Weakly Supervised Instance Segmentation by Learning Annotation Consistent Instances

Aditya Arun[1], C.V. Jawahar[1], and M. Pawan Kumar[2]

[1] CVIT, KCIS, IIIT Hyderabad
[2] OVAL, University of Oxford

Abstract. Recent approaches for weakly supervised instance segmentations depend on two components: (i) a pseudo label generation model which provides instances that are consistent with a given annotation; and (ii) an instance segmentation model, which is trained in a supervised manner using the pseudo labels as ground-truth. Unlike previous approaches, we explicitly model the uncertainty in the pseudo label generation process using a conditional distribution. The samples drawn from our conditional distribution provide accurate pseudo labels due to the use of semantic class aware unary terms, boundary aware pairwise smoothness terms, and annotation aware higher order terms. Furthermore, we represent the instance segmentation model as an annotation agnostic prediction distribution. In contrast to previous methods, our representation allows us to define a joint probabilistic learning objective that minimizes the dissimilarity between the two distributions. Our approach achieves state of the art results on the PASCAL VOC 2012 data set, outperforming the best baseline by 4.2% $\text{mAP}^r_{0.5}$ and 4.8% $\text{mAP}^r_{0.75}$.

1 Introduction

The instance segmentation task is to jointly estimate the class labels and segmentation masks of the individual objects in an image. Significant progress on instance segmentation has been made based on the convolutional neural networks (CNN) [9, 17, 24, 26, 28]. However, the traditional approach of learning CNN-based models requires a large number of training images with instance-level pixel-wise annotations. Due to the high cost of collecting these supervised labels, researchers have looked at training these instance segmentation models using weak annotations, ranging from bounding boxes [18, 20] to image-level labels [1, 10, 13, 23, 42, 43].

Many of the recent approaches for weakly supervised instance segmentation can be thought of as consisting of two components. First, a pseudo label generation model, which provides instance segmentations that are consistent with the weak annotations. Second, an instance segmentation model which is trained by treating the pseudo labels as ground-truth, and provides the desired output at test time.

Seen from the above viewpoint, the design of a weakly supervised instance segmentation approach boils down to three questions. First, how do we represent

the instance segmentation model? Second, how do we represent the pseudo label generation model? And third, how do we learn the parameters of the two models using weakly supervised data? The answer to the first question is relatively clear: we should use a model that performs well when trained in a supervised manner, for example, Mask R-CNN [17]. However, we argue that the existing approaches fail to provide a satisfactory answer to the latter two questions.

Specifically, the current approaches do not take into account the inherent uncertainty in the pseudo label generation process [1, 23]. Consider, for instance, a training image that has been annotated to indicate the presence of a person. There can be several instance segmentations that are consistent with this annotation, and thus, one should not rely on a single pseudo label to train the instance segmentation model. Furthermore, none of the existing approaches provide a coherent learning objective for the two models. Often they suggest a simple two-step learning approach, that is, generate one set of pseudo labels followed by a one time training of the instance segmentation model [1]. While some works consider an iterative training procedure [23], the lack of a learning objective makes it difficult to analyse and adapt them in varying settings.

In this work, we address the deficiencies of prior work by (i) proposing suitable representations for the two aforementioned components; and (ii) estimating their parameters using a principled learning objective. In more detail, we explicitly model the uncertainty in pseudo labels via a conditional distribution. The conditional distribution consists of three terms: (i) a semantic class aware unary term to predict the score of each segmentation proposal; (ii) a boundary aware pairwise term that encourages the segmentation proposal to completely cover the object; and (iii) an annotation consistent higher order term that enforces a global constraint on all segmentation proposals (for example, in the case of image-level labels, there exists at least one corresponding segmentation proposal for each class, or in the case of bounding boxes, there exists a segmentation proposal with sufficient overlap to each bounding box). All three terms combined enable the samples drawn from the conditional distribution to provide accurate annotation consistent instance segmentations. Furthermore, we represent the instance segmentation model as an annotation agnostic prediction distribution. This choice of representation allows us to define a joint probabilistic learning objective that minimizes the dissimilarity between the two distributions. The dissimilarity is measured using a task-specific loss function, thereby encouraging the models to produce high quality instance segmentations.

We test the efficacy of our approach on the Pascal VOC 2012 data set. We achieve 50.9% $\text{mAP}^r_{0.5}$, 28.5% $\text{mAP}^r_{0.75}$ for image-level annotations and 32.1% $\text{mAP}^r_{0.75}$ for bounding box annotations, resulting in an improvement of over 4% and 10% respectively over the state-of-the-art.

2 Related Work

Due to the taxing task of acquiring the expensive per-pixel annotations, many weakly supervised methods have emerged that can leverage cheaper labels. For

the task of semantic segmentation various types of weak annotations, such as image-level [2, 19, 29, 32], point [6], scribbles [25, 39], and bounding boxes [11, 31], have been utilized. However, for the instance segmentation, only image-level [1, 10, 13, 23, 42, 43] and bounding box [18, 20] supervision have been explored. Our setup considers both the image-level and the bounding box annotations as weak supervision. For the bounding box annotations, Hsu *et al.* [18] employs a bounding box tightness constraint and train their method by employing a multiple instance learning (MIL) based objective but they do not model the annotation consistency constraint for computational efficiency.

