Leveraging Instance-, Image- and Dataset-Level Information for Weakly Supervised Instance Segmentation

Yun Liu*, Yu-Huan Wu*, Peisong Wen, Yujun Shi, Yu Qiu, and Ming-Ming Cheng

Abstract—Weakly supervised semantic instance segmentation with only image-level supervision, instead of relying on expensive pixel-wise masks or bounding box annotations, is an important problem to alleviate the data-hungry nature of deep learning. In this paper, we tackle this challenging problem by aggregating the image-level information of all training images into a large knowledge graph and exploiting semantic relationships from this graph. Specifically, our effort starts with some generic segment-based object proposals (SOP) without category priors. We propose a multiple instance learning (MIL) framework, which can be trained in an end-to-end manner using training images with image-level labels. For each proposal, this MIL framework can simultaneously compute probability distributions and category-aware semantic features, with which we can formulate a large undirected graph. The category of background is also included in this graph to remove the massive noisy object proposals. An optimal multi-way cut of this graph can thus assign a reliable category label to each proposal. The denoised SOP with assigned category labels can be viewed as pseudo instance segmentation of training images, which are used to train fully supervised models. The proposed approach achieves state-of-the-art performance for both weakly supervised instance segmentation and semantic segmentation. The code is available at https://github.com/yun-liu/LIID.

Index Terms—Weakly supervised learning, instance segmentation, semantic segmentation, multiple instance learning, multi-way cut.

1 Introduction

Instance-aware semantic segmentation (instance segmentation for short) focuses on simultaneously detecting and segmenting all object instances in an image. It is one of the most important tasks in computer vision due to its great academic and industrial values. Recent rapid progress on instance segmentation has been driven by powerful baseline systems, such as Fast/Faster/Mask R-CNN [1]–[3] and Fully Convolutional Networks (FCNs) [4]. However, the performance of these deep models heavily relies on a large amount of training data with expensive pixel-wise labeling. Annotating such training data has been a particular bottleneck on the way of applying instance segmentation to real-world applications, where labeling each pixel for a large number of images is particularly time-consuming. For example, densely annotating a single image in the Cityscapes dataset needs “more than 1.5h on average” [5].

To alleviate the demand for expensive pixel-wise annotations, some studies relax the supervision with bounding boxes [6]–[9], where the training data can be just the data used for object detection. Although annotating bounding boxes is cheaper than annotating pixels, weakly supervised object detection is actually a well-studied research field [10]–[12] owing to the labor-intensive bounding box labeling. Our work in this paper follows [13]–[17] to further relax the supervision, *i.e.*, **using only image-level supervision to perform weakly supervised instance segmentation**. Thanks to the low annotation cost of image-level labels, approaches in this category will benefit many real-world applications.

In weakly supervised instance segmentation, one of the main challenges is to assign the image keyword to each semantic instance, *e.g.*, object proposals [18]. Zhou *et al.* [13] attempted to tackle this challenging problem by computing class peak responses in the *class activation maps* (CAM) [19] obtained from image classifiers [20], [21]. These peak responses can be used to query category-agnostic object proposals for the prediction of instance masks. Similar to [13], many other weakly supervised instance segmentation methods [14]–[17] and weakly supervised semantic segmentation methods [22]–[30] also heavily depend on CAM for object recognition. However, CAM tends to focus on the small discriminative region of a target object, and it is also difficult for CAM to correctly localize objects from complex scenarios that contain small objects, multiple objects, and the complex background. Although various techniques [27], [31]–[33] have been introduced to improve CAM, the natural limitations of CAM still hinder the development of weakly supervised learning [16].

Motivated by the above observations, we propose a novel method that can overcome these limitations. Unlike the previous CAM-based weakly supervised segmentation methods that directly use CAM or the improved versions of CAM for object recognition [13]–[17], [22]–[31], our method learns the semantic information of each image in the training process by using CAM as **one of the supervision sources** in a *multiple instance learning* (MIL) framework. Therefore, CAM helps the training of our system by providing approximate coarse information, but the performance of our system does not completely rely on CAM because we have other designs to ensure the training of the MIL framework, as proven in the experiments. Moreover, we propose to integrate the useful information of all training images into a large knowledge graph and explore the information in this graph to bridge the image-level keywords and corresponding semantic

- *Y. Liu, Y.H. Wu, P. Wen, Y. Shi, and M.M. Cheng are with TKLNDST, College of Computer Science, Nankai University, Tianjin 300350, China.*
- *Y. Qiu is with College of Artificial Intelligence, Nankai University, Tianjin 300350, China.*
- *Y. Liu and Y.H. Wu have made equal contribution to this paper. M.M. Cheng is the corresponding author (cmm@nankai.edu.cn).*

instances. In this way, our method takes into consideration not only the intrinsic properties of each image but also the overall data distribution of the training database, so that it breaks the limitations of CAM on weakly supervised segmentation.

Specifically, our effort starts with some generic *segment-based object proposals* (SOP) such as selective search [34], LPO [35], and MCG [18]. Since these methods are category-agnostic, they do not rely on any semantic labels. Therefore, our system can generalize to any category with only image-level information. Given an image with corresponding image tags and object proposals, we aim at assigning correct category labels to each proposal and filtering out noisy proposals. To achieve this goal, we build an MIL framework for image classification using image tags as supervision. In this framework, if a proposal contains an object of a specific category, our model will learn to make this proposal contribute more to the final classification probability of the corresponding category. If a proposal does not contain any objects from the target categories, our model is expected to ignore it. At last, this MIL framework can simultaneously assign a probability distribution across all target categories and compute a semantic feature vector for each proposal.

By viewing all proposals in the training database as *non-terminal* nodes and all target categories (including background) as *terminal* nodes, we can construct an undirected graph using the produced probability distributions and semantic feature vectors. This large graph can well represent the properties of each proposal and the relationships among all proposals in the training database. The optimal *multi-way cut* of this undirected graph can associate each proposal with a proper category label. After removing noisy proposals, the remaining proposals with automatically assigned labels can serve as pseudo instance segmentation to be used for training fully supervised models. Since our method **L**everages **I**nstance-, **I**mage- and **D**ataset-level information, we call it **LIID**.

We perform extensive experiments on PASCAL VOC2012 [36] and MS-COCO [37] datasets to evaluate the proposed method with various experimental settings. The evaluation results demonstrate that the proposed approach achieves state-of-the-art performance for both weakly supervised instance segmentation and semantic segmentation. To sum up, the main contribution of this paper is threefold:

- We propose a novel multiple instance learning (MIL) framework to simultaneously compute the probability distribution and extract the semantic feature vector for each proposal.
- We construct a large undirected graph using the produced probability distributions and semantic features, in which the target categories (including background) are viewed as terminal nodes. We further propose an efficient approximate optimization algorithm to perform a multi-way cut on this graph to obtain pseudo instance segmentation.
- Extensive experiments demonstrate that the proposed LIID consistently achieves state-of-the-art performance for both weakly supervised instance segmentation and semantic segmentation.

2 Related Work

Instance segmentation. Instance segmentation is an active research area for scene understanding. Longstanding efforts have focused on fully supervised settings. Most of the top-performing methods are based on object detection networks to output a ranked list of segments rather than bounding boxes [3], [38]–[41]. Among these methods, Mask R-CNN [3] and its derivatives [40], [41] have dominated the state-of-the-art. Some researchers also contributed approaches based on initial semantic segmentation networks to generate instance masks [42]–[44]. Although fully supervised methods can achieve high accuracy, they usually require large-scale training data with expensive pixel-wise annotations, which makes them inconvenient to be applied to real-world applications.

Weakly supervised instance segmentation. For weakly supervised instance segmentation, Khoreva *et al.* [6] firstly proposed to use labeled bounding boxes as the supervision rather than pixel-wise masks. Specifically, they used a modified version of GrabCut [45] to estimate an object segment from its bounding box. The obtained object segments are further refined by the SOP generated by MCG [18]. Li *et al.* [7] extended [6] by iteratively refining the proxy ground truth. They used the outputs of the network on the training set as the new proxy ground truth. Hsu *et al.* [8] formulated this problem as an MIL task by generating positive and negative bags based on the sweeping lines of each bounding box. This MIL formulation can be integrated into an end-to-end network to learn an instance segmentation model. Hu *et al.* [9] introduced a semi-supervised instance segmentation model using transfer learning, in which some classes have pixel-wise annotations while the other classes only have bounding boxes.

Zhou *et al.* [13] initiated the challenging problem of training neural networks with image-level weak supervision for instance segmentation. They introduced a quite novel concept of *class peak responses* that reflect strong visual cues residing inside each semantic instance. The learned class peak response maps can be utilized to query and rank the SOP. Their method significantly outperforms various baselines. Following [13], Zhu *et al.* [14] presented an instance extent filling approach to collect pseudo supervision from noisy SOP selectively. The pseudo supervision is used to learn a differentiable filling module that predicts a class-agnostic activation map for each instance. Cholakkal *et al.* [15] introduced an image-level supervised approach for both ordinary object counting and image-level supervised instance segmentation by constructing an object category density map. Ahn *et al.* [16] propagated the CAM of an image classification model to discover the entire instance areas that are regarded as proxy ground truth to train a fully supervised model. Ge *et al.* [17] proposed Label-PEnet to progressively transform image-level labels to pixel-wise labels by alternatively training four sequentially cascaded modules including multi-label classification, object detection, instance refinement, and instance segmentation. We follow [13]–[17] to only use image-level supervision for instance segmentation. Instead of using CAM-based models, we attempt to use both the intrinsic properties of each proposal and the overall data distribution of the whole training database to determine the semantic category for each proposal.

Weakly supervised semantic segmentation. Semantic segmentation is highly related to instance segmentation, in that semantic segmentation only recognizes the category of each pixel without differentiating different object instances. Weakly supervised instance segmentation can be applied to semantic segmentation by simply eliminating the discrimination of object instances. We also provide evaluation results for semantic segmentation in this paper, so we broadly review the related work of weakly supervised semantic segmentation here.

