang X G, Liu W Y. Deep graph cut network for weakly-supervised semantic segmentation. Sci China Inf Sci, 2021, 64(3): 130105, https://doi.org/10.1007/s11432-020-3065-4

1 Introduction

Semantic segmentation which aims at assigning a semantic label to each pixel of the image is a fundamental problem in computer vision. Owing to the rapid development of deep convolutional neural network (DCNN) [1], there has made great progress when a large amount of fully-annotated data is available [2,3]. However, fully-annotated samples are scarce because of their expensive labeling costs, which become the bottleneck of the practical development of semantic segmentation in real applications. Weakly-supervised learning methods which use annotations less detailed than the accurate pixel-level annotations are proposed to solve such a problem [4, 5]. The concept of weakly-supervised learning contains a great variety of learning problems for different tasks, such as learning from inexact supervision, inaccurate supervision, and incomplete supervision [6]. It has been proven to be effective for text categorization [7], object detection [8], medical image analysis [9], etc. In this paper, we study the problem of weakly-supervised semantic segmentation, which aims at learning fine-grained visual recognition models, such as semantic segmentation model, when only given coarse annotations, i.e., image-level annotations.

Pixel-level annotations provide precise locations of objects and boundaries, while image-level annotations only indicate whether a particular class of objects exists in the image and do not provide any information about their locations or boundaries. To fill the gap of missing position information between pixel-level and image-level annotations, most weakly-supervised methods utilize image-level labels to train a deep classification network and then use the classification activation maps (CAM) [10] algorithm to select the most discriminative regions as localization maps. The discriminative regions are usually small and sparse because the classification network only can focus on the most significant part of the object,

* Corresponding author (email: xgwang@hust.edu.com)

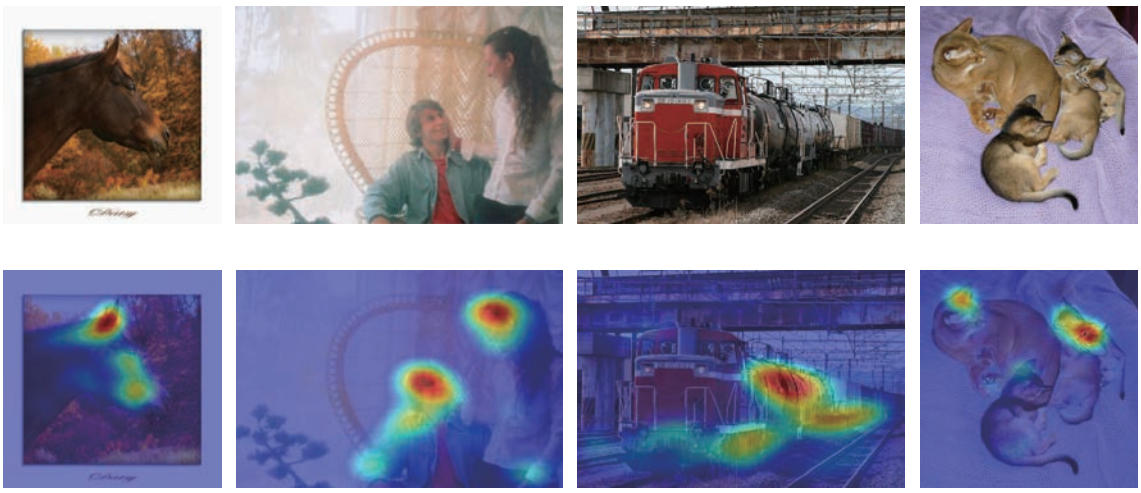

Figure 1 (Color online) The top row shows the original images and the bottom row shows the CAM results which only localize the most discriminative regions of objects.

as shown in Figure 1. But the superior performance on semantic segmentation requires accurate and complete fine-grained supervision to guide segmentation models. Therefore, there are a lot of different strategies emerging to solve the mismatch problem between the found discriminative regions and the ideal supervision. Some methods drive the classification network to focus on the whole object by the erasing strategy [11,12]. Some methods produce dense and reliable object localization maps with multiple dilated convolutional blocks [7]. Others explore the affinities between pixels and then use the obtained relationship to reconstruct the proxy ground truth [13] or pass category messages between pixels [14].

Although some promising results have been obtained, a big performance gap between weakly-supervised approaches and fully-supervised approaches still exits. In this paper, what we try to find is a method expanding from the discriminative but sparse regions to cover the whole objects during the training phase. For each image, we use the found discriminative regions as labeled pixels while other regions as unlabeled pixels. Besides, we observe that the labeled pixels with some context information can be used to better predict the categories of unlabeled pixels. This is based on a smoothness assumption: points that are close to each other are more likely to share the same label. We combine the low-level features and the high-level features, and then embed the pixels to a feature space. Finding the neighbors of each pixel helps passing messages from the labeled pixels to the unlabeled pixels. Then we can regard this extension process as a semi-supervised problem.

We propose to use the classical graph cut algorithm [15] to model this process for generating more accurate pixel-level supervision. Note the task that we are facing is a multi-class segmentation task. However, the graph cut algorithm [15] is a binary segmentation method and divides an image into the foreground part and the background part. Under such circumstances, we carry out the graph cut algorithm for each class existing in the image step by step and then merge all the segmentation results into the final one. For one certain class, firstly we need to build a graph using the deep feature maps. We take the pixel set as the vertex set and the similarity between pixels as the weight of edges between vertexes. Secondly, we can obtain the probability that each pixel belongs to a certain class from the output segmentation map. Besides, we use the features (both low-level and high-level features) extracted from the segmentation network to measure the similarity between pixel points. So, each pixel contains two types of messages: one is the probability from the network and the other one is the label information from adjacent pixels. When judging the class of one pixel, the two types of messages complement each other to make a better prediction. These messages are represented by the weights of the edges in the graph. After that, we perform the max-flow algorithm (also can be called the min-cut algorithm) [16] in the constructed graph to obtain the segmentation result which is used as the proxy ground truth to train semantic segmentation network. We name the proposed method as a deep graph cut network (DGCN) for weakly-supervised semantic segmentation.

