Does Weakly-supervised Data Improve the Performance of Fully-supervised Segmentation Models?

Anonymous CVPR submission

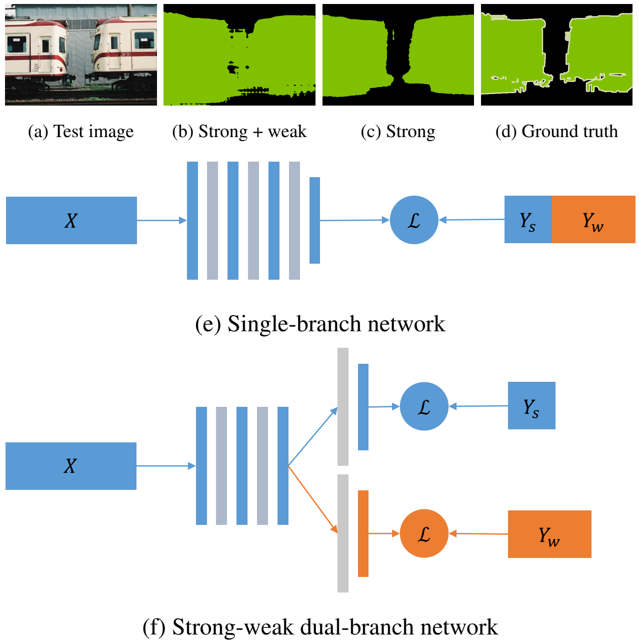
Paper ID 5104

Abstract

Recently many approaches have been proposed to tackle the problem of weakly-supervised semantic segmentation, especially under image-level labels, so as to reduce labeling efforts. They usually exploit the localization cues from a classification network to estimate initial masks, which later supervise the training of a segmentation network. While existing works have explored a variety of techniques to push the envelop of weakly-supervised models, few actually focus on the semi-supervised track where massive amount of extra weakly-supervised data besides a few pixel-level annotations are provided. Current methods simply bundle these two different sets of annotations together to train a segmentation network. However, does this treatment fully explore the potential of weakly-supervised data? This paper devotes to point out the misuse of the weakly-supervised data by previous methods. To this, we propose to impose separate treatments of strong and weak annotations via a strong-weak dual-branch network. This simple architecture requires only slight additional computational costs during training yet brings significant improvements over the previous methods. Experiments on two standard benchmark datasets show the effectiveness of the proposed method.

1. Introduction

Convolutional Neural Networks (CNNs) [17, 30, 11] have proven soaring successes on the semantic segmentation problem. Despite their superior performance, these CNN-based methods are data-hungry and rely on huge amount of pixel-level annotations, whose collections are labor-intensive and time-consuming. Hence researchers have turned to develop segmentation models that could exploit weaker forms of annotation, thus reducing the labeling costs. Although numerous works [15, 35, 13, 19] have been done on learning segmentation models from weak supervisions, especially per-image labels, they still trail the accuracy of their fully-supervised counterparts and thus are not ready for real-world applications.



(a) Test image (b) Strong + weak (c) Strong (d) Ground truth

(e) Single-branch network

(f) Strong-weak dual-branch network

Figure 1. (a) Sample test image; (b) Result using both strong and weak annotations; (c) Result using only the strong annotations; (d) Ground truth; (e) Single-branch network adopted by previous methods [26, 36, 19]; (f) The proposed dual-branch network. The strong annotations Y_s are in blue and the weak ones Y_w are in orange. Images (a)(b)(c)(d) in the first row visually demonstrate that using extra weak annotations brings no improvement over only using the strong annotations when a single-branch network (e) is employed. Refer to Fig.6 for more examples.

In order to achieve good accuracy while still to keep the labeling budget in control, this paper focuses on tackling the problem of semantic segmentation under semi-supervised setting, where a combination of finely-labeled and coarsely-estimated annotations are utilized. Previous methods [26, 36, 19] simply scratch the surface of semi-supervised segmentation by exploring better weakly-supervised strategies to extract more accurate initial pixel-

Table 1. Segmentation accuracy of different methods on PASCAL VOC *val* set.

Methods	Training Set	mIoU (%)
(a) WSSL [26]	1.4k strong + 9k weak	64.6
(b) MDC [36]	1.4k strong + 9k weak	65.7
(c) FickleNet [19]	1.4k strong + 9k weak	65.8
Ours		
(d) Single-branch	1.4k strong	**68.9**
(e) Single-branch	1.4k strong + 9k weak	62.8

level supervisions, which are then mixed together with strong annotations to learn a segmentation network, as in Fig.1(e). However, the bulk of the weak annotations Y_w are of relatively poor quality compared to the strong ones Y_s, hence introducing an inconsistency between the two sets of annotations. Equal treatment of them will overwhelm the handful yet vital fine annotations Y_s. In this regard, we make two key observations that are previously unnoticed concerning the semi-supervised setting:

1. When only trained on small amount of strong data (1.4k in our experiments), the segmentation network doesn't perform as low as people would expect (68.9% in table 1(d)).

