Regularized Loss for Weakly Supervised Single Class Semantic Segmentation

Olga Veksler

University of Waterloo, Waterloo ON N2L3G1, Canada, oveksler@uwaterloo.ca
https://cs.uwaterloo.ca/ oveksler/

Abstract. Fully supervised semantic segmentation is highly successful, but obtaining dense ground truth is expensive. Thus there is an increasing interest in weakly supervised approaches. We propose a new weakly supervised method for training CNNs to segment an object of a single class of interest. Instead of ground truth, we guide training with a regularized loss function. Regularized loss models prior knowledge about the likely object shape properties and thus guides segmentation towards the more plausible shapes. Training CNNs with regularized loss is difficult. We develop an annealing strategy that is crucial for successful training. The advantage of our method is simplicity: we use standard CNN architectures and intuitive and computationally efficient loss function. Furthermore, we apply the same loss function for any task/dataset, without any tailoring. We first evaluate our approach for salient object segmentation and co-segmentation. These tasks naturally involve one object class of interest. In some cases, our results are only a few points of standard performance measure behind those obtained training the same CNN with full supervision, and state-of-the art results in weakly supervised setting. Then we adapt our approach to weakly supervised multi-class semantic segmentation and obtain state-of-the-art results.

1 Introduction

Convolutional Neural Networks (CNNs) [22, 20] lead to a breakthrough in semantic segmentation [11, 35, 4, 57, 5]. Each year new architectures push the state of the art in on fully labeled datasets [6, 49, 39]. However, labeling datasets is expensive. Thus there is an interest in segmentation without fully labeled data.

We consider a new approach to train CNNs in the absence of pixel precise ground truth. Our approach is inspired by [40, 41] who demonstrate the utility of regularized loss[1] for semantic segmentation with partially labeled data. Regularized losses are used in computer vision in the context of Conditional Random Fields (CRFs) [25, 21] for modelling the likely properties of an object, such as short boundary [2], connectivity [45], shape convexity [13], etc.

[1] In the context of CNNs, regularization is a term often used to refer to norm regularization on network weights [12], or other techniques to prevent overfitting. In this work, regularized loss refers to the loss function on the output of CNN.

Our main idea is to supervise training with a regularized loss. Instead of encouraging CNN output to match to ground truth with cross entropy loss, we encourage CNN output to have desired properties with regularized loss. We use *sparse-CRF* loss [2] that encourages shorter object boundaries aligning to intensity edges. We show that sparse-CRF has a high correlation with segmentation accuracy, and thus is a good candidate to use for training. While it is possible to design more complex regularized losses, it is interesting to evaluate the utility of sparse-CRF loss, widely used in classical computer vision, now for CNN training.

Although regularized loss has been used before [40, 41] for settings with partially labeled pixels, our task is significantly more difficult, as we assume no labeled pixels. Consider that in [41] the difference in performance with and without regularized loss is only about 4%, i.e. most of the learning is done by cross entropy on partially labeled pixels. To enable learning in absence of pixel labels, we make an assumption that the training dataset contains a single object class of interest, making our approach a weakly supervised single class segmentation.

Regularized loss by itself is insufficient, since an empty object would be a trivial optimal solution. We need additional priors implemented as "helper" loss functions. Assuming a single object class lets us use the *positional* loss. Positional loss encourages the image border to be assigned to the background, and the image center to the foreground. This prior is reasonable for single class datasets. We also use *volumetric loss* [46] to counteract the well known shrinking bias of sparse-CRF loss. It penalizes segmentations that are either too large or too small in size. Both volumetric and positional loss have a significant weight only in the initial training stage, to push CNN training in the right direction. Their weight is significantly reduced after the initial stage, to allow segmentation of objects that are not centered and have a diverse range of sizes.

Training CNN with regularized loss is hard. We develop an *annealing* strategy [17] that is essential for successful training. The goal of our annealing is different from the traditional one. Our annealing helps CNN to first identify an easy hypothesis for the appearance of the object, and then to slowly refine it.

Training only on the class of interest can result in a CNN that has a good response to similar ("confuser") classes. We can add "confuser" class images to the training set as negative examples, and add a loss function that encourages labeling them as background. This is still in the realm of weakly supervised segmentation since negative examples need only an image-level "negative" tag.

Our method results in a simple end-to-end trainable CNN system. We use standard architectures and just add regularized, volumetric, and positional loss instead of cross entropy. We do not adapt our loss function for any task. Our loss function is easy to interpret, intuitive, and efficient.

First, we evaluate our approach on the salient object segmentation [27] and co-segmentation [32]. These tasks naturally involve a single class of interest. In many cases, our CNN, trained without ground truth is only a few points behind the same CNN trained with full ground truth. For some saliency and cosegmentation datasets we obtain state-of-the art weakly supervised results.

Next we extend our approach to weakly supervised semantic segmentation in a naive fashion: we train on datasets containing a single semantic class and use the other classes as negative examples. The resulting pseudo-ground truth is then used to train a standard multi-class semantic segmentation CNN. We obtain state-of-the art results on Pascal VOC 2012 benchmark in weakly supervised setting. Our code for single class segmentation is publicly available[2].

