Toward Joint Thing-and-Stuff Mining for Weakly Supervised Panoptic Segmentation

Yunhang Shen\textsuperscript{1}, Liujuan Cao\textsuperscript{1,*}, Zhiwei Chen\textsuperscript{1}, Feihong Lian\textsuperscript{1}, Baochang Zhang\textsuperscript{2}
Chi Su\textsuperscript{3}, Yongjian Wu\textsuperscript{4}, Feiyue Huang\textsuperscript{4}, Rongrong Ji\textsuperscript{1,5,6}

\textsuperscript{1}Media Analytics and Computing Lab, Department of Artificial Intelligence, School of Informatics, Xiamen University, 361005, China, \textsuperscript{2}Institute of Artificial Intelligence, Beihang University, Beijing, China, \textsuperscript{3}KingSoft Cloud Co. Ltd., Beijing, China
\textsuperscript{4}Tencent Youtu Lab, Shanghai, China, \textsuperscript{5}Institute of Artificial Intelligence, Xiamen University, Xiamen, China, \textsuperscript{6}Peng Cheng Laborotory, Shenzhen, China

shenyunhang01@gmail.com, caoliujuan@xmu.edu.cn, zhiweichen.xmu@gmail.com
lianfh@stu.xmu.edu.cn, bczhang@buaa.edu.cn, suchi@kingsoft.com
littlekenwu@tencent.com, garyhuang@tencent.com, rrji@xmu.edu.cn

Abstract

Panoptic segmentation aims to partition an image to object instances and semantic content for thing and stuff categories, respectively. To date, learning weakly supervised panoptic segmentation (WSPS) with only image-level labels remains unexplored. In this paper, we propose an efficient jointly thing-and-stuff mining (JTSM) framework for WSPS. To this end, we design a novel mask of interest pooling (MoIPool) to extract fixed-size pixel-accurate feature maps of arbitrary-shape segmentations. MoIPool enables a panoptic mining branch to leverage multiple instance learning (MIL) to recognize things and stuff segmentation in a unified manner. We further refine segmentation masks with parallel instance and semantic segmentation branches via self-training, which collaborates the mined masks from panoptic mining with bottom-up object evidence as pseudo-ground-truth labels to improve spatial coherence and contour localization. Experimental results demonstrate the effectiveness of JTSM on PASCAL VOC and MS COCO. As a by-product, we achieve competitive results for weakly supervised object detection and instance segmentation. This work is a first step towards tackling challenge panoptic segmentation task with only image-level labels.

1. Introduction

Panoptic segmentation focuses on simultaneously segmenting all object instances and semantic content in an image. It is one of the most important tasks in computer vision due to its great academic values and industrial applications. Recent rapid progress on panoptic segmentation has been driven by combining the strength of instance segmentation and semantic segmentation tasks via a multi-branch scheme. However, these deep models heavily rely on a large amount of training data with expensive instance-level and pixel-wise annotations. Collecting such training data has been a particular bottleneck on the way of applying panoptic segmentation to real-world applications, e.g., autonomous driving, robotics, and image editing, where labelling each pixel for numerous images is particularly time-consuming. For example, fully annotating a single image in Cityscapes \cite{cordts2016cityscapes} required more than 1.5 hours on average.

One way to reduce the requirement of strong supervision is the weakly supervised panoptic segmentation (WSPS), which seeks to use weak annotations for model training. To our best knowledge, the only previous work that attempted to address WSPS problem is that of \cite{sun2020weakly}, which requires bounding boxes for thing categories and image-level tags for stuff during training. However, for applications needing very large-scale image sets and categories, bounding-box-level annotations still require enormous human effort. It is

Figure 1: The overall flowchart of our JTSM framework.
thus desirable to learn panoptic segmentation from large-scale datasets with weaker supervision.

We focus on the most extreme case of WSPS where only image-level labels are available, and no instance-level annotations are involved during training. To date, none of the existing work further investigates the problem of learning panoptic segmentation with only image-level labels. An intuitive and strong baseline method is to perform weakly supervised instance segmentation (WSIS) and weakly supervised semantic segmentation (WSSS) independently, and use heuristic post-processing method [3] to merge their results. However, the straightforward combination of such two techniques disregards the underlying relationship and fails to borrow rich contextual cues between things and stuff. As context information is critical to recognize and localize the objects, and foreground objects provide complementary cues to assist background understanding [4, 5].