2. When simply bundling the strong and weak annotations to train a single segmentation network, the segmentation network cannot achieve better performance than that using only the strong ones (62.8% vs. 68.9% in table 1(d)(e) and visually shown in Fig.1(a)(b)(c))

With a small modification to the original DeepLab architecture, our implementation achieves an mIoU of 68.9% using only 1.4k strong annotations. In addition, it is already much better than other methods [26, 36, 19] exploiting extra 9k weak annotations. However, when we bring in the extra 9k weak annotations like current methods [26, 36, 19], the performance drops dramatically by 7%, which marks the weak annotations useless in this case. We argue such treatment underuses the weakly-supervised data and thus introduces limited improvement, or even worse, downgrading the performance achieved by using only the strong annotations.

To better jointly use the strong and weak annotations, we propose a single unified architecture with parallel strong-weak branches (Fig.1(f)), each handling one type of annotation data. The parallel branches share a common convolution backbone in exchange for supervision information of different level without competing with each other. Unlike the popular Mean-teacher approach [33], which maintained both a *teacher* and a *student* network, the shared backbone enables the free flow of the gradient and the parallel branches can discriminate between the accurate and noisy annotations. This simple architecture boosts the segmentation performance by a large margin while introducing negligible overheads.

The main contributions of our paper are three-folds:

1. We for the first time show that segmentation network trained under mixed strong and weak annotations achieves even worse results than using only the strong ones.

2. We propose a simple unified network architecture by designing a strong branch and a weak branch to address the inconsistency problem of annotation data in the semi-supervised setting.

3. The strong-weak network achieves state-of-art performance under semi-supervised setting on both PASCAL VOC and COCO segmentation benchmarks. Remarkably, it even boosts the fully-supervised models when both branches are trained with strong annotations on PASCAL VOC.

2. Related works

In this section, we briefly review weakly-supervised and semi-supervised visual learning, which are most related to our work. Although the idea of multiple branches to capture various context has already been explored in many computer vision tasks, we here highlight the primary difference between previous methods and this work.

Weakly-supervised semantic segmentation To relieve the labeling burden on manual annotation, many algorithms have been proposed to tackle semantic segmentation under weaker supervisions, including points [2], scribbles [21, 32], and bounding boxes [7, 31]. Among them, per-image class labels are most frequently explored to perform pixel-labeling task since their collections require the least efforts, only twenty seconds per image [2]. Class Activation Map (CAM) [39], is a common method to extract from classification network a sparse set of object seeds, which are known to concentrate on small discriminative regions. To mine more foreground pixels, a series of methods have been proposed to apply the erasing strategy, either on the original image [35] or high-level class activations [12]. Erasing strategy is a form of strong attention [37] which suppresses selective responsive regions and forces the network to find extra evidence to support the corresponding task. Some other works [25, 4, 34] also proposed to incorporate saliency prior to ease the localization of foreground objects.

Recently, Huang *et al.* [13] proposed Deep Seeded Region Growing (DSRG) to dynamically expand the discriminative regions along with the network training, thus mining more integral objects for segmentation networks. And Lee *et al.* [19] further improved the segmentation accuracy of DSRG by replacing the original CAM with stochastic feature selection for seed generation. However, the map expansion for seed generation consumes a lot more GPU memory.

Despite the progress on weakly-supervised methods, there is still a large performance gap (over 10%) from their full-supervised versions [5, 6], which marks them unsuitable for the real-world applications.

Semi-supervised learning In general, semi-supervised learning [40] addresses the classification problem by incorporating large amount of extra unlabeled data besides the labeled samples to construct better classifiers. Besides earlier methods, like semi-supervised Support Vector Machine [3], many techniques have been proposed to integrate into deep-learning models, such as Temporal Ensembling [18], Virtual Adversarial Training [24] and Mean Teacher [33].

Recently, such semi-supervised semantic segmentation problem has been paid more attention, where the training data are composed of a small set of finely-labeled data and large amount of coarse annotations, usually estimated from a weakly-supervised methods. In this configuration, current models [27, 20, 36, 19] usually resorted to the sophistication of weakly-supervised models to provide more accurate proxy supervisions and then simply bundled both sets of data altogether to learn a segmentation network. Such treatment, ignoring the annotation inconsistency, overwhelms the handful yet vital minority and consequently produces even worse results compared to using only the fine data.

Multi-branch network Networks with multiple parallel branches have been around for a long time and proven their effectiveness in a variety of vision-related tasks. Object detection models [9, 28, 22] usually ended with two parallel branches, one for the classification and the other for localization. In addition, segmentation networks, such as Atrous Spatial Pyramid Pooling (ASPP) [5] and Pyramid Scene Parsing (PSP) [38] network, explored multiple parallel branches to capture richer context to localize objects of different sizes. Unlike the above works, we instead utilize parallel branches to handle different types of annotation data.

3. Methods

As aforementioned, the proxy supervisions estimated by weakly-supervised methods are of relatively poor quality in contrast to manual annotations. For finely-labeled and weakly-labeled semantic segmentation task, a natural solution for different supervision is to separately train two different networks, whose outputs are then aggregated by taking the average (or maximum). Although this simple ensemble strategy is likely to boost the performance, it is undesirable to maintain two copies of network weights during both training and inference. Besides, separate training prohibits the exchange for supervision information.

