Deep Dual Learning for Semantic Image Segmentation

Ping Luo[2*] Guangrun Wang[1,2*] Liang Lin[1,3] Xiaogang Wang[2]
[1]Sun Yat-Sen University [2]The Chinese University of Hong Kong [3]SenseTime Group (Limited)
pluo@ie.cuhk.edu.hk wanggrun@mail2.sysu.edu.cn linliang@ieee.org xgwang@ee.cuhk.edu.hk

Abstract

Deep neural networks have advanced many computer vision tasks, because of their compelling capacities to learn from large amount of labeled data. However, their performances are not fully exploited in semantic image segmentation as the scale of training set is limited, where per-pixel labelmaps are expensive to obtain. To reduce labeling efforts, a natural solution is to collect additional images from Internet that are associated with image-level tags. Unlike existing works that treated labelmaps and tags as independent supervisions, we present a novel learning setting, namely dual image segmentation (DIS), which consists of two complementary learning problems that are jointly solved. One predicts labelmaps and tags from images, and the other reconstructs the images using the predicted labelmaps. DIS has three appealing properties. 1) Given an image with tags only, its labelmap can be inferred by leveraging the images and tags as constraints. The estimated labelmaps that capture accurate object classes and boundaries are used as ground truths in training to boost performance. 2) DIS is able to clean tags that have noises. 3) DIS significantly reduces the number of per-pixel annotations in training, while still achieves state-of-the-art performance. Extensive experiments demonstrate the effectiveness of DIS, which outperforms an existing best-performing baseline by 12.6% on Pascal VOC 2012 test set, without any post-processing such as CRF/MRF smoothing.

1. Introduction

Deep convolutional networks (CNNs) have improved performances of many computer vision tasks, because they have compelling modeling complexity to learn from large number of supervised data. However, their capabilities are not fully explored in the task of semantic image segmentation, which is to assign a semantic label such as 'person', 'table', and 'cat' to each pixel in an image. This is due to the training set of semantic image segmentation is limited, where the per-pixel labelmaps are difficult and expensive to obtain.

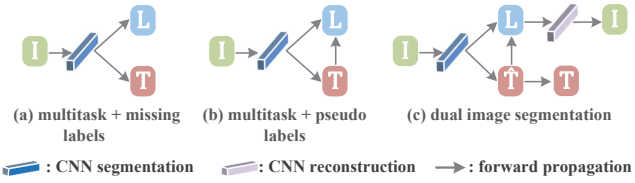

Figure 1: Comparisons of recent semi-supervised learning settings. I, L, and T denote an image, a per-pixel labelmap, and a vector of image-level tags respectively, where the labelmap L can be missing in training. (a) treats L as missing label in multitask learning, where its gradient is not computed in back-propagation (BP). (b) regards L as latent variable that can be inferred by tags and used as ground truth in BP. We propose (c), which infers the missing label L not only by recovering clean tags $\hat{T}$, but also by reconstructing the image to capture accurate object shape and boundary.

To reduce efforts of data annotations, a usual way is to automatically collect images from Internet, which are associated with image-level tags. As a result, the entire dataset contains two parts, including a small number of fully annotated images with per-pixel labelmaps and a large number of weakly annotated images with image-level tags. Learning from this dataset follows a semi-supervised scenario. To disclose the differences among existing works, we introduce necessary notations that will be used throughout this paper. Let I be an image and L, T represent its labelmap and tags. Two superscripts ^{w} and ^{f} are utilized to distinguish the weakly and fully annotated images respectively. For example, I^w represents a weakly labeled image that only has tags T^w, but its labelmap L^w is missing. I^f indicates an image that is fully annotated with both labelmap L^f and tags T^f.

Let θ be a set of parameters of a CNN. The objective function of semi-supervised segmentation is often formulated as maximizing a log-likelihood with respect to θ. We have $\theta^* = \arg\max_\theta \log p(L^f, L^w, T^f, T^w | I^f, I^w; \theta)$, where the probability measures the similarity between the ground truths and the predicted labelmaps and tags pro-

*The first two authors share first-authorship.

duced by the CNN.

Previous works can be divided into two categories according to different factorizations of the above probability. In the first category, methods [13, 23, 28] employed multitask learning with missing labels, where the labelmap L^w is missing and treated as an unobserved variable. In this case, the likelihood is factorized into two terms, $p(L^f, T^f, T^w | I^f, I^w) \propto p(L^f | I^f) \cdot p(T^f, T^w | I^f, I^w)$, which correspond to two tasks as shown in Fig.1 (a). One learns to predict labelmaps L^f using the fully annotated images I^f, and the other is trained to classify tags, T^f and T^w, using both fully and weakly annotated images I^f and I^w. These two tasks are jointly optimized in a single CNN. More specific, back-propagation of the first task is not performed when a weakly labeled image is presented. In general, these approaches learn shared representation to improve segmentation accuracy by directly combining data with strong and weak supervisions.

Unlike the above approaches, L^w is treated as a latent variable in the second category [21, 4, 15]. In this case, the likelihood function can be represented by $p(L^f, L^w, T^f, T^w | I^f, I^w) \propto p(L^f | I^f) \cdot p(L^w | I^w, T^w) \cdot p(T^f, T^w | I^f, I^w)$, where the first and the third terms are identical as above. The second term estimates the missing labelmap given an image and its tags. In other words, when a weakly labeled image I^w is presented, the CNN produces a labelmap L^w, as its parameters are learned to do so from the fully annotated images. The predicted labelmap is then refined with respect to the tags T^w, as shown in Fig.1 (b).