In this paper, we propose a Joint Thing-and-Stuff Mining (JTSM) framework to learn panoptic segmentation with only image-level labels, as illustrated in Fig. 1. Our motivation is to consider foreground things and background stuff as uniform object instances in form of segmentation masks. Particularly, each connected component of stuff content is viewed as an individual instance, which shares the same spirit as thing objects. Different to the baseline that frames the two related tasks at architectural level via a multi-branch scheme, the main advantage of JTSM is to model the correlations between objects and background at instance level.

To this end, we design a novel mask of interest pooling (MoIPool) to extract fixed-size pixel-accurate feature maps for arbitrary-shape segmentations, which provides a uniform representational power for things and stuff. Thus, given a set of segment proposals, a panoptic mining branch leverages multiple instance learning (MIL) to mine all target categories in a unified manner. We further introduce two schemes to refine segmentation masks. First, we collaboratively refine results from panoptic mining with bottom-up object evidence to improve spatial coherence and contour localization. Second, we introduce self-training to refine segmentation masks with parallel instance and semantic segmentation branches. With pseudo-ground-truth masks from its preceding branch, the discriminative power of the image segmentation can be enhanced. Experimental results demonstrate the effectiveness of our proposed JTSM compared to strong baselines on PASCAL VOC [6] and MS COCO [7]. As a by-product, we also achieve competitive results for both weakly supervised object detection and instance segmentation tasks.

The contributions of this work are three folds:

- We propose JTSM to jointly segment things and stuff for weakly supervised panoptic segmentation in a unified framework. To our best knowledge, this work makes the first attempt to tackle challenge panoptic segmentation task with only image-level labels.
- We design a novel mask of interest pooling (MoIPool) to compute fixed-size pixel-accurate feature maps of arbitrary-shape segmentations, which enables JTSM to leverage multiple instance learning (MIL) to mine thing and stuff with a uniform representational power.
- Self-training is further introduced to refine the image segmentation with two parallel instance and semantic segmentation branches, which are supervised by the mined results and bottom-up object evidence to improve spatial coherence and contour localization.

2. Related Work

Weakly Supervised Panoptic Segmentation (WSPS). Although learning panoptic segmentation with only image-level labels is challenging without any existing work in previous literature, one attempt with bounding-box-level annotations has been made [2]. Our work has significant differences to [2]. First, method in [2] required bounding-box supervision and fully-labelled examples of some categories. We use only image-level labels to learn panoptic segmentation for the first time. Second, they [2] heavily relied on external models to pre-compute pseudo-ground-truth masks for thing and stuff categories independently, which failed to model the intrinsic interaction between semantic segmentation and instance segmentation. However, our method simultaneously segments all target categories in a unified manner to achieve a holistic understanding of an image.

Weakly Supervised Object Detection (WSOD). WSOD aims to predict object instance in the form of bounding boxes with weak supervision. Recent widely-used WSOD alternates between localizing object instances and training appearance representation via multiple instance learning (MIL). For example, WSDDN [8] selected box proposals by parallel detection and classification branches in deep convolutional neural networks (CNNs). This method is extended by leveraging contextual information [9], gradient map [10, 11], attention mechanism [12], semantic segmentation [13, 14, 15] to suppress low-quality box proposals. Some work in [16, 17, 18, 19] treated the top-scoring proposals as supervision to train multiple instance refinement classifiers. Other different strategies [20, 21, 22, 23, 24, 25, 26] were also proposed to generate pseudo-ground-truth boxes and assign labels to box proposals. And the above framework was further improved by min-entropy prior [27, 28], continuation MIL [29], utilizing uncertainty [30, 31, 32], knowledge distillation [33], spatial likelihood voting [34], objectness consistent [35, 36] and generative adversarial learning [37]. Methods in [38, 39, 40] trained object detection systems from different supervisions.
Weakly Supervised Instance Segmentation (WSIS). WSIS methods can be categorized into two groups. The first group utilizes bounding-box annotations as weak supervision to training WSIS models. Most methods in this group used box-driven segmentation [41, 42] or multiple instance learning [43] to generate instance-level pseudo-ground-truth labels, which are then refined by recursive training [41, 2]. The second group further challenges WSIS problem with only image-level supervision. The early work [44, 45] utilized class response maps to capture visual cues via back-propagation, which are used to generate instance masks from object segment proposals. WISE [46] and IRNet [47] generated coarse masks from class activation maps [48], which is regarded as pseudo-ground-truth labels to train fully supervised models. S$^2$Net [49] and LIID [50] further leveraged graph partitioning algorithms to learn pseudo-ground-truth labels. Label-PEnet [51] transform image-level labels to pixel-wise predictions with multiple cascaded modules and curriculum learning strategy. Kim et al. [52] proposed multi-task community learning to construct a positive feedback loop and generates pseudo-ground-truth masks using class activation maps [48].