Most of the initial works [42, 43] on weakly supervised instance segmentation using image-level supervision were based on the class activation maps (CAM) [30, 35, 40, 41]. In their work, Zhou *et al.* [42] identify the heatmap as well as its peaks to represent the location of different objects. Although these methods are good at finding the spatial location of each object instance, they focus only on the most discriminative regions of the object and therefore, do not cover the entire object. Ge *et al* [13] uses the CAM output as the initial segmentation seed and refines it in a multi-task setting, which they train progressively. We use the output of [42] as the initial segmentation seed of our conditional distribution but the boundary aware pairwise term in our conditional distribution encourages pseudo labels to cover the entire object.

Most recent works on weakly supervised learning adopt a two-step process - generate pseudo labels and train a supervised model treating these pseudo labels as ground truth. Such an approach provides state-of-the-art results for various weakly supervised tasks like object detection [5, 37, 38], semantic segmentation [11, 20], and instance segmentation [1, 23]. Ahn *et al.* [1] synthesizes pseudo labels by learning the displacement fields and pairwise pixel affinities. These pseudo labels are then used to train a fully supervised Mask R-CNN [17], which is used at the test time. Laradji *et al.* [23] iteratively samples the pseudo segmentation label from MCG segmentation proposal set [3] and train a supervised Mask R-CNN [17]. This is similar in spirit to our approach of using the two distributions. However, they neither have a unified learning objective for the two distribution nor do they model the uncertainty in their pseudo label generation model. Regardless, the improvement in the results reported by these two methods advocates the importance of modeling two separate distributions. In our method, we explicitly model the two distributions and define a unified learning objective that minimizes the dissimilarity between them.

Our framework has been inspired by the work of Kumar *et al.* [22] who were the first to show the necessity of modeling uncertainty by employing two separate distributions in a latent variable model. This framework has been adopted for weakly supervised training of CNNs for learning human poses and object detection tasks [4, 5]. While their framework provides an elegant formulation for weakly supervised learning, its various components need to be carefully constructed for each task. Our work can be viewed as designing conditional and prediction distributions, as well as the corresponding inference algorithms, which are suited to instance segmentation.

3 Method

3.1 Notation

We denote an input image as $\mathbf{x} \in \mathbb{R}^{(H \times W \times 3)}$, where H and W are the height and the width of the image respectively. For each image, a set of segmentation proposals $\mathcal{R} = \{r_1, \ldots, r_P\}$ are extracted from a class-agnostic object proposal algorithm. In this work, we use Multiscale Combinatorial Grouping (MCG) [3] to obtain the object proposals. For the sake of simplicity, we only consider image-level annotations in our description. However, our framework can be easily extended to other annotations such as bounding boxes. Indeed, we will use bounding box annotations in our experiments. Given an image and the segmentation proposals, our goal is to classify each of the segmentation proposals to one of the $C + 1$ categories from the set $\{0, 1, \ldots, C\}$. Here category 0 is the background and categories $\{1, \ldots, C\}$ are object classes.

We denote the image-level annotations by $\mathbf{a} = \{0, 1\}^C$, where $\mathbf{a}^{(j)} = 1$ if image x contains the $j-$th object. Furthermore, we denote the unknown instance-level (segmentation proposal) label as $\mathbf{y} = \{0, \ldots, C\}^P$, where $\mathbf{y}^{(i)} = j$ if the $i-$th segmentation proposal is of the $j-$th category. A weakly supervised data set $\mathcal{W} = \{(\mathbf{x}_n, \mathbf{a}_n) \mid n = 1, \ldots, N\}$ contains N pairs of images $\mathbf{x}_n$ and their corresponding image-level annotations $\mathbf{a}_n$.

3.2 Conditional Distribution

Given the weakly supervised data set $\mathcal{W}$, we wish to generate pseudo instance-level labels $\mathbf{y}$ such that they are annotation consistent. Specifically, given the segmentation proposals $\mathcal{R}$ for an image $\mathbf{x}$, there must exists at least one segmentation proposal for each image-level annotation $\mathbf{a}^{(j)} = 1$. Since the annotations are image-level, there is inherent uncertainty in the figure-ground separation of the objects. We model this uncertainty by defining a distribution $\Pr_c(\mathbf{y} \mid \mathbf{x}, \mathbf{a}; \boldsymbol{\theta}_c)$ over the pseudo labels conditioned on the image-level weak annotations. Here, $\boldsymbol{\theta}_c$ are the parameters of the distribution. We call this a *conditional distribution*.

The conditional distribution itself is not explicitly represented. Instead, we use a neural network with parameters $\boldsymbol{\theta}_c$ which generates samples that can be used as pseudo labels. For the generated samples to be accurate, we wish that they have the following three properties: (i) they should have high fidelity with the scores assigned by the neural network for each region proposal belonging to each class; (ii) they should cover as large a portion of an object instance as possible; and (iii) they should be consistent with the annotation.