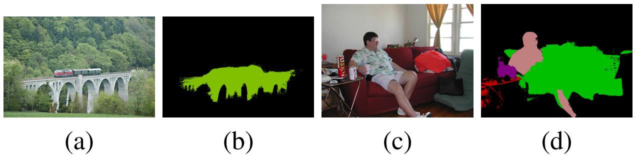

(a) (b) (c) (d)

Figure 2. Inaccurate weak annotations estimated by DSRG. In (a)(b), the *train* mask expands to the background due to color similarity. In (c)(d), large portions of the *human* body are misclassified.

To enable information sharing, we propose a dual-branch architecture to handle different types of supervision, meanwhile eliminating the necessity of keeping two network copies. Our motivation is that the separate treatment of the data will allow the weak branch to absorb the mistakes made by the upstream algorithms and yet still share enough information through the underlying convolutional backbone. Fig.3 presents an overview of the proposed architecture.

Notation Let the training images $X = (x_i)_{i \in [n]}$ be divided into two subsets: the images $X_s = \{(x_1, m_1^s), ..., (x_t, m_t^s)\}$ with strong annotations provided and images $X_w = \{(x_1, m_1^w), ..., (x_k, m_k^w))\}$, the supervisions of which are estimated from a proxy ground-truth generator G:

$$m_i^w = G(x_i) \tag{1}$$

The proxy generator G may need some extra information, such as class labels, to support its decision, but we leave it for general discussion.

The rest of this section is organized as follows. Section 3.1 discusses in depth why training a single-branch network is problematic. Section 3.2 elaborates on the technical details of the proposed strong-weak dual-branch network.

3.1. Single-branch network

Previous works [27, 20, 36, 19] focus on developing algorithm to estimate more accurate initial supervision, but they pay no special attention on how to coordinate the strong and weak annotations. Notably, there are quite a few estimated masks of relatively poor quality (as shown Fig.2) when image scenes become more complex. Equal treatment biases the gradient signal towards the incorrect weak annotations since they are in majority during the computation of the training loss. Consequently, it offsets the correct concept learned from the strong annotations and therefore leads to performance degradation.

One might argue oversampling or weighted loss could be a solution. To this, we conduct experiments via oversampling the strong annotations. As shown in table 2, oversampling improves the final segmentation accuracy

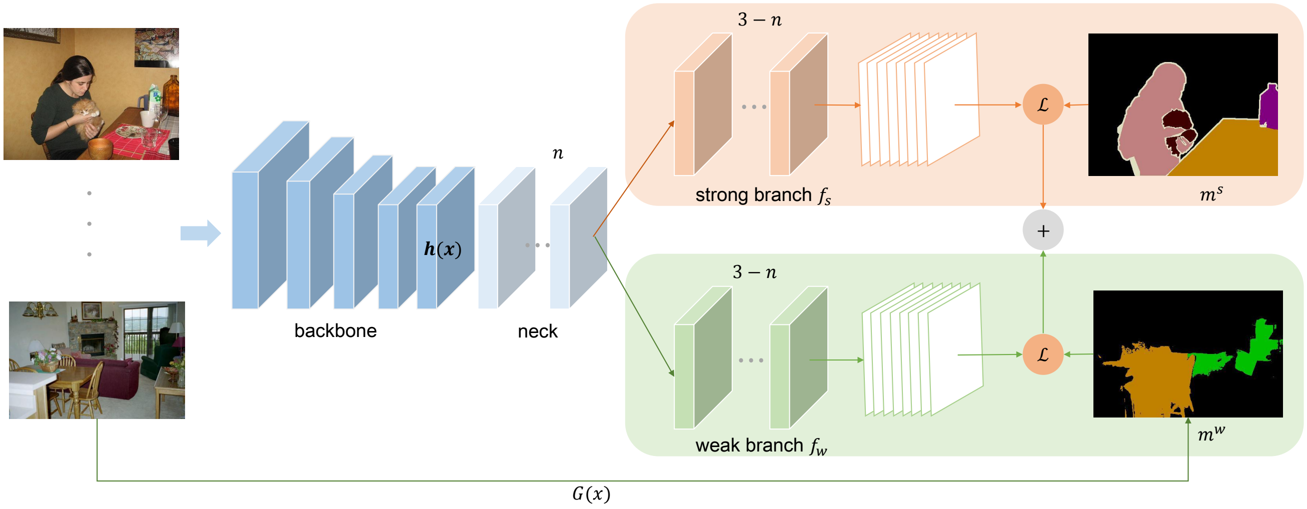


Figure 3. Overview of the proposed dual-branch network. The proposed architecture consists of three individual parts: backbone, neck module and two parallel branches that share an identical structure but differ in the training annotations. The hyparameter n controls the number of individual convolutional layers existing in the parallel branches.

Table 2. Segmentation accuracy concerning different oversampling rates on PASCAL VOC *val* set.