The main contributions of this paper can be summarized as follows: (1) We propose a graph cut neural network, which enables the network to accurately generate new pixel-level labels for weakly-supervised semantic segmentation. (2) The deep graph cut network obtains strong weakly-supervised

semantic segmentation performance on the PASCAL VOC and COCO segmentation benchmarks, which demonstrates the effectiveness of the proposed deep graph cut module for generating supervisions. (3) We build connections between weakly-supervised semantic segmentation and semi-supervised learning. Besides of the instantiation of the deep graph cut algorithm, more semi-supervised learning methods can be applied to solve this problem.

The paper is organized as follows. In Section 2, we first review the related work. Then in Section 3 we describe our proposed method. Experiments will be presented in Section 4. At last, Section 5 provides our conclusion and future work.

2 Related work

In recent years, deep learning has made breakthroughs in semantic segmentation tasks. However, the expensive cost of obtaining pixel-level annotations limits the development of application in real life. Thus weakly-supervised semantic segmentation has been proposed to use more weak annotations to reduce human labeling cost. There has been a lot of weakly-supervised semantic segmentation methods emerging in the last few years with promising performance. Various weak annotations have been adopted in this research field, such as bounding boxes [17–19], scribbles [20], points [21], and image-level labels [5,11,13]. Moreover, some research studies, e.g., [22, 23], improve the performance with additional and unlabeled data. Usually, the data are obtained from the Internet called web data. So these are also called webly-supervised segmentation methods [22]. In this paper, we utilize the image-level labels which are very cheap to obtain and do not provide any localization information about the object in the image.

2.1 Pixel-level semantic segmentation with image-level annotations

Image-level label is one of the weakest annotations. Such kind of annotation can be used widely for its cheap cost. As mentioned above, image-level labels do not give any message about where objects exist. A method named CAM [10] is adopted to identify the discriminative parts of objects and generate localization maps called the "seed". However, the generated seed cannot directly be the proxy ground truth because it has missed most regions of the objects.

There have been several techniques proposed to expand from the seed regions to cover the whole objects. Erasing [11, 12] is a common technique used to drive the classification network to discover other supplementary object regions by erasing the found activated regions. Wei et al. [12] put forward adversarial erasing which trains multiple classifiers with the erased images. By doing this, they force the classification network to seek other discriminative regions in the erased images. Although the method can expand seed to cover the whole object, it requires repetitive training and erasure steps. FickleNet [11] uses the modified dropout technique to randomly select the activation area of feature and has a different receptive field in every forward step. However, to obtain different discriminative regions and cover the whole object, they have to do localization multiple times for one image in a single inference phase. SSENet [24] proposes the scale equivariant regularization to improve the quality of the seed. AffinityNet [13] utilizes the seed generated by CAM as proxy labels to train an affinity network and grows regions based on the found relationship between the pixels. There are also other approaches [14] focusing on exploring the relationship between pixels. RRM [25] proposes a one-step solution to handle the task and also put forward the two-step version which gets a new state-of-the-art performance. The method also considers using multi-scale original images to generate the better seed. Those methods generate static supervision and have no chance to correct the true negative regions during the training phase.

Different from them, DSRG [5] and our proposed method can generate dynamic supervision and fix it during each iteration, thus are more flexible. Moreover, we do have made several improvements in DSRG. First, DSRG expands from the seed cues region and cannot pass messages between two instances of the same class (the two instances are not usually adjacent to each other). However, our proposed method can handle this because we embed each pixel into the feature space and pass messages between neighboring nodes in feature space rather than spatial space. Second, DSRG only does hard thresholding for the segmentation map and then uses the results as the similarity criterion. In contrast, we merge the features from multiple layers of the backbone network to consider the similarity criterion, which seems more accurate.

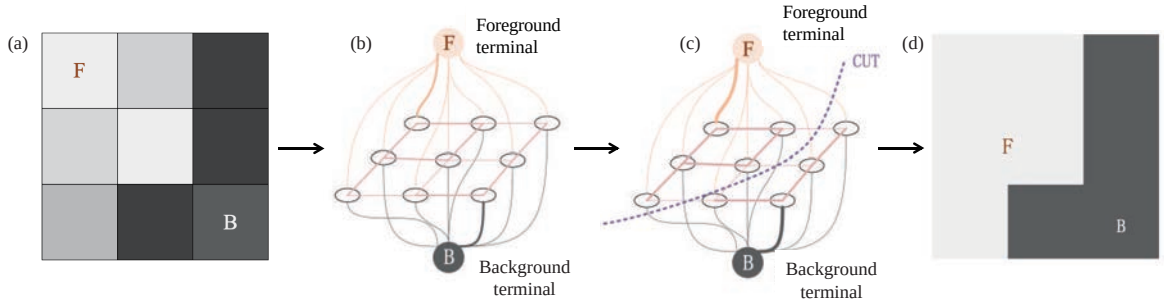

Figure 2 (Color online) (a) A simple 3×3 image with two seed pixels. The pixel marked with "F" represents "foreground" seed while the other one represents "background" seed, and the remaining pixels are unknown. (b) The constructed graph in which the cost of each edge is reflected by the edge's thickness. (c) The min-cut algorithm tends to cut the edges with minimum costs. (d) Segmentation results.