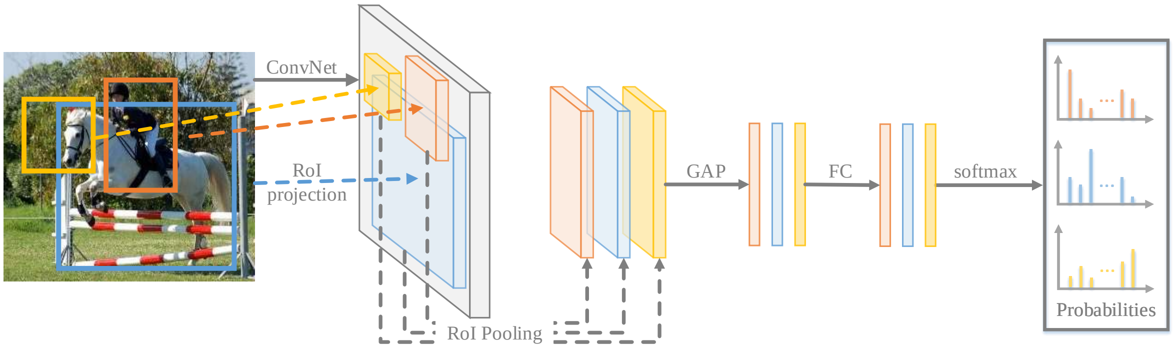


Fig. 1. Our proposed network architecture for MIL-based multi-label image classification. This network is designed to simultaneously compute probability distributions and extract semantic features for each input proposal.

Recent methods have achieved good performance using annotations that provide location information such as points [46], scribbles [47], or bounding boxes [48]. Weakly supervised semantic segmentation with image-level annotations still remains a challenging problem. Given image-level annotations, CAM [19] is a good starting point to discover coarse object locations. However, CAM [19] tends to focus on the small discriminative region of a target object, which makes it improper to train semantic segmentation networks. Most of the current approaches aim at improving CAM to extract complete objects using only image tags. These methods either adopt image hiding and erasure to prevent a classifier from focusing exclusively on the discriminative parts of objects [23], [31], [49], or expand the CAM using feature-level processing [27], [29], [50]–[53] and region growing [30], [32], [54], [55] techniques. These methods often use various auxiliary cues such as saliency maps [56]–[60], edges [61]–[63], and object proposals [18], [64] to improve accuracy [22], [23], [29], [30], [55], [65]–[68].

Besides the above approaches, Saleh *et al.* [69] and Pinheiro *et al.* [70] proposed MIL methods for weakly supervised semantic segmentation, but their methods are limited to pixel-wise classification and cannot learn instance-aware information. In contrast, the introduced MIL framework in this paper focuses on learning instance-aware information that is used to discriminate object instances. More recently, there is a graph-based weakly supervised semantic segmentation model proposed by Fan *et al.* [28] that is relevant to our model. Although we focus on a different task from [28], we analyze the differences between our method and [28], which can be summarized as follows:

1) Our method is different from [28] in proposal information extraction. For each object proposal, Fan *et al.* [28] directly used CAM [19] to estimate the probability distribution and then adopted a pre-trained ImageNet [71] model to extract semantic features. In contrast, we propose an end-to-end MIL framework to simultaneously learn the probability distribution and semantic features from a given image.
2) Our method is different from [28] in graph modeling. Fan *et al.* [28] formulated the category label assignment as an ordinary graph partitioning problem by viewing all proposals as graph nodes, and the initial probabilities were only used as a balanced term in the optimization formula. In contrast, we build an undirected graph by viewing all proposals as ordinary graph nodes and target category tags as *terminals*. The probability distributions and semantic features of proposals are used to calculate weights for different types of edges. We formulate the category assignment as a multi-way cut problem and then propose an efficient approximate optimization algorithm to solve this problem.

Although both our method and [28] use a graph to leverage the dataset-level information, our proposed method is more reasonable and intuitive in model training, probability prediction, feature extraction, graph construction, and graph partitioning, which leads to the significantly better performance of our method as demonstrated in the experiments.

3 Problem Formulation

Suppose we have a training image set $\mathcal{I} = \{I_1, I_2, \cdots, I_N\}$ with corresponding image-level tags $\mathcal{Y} = \{Y_1, Y_2, \cdots, Y_N\}$, where N is the number of training images. Let $\mathcal{K} = \{0, 1, 2, \cdots, K\}$ be the set of categories, in which 0 represents background and K is the number of target semantic categories. Under a mild assumption that every image has background regions, we have $0 \in Y_i$ and $Y_i \subseteq \mathcal{K}$ $(i = 1, 2, \cdots, N)$. For convenience, we define $\mathcal{K}' = \{1, 2, \cdots, K\}$ that excludes the background category[1]. We can input images $\mathcal{I}$ into any bottom-up proposal generation methods [18], [34], [35], [72]–[75] (*i.e.*, MCG [18] here) to obtain *generic* SOP $\mathcal{S} = \{S_1, S_2, \cdots, S_N\}$. Suppose $S_i = \{s_i^1, s_i^2, \cdots, s_i^{|S_i|}\}$ $(i = 1, 2, \cdots, N)$, and s_i^j $(j = 1, 2, \cdots, |S_i|)$ is a binary segment mask. Note that $|\cdot|$ represents the number of elements in a set. We can easily obtain the corresponding bounding boxes of these SOP, which can be denoted as $\mathcal{B} = \{B_1, B_2, \cdots, B_N\}$ and $B_i = \{b_i^1, b_i^2, \cdots, b_i^{|S_i|}\}$.

Each of these category-agnostic SOP may contain no semantic objects, multiple or one semantic object. The proposals containing no complete semantic objects and multiple objects are deemed as *noisy proposals* in this paper. In order to perform instance segmentation, our primary objective in this paper is to remove noisy proposals and assign a correct category label to the proposals tightly containing one complete object. Therefore, our objective can be formulated as

$$F(s_i^j) = \begin{cases} 0 & \text{if } s_i^j \text{ is a noisy proposal} \\ k' & \text{if } s_i^j \text{ belongs to category } k' \end{cases}, \tag{1}$$

where $k' \in \mathcal{K}'$ and s_i^j denotes the j^{th} proposal in the i^{th} image. The SOP s_i^j with $F(s_i^j) > 0$ will serve as our pseudo instance segmentation. An overview of the proposed solution to compute $F(s_i^j)$ is illustrated in Fig. 2.

[1] For clarity, we use $k \in \mathcal{K}$ and $k' \in \mathcal{K}'$ to represent a category including background and excluding background, respectively.

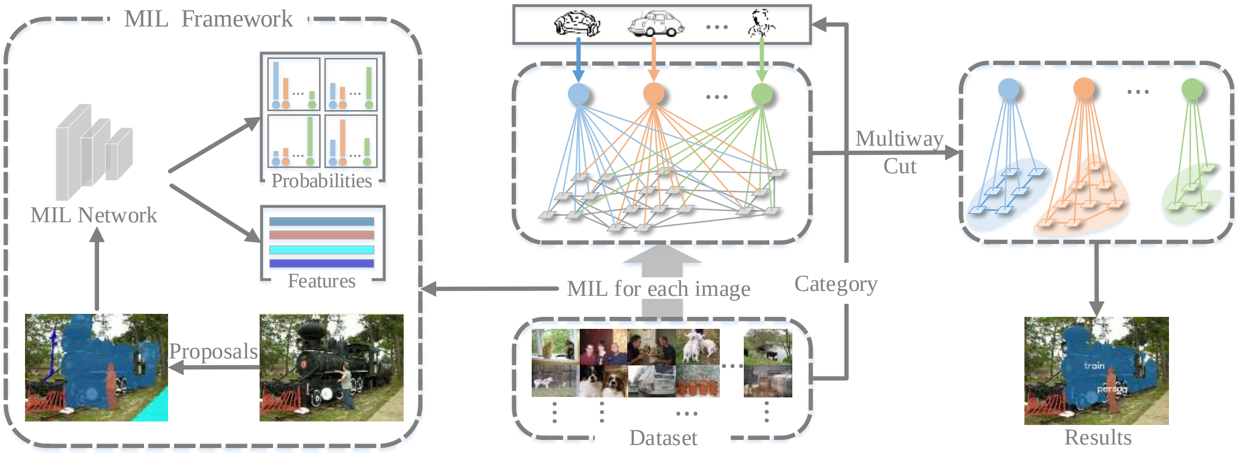


Fig. 2. An overview of the proposed method. The training images with image-level labels are used to train our MIL-based multi-label image classification network, as in Section 4. All training images, together with corresponding proposals, are fed into the MIL network to calculate the probability distributions and semantic features. A large knowledge graph is then constructed using all training images. The pseudo instance segmentation can be obtained using an improved multi-way cut algorithm.

4 Proposal-Based MIL Framework

Given images with image-level labels, previous studies [13]–[15] usually train multi-label image classifiers that are used to compute CAM for object localization. Then, they combine CAM and SOP to produce pseudo segmentation. Due to the natural limitations of CAM, as discussed above, the training data are not fully used. In contrast, we consider to incorporate SOP into the training process, and each SOP is expected to learn useful information. Given an input image I_i with image tags Y_i, we would know the corresponding proposals S_i/B_i contain categories Y_i, but each proposal individually corresponds to which category is unknown. This is actually a case of *multiple instance learning* (MIL). Therefore, we build an MIL framework that takes images and generic object proposals as inputs and views image-level tags as the supervision. Through the training, the model is expected to learn to produce a class probability distribution and a semantic feature vector for each proposal, which will be used for the subsequent multi-way cut. In this section, we first introduce the proposed network architecture and then present several loss functions for the proposal-based MIL framework.

4.1 Network Architecture

In this part, we introduce the designed network for MIL-based multi-label image classification. The proposed network architecture is shown in Fig. 1. Here, the category-agnostic proposals are generated by the MCG algorithm [18]. An input image I_i first passes through the backbone network, *i.e.*, ResNet50 [21] here. We perform ROI pooling [1] on the produced feature maps using the bounding boxes B_i of the SOP S_i. A global average pooling (GAP) layer follows this ROI pooling to convert the feature map into a 2048-dimensional feature vector $\mathbf{f}_i^j$ $(j = 1, 2, \cdots, |S_i|)$ for each proposal. Then, we connect a fully connected layer with $(K + 1)$ outputs $\mathbf{a}_i^j$ $(|\mathbf{a}_i^j| = K + 1)$ representing the produced scores for K target categories and the background. Finally, let $(\mathbf{p}_i^j)_k$ be the probability of class k obtained after a *softmax* layer, so we have

$$(\mathbf{p}_i^j)_k = \frac{exp((\mathbf{a}_i^j)_k)}{\sum_{m=0}^{K} exp((\mathbf{a}_i^j)_m)}, \tag{2}$$

where $k \in \mathcal{K}$. With such a designing pattern, we can calculate a feature vector $\mathbf{f}_i^j$ and a probability distribution $\mathbf{p}_i^j$ for each proposal. Through exposing proper loss functions, the proposed network is expected to learn category-aware information for each proposal.