2 Related Work

In this section, we review prior work on CNN segmentation without full supervision for salient object, co-segmentation and semantic segmentation.

There is some prior works for CNN based salient object segmentation without human annotation [55, 56, 28]. All three are based on exploiting multiple saliency segmentation methods that are not based on machine learning, i.e. "weak methods". These approaches are very specific to saliency segmentation, whereas our method is generic an applies to any single object segmentation task.

There is a prior work on unsupervised co-segmentation in [14]. Their co-attention loss function is expensive to compute, it relies on all pairs of images in the training dataset. Again, the advantage of our method is generic architecture and generic computationally efficient loss function.

The prior work on weakly and partially supervised semantic segmentation can be divided in two groups, depending on the supervision type. In the first group are weakly supervised approaches based on image-level tags [30, 29, 18, 1, 23]. Most successful of these methods use CAM (class activation maps) [58, 34] to produce pseudo-ground truth which is then used to train a semantic segmentation CNN. The drawback of these approaches is their reliance on CAMs, which can vary in accuracy. Our approach is does not require CAMs.

In the second group are methods based on imprecise labeling (i.e. bounding boxes) or partial labeling of the data. Approaches in [29, 51, 7, 15] assume a bounding box is placed around semantic objects. Approaches in [51, 26, 44, 40, 41] are based on partially labeled data or "scribbles". The earlier of these approaches first use scribbles or boxes to generate full pseudo ground truth, which is then used to train CNNs. The later approaches [40, 41] argue against using pseudo ground truth and show more accurate results by using cross entropy loss only on the labeled pixels and regularized loss for the rest of the pixels. The advantage of our approach is that it does not require any labeling or annotation of the data. Note that our approach is different from training with bounding boxes. A bounding box annotation is typically placed relatively tight around the object. In the datasets we use, objects vary in size significantly. Thus if we regard image as a "box", it is anything but tight for the majority of dataset images.

[2] https://github.com/morduspordus/SingleClassRL

3 Our Approach

We now describe our approach. We explain regularized losses in Sec. 3.1, evaluate correlation between regularized loss and segmentation accuracy in Sec. 3.2, give our complete loss function in Sec. 3.3, and describe training strategy in Sec. 3.4.

3.1 Regularized Losses

A variety of regularized loss functions are used in CRFs [25, 21]. The main task in CRF is to assign some label x_p, for example, a class label, to each image pixel p. Here we assume $x_p \in \{0, 1\}$. If $x_p = 1$ then pixel p is assigned the object class, and if $x_p = 0$ then p is assigned the background. A regularized loss is computed for $\boldsymbol{x} = (x_p \mid p \in \mathcal{P})$, an assignment of labels to all pixels in $\mathcal{P}$, where $\mathcal{P}$ is the set of image pixels. We use color edge sensitive Potts [2] regularized loss:

$$L_{reg}(\boldsymbol{x}) = \sum_{p,q \in \mathcal{P}} w_{pq} \cdot [x_p \neq x_q], \tag{1}$$

The pairwise term $w_{pq} \cdot [x_p \neq x_q]$ in Eq. (1) gives a penalty of w_{pq} whenever pixels p, q are not assigned to the same label. To align object boundaries to image edges, we use:

$$w_{pq} = e^{-\frac{||C_p - C_q||^2}{2\sigma^2}}, \tag{2}$$

where σ controls "edge detection". Decreasing σ leads to an increased number of weights w_{pq} that are small enough to be considered a color edge.

The sum in Eq. (1) is over all neighboring pixels p, q. We assume a 4-connected neighborhood. The loss function in Eq. (1) with 4-connected neighborhood is called sparse-CRF. Sparse-CRF regularizes a labeling by encouraging shorter object boundaries that align to image edges [3].

We also test dense-CRF regularized loss from [19], which has the same form as in Eq. (1) but the summation is over all (thus dense) pixel pairs and w_{pq} is a Gaussian weight that depends both on the color difference and the distance between pixels p, q. Unlike sparse-CRF, the regularization properties of dense-CRF are not well understood, although there is some work [43] that analyzes the properties of an approximation to a dense-CRF.

In the context of CNNs, the output label x_p is no longer discrete but in continuous range $(0, 1)$, assuming last layer is softmax. We use the absolute value relaxation of Eq. (1):

$$L^{abs}_{reg}(\boldsymbol{x}) = \sum_{p,q \in \mathcal{P}} w_{pq} \cdot |x_p - x_q|. \tag{3}$$

3.2 Loss and Segmentation Accuracy Correlation

If regularized loss is a good criterion for single class segmentation, then the accuracy of segmentation should be negatively correlated with the value of the

loss. While this may not be a sufficient condition, it is a necessary one. We measure correlation experimentally, using four single-object datasets: OxfordPet [42], MSRA-B [27], ECSSD [52], and DUT-OMRON [53]. The first dataset has ground truth segmentation for two object classes: *cat* and *dog*. We merge them into a single class *pet*. The last three datasets are for salient object segmentation and contain one class. We use F-measure as the accuracy metric.