We take Pascal VOC12 [6] as an example, which has 20 object classes and 1 class of background. A labelmap L^w is of $n \times m \times 21$, where n, m denote width and height of the response map, and each entry in L^w indicates the possibility of the presence of a class on a pixel. For instance, when L^w is confused with 'cat' and 'dog' by assigning them similar probabilities, and T^w tells that only 'dog' is appeared in the image but not 'cat', we can refine L^w by decreasing the probability of 'cat'. An implementation to achieve this is by convolving the $n \times m \times 21$ labelmap with a $1 \times 1 \times 21$ kernel (a vector of tags). After refinement, L^w is used as ground truth in back-propagation, which significantly boosts performance as demonstrated in [21].

However, the recent approaches still have two weaknesses. (i) The tags T^w helps refine the misclassified pixels of L^w, but not the object boundary and shape, because these information are not captured in image-level tags. (ii) T^w may have noises when the images are automatically downloaded from the Internet. These noisy tags will hamper the learning procedure.

To resolve the above issues, this work presents Dual Image Segmentation (DIS), which is inspired by the dual learning in machine translation [9], where exists a small number of bilingual sentence pairs, such as a pair of 'have a good day' in English (En) and 'bonne journée' in French (Fr), but there exists unlimited monolingual sentences in En and Fr that are not labeled as pairs on the Internet. [9] leveraged the unlabeled data to improve performance of machine translation from En to Fr, denoted as En→Fr. This is achieved by designing two translation models, including a model of En→Fr and a reverse model of Fr→En.

In particular, when a pair of bilingual sentences is presented, both models can be updated in a fully-supervised scenario. However, when a monolingual En sentence is presented, a Fr sentence is estimated by first applying En→Fr and then Fr→En′. If the input sentence En and the reproduced sentence En′ are close, the estimated Fr sentence is likely to be a good translation of En. Thus, they can be used as a bilingual pair in training. Training models on these large number of synthesized pairs, performance can be greatly improved as shown in [9]. In general, the above two models behave as a loop to estimate the missing Fr sentences.

In this work, DIS extends [9] to four tuples, I, L, T, and $\hat{T}$, where $\hat{T}$ denotes the clean tags that are recovered from the noisy tags T. As shown in Fig.1 (c), DIS has three "translation" models as a loop, including 1) $I \rightarrow (L, \hat{T})$, 2) $\hat{T} \rightarrow (L, T)$, and 3) $L \rightarrow I$, where the first one predicts the labelmap L and clean tags $\hat{T}$ given an image I, the second one refines L according to $\hat{T}$, and the last one reconstructs I using the refined labelmap L.

Intuitively, DIS treats both L^w and $\hat{T}^w$ as latent variables, other than only one latent variable L^w as in Fig.1 (b). For example, when a weakly labeled image I^w is presented, we can estimate L^w and $\hat{T}^w$ by performing $I^w \rightarrow (L^w, \hat{T}^w)$, $\hat{T}^w \rightarrow (L^w, T^w)$, and $L^w \rightarrow I^{w\prime}$. As a result, when I^w and $I^{w\prime}$ are close, L^w and $\hat{T}^w$ can be used as a strong and a weak supervisions for training, because they not only capture precise object classes that present in image, but also capture accurate object boundary and shape in order to reconstruct the image. Leveraging these synthesized ground truths in training, DIS is able to substantially reduce the number of fully annotated images, while still attaining state-of-the-art performance.

Different from Fig.1 (a,b), DIS iteratively optimizes two learning problems as visualized in (c). Its objective function contains two parts, $\log p(L^f, L^w, T^f, T^w, \hat{T}^w | I^f, I^w) + \log p(I^f, I^w | L^f, L^w)$, where the first part is for segmentation and the second part is for image reconstruction. Compared to (b), the first part has one additional variable $\hat{T}^w$ and is factorized as $p(L^f, L^w, T^f, T^w, \hat{T}^w | I^f, I^w) \propto p(L^f | I^f) \cdot p(L^w | I^w, \hat{T}^w) \cdot p(T^f, T^w | I^f, \hat{T}^w) \cdot p(\hat{T}^w | I^w)$. The first three terms are similar to those in (b), whilst the fourth term estimates the clean tags. The second part of the objective function reconstructs images using the predicted labelmaps.

In general, we propose a novel framework for semantic image segmentation, namely dual image segmentation

Table 1: The numbers of training samples of recent weakly-, semi-, and fully-supervised segmentation methods are compared from top to bottom. We take VOC12 as an example to compare their experimental settings. 'Pixel', 'Tag', 'Bbox', and 'Scribble' indicate different types of supervisions, including per-pixel labelmaps, image-level tags, bounding boxes, and scribbles. 'V', 'VA', 'I', 'V7', and 'C' represent images are come from different data sources, where V=VOC12 [6], VA=VOC12+VOC Augment [8], I=ImageNet [25], V7=VOC07 [5], and C=COCO [16]. All these data are manually labeled, but not the data from 'W', which are collected from the Internet.

	Pixel	Tag	Bbox	Scribble
MIL-FCN [23]		10k,VA		
MIL-sppxl [24]		760k, I		
CCNN [22]		10k,VA		
WSSL (semi) [21]	15k,VA+C	118k, C		
BoxSup [4]	10k,VA		133k, C	
ScribbleSup [13]	11k,VA			10k,V7
DIS (ours)	2.9k,V	40k,W+10k,VA		
WSSL (full) [21]	133k, C			
DeepLabv2-CRF [3]	12k,VA			
CentraleSupelec [2]	12k,VA			
DPN [17]	12k,VA			
RNN [30]	12k,VA			

(DIS), which significantly reduces number of fully annotated images in training, while still achieves state-of-the-art performance. For example, we demonstrate the effectiveness of DIS on Pascal VOC 2012 (VOC12) test set, outperforming a best-performing baseline by more than 12%. When adapting to VOC12, DIS reduces the number of fully annotated images by more than 75%.