Training data	mIoU (%)
1.4k strong + 9k weak	62.8
1.4k*2 strong + 9k weak	63.5
1.4k*3 strong + 9k weak	64.2
1.4k*6 strong + 9k weak	65.9
1.4k strong	**68.9**

steadily as more strong annotations are duplicated, but it still fails to outperform the result using only the strong annotations. We argue that single-branch network in its nature is incapable of handling the inconsistency due to the competing signals sent by the strong and weak annotations. The following section describes how our dual-branch network is designed to address such issue.

3.2. Strong-weak dual-branch network

3.2.1 Network architecture

The strong-weak dual-branch network consists of three individual parts: convolutional backbone, neck module and two parallel branches with identical structure. Since our main experiments centre around the VGG16 network, we here give a detailed discussion of the architecture based on VGG16.

Backbone The backbone is simply the components after removing the fully-connected layers. As in [5], the last two pooling layers are dropped and the dilation rates of the subsequent convolution layers are raised accordingly to obtain features of output stride 8.

Neck module Similarly to the *Conv6* block in the single-branch network, the neck module is a series of convolution layers added for better adaptation of the specific task. The neck module could be shared between or added separately into subsequent parallel branches. Let n be the number of convolution layers in the neck module. The total number of convolution layers in the neck and subsequent branch is fixed, but the hyper-parameter $n \in [0, 3]$ offers greater flexibility to control the information sharing. When n is 0, each downstream branch has its own neck module. We denote the network up until the neck module as $h(\cdot)$ and its output as $Z \in R^{h \times w \times k}$.

Strong-weak branches These two parallel branches have the same structure while differ in the training annotations they receive. The strong branch is supervised by the fine annotation X_s while the weak branch is trained by the coarse supervisions X_w. The branches $f(Z; \theta_s)$ and $f(Z; \theta_w)$ are governed by independent sets of parameters. For brevity, we will omit the parameters in our notation and simply write $f_s(Z)$ and $f_w(Z)$. The normal cross entropy loss has the following form:

$$\mathcal{L}(s, m) = -\frac{1}{|m|} \sum_{c} \sum_{u \in m_c} \log s_{u,c} \qquad (2)$$

where tensor s is the network outputs, m is the annotation mask and m_c denotes the set of pixels assigned to category

c. Then the total training loss of our method is:

$$\begin{aligned} s^s &= f_s(h(x^s)) \\ s^w &= f_w(h(x^w)) \\ \mathcal{L}_{total} &= \mathcal{L}\left(s^s, m^s\right) + \mathcal{L}\left(s^w, m^w\right) \end{aligned} \tag{3}$$

3.2.2 How does the dual-branch network help?

Since we aim to verify the effectiveness of the proposed architecture, the estimated weak annotations remain unchanged during network training. Firstly, our dual-branch network imposes separate treatment on the strong and weak annotations and therefore prevents direct interference of different supervision information, so the coarse ones leave no direct influence on the strong branch, which determines the final prediction. Secondly, it eliminates the tedious process of oversampling or weighted loss even though there exists sample imbalance between strong and weak annotations. Besides, the extra weak annotations provide approximate location of objects and training them on a separate branch introduces regularization into the underlying backbone to some extent, hence improving the network's generalization capability.

3.2.3 Implementation detail

Training Here we introduce an efficient way to train the strong-weak dual-branch network. A presentation of the processing details can be found in Fig.4. During training, a batch of $2n$ images $X = [(x_1^s, m_1^s), ..., (x_{n+1}^w, m_{n+1}^w)...]$ are sampled, usually the first half from X_s and the second from X_w. Since the number of weak annotations is usually much bigger than that of strong ones, we are essentially performing an oversampling of the strong annotations X_s. For the image batch $X \in R^{2n \times h \times w}$, we make no distinction of the images and simply obtain the network logits in each branch, namely $S^s, S^w \in R^{2n \times h \times w}$, but half of them (in color gray Fig.4) have no associated annotations and are thus discarded. The remaining halves are concatenated to yield the final network output $S = [S^s[1:n], S^w[n+1:2n]]$, which are then used to calculate the cross entropy loss irrespective of the annotations employed. We find that this implementation eases the training and inference processes.

Inference When the network is trained, the weak branch is no longer needed since the information from weak annotations has been embedded into the convolution backbone. So at inference stage, only the strong branch is utilized to generate final predictions. For ablation studies, we also evaluate the outputs from the weak branch to provide more insight into the proposed architecture.

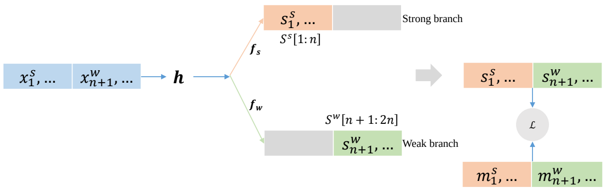


Figure 4. All images are first forwarded through the network and half of the outputs (in color gray) are dropped before they are concatenated to compute the final loss (only the batch dimension is shown).