4.2 Proposal-Based MIL Loss

For the training of the MIL framework, we propose several loss functions to simultaneously infer the probability distributions and extract the semantic features for object proposals. Considering that the proposal labels are unknown, we design a CAM-based loss function to approximate a pseudo label for each proposal, and we also design an MIL-based image classification loss function to compute the aggregated probability for each image so that we can adopt image labels for supervision. These loss functions are imposed to supervise the probability distribution $\mathbf{p}_i^j$ for the convergence of the network. Besides, we design an MIL-based center loss function to concentrate the semantic feature vectors $\mathbf{f}_i^j$ with the same category, so that the proposals belonging to the same category will have small feature distance with respect to $\mathbf{f}_i^j$.

4.2.1 CAM-Based Loss

Instead of relying on CAM to localize objects like previous approaches [13]–[15], we apply CAM as one of the supervision sources for training by approximating a pseudo label for each proposal. Specifically, using standard ResNet50 [21] network with K independent cross-entropy loss functions, we can train a multi-label image classification model. Then, we can use the well-known CAM algorithm [19] to compute the CAM $A_i^{k'}$ $(k' \in \mathcal{K}')$ for an image I_i. $A_i^{k'}$ is normalized into the range of $[0, 1]$. Let $\widetilde{y}_i^j$ denote the approximate category label for the j^{th} proposal $(j = 1, 2, \cdots, |S_i|)$. Suppose we have $(R_i^j)_{k'} = mean(A_i^{k'}[b_i^j]) + max(A_i^{k'}[b_i^j])$ and $(R_i^j)_{k'} \in [0, 2]$, where $A_i^{k'}[b_i^j]$ means the corresponding region of the proposal b_i^j in $A_i^{k'} \in [0, 1]$. We approximate $\widetilde{y}_i^j$ using the computed CAM as

$$\widetilde{y}_i^j = \begin{cases} 0 & if\ \forall k', (R_i^j)_{k'} < \eta \\ \underset{k'}{\arg\max}\ (R_i^j)_{k'} & otherwise \end{cases}, \tag{3}$$

where η is a threshold. Therefore, $\widetilde{y}_i^j$ can be viewed as the pseudo label of the j^{th} proposal b_i^j, and $\mathcal{K}\backslash\{\widetilde{y}_i^j\}$ is the category set other than $\widetilde{y}_i^j$. We define the CAM-based loss function as

$$\begin{aligned} L_{\mathit{Att}}^{(i)} = & -\frac{1}{|S_i|}\sum_{j=1}^{|S_i|}\Bigg[log(\mathbf{p}_i^j)_{\widetilde{y}_i^j} \\ & +\frac{1}{K}\sum_{k\in\mathcal{K}\backslash\{\widetilde{y}_i^j\}} log(1-(\mathbf{p}_i^j)_k)\Bigg]. \end{aligned} \tag{4}$$

In this way, the pre-trained multi-label image classification model can help the MIL training through CAM. For the calculation of $(R_i^j)_{k'}$, we use the box-level pooling rather than mask-level pooling in $A_i^{k'}[b_i^j]$, because proposal boxes are more reliable than proposal segmentation masks (*i.e.*, SOP). As shown in related research [18], [34], [35], [72], proposal box generation is much easier than mask generation and thus achieves higher accuracy. It is difficult for bottom-up methods to accurately segment general objects, and the inaccurate masks would be harmful to the MIL training. We will further demonstrate this design in Section 6.2 through ablation studies.

4.2.2 MIL-Based Image Classification Loss

Although the proposal labels are unknown, the aggregation of the learned probability distributions $\mathbf{p}_i^j$ for all proposals in an image can reflect the classification ability of the network. In other words, we cannot directly supervise the probability $\mathbf{p}_i^j$ for each proposal, but we can supervise the overall probability aggregation for each image. Suppose the aggregation score for each class in the image I_i is $(\mathbf{Z}_i)_k$ ($k \in \mathcal{K}$), which can be inferred from $(\mathbf{p}_i^j)_k$. Instead of a simple maximum or average across $(\mathbf{p}_i^j)_k$, we use Log-Sum-Exp (LSE) function [76] to compute a smooth approximation to the maximum value of $(\mathbf{p}_i^j)_k$ ($j = 1, 2, \cdots, |S_i|$), which can be formulated as

$$(\mathbf{Z}_i)_k = \frac{1}{r}log\Bigg[\frac{1}{|S_i|}\sum_{j=1}^{|S_i|} exp(r\ (\mathbf{p}_i^j)_k)\Bigg], \tag{5}$$

where r is a parameter allowing LSE function to behave in a range between the maximum and the average. We empirically set r to 5 in this paper [70]. Compared with a simple maximum, LSE function can not only approximate the maximum but also take into consideration all elements of $(\mathbf{p}_i^j)_k$. With approximated $(\mathbf{Z}_i)_k$, we define the MIL-based image classification loss function as

$$L_{\mathit{MIL}}^{(i)} = -\frac{1}{|Y_i|}\sum_{k\in Y_i} log((\mathbf{Z}_i)_k) - \frac{1}{|\overline{Y_i}|}\sum_{k\in\overline{Y_i}} log(1-(\mathbf{Z}_i)_k), \tag{6}$$

where $\overline{Y_i}$ is the complementary set of Y_i. It is consistent with the intuition that the present categories should appear in the proposals and the proposals having high probabilities for absent categories should be penalized.

As described in Section 3, we assume that every image has background regions, *i.e.*, $0 \in Y_i$ $(i = 1, 2, \cdots, N)$. This mild assumption is essential for Equ. (6). On the one hand, the proposals generated by bottom-up algorithms usually contain many noisy proposals that fall outside the target object categories, covering other categories of objects or even non-object regions, so we have to include the background class for each image to ensure the network training. On the other hand, our objective requires recognizing and filtering out these noisy proposals as formulated in Equ. (1), so we have to incorporate the background class in training to learn proper information for noisy proposals and thus filter out them using the techniques in Section 5.

4.2.3 MIL-Based Center Loss

The next loss function is designed for the semantic feature extraction. The training is expected to maximize the similarity score of semantic features of proposals with the same category and minimize the similarity score of proposals with different categories. To this end, we introduce an MIL-based center loss function to concentrate the semantic features with similar semantic meanings:

$$\begin{aligned} \widehat{y}_i^j &= \arg\max_k\ (\mathbf{p}_i^j)_k, \\ L_{\mathit{Cent}}^{(i)} &= \frac{1}{|S_i|}\sum_{j=1}^{|S_i|}\Bigg[1 - \frac{\mathbf{f}_i^j\cdot\mathbf{c}_{\widehat{y}_i^j}}{\|\mathbf{f}_i^j\|_2\|\mathbf{c}_{\widehat{y}_i^j}\|_2}\Bigg], \end{aligned} \tag{7}$$

where $\mathbf{c}_k$ is the learned center for the k^{th} category of input samples, and $\|\cdot\|_2$ is the ℓ^2-norm for a vector. This loss measures the *cosine* similarity between a feature vector $\mathbf{f}_i^j$ and the learned category center $\mathbf{c}_k$. In every training iteration, $\mathbf{c}_k$ is updated with respect to the semantic feature vector $\mathbf{f}_i^j$ as

$$\begin{aligned} \mathbf{c}_{\widehat{y}_i^j}^{new} &= \mathbf{c}_{\widehat{y}_i^j}^{old} + \theta\cdot(\mathbf{f}_i^j - \mathbf{c}_{\widehat{y}_i^j}^{old}), \\ & \mathit{for}\ j = 1, 2, \cdots, |S_i|, \end{aligned} \tag{8}$$

where θ is the update rate. Therefore, the similarity distances between proposal pairs can thus be computed through their learned feature vectors $\mathbf{f}_i^j$.

With above definitions, the overall loss function for the MIL-based multi-label image classification problem can be formulated by

$$L^{(i)} = \alpha L_{\mathit{Att}}^{(i)} + \beta L_{\mathit{MIL}}^{(i)} + \gamma L_{\mathit{Cent}}^{(i)}. \tag{9}$$

In practice, we empirically set α, β and γ to 0.5, 0.5 and 0.1, respectively. Our proposed $L_{\mathit{Att}}^{(i)}$ can leverage the pre-trained multi-label image classification model to help the MIL training, and $L_{\mathit{MIL}}^{(i)}$ is naturally suitable for MIL training here. Hence both the coefficients of $L_{\mathit{Att}}^{(i)}$ and $L_{\mathit{MIL}}^{(i)}$ are set to 0.5. For loss $L_{\mathit{Cent}}^{(i)}$, it is designed to minimize intra-class variations, which is irrelevant to image classification, so we set a small coefficient of 0.1 to avoid its effect on the classification results.

5 Label Assignment via Multi-way Cut

Intuitively, the prediction in consideration of the data distribution of all training samples would be better than the prediction in consideration of only a single sample. This is because a single sample may have a bias or random error, but the overall data distribution is more reliable. Although the training process of the MIL framework has utilized all training data, this is only an indirect use of the overall data distribution. Here, we consider a direct way. Specifically, we exploit a large knowledge graph that comprises proposals in all training images for a global solution.