Modeling: In order for the conditional distribution to be annotation consistent, the instance-level labels $\mathbf{y}$ need to be compatible with the image-level annotation $\mathbf{a}$. This constraint cannot be trivially decomposed over each segmentation proposal. As a result, it would be prohibitively expensive to model the conditional distribution directly as one would be required to compute its partition function. Taking inspiration from Arun *et al.* [5], we instead draw representative samples

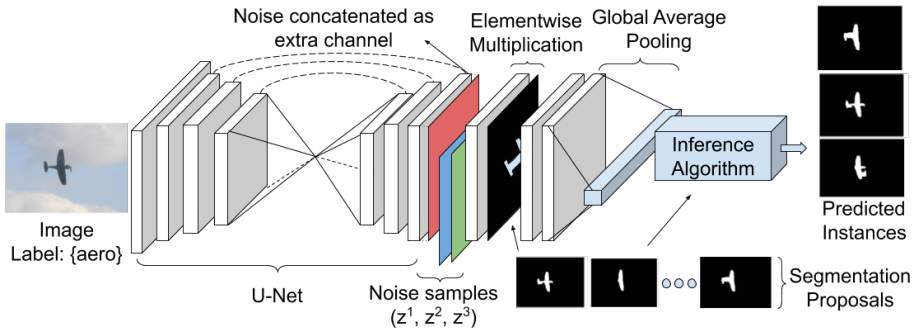

Fig. 1. *The conditional network: a modified U-Net architecture is used to model the conditional network. For a single input image and three different noise samples* $\{z^1, z^2, z^3\}$ *(represented as red, green, and blue matrix) and a pool of segmentation proposals, three different instances are predicted for the given weak annotation (aeroplane in this example). Here the noise sample is concatenated as an extra channel to the final layer of the U-Net. The segmentation proposals are multiplied element-wise with the global feature to obtain the proposal specific feature. A global average pooling is applied to get class specific score. Finally, an inference algorithm generates the predicted samples.*

from the conditional distribution using the Discrete Disco Nets [7]. We will now describe how we model the conditional distribution through a Discrete Disco Nets, which we will now call a *conditional network*.

Consider the modified fully convolutional U-Net [34] architecture shown in figure 1 for the conditional distribution. The parameters of the conditional distribution $\boldsymbol{\theta}_c$ are modeled by the weights of the conditional network. Similar to [21], noise sampled from a uniform distribution is added after the U-Net block (depicted by the colored filter). Each forward pass through the network takes the image $\mathbf{x}$ and noise sample $\mathbf{z}^k$ as input and produces a score function $F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$ for each segmentation proposal u and the corresponding putative label $\mathbf{y}_u$. We generate K different score functions using K different noise samples. These score functions are then used to sample the segmentation region proposals $\mathbf{y}^k_c$ such that they are annotation consistent. This enables us to efficiently generate the samples from the underlying distribution.

Inference: Given the input pair $(\mathbf{x}, \mathbf{z}^k)$ the conditional network outputs K score functions for each of the segmentation proposal $F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$. We redefine these score functions to obtain a final score function such that it is then used to sample the segmentation region proposals $\mathbf{y}^k_c$. The final score function has the following three properties.

1. The score of the sampled segmentation region proposal should be consistent with the score function. This *semantic class aware unary term* ensures that the final score captures the class specific features of each segmentation proposal. Formally, $G^k_{u,\mathbf{y}_u}(\mathbf{y}_c) = F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$.

2. The unary term alone is biased towards segmentation proposals that are highly discriminative. This results in selecting a segmentation proposal which does not cover the object in its entirety. We argue that all the neighboring segmentation proposals must have the same score discounted by the edge weights between them. We call this condition *boundary aware pairwise term*. In order to make the score function $G^k_{u,\mathbf{y}_u}(\mathbf{y}_c)$ pairwise term aware, we employ a simple but efficient iterative algorithm. The algorithm proceeds by iteratively updating the scores $G^k_{u,\mathbf{y}_u}(\mathbf{y}_c)$ by adding the contribution of their neighbors discounted by the edge weights between them until convergence. In practice, we fix the number of iteration to 3. Note that, it is possible to backpropagate through the iterative algorithm by simply unrolling its iterations, similar to a recurrent neural networks (RNN). Formally,

$$G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) = G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) + \frac{1}{H^{k,n-1}_{u,v}(\mathbf{y}_c) + \delta} \exp\left(-I_{u,v}\right). \tag{1}$$

   Here, n denotes the iteration step for the iterative algorithm and δ is a small positive constant added for numerical stability. In our experiments, we set $\delta = 0.1$. The term $H^{k,n-1}_{u,v}(\mathbf{y}_c)$ is the difference between the scores of the neighboring segmentation proposal. It helps encourage same label for the neighboring segmentation proposals that are not separated by the edge pixels. It is given as,

$$H^{k,n-1}_{u,v}(\mathbf{y}_c) = \sum_{u,v \in \mathcal{N}_u} \left(G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) - G^{k,n-1}_{v,\mathbf{y}_u}(\mathbf{y}_c)\right)^2. \tag{2}$$

   The term $I_{u,v}$ is the sum of the edge pixel values between the two neighboring segmentation regions. Note that the pairwise term is a decay function weighted by the edge pixel values. This ensures a high contribution to the pairwise term is only from the pair of segmentation proposals that does not share an edge.
3. In order to ensure that at there must exist at least one segmentation proposal for every image-level annotation,a higher order penalty is added to the score. We call this *annotation consistent higher order term*. Formally,

$$S^k(\mathbf{y}_c) = \sum_{u=1}^{P} G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) + Q^k(\mathbf{y}_c). \tag{3}$$

   Here,

$$Q^k(\mathbf{y}_c) = \begin{cases} 0 & \text{if } \forall j \in \{1,\ldots,C\} \text{ s.t. } \mathbf{a}^{(j)} = 1, \\ & \exists i \in \mathcal{R} \text{ s.t. } \mathbf{y}^{(i)} = j, \\ -\infty & \text{otherwise.} \end{cases} \tag{4}$$