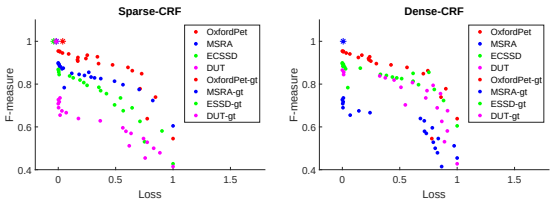

Fig. 1. Scatter plots of regularized loss vs. F-measure on four datasets (different colors): sparse-CRF (left), dense-CRF (right). See text for details.

	OxfordPet	MSRA	ECSSD	DUT-O
sparse-CRF	-0.862	-0.895	-0.956	-0.957
dense-CRF	-0.830	-0.758	-0.772	-0.896

Table 1. Correlation coefficients between F-measure and regularized loss for sparse-CRF and dense-CRF.

To measure correlation between regularized loss and F-measure, for each dataset, we need several segmentation sets of varying accuracy. We are interested in segmentations that can be learned with a CNN, not just any random segmentations, since we assume a dataset has a concept that can be learned with a CNN. Therefore we train a CNN with ground truth and obtain several early/intermediate segmentation results in addition to the final one. We use 20 time steps to obtain 20 different accuracy segmentation sets for each dataset.

We average sparse-CRF losses and F-measures over segmentations of each dataset, separately for each time step. We do the same for the dense-CRF. The scatter plots of regularized loss vs. F-measure are in Fig. 1. Each dataset is in different color. We linearly normalize (no effect on correlation) the loss into range $(0, 1)$ for visualization. The correlation coefficients for sparse-CRF and dense-CRF are in Table 1. Both losses show significant negative correlation, however sparse-CRF correlation is better for all datasets. Given high negative correlation, the task of performing CNN training with regularized loss is promising.

We also measure regularized loss of ground truth, shown with asterisks in Fig. 1. For dense-CRF only one asterisk is visible because the loss value is almost the same for all datasets. Except OxfordPet, for all datasets the ground truth has the lowest value of the regularized loss. For the OxfordPet, the ground truth has a slightly higher value than the best solution obtained through training CNN.

3.3 Complete Loss Function

Sparse-CRF has a shrinking bias [46], it favors segments with a shorter boundary. To counteract, we add volumetric loss, also frequently used in MRFs [46]. Let $\boldsymbol{x}$ be an output of the network, and let $\bar{\boldsymbol{x}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} x_p$, i.e. normalized object size. Our volumetric loss is defined for a batch of m images with outputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$:

$$L_v(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) = \big(\frac{1}{m}\sum_i \bar{\boldsymbol{x}}_i - 0.5\big)^2 + \lambda_s \sum_i \big[\bar{\boldsymbol{x}}_i < 0.15\big] \cdot \big(\bar{\boldsymbol{x}}_i - 0.15\big)^2, \quad (4)$$

The first sum in Eq. (4) encourages the average size of the object in a batch to be 0.5, i.e. roughly half of the image size. We average object size over a batch to allow for a large range of object sizes in a batch: some objects in can be small, some can be large, but the average size is expected to be 0.5. We could encourage any object/background size ratio by replacing 0.5 appropriately. However, we found no need for more precise modeling as volumetric loss is essential only in the first, *annealing* stage of training, see Sec. 3.4. Batch size is set to $m = 16$.

The second sum in Eq. (4) encourages each object inside the batch to have normalized size of at least 0.15. This is a conservative measure of the minimum object size to prevent collapse to an empty object. Here $\big[\cdot\big]$ is 1 if the argument is true and 0 otherwise. The weight λ_s balancing these two sums is set to 5 in all experiments. In practice, this term is important only during the second training stage, when regularization loss has a much higher weight than volumetric loss.

In the beginning of the training, it is important to guide the network in the right direction. For a dataset that contains only one object class of interest, for many images the border pixels do not contain the object, and the center of the image contains the object. We incorporate this prior through *positional* loss. Let $\mathcal{B}$ be the set of pixels on the image border of width w, and $\mathcal{C}$ be the set of image pixels in the center box of side size c. We formulate positional loss $L_p(\boldsymbol{x})$ as:

$$L_p(\boldsymbol{x}) = \Big(\frac{1}{|\mathcal{B}|}\sum_{p\in\mathcal{B}} x_p\Big)^2 + \Big(\frac{1}{|\mathcal{C}|}\sum_{p\in\mathcal{C}} x_p - 1\Big)^2. \quad (5)$$

For all experiments, we set $w = c = 3$, so the number of effected pixels is small. However, positional loss is enough to push the network in the right direction in the beginning of training, seeking object segments that are preferably in the image center and do not overlap the border. Just as with volumetric loss, positional loss is important only in the *annealing* stage. In the *normal* stage, the weight of positional loss is low, thus allowing segmenting objects that overlap with the border and do not contain the central four pixels of the image.