5.1 Review of the Multi-way Cut Problem

Before introducing our method for SOP label assignment, we provide a brief review of the multi-way cut problem in this part. Let us first describe the conventional graph cut. Suppose we have a connected and undirected graph $G = (V, E)$ where the node set is V and the edge set is E. The weight function of this graph G can be formulated as $w : E \rightarrow \mathbb{R}^+$, in which $\mathbb{R}^+$ denotes the set of non-negative real numbers. The commutative property holds for any pair of nodes $u \in V$, $v \in V$, *i.e.*, $w(u, v) = w(v, u)$. A *graph cut* is defined by a partition of V into disjoint subsets V_1 and V_2, resulting in an edge subset $E' \subseteq E$ that have one vertex in V_1 and the other vertex in V_2. Hence the edge subset E' can be used to represent this graph cut. The *cost* of this cut is defined as $\sum_{(u,v)\in E'} w(u, v)$. The typical *minimum cut* problem is to find the minimum-cost cut that separates two given nodes $\dot{u}$ and $\dot{v}$ (we call these nodes *terminals*), *i.e.*, $\dot{u} \in V_1$ and $\dot{v} \in V_2$. This minimum cut problem is the dual of the *maximum flow* problem, which can be solved in polynomial time.

The *multi-way cut* problem is one generalization of the minimum cut, which is also known as the *multi-terminal cut* problem [77]–[79]. Given a set of *terminals* $\hat{E} \subseteq E$, the multi-way cut is to find the minimum-cost subset of edges $E' \subseteq E$ whose removal separates each pair of terminals. In other words, no connected component of the graph $(V, E - E')$ contains two terminals from $\hat{E}$. When there are only two terminals, *i.e.*, $|\hat{E}| = 2$, this problem is equivalent to the above minimum cut problem that is solvable in polynomial time. When there are three or more terminals, *i.e.*, $|\hat{E}| \geq 3$, the multi-way cut becomes NP-hard and requires approximation algorithms to solve it. In the following subsections, we formulate the SOP label assignment as a multi-way cut problem and propose a simple solution to solve this complex problem.

5.2 Knowledge Graph Construction

In order to compute $F(s_i^j)$ in Equ. (1), we construct a large knowledge graph that incorporates not only the intrinsic properties of each proposal but also the relationship between different proposals in the whole training database. We use all training images to construct this graph. Exploiting this knowledge graph will assign a reliable category label for each proposal. We formulate the label assignment process as a multi-way cut problem and introduce an effective approximate solution to this problem. The graph cut results for training images are our pseudo instance segmentation that can be used to train fully supervised models.

As in Section 5.1, we construct a connected and undirected graph $G = (V, E)$. Specifically, we view all proposals s_i^j ($i = 1, 2, \cdots, N; j = 1, 2, \cdots, |S_i|$) and target categories $\mathcal{K}$ ($\mathcal{K} = \{0, 1, 2, \cdots, K\}$) as graph nodes, so we have $V = \mathcal{K} \cup S_1 \cup S_2 \cup \cdots \cup S_N$. Moreover, let $\mathcal{K}$ be the set of *terminals*, *i.e.*, $\hat{E} = \mathcal{K}$. Each edge $(u, v) \in E$ has a non-negative weight $w(u, v)$

$$= \begin{cases} (\mathbf{p}_i^j)_k & if\ \exists i, j\ \ u = s_i^j; v \in \mathcal{K} \\ 0 & if\ u \in \mathcal{K}, v \in \mathcal{K} \\ \delta \cdot \dfrac{|\mathbf{f}_i^j \cdot \mathbf{f}_{i'}^{j'}|}{\|\mathbf{f}_i^j\|_2 \|\mathbf{f}_{i'}^{j'}\|_2} & if\ \exists i, j\ \ u = s_i^j;\ \exists i', j'\ \ v = s_{i'}^{j'} \end{cases}, \tag{10}$$

where δ is a balance factor. Therefore, the edge weight between terminal nodes is 0, which is the minimum edge weight in the knowledge graph G. The edge weight between a proposal node and a terminal node is just the predicted probability of that proposal falling into the corresponding category. The edge weight between two proposal nodes is the *cosine* similarity of their feature vectors [28], [80], so proposal pairs having similar semantic content will have large *cosine* similarities. In this manner, graph G has the knowledge of the whole training database by incorporating the probability distributions and semantic features of all training images learned in Section 4.

5.3 Multi-way Cut on the Knowledge Graph

Given the knowledge graph $G = (V, E)$ with a set of *terminals* $\mathcal{K}$, our goal is to find the multi-way cut to disconnect each terminal node from the rest terminals. That is to say, our primary objective here is to find an edge subset $E' \subseteq E$ with the minimum cost such that in the new graph $(V, E - E')$, there is no connected path between any pair of terminals. After the multi-way cut, the corresponding nodes of proposals that have similar semantic information will fall into the same component, because the above multi-way cut has maximized the similarity within each component and minimized the similarity across different components by minimizing the cut cost. There is only one terminal $k \in \mathcal{K}$ in each component, and the pseudo category label of a proposal is just the terminal k in its corresponding component. Here, the category $k = 0$ implies the background or noisy proposals, because it falls outside the target object categories.

The commonly used datasets such as PASCAL VOC2012 [36] and MS-COCO [37] usually have $|\mathcal{K}| \geq 3$, which means there are three or more object categories. As discussed in Section 5.1, we require an approximation algorithm to solve the above multi-way cut problem. Let Δ_K denote the K-simplex, so the K-dimensional convex polytope in $\mathbb{R}^{K+1}$ can be expressed as $\{x \in \mathbb{R}^{K+1} | (x \geq 0) \wedge \sum_k x_k = 1\}$. For $k, \dot{k} \in \mathcal{K}$, $e^k \in \mathbb{R}^{K+1}$ denotes the unit vector with $(e^k)_k = 1$ and $(e^k)_{\dot{k}} = 0$ ($\forall k \neq \dot{k}$). According to [79], we can formulate the following optimization function to solve the multi-way cut problem

$$\begin{aligned} \min_x \frac{1}{2} \sum_{(u,v)\in E} w(u, v) \cdot \|x^u - x^v\|_1 \quad s.t. \\ x^u \in \Delta_K, \ \ \forall u \in V; \\ x^k = e^k, \ \ \forall k \in \mathcal{K}, \end{aligned} \tag{11}$$

where $\|\cdot\|_1$ means the ℓ^1-norm. However, directly solving the linear programming in Equ. (11) is unpractical because of the exponential number of constraints [79], especially in our case where the knowledge graph on the whole training database is very large. The CPU memory and runtime required by the direct solution is unpractical to current devices. Specifically, the space complexity of the direct solution is $O(|E||V|^2)$, and the required CPU memory for PASCAL VOC2012 [36] training set is about $10^3 \sim 10^4$ GB, which is much larger than the memory capacity of existing computers.

To address this problem, we connect each node $u \in (V - \mathcal{K})$ to at most three other nodes v ($v \in (V - \mathcal{K})$ and $v \neq u$) with three largest edge weights instead of connecting each node $u \in V$ to all the other nodes. We observe that in the obtained sparse graph, the overall large knowledge graph will be automatically divided into many small **disconnected components**, each of which can be viewed as a sub-graph $G_t = (V_t, E_t)$:

$$\begin{aligned} \cup_t V_t = V, \\ \cup_t E_t = E. \end{aligned} \tag{12}$$

Each sub-graph is independent of each other in the multi-way cut problem. This can be easily proved by decomposing Equ. (11) into many terms, each of which represents the cut cost of a sub-graph. The common graph nodes among these terms are only terminals, which will not affect the final results because these terminals must fall into different components in the end. Therefore, we can process each sub-graph individually to compute its multi-way cut E'_t. To solve this linear programming problem, we first use the simplex method to solve Equ. (11), whose results are further converted to the solution of the multi-way cut using a branch-and-bound method of IBM-CPLEX [81]. The multi-way cut E' of the original large graph can be obtained by

$$\cup_t E'_t = E'. \tag{13}$$

In this way, we can successfully approximate the multi-way cut of a large graph by computing many small graphs. Here, we chose to connect each node with three edges because connecting each node with four edges will lead to too large sub-graphs that are difficult to be solved as discussed above. With the multi-way cut results, we can easily assign $F(s_i^j)$ in Equ. (1) to the category k, if the proposal s_i^j falls into the same subset with terminal k ($k \in \mathcal{K}$). If we have $F(s_i^j) = 0$, the SOP s_i^j would be a noisy proposal and thus be abandoned. For the rest of SOP with $F(s_i^j) \neq 0$, we apply non-maximum suppression (NMS) using the corresponding bounding boxes b_i^j with an intersection-over-union (IoU) threshold of 0.4, as commonly done in the field of object detection [1]–[3]. This NMS operation solves the case where multiple proposals represent a single object. At last, we view the rest of SOP and the corresponding category labels $F(s_i^j)$ as the proxy ground truth for training images, so that we can train a Mask R-CNN model [3] (with the ResNet50 [21] backbone network) for weakly supervised instance segmentation or train a DeepLab model [82] (with the ResNet101 [21] backbone network) for weakly supervised semantic segmentation.

6 Experiments

6.1 Experimental Setup

Datasets. The proposed approach is evaluated on the PASCAL VOC2012 dataset [36] and MS-COCO dataset [37]. Note that only image-level tags are used for training. VOC2012 dataset [36] consists of 20 semantic categories as well as a background category. We follow [13]–[15] to utilize VOC2012 *main trainval* subset, excluding *segmentation val* images, to train our MIL network (with ∼10K images). We evaluate our approach and baseline models using the 1449 *segmentation val* images. For ablation studies, we adopt the VOC2012 *main trainval* subset, excluding *segmentation train+val* images, for training and *segmentation train* set for validation. MS-COCO dataset [37] consists of 80 semantic categories. We follow [28] to train on the standard *trainval* set and evaluate on the *test-dev* set.

Implementation details. In the training, we adopt the bottom-up MCG [18] algorithm to generate 500 SOP per image, from which we select 20/40 proposals for VOC2012/MS-COCO using the simple filtering method in [83]. We implement our MIL-based multi-label image classification model using the PyTorch framework. We apply the SGD optimization algorithm with the learning rate policy of *step*. For both VOC2012 and MS-COCO datasets, the initial learning rate is 5×10^{-4}, which will be divided by 10 after 5 epochs. We run SGD for 10 epochs in total with the mini-batch of one image. The weight decay and momentum are set to 10^{-4} and 0.9, respectively. In the construction of the graph, we follow [28] to compute salient instances [75] as object proposals. The training of Mask R-CNN [3] and DeepLab [82] follows the default settings.

TABLE 1
Evaluation results of different θ (in Equ. (8)) and γ (in Equ. (9)) values on the VOC2012 *segmentation train* set [36]. Each result pair w_1/w_2 denotes the result without (w_1) and with (w_2) the knowledge graph, respectively.