   Given the scoring function in equation (3), we compute the $k-$th sample of the conditional network as,

$$\mathbf{y}^k_c = \arg\max_{\mathbf{y} \in \mathcal{Y}} S^k(\mathbf{y}_c). \tag{5}$$

Algorithm 1: Inference Algorithm for the Conditional Net

Input : Region masks: R, Image-level labels: a
Output: Predicted instance level instances: $\mathbf{y}_c^k$

/* Iterative Algorithm */
$G^k_{u,\mathbf{y}_u}(\mathbf{y}_c) = F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$
repeat
  for $v \in \mathcal{N}_u$ **do**
    $H^{k,n-1}_{u,v}(\mathbf{y}_c) = \sum_{u,v\in\mathcal{N}_u} \left(G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) - G^{k,n-1}_{v,\mathbf{y}_v}(\mathbf{y}_c)\right)^2.$
    $G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) = G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) + \frac{1}{H^{k,n-1}_{u,v}(\mathbf{y}_c)+\delta} \exp\left(-I_{u,v}\right)$
until $G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c)$ *has coverged*
/* Greedily select highest scoring non-overlapping proposal */
$Y \leftarrow \phi$
for $j \leftarrow \{1,\ldots,C\} \wedge \mathbf{a}^{(j)} = 1$ **do**
  $Y_j \leftarrow \phi$
  $R_j \leftarrow sort(G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c))$
  for $i \in 1,\ldots,P$ **do**
    $Y_j \leftarrow r_i$
    $R_j \leftarrow R_j - r_i$
    for $l \in R_j \wedge \frac{r_i \cap r_l}{r_l} > t$ **do**
      $R_j \leftarrow R_j - r_l$
  $Y \leftarrow Y_j$
return $\mathbf{y}_c^k = Y$

Observe that in equation (5), the arg max is computed over the entire output space $\mathcal{Y}$. A naïve brute force algorithm is therefore not feasible. We design an efficient greedy algorithm that selects the highest scoring non-overlapping proposal. The inference algorithm is described in Algorithm 1.

3.3 Prediction Distribution

The task of the supervised instance segmentation model is to predict the instancemask given an image. We employ Mask R-CNN [18] for this task. As predictions for each of the regions in the Mask R-CNN is computed independently, we can view the output of the Mask R-CNN as the following fully factorized distribution,

$$\Pr_p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}_p) = \prod_{i=1}^{R} \Pr(\mathbf{y}_i \mid \mathbf{r}_i, \mathbf{x}_i; \boldsymbol{\theta}_p). \tag{6}$$

Here, R are the set of bounding box regions proposed by the region proposal network and $\mathbf{r}_i$ are its corresponding region features. The term $\mathbf{y}_i$ is the corresponding prediction for each of the bounding box proposals. We call the above distribution a *prediction distribution* and the Mask R-CNN a *prediction network*.

4 Learning Objective

Given the weakly supervised data set $\mathcal{W}$, our goal is to learn the parameters of the prediction and the conditional distribution, $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_c$ respectively. We observe that the task of both the prediction and the conditional distribution is to predict the instance segmentation mask. Moreover, the conditional distribution utilizes the extra information in the form of image-level annotations. Therefore, it is expected to produce better instance segmentation masks. Leveraging the task similarity between the two distribution, we would like to bring the two distribution close to each other. Inspired by the work of [4, 5, 8, 22], we design a joint learning objective that can minimize the dissimilarity coefficient [33] between the prediction and the conditional distribution. In what follows, we briefly describe the dissimilarity coefficient before applying it to our setting.

Dissimilarity Coefficient: The dissimilarity coefficient between any two distributions $\Pr_1(\cdot)$ and $\Pr_2(\cdot)$ is determined by measuring their diversities. Given a task-specific loss function $\Delta(\cdot,\cdot)$, the diversity coefficient between the two distribution $\Pr_1(\cdot)$ and $\Pr_2(\cdot)$ is defined as the expected loss between two samples drawn randomly from the two distributions respectively. Formally,

$$DIV_{\Delta}(\Pr_1, \Pr_2) = \mathbb{E}_{\mathbf{y}_1 \sim \Pr_1(\cdot)} \left[\mathbb{E}_{\mathbf{y}_2 \sim \Pr_2(\cdot)} [\Delta(\mathbf{y}_1, \mathbf{y}_2)] \right]. \tag{7}$$

If the model brings the two distributions close to each other, we could expect the diversity $DIV_{\Delta}(\Pr_1, \Pr_2)$ to be small. Using this definition, the dissimilarity coefficient is defined as the following Jensen difference,

$$\begin{aligned} DISC_{\Delta}(\Pr_1, \Pr_2) = DIV_{\Delta}(\Pr_1, \Pr_2) - \gamma DIV_{\Delta}(\Pr_2, \Pr_2) \\ - (1-\gamma) DIV_{\Delta}(\Pr_1, \Pr_1), \end{aligned} \tag{8}$$

where, $\gamma = [0, 1]$. In our experiments, we use $\gamma = 0.5$, which results in dissimilarity coefficient being symmetric for the two distributions.

4.1 Task-Specific Loss Function:

The dissimilarity coefficient objective requires a task-specific loss function. To this end, we use the multi-task loss defined by Mask R-CNN [17] as,

$$\Delta(\mathbf{y}_1, \mathbf{y}_2) = \Delta_{\text{cls}}(\mathbf{y}_1, \mathbf{y}_2) + \Delta_{\text{box}}(\mathbf{y}_1, \mathbf{y}_2) + \Delta_{\text{mask}}(\mathbf{y}_1, \mathbf{y}_2). \tag{9}$$

Here, Δ_{cls} is the classification loss defined by the log loss, Δ_{box} is the bounding box regression loss defined as the smooth-L1 loss, and Δ_{mask} is the segmentation loss for the mask defined by pixel-wise cross entropy, as proposed by [17].