4. Experiments

4.1. Experimental setup

Dataset and evaluation metric The proposed method is evaluated on two segmentation benchmarks, PASCAL VOC [8] and COCO dataset [23]. **PASCAL VOC**: There are 20 foreground classes plus 1 background category in PASCAL VOC dataset. It contains three subsets for semantic segmentation task, *train* set (1464 images), *val* set (1449 images) and *test* set (1456 images). As a common practice, we also include the additional annotations from [10] and end up with a *trainaug* set of 10582 images. For semi-supervised learning, we use the *train* set as the strong annotations and the remaining 9k images as weak annotations. We report segmentation results on both val and $test$ set. **COCO**: We use the train-val split in the 2017 competition, where 118k images are used for training and the remaining 5k for testing. We report the segmentation performance on the 5k testing images.

The standard interaction-over-union (IoU) averaged across all categories is adopted as evaluation metric for all the experiments.

Proxy supervision generator G To verify the effectiveness of the proposed architecture, we choose the recently popular weakly-supervised method, Deep Seeded Region Growing (DSRG) [13], as the proxy supervision generator G. We use the DSRG model before the retraining stage to generate proxy ground truth for our experiments. Further details could be found in the original paper.

Training and testing settings We use the parameters pretrained on the 1000-way ImageNet classification task to initialize our backbones (either VGG16 or ResNet101). We use Adam optimizer [14] with an initial learning rate of 1e-4 for the newly-added branches and 5e-6 for the backbone. The learning rate is decayed by a factor of 10 after 12 epochs. The network is trained under a batch size of 16 and a weight decay of 1e-4 for 20 epochs. We use random scaling and horizontal flipping as data augmentation and the image batches are cropped into a fixed dimension of

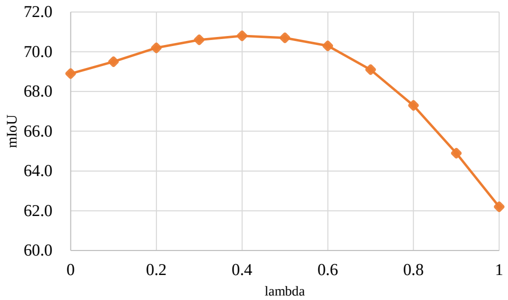


Figure 5. The segmentation mIoU (%) using two separate networks with respect to different λ's.

328×328.

In test phase, we use the strong branch to generate final segmentation for the testing images. Since fully-connected CRF [16] brings negligible improvements when the network predictions are accurate enough, we do not apply CRF as post refinement in our experiments.

4.2. Ablation study

To provide more insight into the proposed architecture, we conduct a series of experiments on PASCAL VOC using different experimental settings concerning network architecture, training data and network backbone. We use VGG16 as backbone unless stated otherwise.

Two separate networks As aforementioned, the single-branch network trained under the mixture of strong and weak annotations achieves no better performance than using only the strong ones. Therefore, it is natural to train two different networks on two sets of data since there exists an obvious annotation inconsistence. Specifically, we train two networks, the first supervised by the strong annotations and the second by extra weak annotations. Then their outputs are aggregated through the following equation:

$$F(x) = \lambda * F_w(x) + (1 - \lambda) * F_s(x) \tag{4}$$

where F_w and F_s denote the weak and strong network respectively. It's a simple ensemble strategy so we expect better performance than using only the output from the strong network. Fig.5 shows the segmentation accuracy under different λ values. Simply training on the strong annotations yields an accuracy of 68.9%, 6.1% higher than the weak one. The result could be improved up to 70.8% with λ equal to 0.4, a 1.9% boost over the strong network. However, separate networks double the computation overhead during both training and inference.

Different n The hyper-parameter n controls the number of shared layers besides the underlying backbone. As shown in table 3, the segmentation accuracy of the strong branch is robust to different choice of n's and is consistently better than combining the result from two separate networks. The performance on the weak branch increases steadily as n becomes bigger. In short, both the strong and weak branches gain performance boost over simply training them separately. We use $n = 3$ in the subsequent experiments.

Table 3. Segmentation accuracy (%) of dual-branch network using different n's and network branches to generate predictions.

n	mIoU	
	Strong branch	Weak branch
0	70.9	62.3
1	71.5	63.1
2	72.2	63.8
3	72.2	64.0

Table 4. Ablation experiments concerning network architectures and training data. Rows marked with "*" are results from the proposed dual-branch network and others are from the single-branch network.

Backbone	Strong branch	Weak branch	mIoU
VGG16	10k weak	-	57.0
VGG16	10k weak (retrain)	-	60.1
VGG16	1.4k strong + 9k weak	-	62.8
VGG16	1.4k strong	-	68.9
VGG16	10k strong	-	71.4
*VGG16	1.4k strong	1.4k strong + 9k weak	**72.2**
*VGG16	1.4k strong	10k strong	**73.9**
ResNet101	10k weak	-	59.0
ResNet101	10k weak (retrain)	-	61.2
ResNet101	1.4k strong + 9k weak	-	63.0
ResNet101	1.4k strong	-	72.4
ResNet101	10k strong	-	75.0
*ResNet101	1.4k strong	1.4k strong + 9k weak	**76.6**
*ResNet101	1.4k strong	10k strong	**78.7**

Single branch vs. dual branch Our VGG16-based implementation of the DSRG method achieves mIoU of 57.0% and 60.1% after retraining, presented in table 4. With a combination of 1.4k strong annotations and 9k weak annotations estimated from DSRG, the single-branch network only improves the segmentation accuracy by 2.7%. However, the single-branch network already achieves much higher accuracy of 68.9% under only 1.4k strong annotations, which means the extra 9k weak annotations bring no benefits but actually downgrade the performance dramatically, nearly 6% drop. This phenomenon verifies our hypothesis that equal treatment of strong and weak annotations are problematic as large amount inaccurate weak annotations mislead the network training.