Our complete loss function for training is a weighted sum of regularized, volumetric, and positional losses, and is applied to $\boldsymbol{x}$, the output of CNN[3]:

$$L(\boldsymbol{x}) = \lambda_{reg} L_{reg}(\boldsymbol{x}) + \lambda_v L_v(\boldsymbol{x}) + \lambda_p L_p(\boldsymbol{x}). \tag{6}$$

All parts of the loss in Eq. (6) are important. Training fails if any part is omitted. When we say we train with "regularized loss", we mean training with the complete loss in Eq. (6). Saying "train with regularized loss" emphasizes the largest important component of our complete loss function.

Sometimes we have images containing a class similar to the class of interest but not the class we want to segment. It helps to include these images as negative examples. Negative images are assumed not to contain the object of interest, and thus no ground truth is required for training. Therefore, including negative images is, again, a form of a weak supervision. If negative images are available, we apply the following *negative* loss to $\boldsymbol{x}$, the output of CNN on negative images:

$$L_{neg}(\boldsymbol{x}) = \lambda_{neg}\big(\frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} x_p - 1\big)^2 \tag{7}$$

3.4 Training

Training CNN with the loss function in Eq. (6) is difficult. Out-of-the-box training produces results with either all pixels assigned to the foreground or background. We found that *annealing* [17] (i.e. slowly increasing) parameter σ in Eq. (2) is essential for obtaining good results. Thus there are two training stages in our approach: the *annealing* stage and the *normal* stage.

The annealing stage has $n = 15$ iterations during which parameter σ changes according to the following schedule, where i is the iteration number:

$$\sigma(i) = 0.05 + \frac{i}{n} * 0.1, \quad 1 \le i \le 15 \tag{8}$$

The other parameters are not changed and are set to: $\lambda_{reg} = \lambda_v = \lambda_p = 1$.

Traditionally, annealing is used to increase parameter λ_{reg}, increasing the difficulty of loss function optimization. The larger is λ_{reg}, the more difficult the loss function tends to be. We anneal parameter σ that controls edge sensitivity, and its annealing seems to serve a different purpose. Intuitively, annealing σ helps CNN to find an easy object hypothesis first, and then refine it.

Consider Fig. 2. The input image is shown on the left. In the middle, for several annealing iterations, we show the segmentation result together with w_{pq} weights (below the result). The lower weights w_{pq} are illustrated with darker intensities. The last column in the figure is the final result after the normal training stage, and the ground truth is below it. Consider the result at iteration one. The value of σ is low, and, therefore, there are many areas in the image with

[3] Note that our volumetric prior is actually defined on batches of images. However for the simplicity of notation, we write $L_v(\boldsymbol{x})$ in this equation.

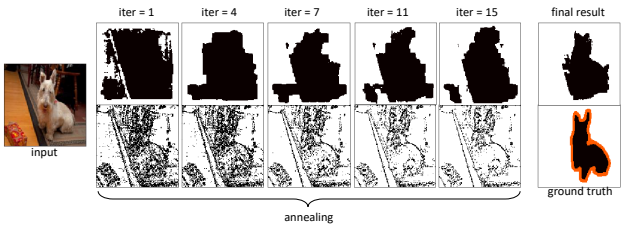

Fig. 2. Illustration of annealing. Left: input image. Middle: segmentation results for several iterations of annealing together with w_{pq} weights at these iterations, shown below each segmentation result; darker intensities correspond to lower w_{pq}. Right, top: the final segmentation after the normal training stage. Right, bottom: the ground truth, with "void" class in orange.

small w_{pq}. At this first iteration, CNN learns to classify most of the areas with low w_{pq} as the object. This happens because placing object/background boundaries around pixels with lower w_{pq} is cheap. It also happens to be a reasonable start as objects usually have edges on their boundary, and often they have texture edges in the interior. Of course, many of the pixels included in the object mask in the first iteration are errors. As annealing proceeds, σ is raised, and there are fewer areas with small w_{pq} weights, making the old placements of boundaries on the weaker edges more expensive. Increasingly more erroneous object areas are discarded as CNN learns the common object appearance from the "surviving" areas. The final annealing result (iter = 15) is a crude approximation to the actual object, and it gets refined further during the normal training stage (last column, right), as the weight of regularized loss is increased significantly.

After the annealing stage, we have the *normal* training stage. The parameters are set to: $\lambda_{reg} = 100,\ \lambda_v = \lambda_p = 1,\ \sigma = 0.15$. Thus during normal training stage, regularized loss is the major contributing component. After annealing, we have a good approximate solution and can relax the penalty for objects that deviate widely from half of the image size, do not overlap the image center, and overlap image border. We do not set λ_v and λ_p to zero because then the optimal solution is a trivial one, everything assigned to the object (or background).

If negative loss in Eq. 7 is used, we set $\lambda_{neg} = 2$. Also, we first train in the annealing and normal stages without negative loss, and then train with the negative loss in the normal stage for an additional 200 epochs.