1.1. Related Works

Semantic Image Segmentation. We review recent advanced segmentation methods by dividing them into three groups, including weakly-, semi-, and fully-supervised methods, which are listed in Table 1 from top to bottom respectively. We take VOC12 as an example to compare their experimental settings. As an approach may have multiple settings, we choose the best-performing one for each approach. We can see that DIS reduces number of labelmaps by 76% and 97% compared to the fully-supervised methods [3, 2, 17, 30, 18] and [21] respectively. In comparison with the weakly- and semi-supervised methods, the number of images of DIS is also smaller than those of many previous works [24, 21, 4]. Unlike existing works that employed manually labeled data, the image-level tags in DIS are mainly collected from the Internet without manual cleaning and refinement.

Image/Labelmap Generation. Transforming between a labelmap and an image has been explored in the literature [11, 1, 29]. Table 2 compares these models with DIS, where [11] aimed at generating realistic images given labelmaps, whilst [1, 29] produced labelmaps using encoder-decoder (AE) and variational autoencoder (VAE). All these approaches treated the feature coding as latent representation to improve the quality of image or labelmap generation in a fully-supervised scenario, where large number of fully annotated images is required. In contrast, DIS infers the latent labelmaps and clean tags to improve segmentation accuracy in a semi-supervised scenario.

Table 2: Comparisons of different image/labelmap generation models, in terms of 'network input', 'network output', and 'latent representation'. In general, [11] generated realistic images given labelmaps as inputs, while [1, 29] output labelmaps. Different from DIS, all these methods learned latent feature coding to improve quality of image/labelmap generation in a fully-supervised scenario, which requires plenty of fully annotated images.

	Network Input	Network Output	Latent Representation
GAN [11]	labelmap	image	coding
AE [1]	image	labelmap	coding
VAE [29]	coding + image	labelmap	coding
DIS (ours)	image	labelmap + image	labelmap + clean tags

Figure 2: Some pairs of images and sentences in IDW. Different tags are highlighted using different colors, where words in 'blue', 'green', 'red', and 'orange' respectively indicate synonyms of tags of VOC12, tags not presented in images, tags of VOC12, and tags appeared in images but missed in sentences. Best viewed in color.

2. Our Approach

Weak Supervisions. To improve segmentation accuracy using weak supervisions, we empoly the Image Description in the Wild (IDW) dataset [26], which has 40 thousand images selected from the Internet. These images are associated with image-level tags, object interactions, and image descriptions. In this work, we only leverage the image-level tags, but not object interactions. To improve image segmentation using object interactions and descriptions, we refer the readers to [26]. Some images and descriptions of IDW are given in Fig.2.

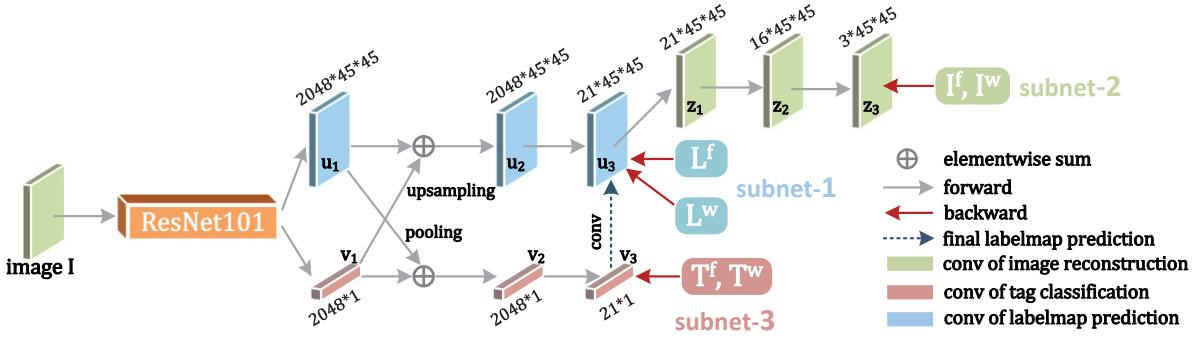

Figure 3: Pipeline of DIS, which contains four key components, including a ResNet101 for feature extraction, and three subnets denoted as 'subnet-1, -2, -3' for labelmap prediction, image reconstruction, and tag classification respectively. Best viewed in color.

An appealing property of IDW is that each image can be associated with multiple tags by parsing the sentences. A deficiency is that these sentences are not sufficiently accurate, where important tags are missing or not presented in images as highlighted in orange and green in Fig.2. For instance, as shown in the top-right image, 'birds' are too small to be observed in image and a 'bicycle' is missing in the sentence. Leveraging them as supervisions may hinder the training procedure. Nevertheless, DIS is able to recover the missing labelmaps and clean tags, to boost the segmentation performance.

2.1. Network Overview

Fig.3 illustrates the diagram of DIS, which has four important components, including a building block of ResNet101 for feature extraction, and three subnets marked as '1', '2', and '3' for labelmap prediction (blue), image reconstruction (green), and tag classification (pink), respectively. The convolutional feature maps in these subnets are distinguished by using u, z, and v. For example, u_1 indicates the first feature map of subnet-1.

Baseline. To evaluate the effectiveness of DIS, we use the ResNet101 network architecture as baseline model. As shown in Fig.3, given an image I, ResNet101 produces a feature map of $2048 \times 45 \times 45$ and a feature vector of 2048×1, denoted as u_1 and v_1 respectively.

Forward Propagation. Fig.3 illustrates the forward flows of the three subnets by using solid arrows in gray, which are explained in detail below. Subnet-1 contains an elementwise-sum layer and a convolutional layer. In particular, the elementwise-sum layer takes u_1 and v_1 as inputs and produces $u_2 = u_1 \oplus \text{up}(v_1)$, where $\text{up}(\cdot)$ represents an upsampling operation that concatenates the 2048×1 feature vector, v_1, into a feature map of $2048 \times 45 \times 45$, and $\oplus$ denotes entry-wise sum between u_1 and $\text{up}(v_1)$. In this case, the pixel-level features u_1 can borrow information from the image-level features v_1 to improve segmentation. After that, a convolutional layer applies a $2048 \times 3 \times 3 \times 21$ kernel on u_2 to produce u_3, which represents the response maps of 21 categories of VOC12. Each entry of u_3 indicates the probability of a category appeared at a specific location.