2.2 Classical graph cut algorithm

The graph cut algorithm [15] can be seen as a semi-supervised approach to segment image into the foreground part and background part. To identify the two categories, there are some pixels marked beforehand as "foreground" and "background" seeds. We construct an undirected graph that contains a set of nodes (pixels) and a set of edges. There are two special nodes called terminals, while others called normal nodes. The normal nodes represent pixels and the edges represent the relationship between those pixels. Besides, the weights of edges reflect the similarity between pixels and it usually takes the position and intensity feature of pixels into account. Then we use the max-flow algorithm (equivalent to the min-cut algorithm) [16] to generate a segmentation result that is optimal based on the edge weights. Figure 2 shows a simple example of utilizing the graph cut algorithm. Gao et al. [26] proposed graph-without-cut (GWC) for image segmentation. GWC first generates a set of superpixels and then learns the similarity matrix between each superpixel based on different features as well as the spatial information. The features are extracted from those superpixels. Finally, it uses the rank constraint on the Laplacian matrix of the learned similarity graph to achieve the optimal segmentation results. However, we try to build the graph whose weights are based on the features extracted from the neural network. Besides, GWC predicts the categories of all the pixels directly while our graph cut module keeps some pixels unknown to avoid introducing noisy labels in early training iterations.

3 Method

In this section, we introduce the details of our proposed deep graph cut network for weakly-supervised semantic segmentation. At first, we will transfer the weakly-supervised semantic segmentation task into the problem of training a segmentation network in a semi-supervised manner. Secondly, we will introduce how to obtain the "seed" information for semi-supervised learning from a deep classification network. Thirdly, we give the details of graph-cut based semi-supervised learning in the segmentation network.

3.1 Problem transformation

For weakly-supervised semantic segmentation, most methods use CAM to locate the discriminative regions of objects as the localization information. We have observed that using the discriminative regions as segmentation supervision directly will obtain poor performance. This is possible because CAM usually ignores most regions of the object and generates incomplete supervision for the segmentation model. To solve such a problem, the SEC method [4] proposes to use "seeding loss" to implicitly expand from the incomplete supervision to the uncertain regions in the image. But we explicitly expand with semi-supervised learning methods. Moreover, the discriminative object regions are the labeled part, while the others are the unlabeled part. We hope to use the labeled pixels with context information to infer the categories of unlabeled pixels through a semi-supervised method. Based on the above, we change the weakly-supervised semantic segmentation problem into a semi-supervised learning problem.

3.2 Initial seed generation

Image-level annotations cannot explicitly provide the localization and boundary information of objects. But recently, some studies have shown a deep classification network could obtain the discriminative regions of objects with image-level labels [10, 27, 28]. To avoid the missing of spatial information in the full-connected layer, the classification network is fully convolutional and the discriminative object regions' position is preserved in the deep layers of the network.

We use the CAM method to locate the foreground regions. First, we use a modified VGG16 network which is pre-trained on ImageNet as our deep classification network. In this network, we adopt the global average pooling (GAP) on the convolutional feature maps and use the output for a fully-connected layer to produce the final prediction. And we compute the weighted sum of feature maps for the last convolution layer to get our class activation maps. Finally, we apply a hard threshold to the heatmap and obtain the discriminative regions as foreground seed.

In addition to locating the foreground, we also need to get the background regions as seed. To localize background, we utilize the saliency detection technique [29] and select the region whose pixels have lower saliency values in the normalized saliency map as the background. We blend the foreground and background regions to gain the final seed cues used in the paper.

3.3 Graph cut

In this subsection, we describe how to apply the graph cut algorithm in our method. We give the notations as follows.

Let I be one of the training images, and we extract its multi-layer feature maps from the backbone network. For example, we use the VGG16 network as the backbone network and its multi-layer outputs are taken as the features of I. The feature maps are demoted as $\{f_1, f_2, \ldots, f_k\}$ where k is the total number of the feature maps. From 1 to k, the size of f_k becomes smaller. When aggregating the multi-scale feature maps, we normalize the spatial sizes of the feature maps to be the size of f_k through bilinear interpolation.

After that, these feature maps are aggregated to be the output feature map denoted by X, $X \in \mathbb{R}^{(C\times H\times W)}$, as shown in Figure 3. Since each X is a 3D tensor, we can use one subscript to index its spatial coordinates, i.e., $X_i \in \mathbb{R}^C$ denotes the channel-vector at spatial position i, where $i \in \{1, 2, \ldots, HW\}$. In the following, we use the term node to refer X_i.

KNN matrix. To compute the K-nearest neighbors matrix for the nodes of one image, the first step is to calculate the distance between these nodes, and then we can obtain the K-nearest neighbors for each node. For example, given two nodes X_i and X_j which have the same dimension of channel-vector, the distance between them is defined in terms of their ℓ^2 distance:

$$D_{ij} = \|X_i - X_j\|_2. \tag{1}$$

The larger distance they have, the smaller similarity they have. Then we construct the matrix Q by selecting the top-K nearest neighbors of each node:

$$Q_{ij} = \begin{cases} 1, & X_j \in \text{knnd}(X_i), \\ 0, & \text{others}, \end{cases} \tag{2}$$

where $\text{knnd}(X_i)$ represents the set of K-nearest neighbors of the sample X_i. K is determined empirically and $Q \in \mathbb{R}^{(HW\times HW)}$.

Graph cut. According to the characteristic of the classical graph cut algorithm, we take this algorithm for each class (excluding background class) existed in the image and regard the foreground part of the segmentation result as the proxy ground truth of the specific class. As for the background class, we use the seed cues as the proxy ground truth. In the following, we will introduce how this algorithm is used in our method.