Weakly Supervised Semantic Segmentation (WSSS). Recently, lots of WSSS methods have been proposed to alleviate labelling cost. Many early work [53, 54, 55, 44] leveraged CNN built-in pixel-level cues and constraint priors to learn segmentation masks. Pathak et al. [53] proposed a constrained CNN, which applied linear constraints on the structured output space of pixel labels. Saleh et al. [55] extracted the built-in masks directly from the hidden layer activation and incorporated the resulting masks via a weakly supervised loss. Some works derive category-wise saliency maps from intermediate feature maps of CNNs to estimate the segmentation masks [54, 44]. Recently, WSSS methods [56, 57, 58, 59] often treat initial object localization cues as pseudo supervision and train fully supervised segmentation models. Popular methods [60, 61, 49, 62, 63] leveraged object saliency maps and feature activation maps to provide complimentary information. Many regularizations [56, 64, 65, 66, 63] were proposed to improve the segmentation results. There are also works [67, 68, 69, 70, 71] that focused on improving feature learning in iterative frameworks. Various approaches based on iteratively mining common feature [72, 73], region refinement [59, 74], random-walk label propagation [75], dilated convolution [57] and pixel-level semantic affinity [58] were proposed. Work in [73, 76] also explored object boundaries to refine localization maps.

3. The Proposed Method
3.1. Overall Framework

The overview of our proposed Joint Thing-and-Stuff Mining (JTSM) framework is illustrated in Fig. 2 We construct a parallel multi-branch architecture for panoptic mining, instance segmentation and semantic segmentation, re-
pectively. Each branch takes full-image feature maps from the backbone network as input. First, panoptic mining branch leverages multiple-instance learning (MIL) [77] to jointly segment thing and stuff object with multiple panoptic refinement heads. Particularly, we design a novel MoIPool to produce fixed-size pixel-accurate convolutional feature maps for segment proposals, which are generated by unsupervised proposal generation methods [78, 79]. Second, mined masks from panoptic mining are integrated with bottom-up object evidence to improve spatial coherence and contour localization. Third, parallel instance and semantic segmentation branches further refine thing and stuff masks by taking the predictions as supervision. During training, we have the following objective function

\[
L = L_{PM} + L_{IS} + L_{SS},
\]

where \(L_{PM}\) is the loss functions of panoptic mining branch, \(L_{IS}\) and \(L_{SS}\) are the loss functions for instance and semantic segmentation branches, respectively.

### 3.2. Joint Thing-and-Stuff Mining

The panoptic mining branch aims to jointly segment countable thing instances and uncountable stuff content in a unified manner. Recall that background stuff can be partitioned into a set of connected components. Thus, we consider each connected component of background as an individual instance, which shares the same spirit as countable objects. Although distinguishing disconnected components for background is unnecessary, all things and stuff are viewed as uniform object instances. To this end, we follow the MIL pipeline in deep convolutional networks and convert two-stream WSDDN [8] and OICR [18] algorithms to recognize instances for all categories in a unified manner.

Formally, given an image \(I\) and corresponding image-level labels \(t = [t_1, t_2, \ldots, t_n]\) during training, JTSM aims to estimate segmentation mask for each object instance in this image. Let \(t\) be a fixed-length binary vector, where \(t_c = 1\) denotes that image \(I\) contains the \(c^{th}\) target category, and otherwise, \(t_c = 0\). And \(n^c\) is the total number of thing and stuff categories. The backbone network first outputs full-image feature maps \(\Phi^I\) of input \(I\). Then we use MoIPool layer (discussed later) to compute fixed-size pooled feature maps \(\Phi^p\) for segment proposals, which are followed by two fully-connected layers with ReLu activation and dropout layer to extract final proposal features. After that, a MIL head forks the proposal features into two streams to produce two score matrices \(S^C, S^D \in \mathbb{R}^{n^p \times n^c}\) by another two fully-connected layers, respectively, where \(n^p\) is the number of proposals. Finally, we use the element-wise product to compute the final proposal score matrix as \(S^{MIL} = \sigma(S^C) \odot \sigma((S^D)^T)^T\), where \(\sigma(\cdot)\) is the softmax function. To train the MIL head with only image-level supervision, a sum pooling is applied to acquire image-level multi-label classification scores as \(y_c = \sum_{p=1}^{n^p} S^{MIL}_{pc}\). Then we obtain a multi-label cross-entropy objective function

\[
L_{MIL} = -\sum_{c=1}^{n^c} \left\{ t_c \log(y_c) + (1 - t_c) \log(1 - y_c) \right\}.
\]