Subnet-2 leverages u_3 as input and reconstructs the image denoted as z_3, by stacking three convolutional layers. Specifically, the sizes of the kernels from u_3 to z_3 are $21 \times 5 \times 5 \times 21$, $21 \times 3 \times 3 \times 16$, and $16 \times 3 \times 3 \times 3$.

Subnet-3 has an elementwise-sum layer and a convolutional layer similarly to subnet-1. In particular, a feature vector v_2 of length 2048 is produced by fusing v_1 and u_1, such that $v_2 = \text{avgpool}(u_1) \oplus v_1$, where avgpool indicates average pooling. In this case, the image-level features are improved by the pixel-level features to facilitate tag classification. This is extremely useful when the tags have noises. After that, v_2 is projected into a response vector v_3 of 21×1, where each entry indicates the possibility of the presence of a category in an image.

Inference in Test. We introduce the testing procedure of DIS first and then training is discussed in Sec.3. Different from the ordinary ResNet101 [3], DIS enables iterative inference in the testing stage to gradually improve accuracy of the predicted labelmap. This is an important contribution of DIS. In general, it is achieved by minimizing the image reconstruction loss (in subnet-2) with respect to the pixel- and image-level features u_1 and v_1, and keeping the learned network parameters fixed.

More specific, given an image I in test, the objective function of inference can be written as $u_1^*, v_1^* = \arg\min_{u_1, v_1} \|z_3 - I\|_2^2$. Let t be the iteration of optimization. For example, u_1^t denotes a variable at the t-th iteration. As shown in Fig.3, when $t = 0$, u_1^0 and v_1^0 are the features extracted by ResNet101 from the input image I at the beginning of inference. When $t > 0$, these features are forwarded to z_3^t, which can be considered as a function taken u_1^t, v_1^t as inputs. Therefore, by freezing all the network parameters, we can refine these features by treating u_1^t, v_1^t as variables and back-propagating gradients of the above objective function to them. After t iterations,

Figure 4: Inference in test. Two examples are selected from IDW. From left to right, three images represent an input image, the labelmaps when $t = 0$ and $t = 30$ respectively. 'pink', 'blue', and 'green' indicate regions of 'person', 'motorbike', and 'bicycle'. Best viewed in color and 200% zoom.

u_1^* and v_1^* represent the refined features by reconstructing the image to capture accurate object boundary.

Fig.4 compares the results when $t = 0$ and 30. When $t = 0$, the results are the outputs of ResNet101 without inference. These examples show that inference is capable of producing more accurate results when t increases, even though it is difficult to distinguish the appearances of 'person' and 'motorbike' in the first example. Also, the noisy predictions ('motorbike') can be removed from the second one.

Final Prediction. After inference, we propagate u_1^*, v_1^* forward to obtain the predicted labelmap u_3 and tags v_3. The final labelmap prediction is attained by combining them, where v_3 is treated as a convolutional kernel and convolved on u_3. We have $u_3 = \text{conv}(u_3, v_3)$, as shown in the blue dashed arrow of Fig.3.

3. Training Algorithms

DIS is trained with two stages. The first stage pretrains the network using the fully annotated images only. In the second stage, the network is finetuned using both the fully and weakly annotated images.

Fully-supervised Stage. In the first stage, given the strongly labeled data $\{I^f, L^f, T^f\}$, DIS is pretrained in a fully-supervised manner with three loss functions, which are indicated by the solid red arrows at the end of the three subnets as shown in Fig.3, including a softmax loss for labelmap prediction, $\mathcal{L}^{\text{map}}(u_3, L^f)$, an euclidian loss for image reconstruction, $\mathcal{L}^{\text{img}}(z_3, I^f)$, and a softmax loss for tag classification, $\mathcal{L}^{\text{tag}}(v_3, T^f)$. For instance, $\mathcal{L}^{\text{map}}(u_3, L^f)$ denotes the loss function of labelmap prediction, where u_3 and L^f are the predicted and ground-truth labelmaps respectively. More specific, the four components of DIS is progressively trained in three steps. First, we train three components, including ResNet101, subnet-1, and -3 to predict labelmaps and tags. Second, subnet-2 is learned to reconstruct images by freezing the parameters of the above components. Finally, all four components are jointly updated. The rationale behind the above multi-step training scheme is that it encourages a smooth course of multitask minimization. Similar idea has been demonstrated in many previous works [12, 10].

3.1. Semi-supervised Stage

In the second stage, DIS is finetuned by two sources of data, the weakly and fully annotated data, to improve segmentation performance. They are represented by $\{I, L, T\}$, where $I = \{I^f, I^w\}$, $L = \{L^f, L^w\}$, and $T = \{T^f, T^w, \hat{T}^w\}$. The missing labelmap L^w and clean tags $\hat{T}^w$ are inferred as ground truths for training. This is another main contribution of DIS. We first explain the objective function and then discuss the training steps.

Let θ be a set of parameters of the entire network. Then the learning problem can be formulated as

$$\begin{aligned}&\arg\min_{\theta}\Big\{\mathcal{L}^{\text{map}}(u_3, L) + \mathcal{L}^{\text{img}}(z_3, I) + \mathcal{L}^{\text{tag}}(v_3, T)\Big\}\\ &+\arg\min_{u_1,v_1}\Big\{\mathcal{L}^{\text{img}}(z_3, I^w) + \mathcal{L}^{\text{tag}}(v_3, T^w)\Big\},\end{aligned} \tag{1}$$

which contains two parts that are optimized iteratively. The first part is identical to the fully-supervised stage above, where each input image I, regardless of which data source it comes from, is trained to update θ by minimizing the losses between its predictions $\{u_3, z_3, v_3\}$ and ground truths $\{L, I, T\}$. However, for an weakly annotated image I^w, its labelmap L^w and clean tags $\hat{T}^w$ are unobserved. Thus, they are estimated by minimizing the second part of Eqn.(1).