We then train the proposed architecture with the strong branch supervised by the 1.4k strong annotations and the weak one by extra 9k weak annotations. This time the accuracy successfully goes up to 72.2%, a

Table 5. Segmentation results of different methods on PASCAL VOC 2012 *val* and *test* set.

Methods	**Backbone**	**Val**	**Test**
Supervision: 10k scribbles			
Scribblesup [21]	VGG16	63.1	-
Normalized cut [32]	ResNet101	74.5	-
Supervision: 10k boxes			
WSSL [26]	VGG16	60.6	62.2
BoxSup [7]	VGG16	62.0	64.2
Supervision: 10k class			
SEC [15]	VGG16	50.7	51.7
AF-SS [35]	VGG16	52.6	52.7
Multi-Cues [29]	VGG16	52.8	53.7
DCSP [4]	VGG16	58.6	59.2
DSRG [13]	VGG16	59.0	60.4
AffinityNet [1]	VGG16	58.4	60.5
MDC [36]	VGG16	60.4	60.8
FickleNet [19]	VGG16	61.2	61.9
Supervision: 1.4k pixel + 9k class			
DSRG [13]*	VGG16	64.3	-
FickleNet [19]	VGG16	65.8	-
WSSL [26]	VGG16	64.6	66.2
MDC [36]	VGG16	65.7	67.6
Ours	VGG16	**72.2**	**72.3**
Ours	ResNet101	**76.6**	**77.1**

* - Result copied from the FickleNet [13].

3.3% improvement over the 1.4k single-branch model. Remarkably, this result is even better than training a single-branch model with 10k strong annotations, which implies that there is an inconsistency between the official 1.4k annotations and the additional 9k annotations provided by [10]. Based on this observation, we conduct another experiment on our dual-branch network with 1.4k strong annotations for the strong branch and 10k strong annotations for the weak branch. As expected, the accuracy is further increased by 1.7%. With a ResNet101 backbone and 10k strong annotations, our method could reach an accuracy of 78.7%, a 1% improvement over the original result reported in [5], which used extra COCO annotations for pre-training and fully-connected CRF as post-processing.

4.3. Comparison with the state-of-arts

Table 5 compares the proposed method with current state-of-art weakly-and-semi supervised methods: SEC [15], DSRG [13], FickleNet [19], WSSL [26], BoxSup [7], etc. For fair comparison, the result reported in the original paper is listed along with the backbone adopted.

The weakly-supervised methods are provided in the upper part of table 5 as reference since many of them used relatively weak supervision, with AffinityNet (63.7%) and FickleNet (65.3%) achieving the best performance among other baselines using only class labels. However, the elimination of the demand for pixel-level annotations results in significant performance drop, around 11% compared to their fully-supervised counterparts. There are some recent works exploring other weak supervisions, such as Normalized cut loss [32]. They improved the segmentation performance significantly with slightly increasing labeling efforts. Our method actually serves as an alternative direction by using a combination of strong and weak annotations to achieve excellent results.

Table 6. Per-class IoU on COCO val set. (a) Single-branch network using 20k strong annotations; (b) Single-branch network using 98k extra weak annotations; (c) Dual-branch network using 98k extra weak annotations.