To further reduce mis-recognize, we refine MIL scores via multiple panoptic refinement heads, each of which contains a single fully-connected layer. For the \(r^{th}\) refinement head, it reuses proposal features as input and produces new classification scores \(S^r \in \mathbb{R}^{n^p \times (n^c+1)}\), where \(n^c + 1\) indicates the \(n^c\) object categories and 1 background category. During training, for the \(r^{th}\) head and the \(c^{th}\) category that \(t_c = 1\), the highest-score bounding box from previous prediction \(S^{r-1}\) is selected as pseudo-ground-truth labels and assigns positive/negative labels for the rest segment proposals. We also set \(S^0 = S^{MIL}\). Thus, the corresponding panoptic refinement loss is

\[
L^r_{PR} = -\sum_{p=1}^{n^p} y_{tp}^r \log \left( \frac{\exp(S^r_{tp})}{\sum_j \exp(S^r_{jp})} \right),
\]

where \(t_p^r\) denotes the classification targets for the \(p^{th}\) segment proposal in the \(r^{th}\) head, and \(S^r_{tp}^r\) is the corresponding prediction score. Thus, \(L_{PR}\) is the softmax cross-entropy loss weighted by image-level classification scores \(y_{tp}^r\).

With above definitions, the overall objective function for panoptic mining branch is defined as

\[
L_{PM} = L_{MIL} + \sum_{r=1}^{n^r} L^r_{PR},
\]

where \(n^r\) is the number of panoptic refinement heads. During testing, the average output of all heads is used.

### 3.3. Mask of Interest Pooling (MoIPool)

We design a novel Mask of Interest Pooling to compute fixed-size feature maps for segment proposals. Different to RoIPool [80] and RoIAlign [81] that require rectangle proposals, i.e., bounding boxes, MoIPool enables to extract pixel-accurate feature maps of arbitrary-shape segmentations. To this end, we introduce two efficient variants: shape-interpolation and shape-invariant MoIPool.

The first variant is shape-interpolation MoIPool, which only applies the pooling operation inside the segment proposals. Our intuition is that if non-rigid segmentations are transformed into rigid regions, we can reuse traditional methods, e.g., RoIPool and RoIAlign. Therefore, we first interpolate segmentations to rectangle regions by thin plate splines (TPS) algorithm, which has been widely used as the non-rigid transformation model in image alignment and shape matching. TPS produces smooth surfaces, which are
infectently differentiable. Despite its simplicity, experiment results show that shape-interpolation MoIPool achieves competitive performance compared to shape-invariant one.

We further design a shape-invariant MoIPool to maintain accurate contour information of segment proposals. Suppose that the backbone network extracts a full-image feature map \( \Phi I \in \mathbb{R}^{hI \times wI} \), which has a total stride size \( s \). We omit the channels of feature maps for simplification. Each segment proposal is defined by a binary mask \( M \), which has the same spatial size as the input image \( I \). We can easily obtain the corresponding bounding boxes \( B \) of segment proposal, which is defined by a four-tuple \((x^b, y^b, w^b, h^b)\) that specifies its top-left corner \((y^b, x^b)\) and its width and height \((w^b, h^b)\). We also denote the pooled proposal feature map as \( \Phi F \in \mathbb{R}^{hF \times wF} \), where \( hF \times wF \) is the pre-defined spatial size of pooled features. The proposed MoIPool works by dividing the \( hF / s \times wF / s \) cropped proposal feature map into a \( h^b \times w^b \) grid of sub-windows of approximate size \( h^b / s \times h^b / s \times w^b / s \times w^b / s \). Maximum value in each sub-window is assigned into the corresponding output grid cell as

\[
\Phi_{\text{uv}} \equiv \max_{ij} (\Phi_{ij}),
\]

\[
i \in [y^b / s + [h^b / s \cdot u / h^p], y^b / s + [h^b / s \cdot (u + 1) / h^p]],
\]

\[
j \in [x^b / s + [w^b / s \cdot v / w^p], x^b / s + [w^b / s \cdot v + 1) / w^p]],
\]

where \( \Phi_{\text{uv}} \) is an indicator matrix, which equals 1 if the corresponding elements in proposal feature maps are available for max-pooling. Given the \( i^{th} \) row and the \( j^{th} \) column element in proposal feature maps, we crop the corresponding sub-window in binary mask \( M \in \mathbb{R}^{hM \times wM} \) of segment proposals, and acquire maximum values as

\[
\Omega_{ij} \equiv \max_{ij} M_{ij},
\]

\[
i \in [(u \cdot s), ((u + 1) \cdot s)], j \in [(v \cdot s), ((v + 1) \cdot s)].
\]

However, the above definition fills activations to zeros if corresponding sub-windows do not belong to segment proposals. To align the feature activations among different proposals, we further introduce a compensation term

\[
\psi = \frac{hF \times wF}{\sum_{ij} \Phi_{ij}}.
\]

Thus, the final pooled proposal features are scaled up as

\[
\hat{\Phi}_{F} = \psi \Phi_{F}.
\]

In fact, the proposed MoIPool can be regarded as a generalization of RoIPool. As MoIPool degenerates to RoIPool when the segment proposal is a rectangle window.