In general, given an image I^f with strong supervisions, DIS can be simply finetuned by optimizing the first part of Eqn.(1). Otherwise, when I^w is presented, DIS is finetuned by iteratively minimizing both parts of Eqn.(1) following two steps, (i) inferring L^w and $\hat{T}^w$ by freezing θ and updating u_1 and v_1, and (ii) updating the network parameters θ by using the inferred L^w and $\hat{T}^w$ as ground truths.

Inference in Training. We introduce the first step in detail. The inference in training is similar to that in test. The main difference is that u_1 and v_1 are updated by minimizing the losses of both image reconstruction and tag classification, other than only the image reconstruction as we did in test. In this case, u_1 and v_1 receive gradients from two paths as shown in Fig.3. One is from v_3 and the other is from u_3 and z_3. As the subnets have been pretrained on strongly annotated data, iteratively updating u_1 and v_1 makes them captured the accurate spatial representation and clean tags, because noises in them are removed in order to produce the labelmaps and reconstruct the images.

Let t be the iteration of optimization. Then u_1^0 and v_1^0 are the outputs of ResNet101 when $t = 0$ at the beginning of inference. For $t > 0$, we update u_1^t, v_1^t by propagating the gradients of the two loss functions back to them, while keeping the network parameters fixed. After t iterations, we obtain u_1^* and v_1^*, which are forwarded to u_3 and v_3. The final prediction of u_3 is refined by v_3 using convolution following the inference in test, as shown in blue dashed arrows of Fig.3. Then the ground truth labelmap and clean tags can be achieved by applying the softmax function on

u_3 and v_3.

Implementation Details. For a fair comparison, the parameters of ResNet101 are initialized the same as [3] by training on ImageNet and COCO. When adapting to VOC12, our baseline model is trained on 2.9k fully supervised data, including images and their labelmaps. The weakly-supervised data with image-level tags are chosen from IDW and VOC augment datasets. The parameters of three subnets are initialized by sampling from a normal distribution. The entire network is trained by using stochastic gradient descent with momentum.

An Interesting Finding. The number of iteration t in inference of training and test can be different. In general, more iterations are performed in training, less iterations are required in test, and vice versa. In other words, computation time in test can be simply reduced by increasing inference iterations in training. On the contrary, training time can be reduced by growing number of iterations in test. We find that both strategies provide remarkable segmentation accuracies. More details are presented in experiments.

4. Experiments

We evaluate the effectiveness of DIS in three aspects. In sec.4.1, DIS is compared with existing segmentation methods on VOC12 test set [6]. When adapting to VOC12, DIS is trained on 2.9k pixel-level annotations and 50k image-level tags. In particular, 40k tags are from IDW and the other 10k tags are from VOC augment dataset. In contrast, previous works typically combined VOC12 and VOC augment dataset [8], resulting in 12k pixel-level annotations. For initialization, existing methods initialized their networks by pretraining on ImageNet [25] and COCO [16], which typically brings 2$\sim$3% performance gain. For a fair comparisons, we mainly report and compare to results that were pretrained on the above two datasets.

In Sec.4.2, we study the impact of the number of iterations in inference, for both the training and testing stages. In Sec.4.3, we examine the generalization of DIS on a test set of IDW.

4.1. Comparisons between Previous Works

The segmentation accuracy of DIS is compared to those of the state-of-the-art methods on VOC12 test set. We adopt 11 representative fully-supervised methods, including SegNet [1], FCN [19], Zoom-out [20], WSSL (full) [21], RNN [30], Piecewise [14], DPN [17], DeepLabv2 [3], LRR-4x-Res [7], HP [27], and CentraleSupelec [2]. We also employ two best-performing semi-supervised approaches, WSSL (semi) [21] and BoxSup [4].

Results are reported in the upper three blocks of Table 3, where the superscript † indicates methods whose baseline models are pretrained on both ImageNet and COCO. We can see that the recent fully-supervised methods generally achieved better results than those of the semi-supervised methods, such as WSSL (semi) and BoxSup. However, when training WSSL on additional images from IDW, it obtains performance of 81.9% that outperforms all previous works. A new record of 86.8% is achieved by DIS, which significantly outperforms the baseline and WSSL+IDW by 12.6% and 4.9% respectively, and reduces the number of fully annotated images by 75% and 97% compared to them.

Properties of several representative works are compared in Table 4, in terms of number of network parameters, training on manually labeled data, CRF post-processing, multiscale testing, and runtime per image on a Titan-X GPU. We have four main observations. First, fusing multiscale features to improve performance is a common practice, which typically leads to 1$\sim$2% gain [17, 4]. Second, all approaches except DIS are trained on manually labeled data, whilst weakly-supervised data for DIS are automatically collected from Web. Third, methods that have smaller numbers of parameters execute faster, but sometimes sacrifice performance such as [1]. Fourth, most of computation time in existing methods such as [3, 17] is occupied by CRF post-processing. Note that the runtime of CRF is not counted in Table 4.

4.2. Ablation Study

We study the effect of using different iterations in inference, as shown at the bottom of Table 3, where t_{tr} and t_{ts} represent numbers of inference in training and test respectively. We evaluate 10 different setups. In the first setup when $t_{\text{tr}} = 0$, DIS degenerates to the recent semi-supervised learning scheme without refining labelmaps and tags. We can see that it outperforms the baseline by 5.8%. For the remaining setups when $t_{\text{tr}} = 5, 10$, and 30, DIS attains more than 5% gain compared to the first one.