Note that the conditional network outputs the segmentation region $\mathbf{y}$, where there are no bounding box coordinates predicted. Therefore, for the conditional network, only Δ_{cls} and Δ_{mask} is active as the gradients for Δ_{box} is 0. For the prediction network, all three components of the loss functions are active. We construct a tight bounding box around the pseudo segmentation label, which acts as a pseudo bounding box label for Mask R-CNN.

4.2 Learning Objective for Instance Segmentation:

We now specify the learning objective for instance segmentation using the dissimilarity coefficient and the task-specific loss function defined above as,

$$\boldsymbol{\theta}_p^*, \boldsymbol{\theta}_c^* = \arg\min_{\boldsymbol{\theta}_p, \boldsymbol{\theta}_c} DISC_{\Delta}\left(\Pr_p(\boldsymbol{\theta}_p), \Pr_c(\boldsymbol{\theta}_c)\right). \tag{10}$$

As discussed in Section 3.2, modeling the conditional distribution directly is difficult. Therefore, the corresponding diversity terms are computed by stochastic estimators from K samples y_c^k of the conditional network. Thus, each diversity term is written as[3],

$$DIV_{\Delta}(\Pr_p, \Pr_c) = \frac{1}{K} \sum_{k=1}^{K} \sum_{\mathbf{y}_p^{(i)}} \Pr_p(\mathbf{y}_p^{(i)}; \boldsymbol{\theta}_p) \Delta(\mathbf{y}_p^{(i)}, \mathbf{y}_c^k), \tag{11a}$$

$$DIV_{\Delta}(\Pr_c, \Pr_c) = \frac{1}{K(K-1)} \sum_{\substack{k,k'=1 \\ k' \neq k}}^{K} \Delta(\mathbf{y}_c^k, \mathbf{y}_c^{k'}), \tag{11b}$$

$$DIV_{\Delta}(\Pr_p, \Pr_p) = \sum_{\mathbf{y}_p^{(i)}} \sum_{\mathbf{y'}_p^{(i)}} \Pr_p(\mathbf{y}_p^{(i)}; \boldsymbol{\theta}_p) \Pr_p(\mathbf{y'}_p^{(i)}; \boldsymbol{\theta}_p) \Delta(\mathbf{y}_p^{(i)}, \mathbf{y'}_p^{(i)}) \tag{11c}$$

Here, $DIV_{\Delta}(\Pr_p, \Pr_c)$ measures the cross diversity between the prediction and the conditional distribution, which is the expected loss between the samples of the two distribution. Since $\Pr_p$ is a fully factorized distribution, the expectation of its output can be trivially computed. As $\Pr_c$ is not explicitly modeled, we draw K different samples to compute its required expectation.

5 Optimization

As the parameters of the two distribution, θ_p and θ_c are modeled by a neural network, it is ideally suited to be minimized by stochastic gradient descent. We employ a block coordinate descent strategy to optimize the two sets of parameters. The algorithm proceeds by iteratively fixing the prediction network and training the conditional network, followed by learning the prediction network for a fixed conditional network.

The iterative learning strategy results in a fully supervised training of each network by using the output of the other network as the pseudo label. This allows us to readily use the algorithms developed in Mask R-CNN [17] and Discrete Disco Nets [7]. Note that, as the conditional network obtains samples over the arg max operator in equation (5), the objective (10) for the conditional network is non-differentiable. However, the scoring function $S^k(\mathbf{y}_c)$ in equation (3) itself is differentiable. This allows us to use the direct loss minimization strategy [16, 36] developed for computing estimated gradients over the arg max operator [7, 27]. We provide the details of the algorithm in the supplementary.