No.	θ	γ	AP$_{50}$	AP$_{75}$	ABO
1	0.01	0.05	30.3/33.8	14.9/16.3	36.7/38.7
2	0.01	0.1	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
3	0.01	0.5	32.4/33.7	15.1/16.1	37.9/39.2
4	0.03	0.1	32.0/33.7	14.9/16.0	38.0/39.3
5	0.03	0.3	31.9/33.7	14.8/16.3	37.6/39.3
6	0.05	0.1	32.3/33.6	15.1/16.3	37.8/39.2
7	0.005	0.5	29.1/32.3	13.6/15.5	36.3/38.5
8	0.05	0.5	31.5/33.2	14.9/16.2	37.7/39.0
9	-	0	31.3/-	15.0/-	37.8/-

TABLE 2
Evaluation results of different α and β values (in Equ. (9)) on the VOC2012 *segmentation train* set [36]. Each result pair w_1/w_2 denotes the result without (w_1) and with (w_2) the knowledge graph, respectively.

No.	α	β	AP$_{50}$	AP$_{75}$	ABO
1	1.0	0.0	28.7/31.8	13.9/15.7	35.5/37.9
2	0.8	0.2	31.5/34.0	14.7/16.6	37.4/39.2
3	0.5	0.5	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
4	0.2	0.8	31.3/32.8	14.8/16.1	36.2/37.6
5	0.0	1.0	18.7/19.3	8.8/9.2	22.9/23.0

TABLE 3
Evaluation of the existence of $mean$ and max in $(R_i^j)_k$ (Equ. (3)) on the VOC2012 *segmentation train* set [36]. Each result pair w_1/w_2 denotes the result without (w_1) and with (w_2) the knowledge graph, respectively.

No.	mean	max	AP$_{50}$	AP$_{75}$	ABO
1	✔	✗	28.7/32.6	13.1/15.5	34.3/37.7
2	✗	✔	32.4/33.6	15.1/16.1	37.7/38.7
3	✔	✔	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**

TABLE 4
Evaluation for the calculation of $(R_i^j)_{k'}$ in Equ. (3) when using box- or mask-level pooling on the VOC2012 *segmentation train* set [36]. Each result pair w_1/w_2 denotes the result without (w_1) and with (w_2) the knowledge graph, respectively.

No.	Proposal types	AP$_{50}$	AP$_{75}$	ABO
1	Box	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
2	Mask	30.7/32.4	14.5/15.7	36.8/38.0

Evaluation metrics. For the evaluation metrics of instance segmentation, we just follow [13] to employ the region-based mean average precision (AP) at IoU threshold 0.5 (AP$_{50}$) and 0.75 (AP$_{75}$) (see in [37]), as well as the average best overlap (ABO) metric (see in [34]) that provides a different perspective.

6.2 Ablation Study

Before the comparison with other competitors, we perform several ablation studies to evaluate the effectiveness of different design choices and parameter settings. All ablation studies are conducted for weakly supervised instance segmentation on the VOC2012 *segmentation train* set [36] as described above. Here, we do not train Mask R-CNN [3] to save time if not mentioned. When tuning each group of hyper-parameters, other parameters are kept as default.

TABLE 5
Evaluation of different η values in Equ. (3)) on the VOC2012 *segmentation train* set [36]. Each result pair w_1/w_2 denotes the result without (w_1) and with (w_2) the knowledge graph, respectively.

No.	η	AP$_{50}$	AP$_{75}$	ABO
1	0.25	28.3/30.8	13.7/15.3	34.6/36.5
2	0.50	30.0/33.4	14.2/16.0	36.5/38.7
3	0.75	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
4	1.00	30.1/32.5	14.5/16.0	36.4/38.3

TABLE 6
Evaluation results of different δ values (in Equ. (10)) on the VOC2012 *segmentation train* set [36].

No.	δ	AP$_{50}$	AP$_{75}$	ABO
1	1	30.3	14.9	37.0
2	2	34.6	16.3	39.3
3	3	34.8	16.7	39.4
4	5	34.8	16.7	39.4
5	10	34.8	16.6	39.4

TABLE 7
Evaluation for the upper bound of LIID on the VOC2012 *segmentation train* set [36]. The oracle version uses ground truth boxes to filter and label SOP.

No.	GT boxes (Oracle)	AP$_{50}$	AP$_{75}$	ABO
1	✗	34.8	16.7	39.4
2	✔	44.9	23.0	39.1

TABLE 8
Evaluation for each component of LIID after Mask R-CNN training on the VOC2012 *segmentation val* set [36]. The symbol ✗ means to remove a component in LIID. The first line (No. 1) is the default version of LIID.

No.	Strategy	AP$_{50}$	AP$_{75}$	ABO
1	-	**48.4**	**24.9**	**50.8**
2	CAM-Based Loss $L_{Att}^{(i)}$ ✗	38.3	17.1	45.4
3	MIL Loss $L_{MIL}^{(i)}$ ✗	46.9	24.1	48.1
4	Center Loss $L_{Cent}^{(i)}$ ✗	45.8	23.0	48.6
5	Knowledge Graph ✗	46.1	22.8	48.1
6	$(R_i^j)_{k'}$ (Box → Mask)	45.2	22.9	48.9

The parameter setting of center loss $L_{Cent}^{(i)}$. The center loss is designed to concentrate the feature vectors $\mathbf{f}_i^j$. The hyperparameter θ (in Equ. (8)) controls the update speed of the center feature vector of each category, and the parameter γ (in Equ. (9)) controls its influence to the backbone net. Different settings and corresponding results of θ and γ are displayed in Table 1. When we have $\gamma = 0$, the parameter θ and the consequent knowledge graph are omitted (No. 9 in Table 1). We can see that the results of this setting are worse than the best setting without the knowledge graph, demonstrating that the MIL-based center loss (Section 4.2.3) is not only necessary for the construction of the knowledge graph but also helpful for the training of the MIL framework. When we have $\gamma \neq 0$, θ and γ seem not sensitive to different values. The setting of $\theta = 0.01$ and $\gamma = 0.1$ achieves slightly better performance. Therefore, we use 0.01 and 0.1 as the default values for θ and γ, respectively.

The balance factors of loss functions $L_{Att}^{(i)}$ and $L_{MIL}^{(i)}$. We also evaluate the effectiveness of the balance factors α, β for loss functions $L_{Att}^{(i)}$ and $L_{MIL}^{(i)}$ in Equ. (9). The results are shown in Table 2. We can see both $L_{Att}^{(i)}$ and $L_{MIL}^{(i)}$ contribute a lot to the final instance segmentation. When $\alpha = 0.5$ and $\beta = 0.5$, the proposed method performs best, so we use this setting as default.

The $mean$ and max terms of $(R_i^j)_k$. In Equ. (3), we have defined an auxiliary term $(R_i^j)_k = mean(A_i^k[b_i^j]) + max(A_i^k[b_i^j])$ to compute the approximate category label $\widetilde{y}_i^j$ which will be used in Equ. (4) to compute $L_{Att}^{(i)}$. In Table 3, we conduct MIL training using only $mean$ term of $(R_i^j)_k$, only max term, and both $mean$ and max terms. The third experiment clearly outperforms the other two.

The box- or mask-level pooling for $(R_i^j)_{k'}$. In Section 4.2.1, we intuitively analyze the reason why we use the box-level pooling rather than mask-level pooling for the calculation of $(R_i^j)_{k'}$ in Equ. (3). Here, we conduct experiments to demonstrate the superiority of box-level pooling when compared with mask-level pooling on the VOC2012 *segmentation val* set [36]. The results are displayed in Table 4 and Table 8 (No. 6). We can observe that mask-level pooling leads to significant performance degradation. Maybe this is because the inaccurate SOP impairs the training of the MIL framework.

The threshold η for $(R_i^j)_{k'}$. In Table 5, we apply different thresholds η for $(R_i^j)_k$ in Equ. (3). Although we have $\eta \in [0, 2]$, we only test $\eta \leq 1.00$ because $\eta \geq 0.75$ leads to significant performance degradation. The threshold of 0.75 performs best, so we use it as the default setting.

The effectiveness of the knowledge graph. In Section 5, we use the outputs of the MIL framework to construct a knowledge graph whose multi-way cut can assign category labels to corresponding proposals. Without the knowledge graph, we can also use the probabilities learned by MIL to label proposals. In Table 1 - Table 5, we report results before and after the multi-way cut. The knowledge graph can improve performance in all cases. Therefore, we can conclude that the knowledge graph is essential to our system.

The balance factor δ. In Equ. (10), we use a balance factor δ to control the contribution of feature *cosine* similarity for graph edges. In Table 6, we study the effect of different δ values. We achieve similar results when $\delta \geq 2$. According to the results, we set δ to 5 as default because $\delta = 5$ has slightly better performance.

The discussion about CAM. If we set $\alpha = 1.0$ and $\beta = 0.0$ for Equ. (9) and do not use the knowledge graph in Section 5, the model will degenerate to a form that only relies on CAM for training. In Table 2, we can see that the results are 28.7%, 13.9%, and 35.5% in terms of AP$_{50}$, AP$_{75}$, and ABO, respectively. With our other designs, the results are improved to 34.8%, 16.7%, and 39.4% in terms of AP$_{50}$, AP$_{75}$, and ABO, respectively. Note that this simple CAM-based variant of our model also includes some of our effective designs as proven in Table 3 - Table 5. Therefore, our system is not straightforward.