At first, we define the affinity graph in our weakly supervised semantic segmentation network. A graph usually consists of a vertex set and an edge set, denoted by $G = (V, E)$. The vertex set V includes the normal nodes which have been defined in Subsection 2.2 and two special nodes called terminals which can be seen as "Foreground Terminal" and "Background Terminal" shown in Figure 2. The edge set E is defined as the set of undirected edges that connect these vertexes. Each edge $\varepsilon \in E$ in the

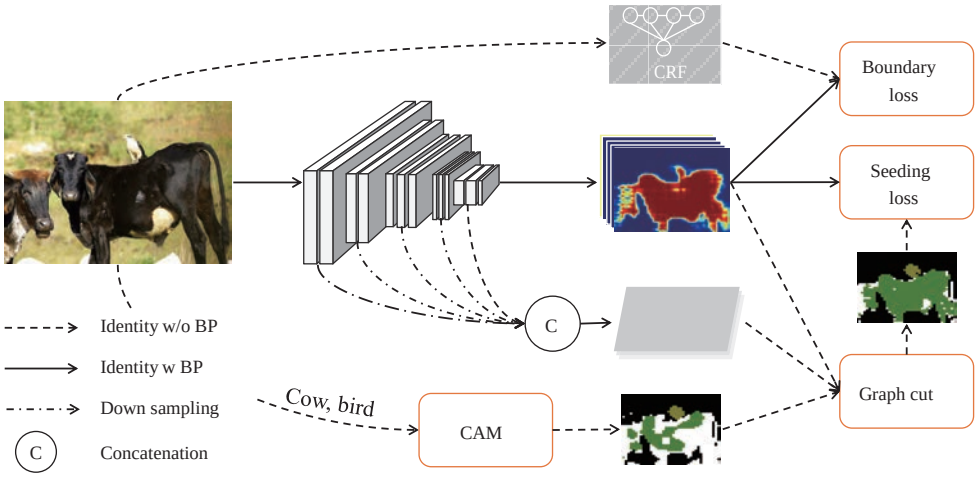

Figure 3 (Color online) The framework of a deep graph cut network for weakly supervised semantic segmentation. Utilizing image-level labels we can train a classification network to localize a part of objects as seed cues. Then the graph cut module takes the seed cues, feature maps extracted from the backbone and segmentation map as input produces the more accurate pixel-level supervision. We train our segmentation network with two loss functions.

Table 1 Weights of edges in E

Edge	Weight	For
$\{p, q\}$	C_0	$W_{pq} \neq 0$
$\{p, S\}$	$-\log(H_u^c)$	$p \in P, p \notin F \cup B$
	C_1	$p \in F$
	0	$p \in B$
$\{p, T\}$	$-\log(1 - H_u^c)$	$p \in P, p \notin F \cup B$
	0	$p \in F$
	C_1	$p \in B$

graph is assigned a nonnegative weight ω_ε. The weights of edges between normal nodes represent the affinities between normal nodes, while the weights of edges between a normal node and terminal reflect the probability of belonging to the specific class. The nodes of the graph can represent pixels and the edges can represent any neighborhood relationship between the pixels.

Secondly, we will discuss the specific numerical setting. For the obtained feature map $X \in \mathbb{R}^{(C\times H\times W)}$, $H\times W$ normal nodes form the vertex set called P. As stated above, there are two additional nodes reflecting the class information: a "foreground" terminal (called S) and a "background" terminal (called T). Then, the vertex set V can be expressed as the union of three types of nodes:

$$V = P \cup \{S, T\}. \tag{3}$$

The set of edges E consists of two types of undirected edges: n-links (neighborhood links) and t-links (terminal links). Each node p has two t-links $\{p, S\}$ and $\{p, T\}$ connecting with each terminal. As for n-links, each node p has K n-links $\{p, \text{knnd}\{p\}\}$. Moreover, n-links edges are constructed with the K-nearest neighbor matrix Q and represent the affinities between different normal nodes. Table 1 gives the weights of edges in E of our graph G_c.

In order to draw the segmentation boundary between the foreground part and the background part, we need to find the minimum cost cut on the graph G_c by the "max-flow" algorithm. This is a binary classification method, but our segmentation task is aimed at multiple classes. Considering the mismatch problem, we propose to treat the multi-class classification task as multiple binary classification tasks to solve. In the method, graphs are constructed for specific classes (excluding background) existing in the image.

In Table 1, C_0 and C_1 are prespecified constants, and H is the segmentation map. H_u denotes the predicted probability at position u. Then the graph G_c is now completely defined (G_c denotes that the graph is made for class c).

After completing the graph cut algorithm for all classes in the image, we generate the final supervision with a given threshold θ in order to avoid excessive segmentation. When the predicted probability of one

node at a certain class is greater than θ and simultaneously the node is assigned to the foreground by the graph cut algorithm, it belongs to the certain class in our generated segmentation supervision.

3.4 Loss function

The loss function used in this paper is consistent with DSRG during the experiment. It consists of two parts: one is a weighted cross-entropy loss between the supervision generated by the graph cut module and the segmentation map which is the output of the segmentation network, called seeding loss; and the other one considers using conditional random field (CRF) to smooth the boundaries of objects. A Kullback-Leibler divergence loss function between the smoothed results and the segmentation map is called boundary loss.

The cross-entropy loss function is commonly used in neural networks to describe the distance between the prediction and the ground truth. And in our settings, the segmentation network's prediction only matches the pixels whose labels have been generated by the graph cut module while ignoring the rest pixels in the image. There is an unbalanced distribution in the amount of foreground and background pixels which may influence the segmentation performance. Therefore, we need to balance loss on the foreground and background with two normalization coefficients. Let C be the set of classes which exist in the image (excluding background) and $\bar{C}$ be the background. And S_c is defined as the set of positions that are labeled as class c. The balanced seeding loss is defined as follows:

$$L_{\text{seed}} = -\frac{1}{\sum_{c\in C}|S_c|}\sum_{c\in C}\sum_{u\in S_c}\log H_{u,c} - \frac{1}{\sum_{c\in \bar{C}}|S_c|}\sum_{c\in \bar{C}}\sum_{u\in S_c}H_{u,c}. \tag{4}$$

Besides, $H_{u,c}$ denotes the probability of class c at position u of the segmentation map.