### 3.4. Segmentation Refinement

As the quality of segmentation results from panoptic mining branch heavily rely on segment proposals, we further take the advantage of self-train to refine masks. To do this we introduce two parallel instance and semantic segmentation branches, which are supervised by pseudo-ground-truth masks generated from panoptic mining.

In details, instance segmentation branch consists of 4 convolutional layers with \( 3 \times 3 \) kernels and 256 channels to extract feature maps, which followed by a deconvolutional layer with \( 2 \times 2 \) kernels and a final prediction layer with \( 1 \times 1 \) kernels. Instance segmentation takes proposal features \( \Phi_{F} \) from RoIAlign [81] as input and produces refined masks \( M_{IS} \) for all \( n^t \) thing categories. Thus, given a set of box proposals, instance segmentation objective function is

\[
L_{IS} = \sum_{p=1}^{n^p} \sum_{t=1}^{n^t} |t| \cdot y_{tIS} \cdot \text{BCE}(M_{pc} \hat{M}^{IS}_{pc}),
\]

where \( M_{pc}^{IS} \) and \( \hat{M}^{IS}_{pc} \) are the predicted and target masks for the \( p^{th} \) proposals and the \( c^{th} \) category, and \( t^{IS}_{pc} \) is the pseudo category labels for the \( p^{th} \) proposals. And \( L_{BCE} \) is the binary cross-entropy loss. As mask head is class-specific, we only compute losses for categories existed in images, which are then weighted by the image-level prediction scores \( y_{tIS} \).

Semantic segmentation branch consists of two convolutional layers with \( 3 \times 3 \) kernels and 256 channels to extract feature maps and a final prediction layer with \( 1 \times 1 \) kernels. Different to instance segmentation, semantic segmentation branch takes full-image feature maps \( \Phi_{F} \) as input and outputs refined masks \( M_{SS} \) for each \( stuff \) category. Thus, the semantic segmentation objective function is defined as

\[
L_{SS} = \mathcal{L}_{CE}(M_{SS}, \hat{M}_{SS}),
\]

where \( \hat{M}_{SS} \) denotes the target segmentation masks, and \( L_{BCE} \) is the binary cross entropy loss function.

To generate pixel-wise supervision \( \hat{M}_{IS} \) and \( \hat{M}_{SS} \) for segmentation refinement, we integrate the mined masks of panoptic mining branch with bottom-up object evidence to improve spatial coherence and contour localization. We employ unsupervised grouping-based segmentation GrabCut [86] algorithm to re-estimate object masks within the corresponding bounding boxes. For each \( thing \) and \( stuff \) category existed in images, we constrain the area within the highest-score segment proposal as firm positive pixels and the areas outside its bounding boxes as firm negative pixels during re-estimation. Such post-process of mined masks helps to reduce ambiguous outline using low-level features such as the pixel colours. We are not restricted with the algorithms that generate object evidence from input images. The re-estimated masks are treated as pseudo-ground-truth masks to learn above segmentation refinement.
4. Quantitative Evaluations
4.1. Experimental Setup

Datasets We evaluate our method on two popular benchmarks, *i.e.*, PASCAL VOC 2012 [6] and MS COCO [7]. PASCAL VOC 2012 consists of 20 target categories as well as one background category. As in full supervised panoptic segmentation [87–82], we generate a training set by merging the Pascal VOC 2012 training set and the additional annotations from the SBD dataset [88]. This results in 10,582 training images. For validation, we evaluate on the Pascal VOC 2012 validation set, as the evaluation server is not available for panoptic segmentation. The MS COCO panoptic segmentation has a greater number of images and categories. It features 118k training images, 5k validation images. There are 133 semantic classes, including 53 *stuff* and 80 *thing* categories. We also evaluate the performance of object detection on PASCAL VOC 2007 [6], which are widely-used benchmark dataset for WSOD. PASCAL VOC 2007 consists of 5,011 *trainval* images and 4,092 *test* images over 20 categories. Note that only image-level labels are used for model training in all our results.