The upper bound of LIID. We also evaluate the upper bound of LIID using ground truth boxes to filter and label SOP. Specifically, if the maximum IoU of a bounding box proposal with any ground truth boxes is larger than 0.5, this proposal is kept, and its assigned label is the same as the ground truth box with the maximum IoU; otherwise, this proposal is abandoned. We show the experimental

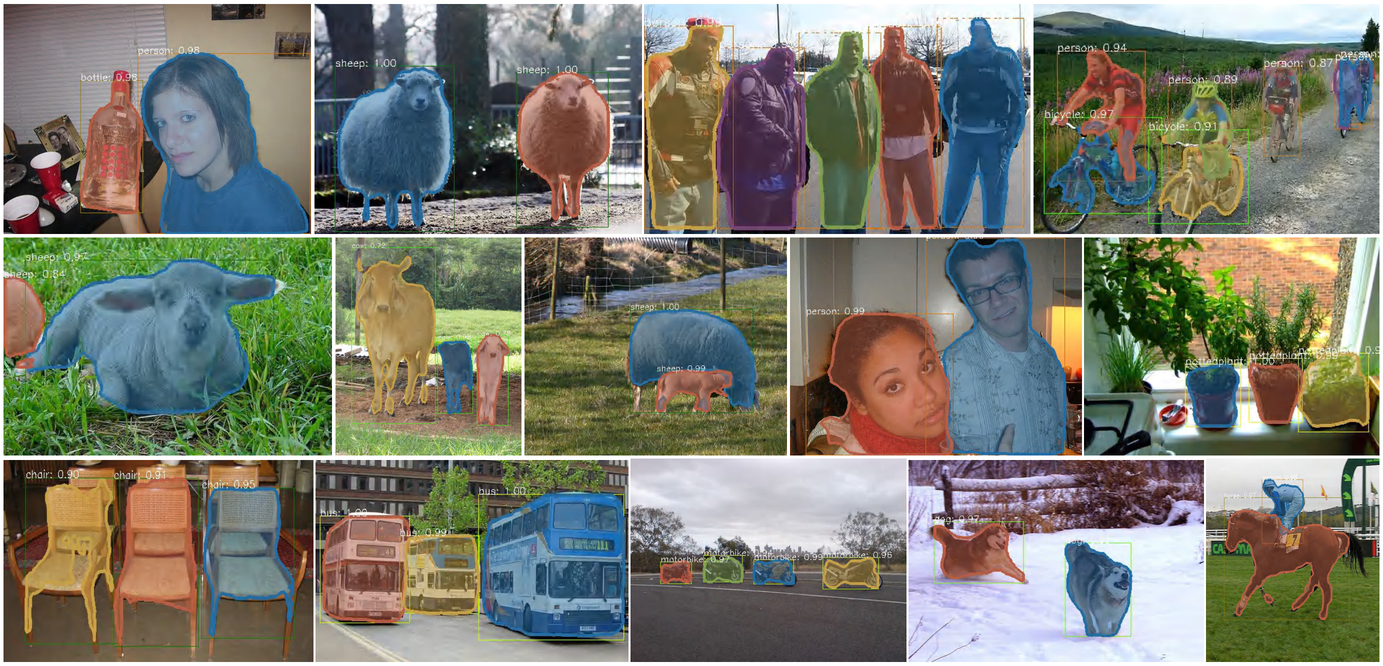


Fig. 3. Qualitative results of instance segmentation on the PASCAL VOC2012 *segmentation val* set [36].

TABLE 9
Comparison of our method and other weakly supervised instance segmentation models on the VOC2012 *segmentation val* dataset [36]. The light-colored method [8] uses bounding boxes as supervision, whereas other methods only use image-level labels as supervision.

Method		AP_{50}	AP_{75}	ABO
CAM [19]	Rect.	2.5	0.1	18.9
	Ellipse	3.9	0.1	20.8
	MCG	7.8	2.5	23.0
SPN [84]	Rect.	5.2	0.3	23.0
	Ellipse	6.1	0.3	24.0
	MCG	12.7	4.4	27.1
MELM [10]	Rect.	14.6	1.9	26.4
	Ellipse	19.3	2.4	27.0
	MCG	22.9	8.4	32.9
PRM [13]		26.8	9.0	37.6
IAM-S5 [14]		28.8	11.9	41.9
Cholakkal *et al.* [15]		30.2	14.4	44.3
Ahn *et al.* [16]		46.7	17.4	-
Hsu *et al.* [8]		58.9	21.6	-
Label-PEnet [17]		30.2	12.9	41.4
LIID (Ours)		**48.4**	**24.9**	**50.8**

results in Table 7. There is a large performance gap between LIID and the oracle version, leaving room for future improvement.

Each component after Mask R-CNN training. We continue by evaluating the effect of each component after Mask R-CNN training on the VOC2012 *segmentation val* set [36]. Specifically, we omit each component of the loss function or the multi-way graph cut and then adopt the produced pseudo ground truth to train Mask R-CNN. The results are summarized in Table 8. We can observe that every component of LIID contributes significantly to the final performance, as removing any component would lead to substantial performance degradation.

6.3 Instance Segmentation on VOC2012

Since weakly supervised instance segmentation with only image-level supervision is a recently initiated problem by Zhou *et al.* [13], the previous study on this topic is very limited [13]–[17]. Hence we follow [13] to construct some baselines based on the bounding boxes generated by several weakly supervised object localization models [10], [19], [84]. To obtain the instance segmentation, we apply three simple mask extraction strategies: i) Rect, *i.e.*, just using the bounding boxes as the segmentation results; ii) Ellipse, *i.e.*, simply filling the largest ellipse enclosed in each bounding box; iii) MCG, *i.e.*, retrieving an MCG SOP [18] with the maximum IoU with each bounding box. We train a Mask R-CNN model [3] using the pseudo instance segmentation of the training set and compare the test results with [8], [13]–[17] and these nine baseline models.

The numeric experimental results on the VOC2012 *segmentation val* set [36] are summarized in Table 9. Note that Hsu *et al.* [8] used bounding boxes as supervision, so it is unfair to compare other methods with it directly. Nevertheless, the proposed LIID achieves a 3.3% improvement compared with [8] in terms of the AP_{75} metric, which demonstrates the effectiveness of LIID for accurately segmenting object instances. It is not surprising to see that [8] outperforms LIID in terms of the AP_{50} metric, because the bounding box priors used by [8] would greatly help to find object instances and thus roughly segment them with a small overlap with ground truths. For image-level supervised methods, the proposed LIID achieves the best performance under various evaluation metrics. Compared with the second-best method, *i.e.*, [16], LIID is 1.7% and 7.5% higher in terms of AP_{50} and AP_{75}, respectively. Note that AP_{75} is the most important measure metric for instance segmentation, because it reflects the ability of detections to cover objects tightly. The significant improvement in terms of AP_{75} indicates that LIID is good at correctly segmenting objects that have a high overlap with ground truth. The recent weakly supervised object detection model MELM [10] with SOP generated by

TABLE 10
Instance segmentation *mask* AP on COCO *test-dev* [37]. The details of metrics can be found in [37]. The light-colored methods are fully supervised, while [28], [85] and our LIID are weakly supervised.

Method	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
MNC [39]	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [86]	29.2	49.5	-	7.1	31.3	50.0
Mask R-CNN [3]	35.7	58.0	37.8	15.5	38.1	52.4
Fan *et al.* [28]	13.7	25.5	13.5	0.7	15.7	26.1
WS-JDS [85]	6.1	11.7	5.5	1.5	7.1	12.2
LIID (Ours)	**16.0**	**27.1**	**16.5**	**3.5**	**15.9**	**27.7**

MCG can perform pretty well but worse than PRM [13] and LIID. This demonstrates weakly supervised object detection is highly related to, but can not be directly applied to weakly supervised instance segmentation. We display some examples of our instance segmentation results in Fig. 3. We can see that LIID can produce pretty good instance segmentation. Even for images containing multiple instances of the same category, each instance can be segmented very well.

Runtime and memory consumption. For the runtime and memory footprint, the multi-way cut needs about 5 minutes and 26 GB CPU memory for VOC2012 training images. The MIL framework needs about 0.02 seconds to process an image. Hence the average runtime for a training image is $5\times60/10\text{K}+0.02 = 0.05$ second. The runtime for a test image is the same as Mask R-CNN [3] because we adopt our pseudo ground truth to train Mask R-CNN for testing.

6.4 Instance Segmentation on MS-COCO

In this part, we compare with [28], [85] that have reported weakly supervised instance segmentation results on the MS-COCO dataset [37]. We use the same experimental settings to assign category labels to SOP as on the VOC2012 dataset and train a Mask R-CNN [3] model. Besides [28], [85], we also report the results of three fully supervised methods, including MNC [39], FCIS [86], and Mask R-CNN [3]. The evaluation results are summarized in Table 10. The proposed LIID performs significantly better than [28], [85], which demonstrates that the proposed LIID is robust to different datasets. Compared with [28], LIID achieves 2.3%, 1.6%, and 3.0% better performance in terms of AP, AP$_{50}$, and AP$_{75}$, respectively. This proves that the improvement of LIID over [28] is nontrivial.

6.5 Weakly Supervised Semantic Segmentation

The above experiments evaluate our approach for instance segmentation, while another challenging task highly related to us is weakly supervised semantic segmentation with only image-level supervision. Semantic segmentation can be viewed as a pixel-wise classification, in which each pixel is assigned with a category label. Unlike instance segmentation, semantic segmentation need not recognize objects with the same category. For training images, we merge our instance segmentation masks of the same semantic category in each image. Then, we view the resulting semantic segmentation as proxy ground truth and adopt the same settings as in previous methods [26], [28], [54], [90], [92] to train a DeepLab [82] model.

In Table 11, we compare with recent state-of-the-art methods [16], [17], [22]–[32], [46], [47], [54], [55], [65], [67], [70], [87]–[93] on the PASCAL VOC2012 [36] *segmentation val* and *test* sets in terms of *mean intersection-over-union* (mIoU). For a fair comparison, we report the results of these methods with the ResNet101 [21] backbone network if provided by the original paper (recent methods usually report ResNet101 results). Besides the 10K VOC2012 training images, some methods [22], [25], [30], [31], [67], [91] also use extra training data, such as web-crawled images [22], [67], [91], web-crawled videos [25], [30], and pixel-level labels [31], to improve performance, which has been visualized in Table 11. We provide two versions of LIID: one is without extra training data, and the other is pre-trained on the simple ImageNet dataset [68]. The simple ImageNet dataset [68] selects 24K images that have the same categories as PASCAL VOC from ImageNet dataset [71]. LIID outperforms all recent competitors with or without extra data. Compared with [28] that is designed for both instance segmentation and semantic segmentation, LIID has 3.3% and 2.7% higher mIoU on the *val* set and *test* set, respectively, when both methods use the 24K simple ImageNet images [68] as extra training data. This again demonstrates the improvement of LIID over [28] is neither trivial nor straightforward. The recent state-of-the-art method [30] uses 4.6K videos [25] that contain 960K video frames as extra training data, which is $40\times$ more than LIID. However, LIID still performs

TABLE 11
Comparison for weakly supervised semantic segmentation on the PASCAL VOC2012 [36] *segmentation val* and *test* sets. Besides the 10K VOC2012 training images, some methods also use extra data for training. *24K ImageNet* means the simple ImageNet images in [68]. *4.6K Videos* are from the Web-Crawl dataset [25], including 960K video frames. Besides the image-level supervision, semi-supervised methods, [31], [46], [47], also use pixel-level labels, points, and scribbles as the supervision, respectively. For a fair comparison, we report the results of various competitors with the ResNet101 [21] backbone network if provided by the original paper. "†" indicates results with the Res2Net101 [33] backbone.