We have several important observations. First, when $t_{\text{ts}} = 0$, setup id '2', '5', and '8' outperform '1' and the baseline by more than 5% and 10% respectively, demonstrating the importance of inference in training. Second, when the numbers of t_{tr} are the same, increasing t_{ts} improves performances. For example, the seventh setup outperforms the fifth one by 0.9%, showing the usefulness of inference in test. Third, when the numbers of t_{ts} are the same, larger t_{tr} obtains better performances. For instance, setup id '9' has 1.1% improvement over '3'. Finally, the best performance of 86.8% is achieved when $t_{\text{tr}} = 30$ and $t_{\text{ts}} = 30$.

Fig.5 visualizes some segmentation results. As our purpose is not to generate high quality images, DIS is supervised by the downsampled images of 45×45 in order to save computations. These images are able to catch sufficient details of object shapes and boundaries as shown in the second column. In general, the predicted labelmaps become more accurate when more inferences are performed. Note

Table 3: Comparisons on VOC12 *test* set. Best performance of each category is highlighted in bold.

	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
SegNet [1]	73.6	37.6	62.0	46.8	58.6	79.1	70.1	65.4	23.6	60.4	45.6	61.8	63.5	75.3	74.9	42.6	63.7	42.5	67.8	52.7	59.9
FCN [19]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Zoom-out [20]	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	69.6
WSSL (full)† [21]	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	73.9
RNN† [30]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
Piecewise† [14]	92.3	38.8	82.9	66.1	75.1	92.4	83.1	88.6	41.8	85.9	62.8	86.7	88.4	84.0	85.4	67.4	88.8	61.9	81.9	71.7	77.2
DPN† [17]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
DeepLabv2† [3]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
LRR-4x-Res† [7]	92.4	45.1	**94.6**	65.2	75.8	95.1	89.1	92.3	39.0	85.7	70.4	88.6	89.4	88.6	86.6	65.8	86.2	57.4	85.7	77.3	79.3
HP† [27]	91.9	48.1	93.4	69.3	75.5	94.2	87.5	92.8	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	90.1	59.7	85.5	72.7	79.1
CentraleSupelec† [2]	92.9	61.2	91.0	66.3	77.7	95.3	88.9	92.4	33.8	88.4	69.1	89.8	92.9	87.7	87.5	62.6	89.9	59.2	87.1	74.2	80.2
WSSL (semi)† [21]	80.4	41.6	84.6	59.0	64.7	84.6	79.6	83.5	26.3	71.2	52.9	78.3	72.3	83.3	79.1	51.7	82.1	42.5	75.0	63.4	69.0
BoxSup† [4]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
WSSL† +IDW	**94.7**	62.3	93.3	65.5	75.8	94.6	**89.7**	**93.9**	38.6	93.8	72.2	91.4	**95.5**	89.0	88.4	66.0	**94.5**	60.4	91.3	74.1	81.9
ResNet101† [3]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	74.2
DIS† (ours)	94.4	**73.2**	93.4	**79.5**	**84.5**	**95.3**	89.4	93.4	**54.1**	**94.6**	**79.1**	**93.1**	95.4	**91.6**	**89.2**	**77.6**	93.5	**79.2**	**93.9**	**80.7**	**86.8**
1. $t_{\text{tr}} = 0$	93.3	58.6	90.9	67.9	76.0	94.2	88.6	91.1	35.8	89.7	70.5	87.1	92.2	87.7	86.8	65.3	88.5	60.9	85.9	74.4	80.0
2. $t_{\text{tr}} = 5, t_{\text{ts}} = 0$	93.1	68.2	92.2	73.9	82.1	94.7	87.9	92.9	47.4	93.2	77.0	90.7	92.2	91.1	87.8	75.5	91.9	75.5	92.6	79.8	84.6
3. $t_{\text{tr}} = 5, t_{\text{ts}} = 10$	94.0	69.6	93.1	73.3	83.8	95.2	89.1	93.4	48.8	93.8	77.6	92.0	94.6	91.1	88.3	75.7	93.1	75.8	93.3	**81.1**	85.4
4. $t_{\text{tr}} = 5, t_{\text{ts}} = 30$	93.9	70.7	93.1	77.6	83.4	95.2	89.2	93.3	53.3	94.1	78.7	92.8	94.8	91.4	88.8	77.0	93.0	78.6	93.7	**81.1**	86.2
5. $t_{\text{tr}} = 10, t_{\text{ts}} = 0$	94.2	67.6	93.5	75.9	82.8	95.1	89.5	94.1	48.2	94.5	76.8	93.4	95.2	91.7	88.6	75.3	93.5	76.3	94.1	78.3	85.5
6. $t_{\text{tr}} = 10, t_{\text{ts}} = 10$	94.2	67.9	93.6	74.8	84.3	95.7	89.1	93.3	52.4	94.9	77.9	92.1	95.2	90.5	88.2	76.9	93.7	78.7	93.1	80.3	85.9
7. $t_{\text{tr}} = 10, t_{\text{ts}} = 30$	**94.7**	68.7	**93.8**	75.7	**84.9**	**95.8**	**89.7**	94.3	51.7	**95.2**	78.5	93.2	95.6	91.9	88.8	**77.9**	**93.9**	78.6	**94.3**	80.0	86.4
8. $t_{\text{tr}} = 30, t_{\text{ts}} = 0$	93.6	69.6	**93.8**	76.1	84.4	95.6	89.6	**94.4**	49.7	95.0	78.1	**93.5**	**96.0**	**92.2**	89.0	77.5	93.8	78.6	93.9	80.9	86.3
9. $t_{\text{tr}} = 30, t_{\text{ts}} = 10$	93.3	73.0	93.2	79.1	84.1	95.3	89.5	93.4	53.9	94.3	79.0	92.8	94.9	91.4	89.1	77.5	93.5	**79.2**	93.6	80.5	86.5
10. $t_{\text{tr}} = 30, t_{\text{ts}} = 30$	94.4	**73.2**	93.4	**79.5**	84.5	95.3	89.4	93.4	**54.1**	94.6	**79.1**	93.1	95.4	91.6	**89.2**	77.6	93.5	**79.2**	93.9	80.7	**86.8**

that the original images are put in the first column for better visualization.