Cat.	Class	(a)	(b)	(c)
BG	background	86.2	78.4	86.7
P	person	74.4	60.7	75.2
Vehicle	bicycle	54.2	48.4	55.3
	car	47.4	38.2	49.5
	motorcycle	70.4	63.7	70.6
	airplane	63.3	30.5	66.0
	bus	69.7	64.1	71.5
	train	67.2	46.7	69.8
	truck	43.3	36.4	45.2
	boat	42.5	26.1	41.9
Outdoor	traffic light	42.9	27.6	47.1
	fire hydrant	74.2	47.3	75.5
	stop sign	82.3	53.6	87.3
	parking meter	48.4	42.7	53.8
	bench	32.6	25.3	34.9
Animal	bird	56.6	33.9	62.0
	cat	76.7	65.1	77.5
	dog	68.7	60.6	69.0
	horse	64.4	50.0	66.2
	sheep	70.5	55.5	73.3
	cow	61.7	49.7	65.3
	elephant	79.9	67.6	81.2
	bear	79.7	60.4	81.7
	zebra	81.7	61.2	82.9
	giraffe	74.3	47.0	75.0
Accessory	backpack	11.4	2.5	12.6
	umbrella	57.9	44.3	59.1
	handbag	6.8	0.0	8.2
	tie	34.5	20.6	35.4
	suitcase	53.1	48.4	57.6
Sport	frisbee	48.2	39.4	50.8
	skis	14.6	5.3	11.8
	snowboard	37.8	15.6	39.1
	sports ball	27.0	13.3	29.7
	kite	32.1	23.7	36.2
	baseball bat	10.4	0.0	11.1
	baseball glove	28.4	0.0	37.6
	skateboard	32.0	20.4	31.6
	surfboard	43.7	32.2	44.5
	tennis racket	55.7	47.3	58.1
	bottle	39.7	33.0	39.4
Kitchenware	wine glass	42.5	36.0	45.2
	cup	38.8	30.9	38.9
	fork	16.6	0.0	17.2
	knife	3.4	0.1	6.9
	spoon	5.9	0.0	5.4
	bowl	33.0	22.4	34.7
Food	banana	62.4	53.1	63.3
	apple	36.6	29.8	37.3
	sandwich	44.3	35.1	46.0
	orange	55.3	50.3	57.9
	broccoli	49.9	37.3	53.3
	carrot	34.4	31.8	37.0
	hot dog	38.8	36.0	39.8
	pizza	74.8	68.6	76.6
	donut	49.4	48.6	53.9
	cake	45.6	40.6	45.3
Furniture	chair	24.4	12.3	25.2
	couch	41.0	20.5	42.6
	potted plant	23.4	15.5	24.5
	bed	46.9	38.2	50.4
	dining table	34.8	9.2	35.0
	toilet	61.5	45.3	62.7
Electronics	tv	49.9	22.5	52.5
	laptop	56.2	40.6	57.4
	mouse	38.5	0.7	35.5
	remote	37.8	25.7	30.9
	keyboard	44.3	35.9	47.1
	cell phone	44.1	36.8	42.3
Appliance	microwave	47.2	32.8	44.6
	oven	42.9	29.7	47.9
	toaster	0.0	0.0	0.0
	sink	40.0	30.5	42.4
	refrigerator	55.5	34.8	57.3
Indoor	book	29.9	16.4	29.0
	clock	57.5	16.4	59.5
	vase	45.8	30.1	43.9
	scissors	57.1	34.1	56.4
	teddy bear	64.4	57.9	66.1
	hair drier	0.0	0.0	0.0
	toothbrush	13.2	8.5	17.6
	mean IoU	**46.1**	**33.4**	**47.6***

*- Note current state-of-art by DSRG [13] was only 26.0%.

The lower part of table 5 presents results of the semi-supervised methods. DSRG and FickleNet used the same region growing mechanism to expand the original object

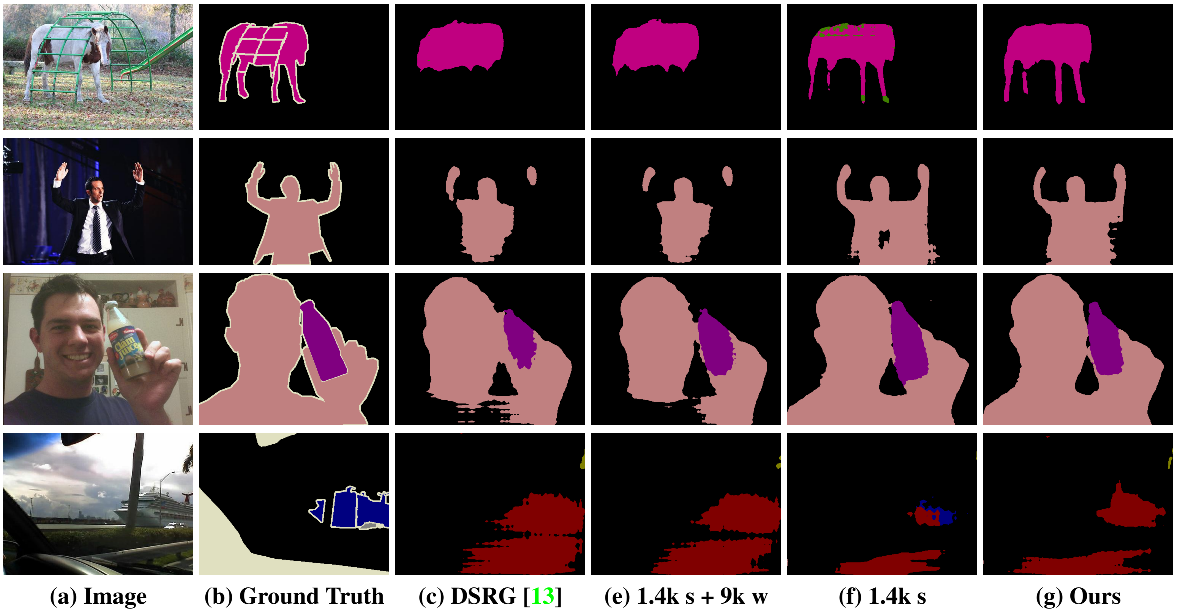


Figure 6. Demonstration of sample images. (a) Original images; (b) Ground truth; (c) DSRG [13]; (e) Mixing 1.4k s + 9k w for training; (f) 1.4k strong annotations; (g) Ours under 1.4k s + 9k w.

seeds. FickleNet explored more accurate initial seeds by stochastic feature selection, which requires a lot more GPU memory to function. As a result, we choose DSRG method as our proxy ground-truth generator. As shown in the table, all previous methods achieved roughly the same and poor performance when learned under 1.4k pixel annotations and 9k class annotations, with the best accuracy 67.6% by MDC approach.