Kullback-Leibler divergence is also a measure of the distance between two probability distributions. We utilize it to calculate the differences between the segmentation map and the result after the CRF's refinement. We apply the fully-connected CRF to smooth the segmentation results with unary potential and pairwise potential, which are determined by the segmentation results and color information respectively. The smoothed results are devoted by $Q_{u,c}(I,H)$. I is the original image which has been cropped to the same size as the segmentation mask. The boundary loss function is defined as follows:

$$L_{\text{boundary}} = \frac{1}{n}\sum_{u=1}^{n}\sum_{c\in C}Q_{u,c}(I,H)\cdot\log\frac{Q_{u,c}(I,H)}{H_{u,c}}. \tag{5}$$

4 Experiments

4.1 Experimental setup

Dataset. We evaluate the proposed method on the PASCAL VOC 2012 image segmentation benchmark dataset [30] and the MS COCO dataset [31].

PASCAL VOC 2012. It contains three parts: training data (1464 images), validation data (1449 images), and testing data (1456 images). Using the common practice of weakly supervised semantic segmentation on this dataset [30], we augment the training part by augmented training images from [32]. In our experiments, only image-level labels are utilized for training. We compare our method with other state-of-the-arts on both the validation and test sets. The standard mean intersection over union (mIoU) criterion and pixel-wise accuracy are adopted for evaluation on the validation set. The result on the test set is obtained on the official PASCAL VOC evaluation server.

COCO. The COCO dataset released in 2014 contains 82783 training and 40504 validation images. We use the pixel IoU averaged on 81 categories as the evaluation metric.

Training/testing settings. We adopt the slightly modified version of the VGG16 network from [4] for the classification network. We choose ResNet-101 and ResNet-38 as the backbone network in our experiments, following the setting of [5,13]. They are all initialized by the model pre-trained on ImageNet. Stochastic gradient descent (SGD) with mini-batch is used for training classification and segmentation networks. We use the momentum of 0.9 and a weight decay of 0.0005. The batch size is 16, and the dropout rate is 0.5. The initial learning rate is 5E−4 and we employ a step learning rate policy where the initial learning rate is multiplied by gamma = 0.3 after every three epochs.

Table 2 Comparison of weakly-supervised semantic segmentation methods on PASCAL VOC 2012 validation and test image sets

Method	Supervision	Training set	Val	Test
FCN† [2]	Pixel-level	9k	–	62.2
Deeplab† [33]		10k	67.6	70.3
BoxSup† [17]	Box-level	10k	62.0	64.6
ScribbleSup† [34]	Scribble-level	10k	63.1	–
SEC† [4]	Image-level	10k	50.7	51.1
AE-PSL† [12]		10k	55.0	55.7
MCOF [35]		10k	60.3	61.2
DCSP [36]		10k	60.8	61.9
SeeNet† [37]		10k	61.1	60.7
SeeNet [37]		10k	63.1	62.8
DSRG [5]		10k	61.4	63.2
AffinityNet† [13]		10k	58.4	60.5
AffinityNet* [13]		10k	61.7	63.7
CIAN [14]		10k	64.1	64.7
SSENet* [24]		10k	63.3	64.9
FickleNet [11]		10k	64.9	65.3
SSDD* [38]		10k	64.9	65.5
RRM [25]		10k	**66.3**	**66.5**
DGCN*		10k	64.0	64.6

For initial seed generation and CRF operation, we use the same setting as DSRG. Besides, in the experiments, $K = 9$. Then we adopt the value 0.7 as the threshold θ in updating seed. In the test phase, we predict a segmentation probability map for each test image using the trained segmentation network. Then we upsample the predicted segmentation probability map with bilinear interpolation to match the size of the input image and apply a fully-connected CRF to refine the segmentation result.

4.2 Comparisons to the state-of-the-arts

In Table 2 [2,4,5,11–14,17,24,25,33–38], we compared our approach with other recently introduced weakly-supervised semantic segmentation methods on the PASCAL VOC validation and test sets. Besides, we give the results of two early fully-supervised methods. Note that FCN [2] uses the PASCAL VOC 2011 dataset which contains 1112 training images and the augmented 8498 images from [32]. Except for FCN [2], other methods use the same dataset as we do. We also give the results of a typical box-based method [17] and a scribble-based method [34]. Methods marked by † use VGG16 as backbone and methods marked by $*$ use ResNet-38 as the backbone network. Others use ResNet-101 as the backbone network.

Our method uses the ResNet-38 as backbone and achieves mIoU values of 64.0 and 64.6 on the validation set and test set, respectively. Other state-of-the-art methods are shown in Table 2. AffinityNet [13] explores pairwise semantic similarity and then utilizes the message to propagate semantic labels from the labeled seed to the full image. But it uses an extra network as the AffinityNet. CIAN [14] passes messages between multiple images which have objects of the same class. SSENet fuses seed from different scale images. FickleNet [11] uses a modified dropout strategy to drive the classification network to focus on different parts of objects rather than the previous discriminative parts. But to obtain the final CAM results it takes hundreds of inferences under random obtained dropout strategies. SSDD [38] takes AffinityNet as the baseline and then proposes a self-supervised difference detection module to explore the differences between seeds and segmentation maps smoothed by CRF. Then it uses the trained module to refine the segmentation results. The whole framework for this approach is complex. RRM [25] firstly obtains the multi-scale CAMs and then uses CRF to refine the seed. Next, it proposes a novel dense energy loss to train the segmentation model. Our proposed method uses the graph cut module to generate the dynamic proxy ground truth as the supervision of the segmentation model. Our model cannot achieve the same performance as the newest state-of-the-art. This is probably due to the precision of the seed. The seed is the prior of the graph cut module and its precision determines the precision of the generated proxy ground truth. The bottleneck in our approach is the precision of seed. Compared to the newest state-of-the-art method (RRM), we use a lower performance seed. So we emphasize that our method can