4 Experimental Results

This section is organized as follows. We start with saliency datasets and OxfordPet dataset in Sec. 4.1. We use these datasets to make choices about CNN architecture and training protocol and then use these choices for all the other experiments. In Sec. 4.2 we test our approach for cosegmentation. In Sec. 4.3 we extend our approach to weakly supervised multi-class semantic segmentation.

The supplementary materials contain additional experiments that experimentaly analyze properties of CNN trained with regularized loss, as well as other minor experiments.

4.1 Saliency Datasets

We use two CNN architectures, both based on Unet [31]. They differ in the module for feature extraction. The first one is based on MobileNetV2 [33] and the second on ResNeXt [50]. The feature extraction modules are pretrained on Imagenet [8]. The pretrained features are used in the encoder part Unet, and they stay fixed during training. We refer to these networks, respectively, as *UMobV2* and *UResNext*. *UResNext* is much bigger than *UMobV2*. For both networks, we add softmax as the last layer to convert the output to the range of $(0, 1)$. We train each network with regularized loss in Eq. (6) and with cross-entropy loss on ground truth for comparison. To distinguish between these training types, we add "gt" to the name of any network that is trained with ground truth, and "reg" if it is trained with regularized loss.

For the experiments in this section, we use three saliency datasets: MSRA-B [27], ECSSD [52], and DUT-OMRON [53]. We split MSRA-B into $3,000$ training and $2,000$ test images, ECSSD into 700 training and 300 test images, and DUT-OMRON into $3,678$ training and $1,490$ test images.

We also use OxfordPet [42] dataset that has segmentations of cats and dogs. To make it appropriate for single class segmentation, we combine the cat and dog classes into one *pet* class. OxfordPet has $3,680$ training and $3,669$ test images.

We use F-measure as an accuracy metric, $F = \frac{(1+\beta^2)precision \times recall}{\beta^2 \times precision + recall}$, with $\beta^2 = 0.3$.

	OxfordPet	MSRA-B	ECSSD	DUT-O
UMobV2-reg	94.62	86.20	57.96	53.79
UMobV2-gt	96.07	89.35	84.78	80.86
gap	**1.45**	3.15	26.82	27.07
UResNext-reg	**94.77**	**88.36**	**73.47**	**64.48**
UResNext-gt	96.53	89.68	86.11	81.47
gap	1.76	**1.32**	**12.64**	**16.99**

Table 2. F-measure for all networks and datasets; for each network and dataset, we also show the gap between training with ground truth and regularized loss.

F-measures are computed on the test fold. However, since our method does not use ground truth, the test and training folds have a similar F-measure.

For efficiency, we decrease resolution of all images on all experiments in this section to 128×128. We use Adam optimizer [16] with a fixed learning rate 0.001 and train all networks for 50 epochs.

The F-measures of *UMobV2* and *UResNext* on the OxfordPet, MSRA-B, ECSSD, and DUT-OMRON are in Table 2, both in the case of training with regularized loss and training with ground truth. We also give the gap between training with ground truth and regularized loss. Whether training with ground truth or regularized loss, *UResNext*, a bigger network, performs better than *UMobV2*, and has a smaller gap from the network trained with ground truth on 3 out of 4 datasets. OxfordPet dataset is the easiest to fit dataset.

Replacing the Annealing Stage The purpose of the annealing training stage is to find a good start for the normal training stage. We found that instead of annealing, using weights from the same network trained on another dataset also gives a good start for the normal training stage. Our training consists of 15 annealing and 50 normal epochs. So skipping 15 epochs of annealing gives only a modest time saving. However, we found that transferring weights from a network trained on an easy-to-fit dataset can lead to a much better result. We use OxfordPet for weight transfer because it is the easiest dataset to fit. Specifically, we train a network on OxfordPet both in the annealing and normal training stage, and then use the resulting weights to start training in the normal stage (skipping annealing) on another dataset.

The results are summarized in Table 3. Most F-measures improve, in some cases by a large margin. *UResNext* is, again, the best, and is less than 3 points of F-measure behind the same network trained with ground truth on all datasets except DUT-OMRON. We also show the new gap between the same networks trained with ground truth and regularized loss. The new best gaps are in bold. For the rest of the experiments in this paper, we always start with the weights pretrained on OxfordPet, skipping the annealing stage.

	MSRA-B	ECSSD	DUT-O
UMobV2-reg	85.51	79.20	54.49
UMobV2-gt	89.35	84.78	80.86
gap	3.84	5.58	26.37
UResNext-reg	**88.64**	**83.49**	**73.42**
UResNext-gt	89.68	86.11	81.47
gap	**1.04**	**2.61**	**8.05**

Table 3. Replacing annealing stage by transferring the weights from OxfordPet.

Comparison to Saliency CNNs trained without ground truth There is prior work that train saliency CNN without ground truth [55, 56, 28]. The F-measure comparison is in Table 4. For fairness of comparison, as done in [55, 56, 28], we use the model trained on MSRA-B to test on ECSSD and DUT-OMRON.

Our method outperforms other methods on DUT-OMRON, and is not far from the best other methods on MSRA-B and ECSSD. Note that in addition, our method is generic and not tuned in architecture or loss function to the saliency segmentation task, where as all three methods in [55] are saliency specific.