Evaluation Protocol. Our main evaluation metric is the panoptic quality (PQ), which is the product of segmentation quality (SQ) and recognition quality (RQ) [3]. SQ captures the average segmentation quality of matched segments, whereas RQ measures the ability of an algorithm to detect objects correctly. For the evaluation metrics of instance segmentation, we also report the standard MS COCO metrics [7], which is mean average precision (AP) over IoU thresholds. For object detection on Pascal VOC, we follow standard PASCAL VOC protocol to report the mAP at 50% Intersection-over-Union (IoU) of the detected boxes with the ground-truth ones. We also report CorLoc to indicate the percentage of images in which a method correctly localizes an object of the target category. For object detection on MS COCO, we report standard COCO metrics, including AP at different IoU thresholds.

Implementation Details We implement our method using PyTorch framework. All backbones are initialized with the weights pre-trained on ImageNet ILSVRC [89]. We use synchronized SGD training on 4 GPUs. A mini-batch involves 1 images per GPU. We use a learning rate of 0.01, momentum of 0.9, and dropout rate of 0.5. We use a step learning rate decay schema with decay weight of 0.1 and step size of 70,000 iterations. The total number of training iterations is 100,000. We adopt 4 training schedules for MS COCO. In the multi-scale setting, we use scales range from 480 to 1,216 with stride 32. To improve the robustness, we randomly adjust the exposure and saturation of the images by up to a factor of 1.5 in the HSV space. We use MCG [79] to generate segment proposals for all experiments. We set the maximum number of proposals in an image to be 4,000. The test scores are the average of scales of {480, 576, 688, 864, 1200} and flips. Detection results are post-processed by NMS with a threshold of 0.5.

We use the following parameter settings in all the experiments unless specified otherwise. We set the number $n^f$ of object refinement branches to 4. For the proposed MolPool, we use the shape-invariant version for default.

4.2. Weakly Supervised Panoptic Segmentation

We first perform several ablation studies to evaluate the effectiveness of different design choices and parameter settings. All ablation studies are conducted on the PASCAL VOC 2012 panoptic segmentation as described above. Here, we use ResNet18-WS [90] as the backbone to save
Table 4: Comparison with the state-of-the-art methods on PASCAL VOC 2012 panoptic segmentation. The terms $\mathcal{M}$, $\mathcal{B}$ and $\mathcal{T}$ denote pixel-level, bounding-box-level and image-level labels, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervision</th>
<th>Backbone</th>
<th>PQ</th>
<th>SQ</th>
<th>RQ</th>
<th>PQ$	ext{Th}$</th>
<th>SQ$	ext{Th}$</th>
<th>RQ$	ext{Th}$</th>
<th>PQ$	ext{St}$</th>
<th>SQ$	ext{St}$</th>
<th>RQ$	ext{St}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeeperLab [82]</td>
<td>$\mathcal{M}$</td>
<td>Xception-71</td>
<td>67.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Panoptic FPN [83]</td>
<td>$\mathcal{M}$</td>
<td>ResNet50</td>
<td>65.7</td>
<td>84.3</td>
<td>77.6</td>
<td>64.5</td>
<td>83.9</td>
<td>76.5</td>
<td>90.8</td>
<td>92.5</td>
<td>98.1</td>
</tr>
<tr>
<td>Li et al. [2]</td>
<td>$\mathcal{B} + \mathcal{T}$</td>
<td>ResNet101</td>
<td>59.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Combination [47, 58]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>37.1</td>
<td>69.8</td>
<td>49.5</td>
<td>35.5</td>
<td>70.5</td>
<td>47.2</td>
<td>74.2</td>
<td>82.6</td>
<td>86.3</td>
</tr>
<tr>
<td>JTSM</td>
<td>$\mathcal{T}$</td>
<td>ResNet18-WS</td>
<td>39.0</td>
<td>74.4</td>
<td>51.5</td>
<td>37.1</td>
<td>73.9</td>
<td>49.5</td>
<td>77.7</td>
<td>85.1</td>
<td>91.2</td>
</tr>
</tbody>
</table>