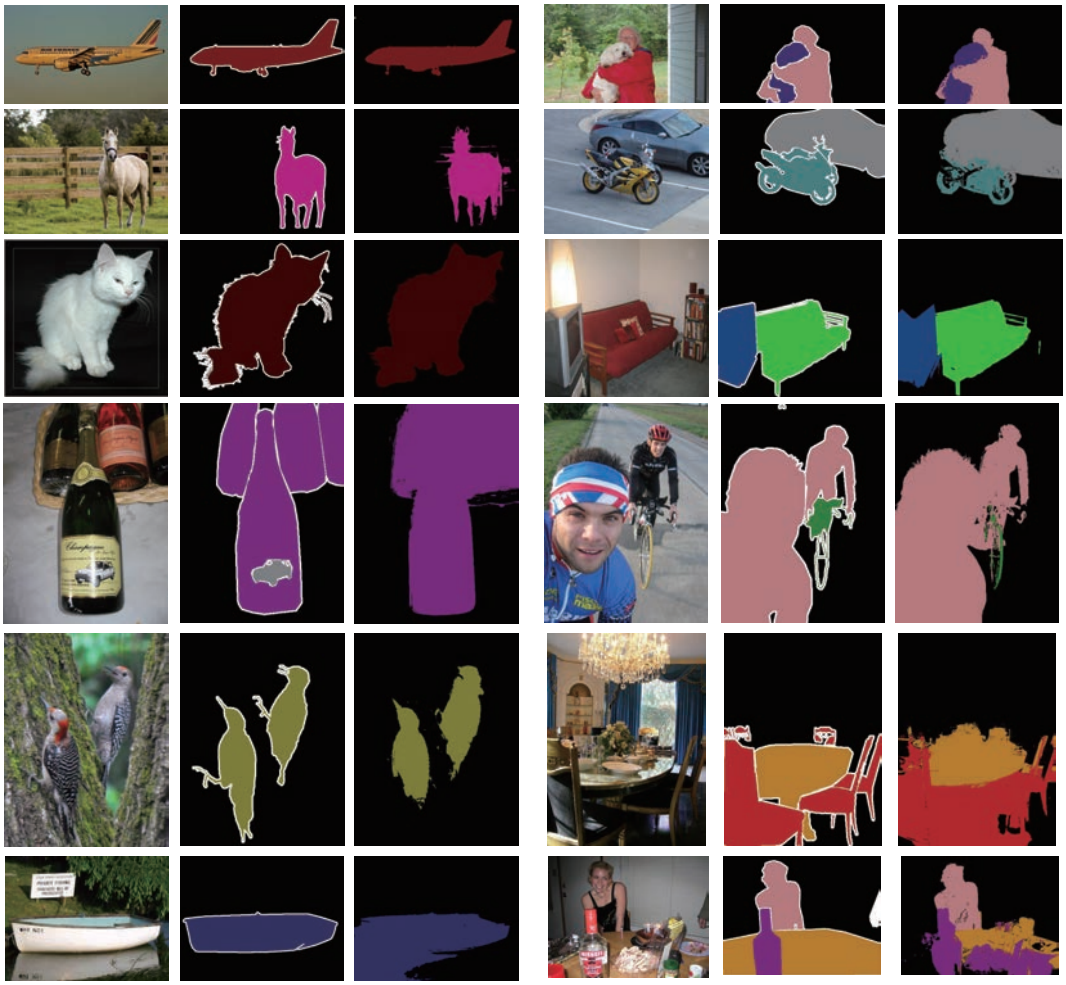

Figure 4 (Color online) Qualitative segmentation results on the PASCAL VOC 2012 validation set. Two failure cases are shown in the last row.

obtain relative improvement compared to our baseline. In addition to this graph cut algorithm, other semi-supervised methods can achieve effective improvements (e.g., label propagation [39]).

Qualitative results. Some successful segmentation results are shown in Figure 4. It can be seen that our method can produce accurate segmentation results only with image-level annotations. We can obtain fine-grained results for most of the images, even some complicated images. However, there are also some failure cases and we give two failure cases in the bottom row of Figure 4. These failure modes contain that the network cannot separate the object region from the background precisely and the boundary between adjacent objects is not clear. The first mode is typical for weakly-supervised systems, strongly relevant categories (such as train and rails, boat and water) cannot be separated without fine-grained annotations. The second mode is because the boundary of adjacent objects becomes obscure after smoothing the segmentation result.

4.3 Ablation studies

To further verify the effectiveness of different components, we conduct multiple ablation experiments with different settings. In Table 3, we reproduce DSRG based on PyTorch [40] as our baseline. The "+ Graph Cut" denotes replacing the seed region growing module with the the proposed graph cut module. The results show that our graph cut module provides 1.5% mIoU performance gain over the DSRG baseline. It can be observed that the graph cut module obtains large improvements in classes of airplane and table, in which object sizes are large. Since the graph cut module can generate larger range pseudo supervision than the DSRG method, we obtain significant performance gain on these classes. The "+ ResNet-38" denotes replacing the backbone ResNet-101 with ResNet-38 [41]. Besides, we follow DSRG [5] which uses

Table 3 Comparison of mIoU of our approach with different settings on PASCAL VOC 2012 validation set