[3] Details in the supplementary material

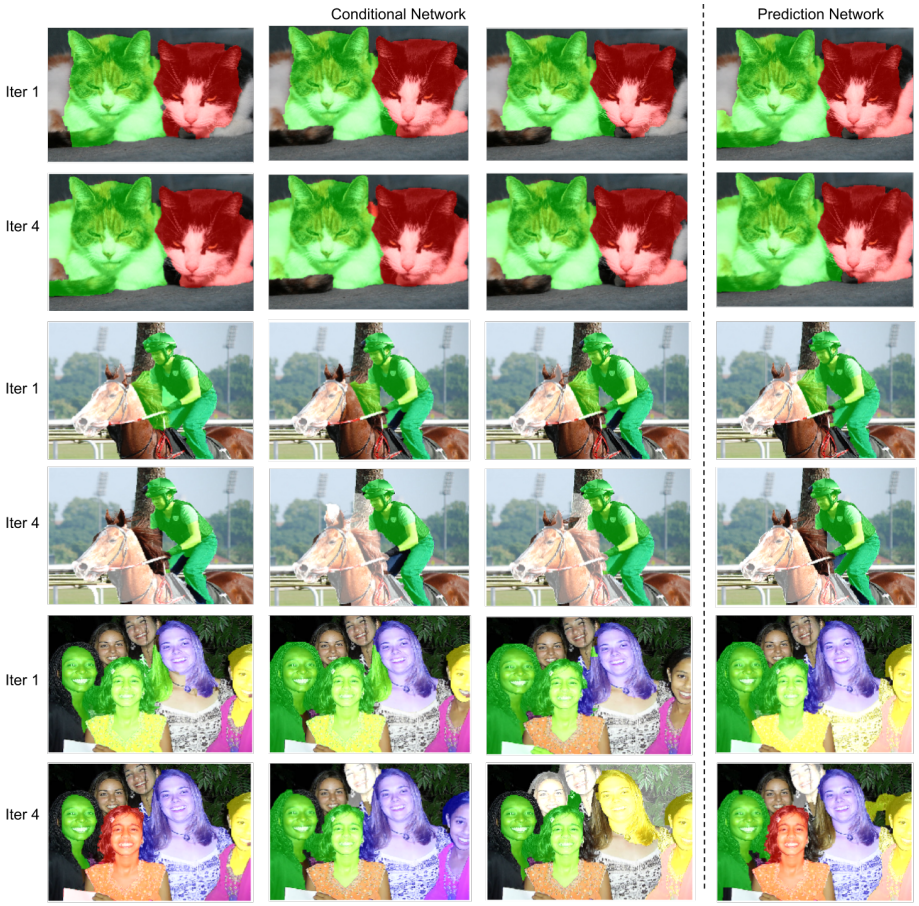

Fig. 2. *Examples of the predictions from the conditional and prediction networks for three different cases of varying difficulty. Columns 1 through 3 are different samples from the conditional network. For each case, its first row shows the output of the two networks after the first iteration and its second row represents the output of the two networks after the fourth (final) iteration. Each instance of an object is represented by different mask color. Best viewed in color.*

5.1 Visualization of the learning process

Figure 2 provides the visualization of the output of the two networks for the first and the final iterations of the training process. The first three columns are the three output samples of the conditional distribution. Note that in our experiments, we output 10 samples corresponding to 10 different noise samples. The fourth column shows the output of the prediction distribution. The output for the prediction network is selected by employing a non-maximal suppression (NMS) with its score threshold kept at 0.7, as is the default setting in [17]. The

first row represents the output of the two networks after the first iteration and the second row shows their output after the fourth (final) iteration.

The first case demonstrates an easy example where two cats are present in the image. Initially, the conditional distribution samples the segmentation proposals which do not cover the entire body of the cat but still manages to capture the boundaries reasonably well. However, due to the variations in these samples, the prediction distribution learns to better predict the extent of the cat pixels. This, in turn, encourages the conditional network to generate a better set of samples. Indeed, by the fourth iteration, we see an improvement in the quality of samples by the conditional network and they now cover the entire body of the cat, thereby improving the performance. As a result, we can see that finally the prediction network successfully learns to segment the two cats in the image.

The second case presents a challenging scenario where a person is riding a horse. In this case, the person is occluding the front and the rear parts of the horse. Initially, we see that the conditional network only provides samples for the most discriminative region of the horse - its face. The samples generated for the person class, though not accurate, covers the entire person. We observe that over the subsequent iterations, we get an accurate output for the person class. The output for the horse class also expands to cover its front part completely. However, since its front and the rear parts are completely separated, the final segmentation could not cover the rear part of the horse.

The third case presents another challenging scenario where there are multiple people present. Four people standing in front and two are standing at the back. Here, we observe that initially, the conditional network fails to distinguish between the two people standing in the front-left of the image and fails to detect persons standing at the back. The samples for the third and the fourth persons standing in front-center and front-right respectively are also not accurate. Over the iterations, the conditional network improves its predictions for the four people standing in front and also sometimes detect the people standing at the back. As a result, prediction network finally detects five of the six people in the image.

6 Experiments

6.1 Data set and Evaluation Metric

Data Set: We evaluate our proposed method on Pascal VOC 2012 segmentation benchmark [12]. The data set consists of 20 foreground classes. Following previous works [1, 13, 18, 20], we use the augmented Pascal VOC 2012 data set [14], which contains $10,582$ training images.

From the augmented Pascal VOC 2012 data set, we construct two different weakly supervised data sets. The first data set is where we retain only the image-level annotations. For the second data set, we retain the bounding box information along with the image-level label. In both the data sets, the pixel-level labels are discarded.

Evaluation Metric: We adopt the standard evaluation metric for instance segmentation, mean average precision (mAP) [15]. Following the same evaluation protocol from other competing approaches, we report mAP with four intersection over union (IoU) thresholds, denoted by mAP_k^r where k denotes the different values of IoU and $k = \{0.25, 0.50, 0.70, 0.75\}$.

6.2 Initialization

We now discuss various strategies to initialize our conditional network for different levels of weakly supervised annotations.

Image Level Annotations: Following the previous works on weakly supervised instance segmentation from image-level annotations [1, 23, 43], we use the Class Activation Maps (CAMs) to generate the segmentation seeds for each image in the training set. Specifically, like [1, 23, 43], we rely on the Peak Response Maps (PRM) [42] to generate segmentation seeds that identify the salient parts of the objects. We utilize these seeds as pseudo segmentation labels to initially train our conditional network. We also filter the MCG [3] segmentation proposal such that each selected proposal has at least a single pixel overlap with the PRM segmentation seeds. This helps us reduce the number of segmentation proposals needed thereby reducing the memory requirement. Once the initial training for the conditional network is over, we proceed with the iterative optimization strategy, described in section 5.