Method	Year	Extra Data	mIoU (%)	
			val	test
CCNN [87]	ICCV'15	✗	35.3	-
EM-Adapt [88]	ICCV'15	✗	38.2	39.6
MIL [70]	CVPR'15	✗	42.0	-
SEC [55]	ECCV'16	✗	50.7	51.7
AugFeed [65]	ECCV'16	✗	54.3	55.5
Bearman *et al.* [46]	ECCV'16	Points	49.1	-
ScribbleSup [47]	CVPR'16	Scribbles	63.1	-
STC [22]	PAMI'17	40K Web	49.8	51.2
Roy *et al.* [24]	CVPR'17	✗	52.8	53.7
Oh *et al.* [89]	CVPR'17	✗	55.7	56.7
AE-PSL [23]	CVPR'17	✗	55.0	55.7
WebS-i2 [67]	CVPR'17	19K Web	53.4	55.3
Hong *et al.* [25]	CVPR'17	4.6K Videos	58.1	58.7
DCSP [26]	BMVC'17	✗	60.8	61.9
DSRG [54]	CVPR'18	✗	61.4	63.2
MCOF [90]	CVPR'18	✗	60.3	61.2
AffinityNet [32]	CVPR'18	✗	61.7	63.7
Wei *et al.* [27]	CVPR'18	✗	60.4	60.8
GAIN [31]	CVPR'18	1464 Pixel	60.5	62.1
Shen *et al.* [91]	CVPR'18	80K Web	63.0	63.9
Fan *et al.* [28]	ECCV'18	✗	63.6	64.5
Fan *et al.* [28]	ECCV'18	24K ImageNet	64.5	65.6
Ahn *et al.* [16]	CVPR'19	✗	63.5	64.8
FickleNet [92]	CVPR'19	✗	64.9	65.3
Label-PEnet [17]	ICCV'19	✗	-	57.2
Lee *et al.* [30]	ICCV'19	4.6K Videos	66.5	67.4
SSDD [93]	ICCV'19	✗	64.9	65.5
OAA [29]	ICCV'19	✗	65.2	66.4
LIID (Ours)	-	✗	**66.5**	**67.5**
LIID (Ours)	-	**24K ImageNet**	**67.8**	**68.3**
LIID† (Ours)	-	✗	**69.4**	**70.4**

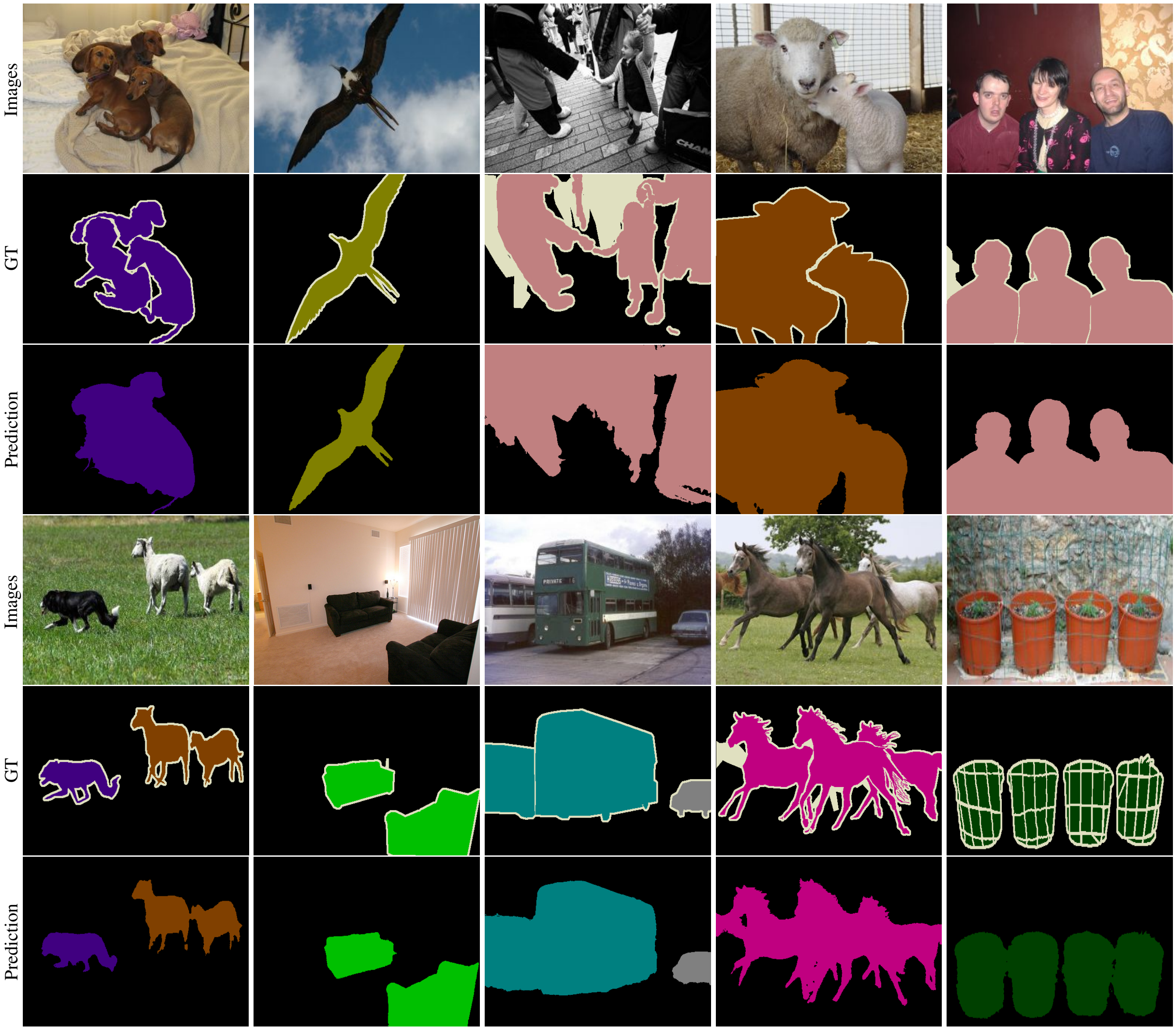


Fig. 4. Qualitative results of semantic segmentation on the PASCAL VOC2012 *segmentation val* set [36]. **From Top to Bottom**: Original images, ground truth, and the predicted results by LIID, repeated by the bottom three rows.

better than it, demonstrating the superiority of LIID. We display some examples of our semantic segmentation results in Fig. 4. Combined with the experiments in Section 6.3 and Section 6.4, we can come to the conclusion that LIID achieves the state-of-the-art performance for both weakly supervised instance segmentation and semantic segmentation.

7 Conclusion

In this paper, we tackle the problem of weakly supervised instance segmentation with only image-level supervision. Our effort starts with some generic SOP. With these proposals, we first propose an MIL framework which can simultaneously predict probability distributions and extract semantic feature vectors. Then, we construct a large knowledge graph for all training images with the obtained information. At last, an improved multi-way cut algorithm is proposed to classify each proposal into a category. The proposals falling into the background category will be viewed as noisy data and be removed. Therefore, the proposed approach leverages instance-, image- and dataset-level information to retrieve object proposals and assign correct labels to them. Compared with previous competitors, the proposed approach can achieve better performance for both weakly supervised instance segmentation and semantic segmentation. Moreover, we use the same hyper-parameters for PASCAL VOC2012 and COCO datasets, which indicates that the hyperparameters of our approach are robust to different datasets. In the future, we would try to apply the proposed proposal-based MIL framework and multi-way cut formulation to other weakly supervised vision tasks.

Acknowledgments

This research was supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, NSFC (61922046), Tianjin Natural Science Foundation(18ZXZNGX00110), and S&T innovation project from Chinese Ministry of Education.

References

[1] R. Girshick, "Fast R-CNN," in *Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inform. Process. Syst.*, 2015, pp. 91–99.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3213–3223.

[6] A. Khoreva, R. Benenson, J. H. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 876–885.

[7] Q. Li, A. Arnab, and P. H. Torr, "Weakly- and semi-supervised panoptic segmentation," in *Eur. Conf. Comput. Vis.*, 2018, pp. 106–124.

[8] C.-C. Hsu, K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, and Y.-Y. Chuang, "Weakly supervised instance segmentation using the bounding box tightness prior," in *Adv. Neural Inform. Process. Syst.*, 2019, pp. 6582–6593.

[9] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick, "Learning to segment every thing," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4233–4241.

[10] F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye, "Min-entropy latent model for weakly supervised object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1297–1306.

[11] X. Zhang, J. Feng, H. Xiong, and Q. Tian, "Zigzag learning for weakly supervised object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4262–4270.

[12] Y. Shen, R. Ji, S. Zhang, W. Zuo, and Y. Wang, "Generative adversarial learning towards fast weakly supervised detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5764–5773.

[13] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3791–3800.

[14] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann, and J. Jiao, "Learning instance activation maps for weakly supervised instance segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3116–3125.

[15] H. Cholakkal, G. Sun, F. S. Khan, and L. Shao, "Object counting and instance segmentation with image-level supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 12 397–12 405.

[16] J. Ahn, S. Cho, and S. Kwak, "Weakly supervised learning of instance segmentation with inter-pixel relations," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2209–2218.

[17] W. Ge, S. Guo, W. Huang, and M. R. Scott, "Label-PEnet: Sequential label propagation and enhancement networks for weakly supervised instance segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 3345–3354.

[18] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 128–140, 2017.

[19] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2921–2929.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[22] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan, "STC: A simple to complex framework for weakly-supervised semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2314–2320, 2017.