Our method significantly outperforms all the weakly-and-semi supervised method by a large margin, with state-of-art 77.1% mIoU on the *test* set when ResNet101 backbone is adopted.

4.4. Visualization result

Fig.6 shows segmentation results of sample images from PASCAL VOC *val* set. As can be seen in the third column, weakly-supervised method (DSRG) generates segmentation maps of relatively poor quality and no improvement is visually significant if combined with 1.4k strong annotations. Our approach manages to remove some of the false positives in the foreground categories, as in the first and second examples. The last line demonstrates a failure case when neither approach is effective to generate correct prediction.

4.5. Results on COCO

To verify the generality of the proposed architecture, we conduct further experiments on the Microsoft COCO dataset, which contains a lot more images (118k) and semantic categories (81 classes), thus posing a challenge even for fully-supervised segmentation approaches. We randomly select 20k images as our strong set and the remaining 98k images as the weak set, whose annotations are estimated from the DSRG method. This splitting ratio is roughly the same compared to PASCAL VOC experiments. We report per-class IoU over all 81 semantic categories on the 5k validation images. As shown in table 6, with 20k strong annotations, the single branch network achieves an accuracy of 46.1%. When we bring in extra 98k weak annotations estimated by DSRG, the performance downgrades by 12.7%, down to only 33.4%, which again verifies our hypothesis. Using our dual-branch network, the performance successfully goes up to 47.6%, which means our approach manages to make use of the weak annotations.

5. Conclusion

We have addressed the problem of semi-supervised semantic segmentation where a combination of finely-labeled masks and coarsely-estimated data are available for training. Weak annotations are cheap to obtain yet not enough to train a segmentation model of high quality. We propose a strong-weak dual-branch network that has fully utilized the limited strong annotations without being overwhelmed by the bulk of weak ones. Our method significantly outperforms the weakly-supervised and almost reaches the accuracy of fully-supervised models. We think semi-supervised approaches could serve as an alternative to weakly-supervised methods by retaining the segmentation accuracy while still keeping labeling budget in control. In future work, we hope to apply our approach to more vision-related tasks.

References

[1] J. Ahn and S. Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, June 2018. 7

[2] A. Bearman, O. Russakovsky, V. Ferrari, and F. Li. What's the Point: Semantic Segmentation with Point Supervision. *ECCV*, 2016. 2

[3] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *NIPs*, pages 368–374, Cambridge, MA, USA, 1999. MIT Press. 3

[4] A. Chaudhry, P. K. Dokania, P. Torr, and P. Toor. Discovering class-specific pixels for weakly-supervised semantic segmentation. In *BMVC*, volume abs/1707.05821, 2017. 2, 7

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40:834–848, 2016. 3, 4, 7

[6] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 3

[7] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *ICCV*, pages 1635–1643, 2015. 2, 7

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012. 5

[9] R. Girshick. Fast r-cnn. *ICCV*, pages 1440–1448, 2015. 3

[10] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 5, 7

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2015. 1

[12] Q. Hou, P. Jiang, Y. Wei, and M. Cheng. Self-erasing network for integral object attention. In *NIPs*, 2018. 2

[13] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, June 2018. 1, 2, 5, 7, 8

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[15] A. Kolesnikov and C. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, volume abs/1603.06098, 2016. 1, 7

[16] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 6

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPs*, pages 1097–1105, USA, 2012. Curran Associates Inc. 1

[18] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, volume abs/1610.02242, 2016. 3

[19] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, June 2019. 1, 2, 3, 7

[20] K. Li, Z. Wu, K. Peng, J. Ernst, and Y. Fu. Tell me where to look: Guided attention inference network. *CVPR*, pages 9215–9223, 2018. 3

[21] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *CVPR*, pages 3159–3167, 2016. 2, 7

[22] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *ICCV*, pages 2999–3007, 2017. 3

[23] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[24] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *TPAMI*, 41(8):1979–1993, Aug 2019. 3

[25] S. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele. Exploiting saliency for object segmentation from image level labels. In *CVPR*, 2017. to appear. 2

[26] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, pages 1742–1750, Dec 2015. 1, 2, 7

[27] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, ICCV '15, pages 1742–1750, Washington, DC, USA, 2015. IEEE Computer Society. 3

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39:1137–1149, 2015. 3

[29] A. Roy and S. Todorovic. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In *CVPR*, July 2017. 7

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[31] C. Song, Y. Huang, W. Ouyang, and L. Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *CVPR*, June 2019. 2

[32] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *CVPR*, June 2018. 2, 7

[33] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *ICLR*, 2017. 2, 3

[34] X. Wang, S. You, X. Li, and H. Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *CVPR*, June 2018. 2

[35] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, July 2017. 1, 2, 7

[36] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T.S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. In *CVPR*, June 2018. 1, 2, 3, 7

[37] J. Zhang, Sarah A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. *IJCV*, 126:1084–1102, 2016. 2

[38] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *CVPR*, pages 6230–6239, 2016. 3

[39] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016. 2

[40] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. 3