Bounding Box Annotations For the weakly supervised data set where bounding box annotations are present, we filter the MCG [3] segmentation proposals such that only those who have a high overlap with the ground-truth bounding boxes are retained. The PRM [42] segmentation seeds are also pruned such that they are contained within each of the bounding box annotations.

6.3 Comparison with other methods

We compare our proposed method with other state-of-the-art weakly supervised instance segmentation methods. The mean average precision (mAP) over different IoU thresholds are shown in table 1. Compared with the other methods, our proposed framework achieves state-of-the-art performance for both image-level and the bounding box labels. We also study the effect of using a different conditional network architecture based on ResNet-50 and ResNet-101. This is shown in the table as 'Ours (ResNet-50)' and 'Ours (ResNet 101)' respectively. Our main result employs a U-Net based architecture for the conditional network and is presented by 'Ours' in the table. The implementation details and the details of the alternative architecture are presented in the supplementary. The encoder-decoder architecture of the U-Net allows us to learn better features. As a result, we observe that our method which adopts U-Net architecture for the conditional network consistently outperforms the one which adopts a ResNet based architecture. In table 1, observe that our approach performs particularly

Table 1. Evaluation of instance segmentation results from different methods with varying level of supervision on Pascal VOC 2012 *val* set. The terms $\mathcal{F}$, $\mathcal{B}$, and $\mathcal{I}$ denotes a fully supervised approach, methods that uses the bounding box labels, and methods that uses the image-level labels respectively. Our prediction network results when using a ResNet based conditional network is presented as 'Ours (ResNet-*) and the results of the prediction network using a U-Net based conditional network is presented as 'Ours'.

Method	Supervision	Backbone	$\text{mAP}^r_{0.25}$	$\text{mAP}^r_{0.50}$	$\text{mAP}^r_{0.70}$	$\text{mAP}^r_{0.75}$
Mask R-CNN [17]	$\mathcal{F}$	R-101	76.7	67.9	52.5	44.9
PRN [42]	$\mathcal{I}$	R-50	44.3	26.8	-	9.0
IAM [43]	$\mathcal{I}$	R-50	45.9	28.8	-	11.9
OCIS [10]	$\mathcal{I}$	R-50	48.5	30.2	-	14.4
Label-PEnet [13]	$\mathcal{I}$	R-50	49.1	30.2	-	12.9
WISE [23]	$\mathcal{I}$	R-50	49.2	41.7	-	23.7
IRN [1]	$\mathcal{I}$	R-50	-	46.7	-	23.5
Ours (ResNet-50)	$\mathcal{I}$	R-50	59.1	49.7	29.2	27.1
Ours	$\mathcal{I}$	R-50	**59.7**	**50.9**	**30.2**	**28.5**
SDI [20]	$\mathcal{B}$	R-101	-	44.8	-	17.8
BBTP [18]	$\mathcal{B}$	R-101	**75.0**	**58.9**	30.4	21.6
Ours (ResNet-101)	$\mathcal{B}$	R-101	73.1	57.7	33.5	31.2
Ours	$\mathcal{B}$	R-101	73.8	58.2	**34.3**	**32.1**

Table 2. Evaluation of the instance segmentation results for the various ablative settings of the conditional distribution on Pascal VOC 2012 data set

$\text{mAP}^r_{0.25}$			$\text{mAP}^r_{0.50}$			$\text{mAP}^r_{0.75}$		
U	U+P	U+P+H	U	U+P	U+P+H	U	U+P	U+P+H
57.9	59.1	59.7	47.6	49.9	50.9	23.1	26.9	28.5

well for the higher IoU thresholds ($\text{mAP}^r_{0.70}$ and $\text{mAP}^r_{0.75}$) for both the image-level and the bounding-box labels. This demonstrates that our model can predict the instance segments most accurately by respecting the object boundaries. The per-class quantitative and qualitative results for our method is presented in the supplementary material.

6.4 Ablation Experiments

Effect of the unary, the pairwise and the higher order terms We study the effect of the conditional distributions unary, pairwise and the higher order terms have on the final output in table 2. We use the terms U, U+P, and U+P+H to denote the settings where only the unary term is present, both the unary and the pairwise terms are present and all three terms are present in the conditional distribution. We see that unary term alone performs poorly across the different IoU thresholds. We argue that this is because of the bias of the unary term for segmenting only the most discriminative regions. The pairwise term helps allay this problem and we observe a significant improvement in the results. This is specially noticeable for higher IoU thresholds that require more accurate seg-

Table 3. Evaluation of the instance segmentation results for the various ablative settings of the loss function's diversity coefficient terms on Pascal VOC 2012 data set

Method mAP^r_k	Pr_p, Pr_c (proposed)	PW_p, Pr_c	Pr_p, PW_c	PW_p, PW_c
$\text{mAP}^r_{0.25}$	59.7	59.5	57.3	57.2
$\text{mAP}^r_{0.50}$	50.9	50.3	46.9	46.6
$\text{mAP}^r_{0.75}$	28.5	27.7	23.4	23.0

mentation. The higher order term helps in improving the accuracy by ensuring that correct samples are generated by the conditional distribution.