In Table 3, we also see that the results of '$t_{\text{tr}} = 30, t_{\text{ts}} = 0$', '$t_{\text{tr}} = 5, t_{\text{ts}} = 30$', and '$t_{\text{tr}} = 10, t_{\text{ts}} = 30$' are comparable, which are 86.3, 86.2, and 86.4 respectively. We find that this is an useful feature of DIS. In particular, the first one indicates we can reduce runtime in test by increasing iterations of inference in training, when computational cost is a priority. To the extreme, inference is not performed in test when $t_{\text{ts}} = 0$, while still maintaining high performance. The last two tell us when model deployment is urgent, we can reduce iterations in training and increase those in test, such as $t_{\text{ts}} = 30$.

Another interesting finding from Table 3 is that different iterations in inference induces diversity among these models, as disclosed by the top performance of each class in bold. In other words, accuracy can be further boosted by ensembling models with different iterations in inference.

4.3. Weakly-supervised Segmentation

We examine the generalization of DIS on IDW test set, which consists of one thousand manually labeled images. This test set is challenging, because the number of object categories per image is more than those in existing datasets, *i.e*. 2.23 compared to 1.48 in VOC12 and 1.83 in COCO. We evaluate 7 setups of DIS similar as above, and compare them to the baseline ResNet101 and WSSL+IDW.

The results are reported in Table 5. In general, since only image-level tags are available in training, the performances in IDW test set are much lower than those in VOC12 test set. However, similar trends can be observed from these two datasets, representing the effectiveness of DIS. It outperforms the baseline and WSSL+IDW by 9.2% and 7.8% respectively, when $t_{\text{tr}} = 30$ and $t_{\text{ts}} = 30$.

Table 4: Comparisons of several representative fully- and semi-supervised segmentation methods. '#params', 'manual', 'CRF', 'multi.', and 'speed (ms)' indicate number of paramters, training on manually labeled data, CRF post-processing, multiscale feature fusion, and runtime per image in millisecond (exclude CRF).

	#params	manual	CRF	multi.	speed (ms)
FCN [19]	134M	√	×	√	160
DPN [17]	134M	√	√	√	175
CentraleSupelec [2]	45M	√	√	√	140
SegNet [1]	16M	√	×	√	75
DeepLabv2 [3]	45M	√	√	√	140
WSSL [21]	134M	√	√	√	200
BoxSup [4]	134M	√	√	√	–
DIS (ours)	45.5M	×	×	√	140

5. Conclusion

This work presented a novel learning setting for semi-supervised semantic image segmentation, namely dual image segmentation (DIS). DIS is inspired by the dual learning in machine translation [9], but has two uniqueness. First, DIS extends two tuples of En and Fr translation in [9] to multiple tuples in the task of semantic image segmentation, by modeling a close loop of generation among images, per-pixel labelmaps, and image-level tags. Second, different "translation" models of DIS can be plugged into a single

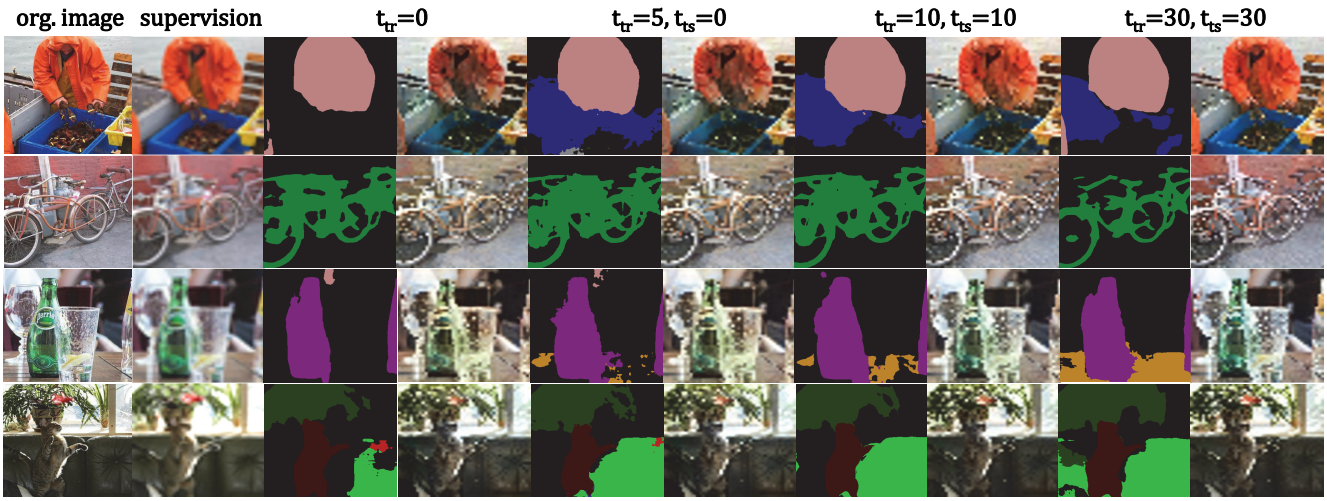

Figure 5: Segmentation examples on VOC12 test set. The columns from left to right show the original images, downsampled images as supervisions, results when '$t_{\text{tr}} = 0$', '$t_{\text{tr}} = 5, t_{\text{ts}} = 0$', '$t_{\text{tr}} = 10, t_{\text{ts}} = 10$', and '$t_{\text{tr}} = 30, t_{\text{ts}} = 30$' respectively. In general, the predicted labelmaps produce better results to capture object classes and boundaries, when more inferences are performed. For example, the regions of 'sofa', 'plant', and 'cat' are correctly identified in the bottom-right labelmap.