Table 5: Comparison with the state-of-the-art methods on MS COCO panoptic segmentation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervision</th>
<th>Backbone</th>
<th>PQ</th>
<th>SQ</th>
<th>RQ</th>
<th>PQ$	ext{Th}$</th>
<th>SQ$	ext{Th}$</th>
<th>RQ$	ext{Th}$</th>
<th>PQ$	ext{St}$</th>
<th>SQ$	ext{St}$</th>
<th>RQ$	ext{St}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panoptic FPN [83]</td>
<td>$\mathcal{M}$</td>
<td>ResNet50</td>
<td>39.0</td>
<td>-</td>
<td>-</td>
<td>45.9</td>
<td>-</td>
<td>-</td>
<td>28.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>JTSM</td>
<td>$\mathcal{T}$</td>
<td>ResNet18-WS</td>
<td>5.3</td>
<td>30.8</td>
<td>7.8</td>
<td>8.4</td>
<td>46.6</td>
<td>11.4</td>
<td>0.7</td>
<td>6.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 6: Comparison with the state-of-the-art methods on PASCAL VOC 2012 instance segmentation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervision</th>
<th>Backbone</th>
<th>mAP$_{0.50}$</th>
<th>mAP$_{0.75}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask R-CNN [81]</td>
<td>$\mathcal{M}$</td>
<td>ResNet101</td>
<td>67.9</td>
<td>44.9</td>
</tr>
<tr>
<td>PRM [44]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>26.8</td>
<td>9.0</td>
</tr>
<tr>
<td>IAM [45]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>28.8</td>
<td>11.9</td>
</tr>
<tr>
<td>IRNet [47]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>46.7</td>
<td>-</td>
</tr>
<tr>
<td>Label-IPinNet [51]</td>
<td>$\mathcal{T}$</td>
<td>VGG16</td>
<td>30.2</td>
<td>12.9</td>
</tr>
<tr>
<td>WISE [46]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>41.7</td>
<td>23.7</td>
</tr>
<tr>
<td>Kim et al. [52]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>35.7</td>
<td>5.8</td>
</tr>
<tr>
<td>Ann et al. [42]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>50.9</td>
<td>28.5</td>
</tr>
<tr>
<td>LIID [50]</td>
<td>$\mathcal{T}$</td>
<td>ResNet50</td>
<td>48.4</td>
<td>24.9</td>
</tr>
<tr>
<td>JTSM</td>
<td>$\mathcal{T}$</td>
<td>ResNet18-WS</td>
<td>44.2</td>
<td>12.0</td>
</tr>
</tbody>
</table>

time if not mentioned. When tuning each group of hyperparameters, other parameters are kept as default.

The number $n^r$ of panoptic refinement heads. The panoptic refinement heads output final mining scores for segmentation during testing, which heavily influence the performance of instance and semantic segmentation. The hyperparameter $n^r$ in Equ. 4 controls the number of panoptic refinement branches. Different settings and corresponding results of $n^r$ are displayed in Tab. 3. When we have $n^r = 0$, the second term of loss function $L_{II}$ in Equ. 4 are omitted. We can see that the results of this setting are worse than using panoptic refinement branches, demonstrating that the panoptic refinement is very helpful for segmentation predictions. When $n^r \geq 4$, the performance gains are margin. We use 4 as the default values for $n^r$.

The shape-invariant vs. shape-interpolation MoIPool. We first use the traditional RoIPool [33] and RoIAlign [81] methods to analyze how performance varies with different proposal pooling methods. As shown in Tab. 1, traditional methods are unable to handle stuff well. As stuff content often has large outline and scale variance, which may also contain other stuff and thing object. Thus, it requires to pooling methods to compute pixel-accurate feature maps of arbitrary-shape regions. The proposed MoIPool achieves large performance gains compared to RoIPool and RoIAlign, as MoIPool only utilizes the features within segment proposals. We also find that shape-invariant MoIPool has superior performance compared to shape-interpolation version. As shape-invariant MoIPool maintains accurate contour information of segment proposals.

The instance and semantic segmentation refinement. We continue by evaluating the effect of segmentation refinement. As shown in Tab. 2, segmentation refinement improve overall performance with large gains. As the quality of original mined masks heavily relies on segment proposals, while the segmentation refinement leverages self-training to improve predicted masks. We observe that the performance can increase significantly with the guidance of bottom-up evidence. It demonstrates that the bottom-up evidence is positively correlated to object segmentation.

With the above ablation study, we perform panoptic segmentation on the PASCAL VOC 2012 and MS COCO with various ResNet backbones. To the best of our knowledge, this is the first work reporting results for image-level supervised panoptic segmentation. Inspired by fully supervised panoptic segmentation, we construct a strong baseline for WSPS, which combines the output of independent WSIS and WSSS tasks via a series of post-processing steps [3] that merge their outputs. Specifically, we use the results from a combination of WSIS algorithm, IRNet [47], and WSSS algorithm, AffinityNet [58]. Note that both IRNet and AffinityNet are competitive approaches in their target tasks. For PASCAL VOC that has only one stuff category, we compute all thing segmentation and treat the rest regions as stuff segmentation. Tab. 4 and 5 show that JTSM significantly outperforms the strong baseline models that use the same setting, i.e., using image-level labels only for model training. The performance improvement for stuff, e.g., PQ$^\text{St}$, shows the validity of joint category mining, while the improvement for thing, e.g., PQ$^\text{Th}$, indicates the effectiveness of MoIPool.