Effect of the probabilistic learning objective To understand the effect of explicitly modeling the two distributions (Pr_p and Pr_c), we compare our approach with their corresponding pointwise network. In order to sample a single output from our conditional network, we remove the self-diversity coefficient term and feed a zero noise vector (denoted by PW_c). For a pointwise prediction network, we remove its self-diversity coefficient. The prediction network still outputs the probability of each proposal belonging to a class. However, by removing the self-diversity coefficient term, we encourage it to output a peakier distribution (denoted by PW_p). Table 3 shows that both the diversity coefficient term is important for maximum accuracy. We also note that modeling uncertainty over the pseudo label generation model by including the self-diversity in the conditional network is relatively more important. The self-diversity coefficient in the conditional network enforces it to sample a diverse set of outputs which helps in dealing with the difficult cases and in avoiding overfitting during training.

7 Conclusion

We present a novel framework for weakly supervised instance segmentation. Our framework efficiently models the complex non-factorizable, annotation consistent and boundary aware conditional distribution that allows us to generate accurate pseudo segmentation labels. Furthermore, our framework provides a joint probabilistic learning objective for training the prediction and the conditional distributions and can be easily extendable to different weakly supervised labels such as image-level and bounding box annotations. Extensive experiments on the benchmark Pascal VOC 2012 data set has shown that our probabilistic framework successfully transfers the information present in the image-level annotations for the task of instance segmentation achieving state-of-the-art result for both image-level and bounding box annotations.

Acknowledgements

This work is partly supported by DST through the IMPRINT program. Aditya Arun is supported by Visvesvaraya Ph.D. fellowship.

References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: CVPR (2019)
2. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: CVPR (2018)
3. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014)
4. Arun, A., Jawahar, C.V., Kumar, M.P.: Learning human poses from actions. In: BMVC (2018)
5. Arun, A., Jawahar, C., Kumar, M.P.: Dissimilarity coefficient based weakly supervised object detection. In: CVPR (2019)
6. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What's the point: Semantic segmentation with point supervision. In: ECCV (2016)
7. Bouchacourt, D.: Task-Oriented Learning of Structured Probability Distributions. Ph.D. thesis, University of Oxford (2017)
8. Bouchacourt, D., Kumar, M.P., Nowozin, S.: DISCO Nets: Dissimilarity coefficients networks. In: NIPS (2016)
9. Chen, L.C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: Masklab: Instance segmentation by refining object detection with semantic and direction features. In: CVPR (2018)
10. Cholakkal, H., Sun, G., Khan, F.S., Shao, L.: Object counting and instance segmentation with image-level supervision. In: CVPR (2019)
11. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: CVPR (2015)
12. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
13. Ge, W., Guo, S., Huang, W., Scott, M.R.: Label-PEnet: Sequential label propagation and enhancement networks for weakly supervised instance segmentation. In: ICCV (2019)
14. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV (2011)
15. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV (2014)
16. Hazan, T., Keshet, J., McAllester, D.A.: Direct loss minimization for structured prediction. In: NIPS (2010)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
18. Hsu, C.C., Hsu, K.J., Tsai, C.C., Lin, Y.Y., Chuang, Y.Y.: Weakly supervised instance segmentation using the bounding box tightness prior. In: NeurIPS (2019)
19. Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: CVPR (2018)
20. Khoreva, A., Benenson, R., Hosang, J., Hein, M., Schiele, B.: Simple does it: Weakly supervised instance and semantic segmentation. In: CVPR (2017)
21. Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J.R., Maier-Hein, K., Eslami, S.A., Rezende, D.J., Ronneberger, O.: A probabilistic u-net for segmentation of ambiguous images. In: NIPS (2018)
22. Kumar, M.P., Packer, B., Koller, D.: Modeling latent variable uncertainty for loss-based learning. In: ICML (2012)
23. Laradji, I.H., Vazquez, D., Schmidt, M.: Where are the masks: Instance segmentation with image-level supervision. BMVC (2019)

24. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: CVPR (2017)
25. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: CVPR (2016)
26. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018)
27. Lorberbom, G., Gane, A., Jaakkola, T., Hazan, T.: Direct optimization through argmax for discrete variational auto-encoder. In: NeurIPS (2019)
28. Novotny, D., Albanie, S., Larlus, D., Vedaldi, A.: Semi-convolutional operators for instance segmentation. In: ECCV (2018)
29. Oh, S.J., Benenson, R., Khoreva, A., Akata, Z., Fritz, M., Schiele, B.: Exploiting saliency for object segmentation from image level labels. In: CVPR (2017)
30. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: CVPR (2015)
31. Papandreou, G., Chen, L.C., Murphy, K., Yuille, A.: Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. In: ICCV (2015)
32. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: CVPR (2015)
33. Rao, C.R.: Diversity and dissimilarity coefficients: a unified approach. Theoretical population biology (1982)
34. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
35. 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: CVPR (2017)
36. Song, Y., Schwing, A., Urtasun, R., et al.: Training deep neural networks via direct loss minimization. In: ICML (2016)
37. Tang, P., Wang, X., Bai, X., Liu, W.: Multiple instance detection network with online instance classifier refinement. In: CVPR (2017)
38. Tang, P., Wang, X., Wang, A., Yan, Y., Liu, W., Huang, J., Yuille, A.: Weakly supervised region proposal network and object detection. In: ECCV (2018)
39. Vernaza, P., Chandraker, M.: Learning random-walk label propagation for weakly-supervised semantic segmentation. In: CVPR (2017)
40. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: CVPR (2017)
41. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016)
42. Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., Jiao, J.: Weakly supervised instance segmentation using class peak response. In: CVPR (2018)
43. Zhu, Y., Zhou, Y., Xu, H., Ye, Q., Doermann, D., Jiao, J.: Learning instance activation maps for weakly supervised instance segmentation. In: CVPR (2019)