Table 5: Comparisons on IDW *test* set. Best performance of each category is highlighted in bold.

	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
1. $t_{\text{tr}} = 0$	48.0	39.7	64.3	23.4	43.2	57.6	55.8	60.8	27.9	61.3	43.6	69.2	74.5	59.2	75.8	31.8	52.4	29.1	30.5	62.6	51.6
2. $t_{\text{tr}} = 10, t_{\text{ts}} = 0$	58.5	43.2	67.1	22.4	52.3	63.9	59.7	71.0	38.0	74.1	45.6	74.9	74.4	67.1	80.4	36.2	47.9	26.7	38.8	64.8	56.6
3. $t_{\text{tr}} = 10, t_{\text{ts}} = 10$	63.7	47.0	**72.5**	23.7	51.5	66.4	58.4	70.3	35.5	**80.0**	47.9	73.1	71.4	67.7	82.1	33.2	55.1	31.2	33.0	66.8	58.0
4. $t_{\text{tr}} = 10, t_{\text{ts}} = 30$	**66.8**	39.6	72.3	21.4	**57.1**	68.8	62.5	72.2	33.6	75.6	50.7	75.6	78.0	68.4	79.9	32.5	**57.3**	33.4	39.0	**70.7**	59.0
5. $t_{\text{tr}} = 30, t_{\text{ts}} = 0$	62.7	44.4	**72.5**	26.8	54.3	65.7	60.7	71.8	38.3	62.6	48.2	75.4	74.7	68.4	80.5	34.4	49.2	26.1	39.1	68.9	57.5
6. $t_{\text{tr}} = 30, t_{\text{ts}} = 10$	61.9	**49.6**	68.2	29.6	48.8	66.4	63.8	**73.7**	35.6	72.1	48.8	76.8	76.7	68.9	81.6	**44.8**	44.8	**33.6**	33.6	66.4	58.8
7. $t_{\text{tr}} = 30, t_{\text{ts}} = 30$	61.7	47.0	68.6	**31.9**	54.2	**72.5**	**66.5**	71.3	**39.1**	66.0	**54.1**	**78.9**	**80.6**	**69.2**	**84.7**	43.0	44.9	32.4	**40.2**	59.1	**59.8**
ResNet101†	50.9	42.0	67.9	17.4	46.4	65.4	59.6	64.8	32.5	21.1	45.8	69.7	74.3	61.2	79.7	25.2	40.0	23.8	34.6	57.6	50.6
WSSL†+IDW	51.4	42.5	61.6	17.0	48.4	62.4	58.3	65.8	34.2	30.8	47.3	70.5	75.1	60.5	80.4	34.8	43.6	24.6	33.4	65.9	52.0

baseline network, which is trained end-to-end, unlike [9] where two translation models are separated.

Different from existing semi-supervised segmentation methods, where a missing labelmap is treated as an unobserved variable, or as a latent variable but inferred only based on the tags, DIS estimates the missing labelmaps by not only satisfying the tags, but also reconstructing the images. These two problems are iteratively solved, making the inferred labelmaps captured the accurate object classes that present in the images, as well as the accurate object shapes and boundaries so as to generate the images. Extensive experiments demonstrated the effectiveness of DIS on VOC12 and IDW test sets, where it establishes new records of 86.8% and 59.8% respectively, outperforming the baseline by 12.6% and 9.2% and reducing number of fully annotated images by more than 75%.

Acknowledgement. This work is supported in part by SenseTime Group Limited, the Hong Kong Innovation and Technology Support Programme, and the National Natural Science Foundation of China (61503366, 91320101, 61472410, 61622214). Correspondence to: Ping Luo and Liang Lin.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv:1511.00561v3*, 2016. 3, 6, 7

[2] S. Chandra and I. Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. *ECCV*, 2016. 3, 6, 7

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 3, 4, 6, 7

[4] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643, 2015. 2, 3, 6, 7

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 3

[6] 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. 2, 3, 6

[7] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, pages 519–534. Springer, 2016. 6, 7

[8] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. *ICCV*, 2011. 3, 6

[9] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma. Dual learning for machine translation. *NIPS*, 2016. 2, 7, 8

[10] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006. 5

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv:1611.07004*, 2016. 3

[12] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. *AISTATS*, 2015. 5

[13] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *CVPR*, 2016. 2, 3

[14] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015. 6, 7

[15] L. Lin, G. Wang, R. Zhang, R. Zhang, X. Liang, and W. Zuo. Deep structured scene parsing by learning with image descriptions. *CVPR*, 2016. 2

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 3, 6

[17] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, pages 1377–1385, 2015. 3, 6, 7

[18] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Deep learning markov random field for semantic segmentation. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017. 3

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 6, 7

[20] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, pages 3376–3385, 2015. 6, 7

[21] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *ICCV*, 2015. 2, 3, 6, 7

[22] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, pages 1796–1804, 2015. 3

[23] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014. 2, 3

[24] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, pages 1713–1721, 2015. 3

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 3, 6

[26] G. Wang, P. Luo, L. Lin, and X. Wang. Learning object interactions and descriptions for semantic image segmentation. *CVPR*, 2017. 3

[27] Z. Wu, C. Shen, and A. v. d. Hengel. High-performance semantic segmentation using very deep fully convolutional networks. *arXiv preprint arXiv:1604.04339*, 2016. 6, 7

[28] Z. Zhang, P. Luo, C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *TPAMI*, 2016. 2

[29] H. Zheng, F. Wu, L. Fang, Y. Liu, and M. Ji. Learning high-level prior with convolutional neural networks for semantic segmentation. *arXiv:1511.06988v1*, 2015. 3

[30] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015. 3, 6, 7