	MSRA-B	ECSSD	DUT-O
[55]	NA	78.70	58.30
[56]	87.70	**87.83**	71.56
[28]	**90.31**	87.42	73.58
Ours *UMobV2-reg*	85.51	83.50	70.47
Ours *UResNext-reg*	88.64	86.10	**74.77**

Table 4. Comparison to other saliency-CNNs that do not use ground truth for training.

4.2 Cosegmentation

The goal of cosegmentation [32] is to segment an object common to all images in a dataset. The setting of this problem is somewhat different from the setting we assume. We assume that the object of interest is the main subject of the image. For co-segmentation datasets, there maybe several salient objects of different classes, and the task is to find the object class that occurs in all images.

We use Pascal-VOC cosegmentation dataset [10] for evaluation, the most challenging cosegmentation benchmark. It has 20 separate subsets, one for each Pascal category. This dataset is particularly challenging for our approach because the number of images per class is only around 50, and our method needs more images for accurate training, compared to training with ground truth, see supplementary materials. We use UnetX network and train at resolution 256×256.

There is one prior works that trains CNN for cosegmentation without full supervision [14]. The comparison with our method is in Table 5. Our performance, marked as $ours_1$, is slightly inferior. However, our method is generic and unchanged between various applications/datasets. The method in [14] is handcrafted for co-segmentation, has a costly loss function that depends on pairs of images, and thus has quadratic complexity, which is infeasible for large datasets. Furthermore, in [14] they use post-processing with dense-CRF and do not report results without post-processing. We do not use any post-processing. For comparison, in this table, we also include the most recent prior work [24] on cosegmentation that does use full ground truth for training. It performs only slightly better than [14] and $ours_1$, despite using full supervision.

Next we add negative loss from Eq. 7. Namely, when training for, say, 'airplane' category, images in all the other categories are used as negative examples. Note that negative images can also have objects of class 'airplane', since the categories in cosegmentation dataset are not "pure". That is the category "person" has an object of type person in all examples, but may also have "airplane" objects in some examples. Still, negative loss improves results, see column $ours_2$ in Fig. 5. We now outperform [14]. Although we are using the same dataset as [14], this comparison is somewhat unfair since [14] does not use any loss term that depends on data across different object categories. Still, it is reasonable to assume that the user provides examples of images without the object of interest as negative examples. Fig. 3 illustrates how negative loss improves performance.

Cosegmentation dataset has a small number of images, a problem for our method. If we use Web dataset from Sec. 4.3 for training and evaluate on Pascal cosegmentation, then our results are much improved, under $ours_3$ in Fig. 5. Training with more images does give us an advantage over [14]. But the method in [14], with its quadratic loss function, is infeasible for large datasets.

	[14]	[24]*	$ours_1$	$ours_2$	$ours_3$
airplane	0.77	0.78	0.78	0.82	**0.83**
bike	0.27	0.29	0.27	0.28	**0.32**
bird	0.70	**0.71**	0.68	0.70	0.67
boat	0.61	0.66	0.53	0.61	**0.68**
bottle	0.58	0.58	0.58	0.61	**0.72**
bus	0.79	0.82	0.76	0.84	**0.84**
car	0.76	0.79	0.76	0.79	**0.82**
cat	0.79	0.81	0.81	0.82	**0.86**
chair	0.29	0.35	0.34	0.35	**0.38**
cow	0.75	0.78	0.77	0.80	**0.83**
dining table	**0.28**	0.26	0.15	0.16	0.21
dog	0.63	0.65	0.70	0.70	**0.76**
horse	0.66	0.78	0.75	0.77	**0.79**
motorbike	0.65	0.69	0.69	0.71	**0.73**
person	0.37	0.39	0.44	0.46	**0.60**
potted plant	0.42	0.45	0.41	0.45	**0.60**
sheep	0.75	0.77	0.77	0.80	**0.81**
sofa	**0.67**	0.70	0.55	0.60	**0.67**
train	0.68	0.73	0.68	0.75	**0.82**
tv	0.51	**0.55**	0.33	0.40	0.50
mean	0.60	0.63	0.59	0.62	**0.67**

Table 5. Results on Pascal VOC Cosegmentation, evaluation metric is Jaccard. Here '*' indicates that a method uses full ground truth for training.

4.3 Semantic Segmentation

We extend our approach to handle weakly supervised semantic segmentation following a simpler version of [36]. We use Web image search on 20 class names in Pascal VOC [9] to automatically collect about 800 images per class. Let us call these datasets as 'airplane'-Web, 'bicycle'-Web, etc. Then we train our *UMobV2-reg* on each class separately. We train at resolution 256×256. Let the resulting trained CNNs be called 'airplane'-CNN, 'bicycle'-CNN, etc.