Method	mIoU	bkg	airplane	bike	bird	boat	bottle	bus	car	cat	chair
DSRG	59.3	87.0	65.3	32.5	71.1	38.2	66.9	78.1	68.4	80.4	**27.6**
+ Graph Cut	60.8	87.6	72.0	34.5	71.5	39.1	67.3	80.3	70.5	**80.7**	24.7
+ ResNet-38	62.7	88.0	76.4	35.2	76.9	44.7	72.8	**82.5**	74.6	79.0	25.4
+ Retrain	**64.0**	**88.7**	**77.8**	**35.9**	**78.5**	**45.8**	**73.7**	82.1	**76.1**	79.6	26.6
Method	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
DSRG	63.1	19.1	**76.4**	67.9	68.7	70.8	42.0	76.2	33.8	52.1	59.1
+ Graph Cut	63.8	29.9	73.1	69.5	69.9	71.8	43.1	79.2	33.9	**55.7**	59.0
+ ResNet-38	71.6	29.2	74.4	71.4	**71.9**	70.8	49.5	76.5	34.0	51.5	60.1
+ Retrain	**73.2**	**30.5**	75.6	**72.9**	71.2	**73.1**	**50.8**	**79.7**	**35.3**	55.2	**61.3**

Table 4 Performance with different values of K on PASCAL VOC 2012 validation set with $\theta = 0.7$

	$K = 6$	$K = 8$–12	$K = 14$
Our method	62.5	**62.7**	62.4

Table 5 Performance with different values of θ on PASCAL VOC 2012 validation set with $K = 9$

	$\theta = 0.0$	$\theta = 0.3$	$\theta = 0.5$	$\theta = 0.7$	$\theta = 0.9$
Our method	62.4	62.3	62.6	**62.7**	62.0

extra training steps to further increase the performance. Firstly, we use the trained segmentation model to generate masks for all training images. Secondly, the predicted segmentation masks are regarded as the proxy ground truths to train the previous network in a fully-supervised manner.

Analysis of K. K is the number of neighbors of the specific node and the value of it is usually determined experimentally. In Table 4, we show the performance with different values of K. It can be observed that results are stable within a certain range of values of K. At last, we choose $K = 9$.

Analysis of θ. We have tried different values of hyper-parameter θ to find the most suitable value in our method. And the results are shown in the simple Table 5. The parameter θ is used in a deep graph cut module to prevent over-segmentation for one certain class. When $\theta = 0.0$, it means that we do not use the threshold constraint. However, from the result, we can observe that the experimental results are not sensitive to the variation of parameter θ. Moreover, we have considered utilizing different thresholds for the foreground category and background. But such an attempt has not worked. Based on the above experiments, we choose to set θ to 0.7 finally.

4.4 COCO results

We conduct a set of experiments on the MS COCO dataset [31] to further demonstrate the effectiveness of our proposed method. We use the steps in Subsection 3.2 to generate the seed. Then we compare our proposed method with our reproduced DSRG on the COCO validation set. It should be noted that we use VGG16 as the backbone to be consistent with DSRG [5]. The per-class IoU of DSRG and our approach are provided in Table 6. DSRG has achieved mIoU values of 26.2 and our proposed method has achieved mIoU values of 27.1 on COCO validation set.

5 Conclusion and future work

We have proposed a method to train a new semantic segmentation network only with image-level annotations. Although seed cues can be obtained from a deep classification network, it is hard to cover the whole object. The proposed graph cut module trains the segmentation network with dynamic supervision and can generate better proxy ground truths than previous methods, e.g., DSRG. The experiment results on PASCAL VOC and COCO demonstrate that our method obtains substantial improvement over the previous weakly supervised approaches under the same experimental conditions. Besides, we have built a bridge between weakly-supervised learning and semi-supervised learning, and then other semi-supervised

Table 6 Comparisons of per-class IoU on COCO validation set

Class	DSRG	Ours	Class	DSRG	Ours	Class	DSRG	Ours
background	78.3	**81.1**	handbag	**4.1**	3.6	pizza	**16.8**	16.7
person	**58.7**	58.3	tie	**7.1**	3.5	donut	**23.5**	22.8
bicycle	28.2	**32.6**	suitcase	27.0	27.0	cake	9.0	**14.6**
car	30.0	**30.6**	frisbee	**21.1**	17.8	chair	**16.8**	15.7
motorcycle	44.2	**47.5**	skis	8.4	**10.4**	couch	18.3	**19.3**
airplane	41.0	**45.3**	snowboard	10.4	**13.6**	potted plant	**19.6**	12.6
bus	47.0	**49.3**	sports ball	18.3	**19.0**	bed	31.7	**32.3**
train	46.2	**47.9**	kite	21.3	**22.1**	dining table	**20.0**	10.6
truck	30.6	**31.6**	baseball bat	4.6	**7.6**	toilet	**49.1**	47.4
boat	21.8	**27.2**	baseball glove	5.1	**8.5**	tv	17.8	**20.3**
traffic light	**23.6**	21.8	skateboard	15.0	**16.4**	laptop	**33.6**	29.5
fire hydrant	40.5	**43.8**	surfboard	23.2	**25.1**	mouse	18.7	**21.5**
stop sign	**59.2**	58.5	tennis racket	23.4	**34.0**	remote	**26.6**	21.2
parking meter	29.8	**33.9**	bottle	12.6	**20.4**	keyboard	28.4	**37.5**
bench	20.2	**25.7**	wine glass	21.7	**23.3**	cell phone	**32.8**	26.2
bird	30.0	**33.3**	cup	16.7	**21.3**	microwave	19.1	**20.7**
cat	**58.0**	55.4	fork	**10.7**	9.6	oven	22.2	**25.9**
dog	**43.0**	39.8	knife	3.9	**4.3**	toaster	0.0	0.0
horse	**39.9**	38.2	spoon	2.3	**3.3**	sink	**28.5**	28.4
sheep	40.4	**40.7**	bowl	**23.7**	19.2	refrigerator	21.2	**30.2**
cow	33.7	**37.7**	banana	**37.8**	31.6	book	**12.1**	10.1
elephant	62.0	**62.9**	apple	19.9	**24.0**	clock	33.4	**40.5**
bear	**50.1**	48.1	sandwich	**7.1**	6.5	vase	**22.6**	20.1
zebra	68.4	**75.0**	orange	**18.0**	15.8	scissors	13.3	**15.5**
giraffe	63.7	**66.1**	broccoli	24.8	**29.5**	teddy bear	**37.2**	37.1
backpack	**11.6**	9.1	carrot	**11.8**	3.1	hair drier	0.0	0.0
umbrella	33.2	**37.5**	hot dog	**12.2**	11.2	toothbrush	1.5	**3.6**

learning methods also can be applied to handle this weakly-supervised task. In future work, we will focus on generating more accurate seeds for weakly-supervised semantic segmentation.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61876212, 61733007) and Zhejiang Lab (Grant No. 2019NB0AB02).