[23] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1568–1576.

[24] A. Roy and S. Todorovic, "Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3529–3538.

[25] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han, "Weakly supervised semantic segmentation using web-crawled videos," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 7322–7330.

[26] A. Chaudhry, P. K. Dokania, and P. H. Torr, "Discovering class-specific pixels for weakly-supervised semantic segmentation," in *Brit. Mach. Vis. Conf.*, 2017.

[27] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7268–7277.

[28] R. Fan, Q. Hou, M.-M. Cheng, G. Yu, R. R. Martin, and S.-M. Hu, "Associating inter-image salient instances for weakly supervised semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2018, pp. 371–388.

[29] P.-T. Jiang, Q. Hou, Y. Cao, M.-M. Cheng, Y. Wei, and H.-K. Xiong, "Integral object mining via online attention accumulation," in *Int. Conf. Comput. Vis.*, 2019, pp. 2070–2079.

[30] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 6808–6818.

[31] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 9215–9223.

[32] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4981–4990.

[33] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[34] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[35] P. Krahenbuhl and V. Koltun, "Learning to propose objects," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1574–1582.

[36] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[38] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Eur. Conf. Comput. Vis.*, 2014, pp. 297–312.

[39] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3150–3158.

[40] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 8759–8768.

[41] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 4974–4983.

[42] A. Arnab and P. H. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 441–450.

[43] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2858–2866.

[44] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "InstanceCut: from edges to instances with multicut," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 5008–5017.

[45] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[46] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *Eur. Conf. Comput. Vis.*, 2016, pp. 549–565.

[47] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3159–3167.

[48] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3136–3145.

[49] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *Int. Conf. Comput. Vis.*, 2017, pp. 3544–3553.

[50] D. Kim, D. Cho, D. Yoo, and I. So Kweon, "Two-phase learning for weakly supervised object localization," in *Int. Conf. Comput. Vis.*, 2017, pp. 3534–3543.

[51] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, “Adversarial complementary learning for weakly supervised object localization,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1325–1334.

[52] T. Durand, T. Mordan, N. Thome, and M. Cord, “WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 642–651.

[53] T. Durand, N. Thome, and M. Cord, “Exploiting negative evidence for deep latent structured models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 337–351, 2018.

[54] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, “Weakly-supervised semantic segmentation network with deep seeded region growing,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7014–7023.

[55] A. Kolesnikov and C. H. Lampert, “Seed, expand and constrain: Three principles for weakly-supervised image segmentation,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 695–711.

[56] Y. Liu, M.-M. Cheng, X. Zhang, G.-Y. Nie, and M. Wang, “DNA: Deeply-supervised nonlinear aggregation for salient object detection,” *IEEE Transactions on Cybernetics*, 2020.

[57] Y. Qiu, Y. Liu, H. Yang, and J. Xu, “A simple saliency detection approach via automatic top-down feature fusion,” *Neurocomputing*, vol. 388, pp. 124–134, 2020.

[58] Y. Qiu, Y. Liu, X. Ma, L. Liu, H. Gao, and J. Xu, “Revisiting multilevel feature fusion: A simple yet effective network for salient object detection,” in *IEEE Int. Conf. Image Process.*, 2019, pp. 4010–4014.

[59] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, “Salient object detection: A survey,” *Computational Visual Media*, vol. 5, no. 2, pp. 117–150, 2019.

[60] M.-M. Cheng, N. Mitra, X. Huang, and S.-M. Hu, “SalientShape: group saliency in image collections,” *The Vis. Comput.*, vol. 30, no. 4, pp. 443–453, 2014.

[61] S. Xie and Z. Tu, “Holistically-nested edge detection,” *Int. J. Comput. Vis.*, vol. 125, no. 1-3, p. 3, 2017.

[62] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang, “Richer convolutional features for edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1939–1946, 2019.

[63] Y. Liu, M.-M. Cheng, D.-P. Fan, L. Zhang, J. Bian, and D. Tao, “Semantic edge detection with diverse deep supervision,” *arXiv preprint arXiv:1804.02864*, 2018.

[64] Y. Liu, S.-J. Li, and M.-M. Cheng, “RefinedBox: Refining for fewer and high-quality object proposals,” *Neurocomputing*, 2020.

[65] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia, “Augmented feedback in semantic segmentation under image level supervision,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 90–105.

[66] W. Shimoda and K. Yanai, “Distinct class-specific saliency maps for weakly supervised semantic segmentation,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 218–234.

[67] B. Jin, M. V. O. Segovia, and S. Süsstrunk, “Webly supervised semantic segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1705–1714.

[68] Q. Hou, D. Massiceti, P. K. Dokania, Y. Wei, M.-M. Cheng, and P. H. Torr, “Bottom-up top-down cues for weakly-supervised semantic segmentation,” in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2017, pp. 263–277.

[69] F. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez, “Built-in foreground/background prior for weakly-supervised semantic segmentation,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 413–432.

[70] P. O. Pinheiro and R. Collobert, “From image-level to pixel-level labeling with convolutional networks,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1713–1721.

[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[72] P. Krähenbühl and V. Koltun, “Geodesic object proposals,” in *Eur. Conf. Comput. Vis.*, 2014, pp. 725–739.

[73] M.-M. Cheng, Y. Liu, W.-Y. Lin, Z. Zhang, P. L. Rosin, and P. H. Torr, “BING: Binarized normed gradients for objectness estimation at 300fps,” *Computational Visual Media*, vol. 5, no. 1, pp. 3–20, 2019.

[74] Z. Zhang, Y. Liu, X. Chen, Y. Zhu, M.-M. Cheng, V. Saligrama, and P. H. Torr, “Sequential optimization for efficient high-quality object proposal generation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1209–1223, 2017.

[75] R. Fan, M.-M. Cheng, Q. Hou, T.-J. Mu, J. Wang, and S.-M. Hu, “S4Net: Single stage salient-instance segmentation,” *Computational Visual Media*, vol. 6, no. 2, pp. 191–204, 2020.

[76] K. P. Murphy, *Machine learning: A probabilistic perspective*. MIT press, 2012.

[77] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, “The complexity of multiterminal cuts,” *SIAM Journal on Computing (SICOMP)*, vol. 23, no. 4, pp. 864–894, 1994.

[78] N. Garg, V. V. Vazirani, and M. Yannakakis, “Approximate max-flow min-(multi) cut theorems and their applications,” *SIAM Journal on Computing (SICOMP)*, vol. 25, no. 2, pp. 235–251, 1996.

[79] G. Călinescu, H. Karloff, and Y. Rabani, “An improved approximation algorithm for multiway cut,” *Journal of Computer and System Sciences (JCSS)*, vol. 60, no. 3, pp. 564–574, 2000.

[80] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.

[81] C. Bliek1ú, P. Bonami, and A. Lodi, “Solving mixed-integer quadratic programming problems with IBM-CPLEX: A progress report,” in *RAMP Symposium*, 2014, pp. 16–17.

[82] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[83] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, “HCP: A flexible CNN framework for multi-label image classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1901–1907, 2016.

[84] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, “Soft proposal networks for weakly supervised object localization,” in *Int. Conf. Comput. Vis.*, 2017, pp. 1841–1850.

[85] Y. Shen, R. Ji, Y. Wang, Y. Wu, and L. Cao, “Cyclic guidance for weakly supervised joint detection and segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 697–707.

[86] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2359–2367.

[87] D. Pathak, P. Krahenbuhl, and T. Darrell, “Constrained convolutional neural networks for weakly supervised segmentation,” in *Int. Conf. Comput. Vis.*, 2015, pp. 1796–1804.

[88] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, “Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation,” in *Int. Conf. Comput. Vis.*, 2015, pp. 1742–1750.

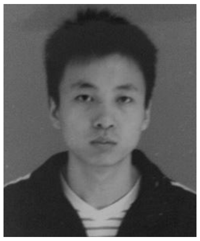
[89] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, B. Schiele *et al.*, “Exploiting saliency for object segmentation from image level labels,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[90] X. Wang, S. You, X. Li, and H. Ma, “Weakly-supervised semantic segmentation by iteratively mining common object features,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1354–1362.

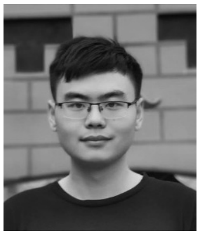
[91] T. Shen, G. Lin, C. Shen, and I. Reid, “Bootstrapping the performance of webly supervised semantic segmentation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1363–1371.

[92] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, “FickleNet: Weakly and semi-supervised semantic image segmentation using stochastic inference,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 5267–5276.

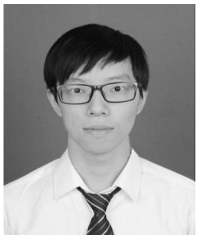
[93] W. Shimoda and K. Yanai, “Self-supervised difference detection for weakly-supervised semantic segmentation,” in *Int. Conf. Comput. Vis.*, 2019, pp. 5208–5217.



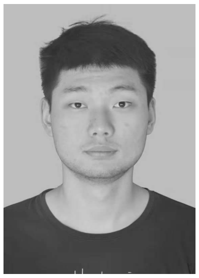
Yun Liu is a PhD candidate at College of Computer Science, Nankai University. He received his bachelor’s degree from Nankai University in 2016. His research interests include computer vision and machine learning.



Yu-Huan Wu is currently a Ph.D. candidate with College of Computer Science at Nankai University, supervised by Prof. Ming-Ming Cheng. He received his bachelor's degree from Xidian University in 2018. His research interests include computer vision and machine learning.



Peisong Wen is a senior undergraduate student at Nankai University. His research interests are semantic segmentation and video object segmentation.



Yujun Shi is currently a research assistant at National University of Singapore, supervised by Dr. Jiashi Feng. His research interests include the adversarial behavior of deep neural networks and applying the theory of optimal control in deep learning.

Yu Qiu is a PhD candidate at the College of Artificial Intelligence, Nankai University. She received her bachelor's degree from Northwest University of A&F Science and Technology in 2017. Her research interests include machine learning, computer vision, and software engineering.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did two years research fellow with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards, including ACM China Rising Star Award, IBM Global SUR Award, and CCF-Intel Young Faculty Researcher Program. He is on the editorial boards of IEEE TIP.