We use the obtained segmentations to generate pseudo-ground truth. We run airplane-CNN on airplane-Web. Each pixel that is labelled 'salient' is assigned to label 'airplane'. Each pixel that is labeled 'background' stays assigned 'background'. There are no ambiguity issues as images in airplane-Web dataset go only through airplane-CNN. We process all other classes similarly. Note that

Fig. 3. Sample results on Pascal VOC cosegmentation dataset. The columns are: input image, ground truth, our results without negative loss, our results with negative loss. Negative loss helps to rule out the salient objects that are not of the class of interest.

our pseudo-ground truth is multi-label, but each individual image has only 2 labels: background and one of the object classes. Similarly to cosegmentation, we train both with and without negative loss in Eq. (7).

Method	[36]*	[47]#	[48]#	[1]#	[37]*	[54]	[23]	[38]	$ours_1^*$	$ours_2^*$	$ours_3^*$
mIoU	56.4	60.3	60.4	61.7	63.0	63.3	64.9	64.9	64.0	66.7	67.1

Table 6. Comparison (mIoU metric) to other weakly supervised semantic segmentation methods on Pascal VOC 2012 validation fold.

Our Web dataset has around 16K images and thus is larger than the augmented Pascal training dataset. However, our Web images are of lower similarity to Pascal validation images. Therefore we also train on the subset of augmented Pascal training dataset that has only one class per image. We call this dataset PascalSingle, it has around 5K images. We generate pseudo-ground truth for PascalSingle using the same procedure as for our Web dataset, described above.

Next we train a multi-class CNN on our pseudo-ground truth. We use DeepLab-ResNet101 [5] pretrained on ImageNet [8]. We follow the same training strategy as in [5], except our initial learning rate is 0.001. DeepLab is trained on images of size 513×513. We also use random horizontal flip, Gaussian blur, and rescaling for data augmentation. We train for 100 epochs.

Table 6 compares the performance of our method with the most related or most recent work. Methods marked with "*" use additional data for training

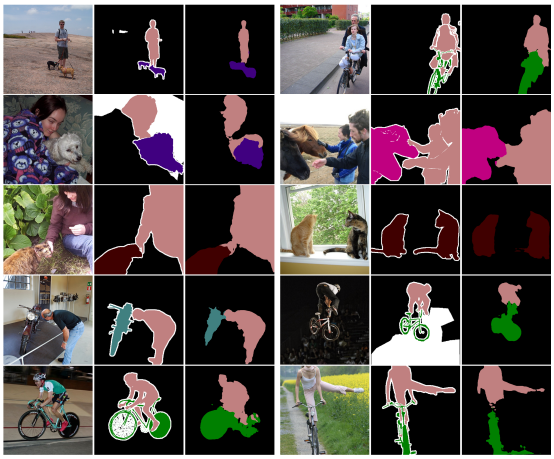

Fig. 4. Sample results on Pascal VOC 2012 validation fold.

(with image-level tags), usually obtained by Web search. Methods marked with "#" use saliency datasets together with their pixel precise ground truth. We compare three versions of our method. $Ours_1$ is trained on Web images. $Ours_2$ and $Ours_3$ are trained on PascalSingle dataset, without and with negative loss, respectively. Our method trained with negative loss outperforms all prior work.

Fig. 4 shows some sample segmentations. Despite being trained on images each containing only one semantic class, we are able to segment images containing multiple semantic classes. Note that we do not use any post-processing, and therefore our results are likely to improve with CRF-based refinement.

Conclusions

We presented a new method of training segmentation CNNs using regularized loss instead of ground truth. Our approach is a simple end-to-end trainable system with no need to adapt architecture or loss for a specific dataset/task. In some cases, our method obtains results that are only a few points behind the results when training the same CNN with pixel precise ground truth. There are several future research directions. First our regularized loss is rather simple and cannot work well for all object classes. It will be interesting to research other loss functions that can be useful for training. Second research direction is extending our approach to a multi-class setting in a unified manner, not as a separate pseudo-ground truth generating stage, as we do currently.

References

1. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: Conference on Computer

Vision and Pattern Recognition. pp. 4981–4990 (2018)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(11), 1222–1239 (November 2001)
3. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: 9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France. pp. 26–33 (2003)
4. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: International Conference on Learning Research (2015)
5. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2018)
6. Chen, L., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision. pp. 833–851 (2018)
7. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: International Conference on Computer Vision. pp. 1635–1643 (2015)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (June 2010)
10. Faktor, A., Irani, M.: Co-segmentation by composition. In: International Conference on Computer Vision. pp. 1297–1304 (2013)
11. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013), in press
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT Press (2016)
13. Gorelick, L., Veksler, O., Boykov, Y., Nieuwenhuis, C.: Convexity shape prior for segmentation. In: European Conference on Computer Vision. pp. 675–690 (2014)
14. Hsu, K.J., Lin, Y.Y., Chuang, Y.Y.: Co-attention cnns for unsupervised object co-segmentation. In: IJCAI. pp. 748–756 (2018)
15. Khoreva, A., Benenson, R., Hosang, J.H., Hein, M., Schiele, B.: Simple does it: Weakly supervised instance and semantic segmentation. In: Conference on Computer Vision and Pattern Recognition. pp. 1665–1674 (2017)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
18. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: European Conference on Computer Vision. pp. 695–711 (2016)
19. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: Neural Information Processing Systems. pp. 109–117 (2011)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems (2012)
21. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields. In: International Conference on Machine Learning (2001)

22. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (December 1989)
23. Lee, J., Kim, E., Lee, S., Lee, J., Yoon, S.: Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In: Conference on Computer Vision and Pattern Recognition. pp. 5267–5276 (2019)
24. Li, B., Sun, Z., Li, Q., Wu, Y., Hu, A.: Group-wise deep object co-segmentation with co-attention recurrent neural network. In: International Conference on Computer Vision (October 2019)
25. Li, S.: Markov Random Field Modeling in Computer Vision. Springer-Verlag (1995)
26. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: Conference on Computer Vision and Pattern Recognition. pp. 3159–3167 (2016)
27. Liu, T., Sun, J., Zheng, N.N., Tang, X., Shum, H.Y.: Learning to detect a salient object. In: Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
28. Nguyen, T., Dax, M., Mummadi, C.K., Ngo, N., Nguyen, T.H.P., Lou, Z., Brox, T.: Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In: Advances in Neural Information Processing Systems. pp. 204–214 (2019)
29. Papandreou, G., Chen, L.C., Murphy, K.P., Yuille, A.L.: Weakly and semi supervised learning of a deep convolutional network for semantic image segmentation. In: International Conference on Computer Vision. pp. 1742–1750 (2015)
30. Pinheiro, P.H.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: Conference on Computer Vision and Pattern Recognition. pp. 1713–1721 (2015)
31. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. pp. 234–241 (2015)
32. Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In: Conference on Computer Vision and Pattern Recognition. vol. 1, pp. 993–1000 (2006)
33. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
34. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: International Conference on Computer Vision. pp. 618–626 (2017)
35. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. Transactions on Pattern Analysis and Machine Intelligence **39**(4), 640–651 (2017)
36. Shen, T., Lin, G., Liu, L., Shen, C., Reid, I.D.: Weakly supervised semantic segmentation based on co-segmentation. In: BMVC (2017)
37. Shen, T., Lin, G., Shen, C., Reid, I.: Bootstrapping the performance of webly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1363–1371 (2018)
38. Shimoda, W., Yanai, K.: Self-supervised difference detection for weakly-supervised semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5208–5217 (2019)
39. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. In: International Conference on Computer Vision (2019)

40. Tang, M., Djelouah, A., Perazzi, F., Boykov, Y., Schroers, C.: Normalized cut loss for weakly-supervised CNN segmentation. In: Conference on Computer Vision and Pattern Recognition. pp. 1818–1827 (2018)
41. Tang, M., Perazzi, F., Djelouah, A., Ayed, I.B., Schroers, C., Boykov, Y.: On regularized losses for weakly-supervised CNN segmentation. In: European Conference on Computer Vision. pp. 524–540 (2018)
42. Vedaldi, A.: Cats and dogs. In: Conference on Computer Vision and Pattern Recognition (2012)
43. Veksler, O.: Efficient graph cut optimization for full crfs with quantized edges. Transactions on Pattern Analysis and Machine Intelligence (2018)
44. Vernaza, P., Chandraker, M.: Learning random-walk label propagation for weakly-supervised semantic segmentation. In: Conference on Computer Vision and Pattern Recognition. pp. 2953–2961 (2017)
45. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008)
46. Vogiatzis, G., Torr, P., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: Conference on Computer Vision and Pattern Recognition. pp. II: 391–398 (2005)
47. Wang, X., You, S., Li, X., Ma, H.: Weakly-supervised semantic segmentation by iteratively mining common object features. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1354–1362 (2018)
48. Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., Huang, T.S.: 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. pp. 7268–7277 (2018)
49. Wu, H., Zhang, J., Huang, K., Liang, K., Yu, Y.: Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. CoRR **abs/1903.11816** (2019)
50. Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Conference on Computer Vision and Pattern Recognition. pp. 5987–5995 (2017)
51. Xu, J., Schwing, A.G., Urtasun, R.: Learning to segment under various forms of weak supervision. In: Conference on Computer Vision and Pattern Recognition. pp. 3781–3790 (2015)
52. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: Conference on Computer Vision and Pattern Recognition. pp. 1155–1162. IEEE (2013)
53. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: Conference on Computer Vision and Pattern Recognition. pp. 3166–3173. IEEE (2013)
54. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L.: Joint learning of saliency detection and weakly supervised semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7223–7233 (2019)
55. Zhang, D., Han, J., Zhang, Y.: Supervision by fusion: Towards unsupervised learning of deep salient object detector. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4048–4056 (2017)
56. Zhang, J., Zhang, T., Dai, Y., Harandi, M., Hartley, R.: Deep unsupervised saliency detection: A multiple noisy labeling perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9029–9038 (2018)
57. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: International Conference on Computer Vision. pp. 1529–1537 (2015)

58. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Conference on Computer Vision and Pattern Recognition. pp. 2921–2929 (2016)