References

1 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2012. 1097–1105
2 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 3431–3440
3 Huang Z L, Wang X G, Wei Y C, et al. CCNet: criss-cross attention for semantic segmentation. 2020. arXiv:1811.11721
4 Kolesnikov A, Lampert C H. Seed, expand and constrain: three principles for weakly-supervised image segmentation. In: Proceedings of European Conference on Computer Vision. Berlin: Springer, 2016. 695–711
5 Huang Z, Wang X, Wang J, et al. Weakly-supervised semantic segmentation network with deep seeded region growing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7014–7023
6 Zhou Z H. A brief introduction to weakly supervised learning. Natl Sci Rev, 2018, 5: 44–53
7 Wei Y, Xiao H, Shi H, et al. Revisiting dilated convolution: a simple approach for weakly-and semi-supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7268–7277
8 Tang P, Wang X G, Bai S, et al. PCL: proposal cluster learning for weakly supervised object detection. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 176–191
9 Wang X G, Deng X B, Fu Q, et al. A weaklysupervised framework for COVID-19 classiffication and lesion localization from chest CT. IEEE Trans Medical Imaging, 2020, 39: 2615–2625
10 Zhou B, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2921–2929
11 Lee J, Kim E, Lee S, et al. FickleNet: weakly and semi-supervised semantic image segmentation using stochastic inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 5267–5276
12 Wei Y, Feng J, Liang X, et al. Object region mining with adversarial erasing: a simple classification to semantic segmentation approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 1568–1576

13 Ahn J, Kwak S. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 4981–4990
14 Fan J S, Zhang Z X, Tan T N. CIAN: cross-image affinity net for weakly supervised semantic segmentation. 2018. ArXiv:1811.10842
15 Boykov Y Y, Jolly M P. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In: Proceedings of the 8th IEEE International Conference on Computer Vision, 2001. 105–112
16 Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Trans Pattern Anal Machine Intell, 2004, 26: 1124–1137
17 Dai J, He K, Sun J. BoxSup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, 2015. 1635–1643
18 Tang M, Perazzi F, Djelouah A, et al. On regularized losses for weakly-supervised CNN segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 507–522
19 Papandreou G, Chen L C, Murphy K, et al. Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. 2015. ArXiv:1502.02734
20 Tang M, Djelouah A, Perazzi F, et al. Normalized cut loss for weakly-supervised CNN segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1818–1827
21 Bearman A, Russakovsky O, Ferrari V, et al. What's the point: semantic segmentation with point supervision. In: Proceedings of European Conference on Computer Vision. Berlin: Springer, 2016. 549–565
22 Lee J, Kim E, Lee S, et al. Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 6808–6818
23 Yang J, Sun X, Lai Y K, et al. Recognition from web data: a progressive filtering approach. IEEE Trans Image Process, 2018, 27: 5303–5315
24 Wang Y, Zhang J, Kan M, et al. Self-supervised scale equivariant network for weakly supervised semantic segmentation. 2019. ArXiv:1909.03714
25 Zhang B, Xiao J, Wei Y, et al. Reliability does matter: an end-to-end weakly supervised semantic segmentation approach. 2019. ArXiv:1911.08039
26 Gao L, Song J, Nie F, et al. Graph-without-cut: an ideal graph learning for image segmentation. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016
27 Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. 2013. ArXiv:1312.6034
28 Selvaraju R R, Cogswell M, Das A, et al. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 618–626
29 Jiang H, Wang J, Yuan Z, et al. Salient object detection: a discriminative regional feature integration approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013. 2083–2090
30 Everingham M, Eslami S M A, van Gool L, et al. The pascal visual object classes challenge: a retrospective. Int J Comput Vis, 2015, 111: 98–136
31 Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: common objects in context. In: Proceedings of European Conference on Computer Vision. Berlin: Springer, 2014. 740–755
32 Hariharan B, Arbeláez P, Bourdev L, et al. Semantic contours from inverse detectors. In: Proceedings of 2011 International Conference on Computer Vision, 2011. 991–998
33 Chen L C, Papandreou G, Kokkinos I, et al. Semantic image segmentation with deep convolutional nets and fully connected CRFs. 2014. ArXiv:1412.7062
34 Lin D, Dai J, Jia J, et al. Scribblesup: scribble-supervised convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 3159–3167
35 Wang X, You S, Li X, et al. Weakly-supervised semantic segmentation by iteratively mining common object features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1354–1362
36 Chaudhry A, Dokania P K, Torr P H. Discovering class-specific pixels for weakly-supervised semantic segmentation. 2017. ArXiv:1707.05821
37 Hou Q, Jiang P, Wei Y, et al. Self-erasing network for integral object attention. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 549–559
38 Shimoda W, Yanai K. Self-supervised difference detection for weakly-supervised semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 5208–5217
39 Zhou D, Bousquet O, Lal T N, et al. Learning with local and global consistency. In: Proceedings of Advances in Neural Information Processing Systems, 2004. 321–328
40 Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. In: Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, 2017
41 Wu Z, Shen C, van den Hengel A. Wider or deeper: revisiting the ResNet model for visual recognition. Pattern Recogn, 2019, 90: 119–133