4.3. Weakly Supervised Instance Segmentation

We also report instance segmentation performance in terms of $AP$ and compare with other WSIS methods on
Table 7: Comparison with the state-of-the-art methods on MS COCO instance segmentation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervision</th>
<th>Backbone</th>
<th>PASCAL VOC 2007 mAP (%)</th>
<th>PASCAL VOC 2012 mAP (%)</th>
<th>MS COCO Avg. Precision, IoU: 0.5 0.5 0.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask R-CNN [6]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS-JDS [15]</td>
<td></td>
<td>VGG16</td>
<td>6.1</td>
<td>11.7</td>
<td>5.5</td>
</tr>
<tr>
<td>JTSM</td>
<td></td>
<td>ResNet18-WS</td>
<td>6.1</td>
<td>12.1</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Table 8: Comparison with the state-of-the-art methods on PASCAL VOC 2007, 2012 and MS COCO object detection.

<table>
<thead>
<tr>
<th>Method</th>
<th>Supervision</th>
<th>Backbone</th>
<th>PASCAL VOC 2007 mAP (%)</th>
<th>PASCAL VOC 2012 mAP (%)</th>
<th>MS COCO Avg. Precision, IoU: 0.5 0.5 0.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster RCNN [8]</td>
<td></td>
<td>VGG16</td>
<td>69.9</td>
<td>67.0</td>
<td>21.2</td>
</tr>
<tr>
<td>WSDDN [8]</td>
<td></td>
<td>VGG16</td>
<td>34.8</td>
<td>53.5</td>
<td>9.5</td>
</tr>
<tr>
<td>OPG [10]</td>
<td></td>
<td>VGG16</td>
<td>28.8</td>
<td>43.5</td>
<td>19.2</td>
</tr>
<tr>
<td>CSC C5 [11]</td>
<td></td>
<td>VGG16</td>
<td>43.0</td>
<td>62.2</td>
<td>38.1</td>
</tr>
<tr>
<td>WS-JDS [15]</td>
<td></td>
<td>VGG16</td>
<td>45.6</td>
<td>64.5</td>
<td>50.5</td>
</tr>
<tr>
<td>OICR [18]</td>
<td></td>
<td>VGG16</td>
<td>41.2</td>
<td>60.6</td>
<td>50.5</td>
</tr>
<tr>
<td>MELM [28]</td>
<td></td>
<td>VGG16</td>
<td>47.3</td>
<td>61.4</td>
<td>50.5</td>
</tr>
<tr>
<td>Kosugi et al. [21]</td>
<td></td>
<td>VGG16</td>
<td>47.6</td>
<td>66.7</td>
<td>50.5</td>
</tr>
<tr>
<td>C-MIL [29]</td>
<td></td>
<td>VGG16</td>
<td>50.5</td>
<td>65.0</td>
<td>50.5</td>
</tr>
<tr>
<td>Pred Net [30]</td>
<td></td>
<td>VGG16</td>
<td>52.9</td>
<td>70.9</td>
<td>57.4</td>
</tr>
<tr>
<td>WSOD² [33]</td>
<td></td>
<td>VGG16</td>
<td>53.6</td>
<td>69.5</td>
<td>57.4</td>
</tr>
<tr>
<td>Yang et al. [19]</td>
<td></td>
<td>VGG16</td>
<td>48.6</td>
<td>85.6</td>
<td>57.4</td>
</tr>
<tr>
<td>C-MIDN [14]</td>
<td></td>
<td>VGG16</td>
<td>52.6</td>
<td>68.7</td>
<td>57.4</td>
</tr>
<tr>
<td>Ren et al. [24]</td>
<td></td>
<td>VGG16</td>
<td>54.9</td>
<td>68.8</td>
<td>57.4</td>
</tr>
<tr>
<td>UWSD [26]</td>
<td></td>
<td>ResNet18-WS</td>
<td>45.0</td>
<td>63.8</td>
<td>57.4</td>
</tr>
<tr>
<td>JTSM</td>
<td></td>
<td>ResNet18-WS</td>
<td>53.4</td>
<td>71.4</td>
<td>57.4</td>
</tr>
</tbody>
</table>

PASCAL VOC 2012. In details, JTSM only mines thing categories and ignores the segmantic branch. As shown in Tab. 6 and 7, our JTSM largely outperforms previous state-of-the-art that also uses image-level supervision. Some previous methods achieved high performance, thanks to the specially designed inter-pixel relation module [47], graph partition algorithm [50], salient detector [49], fully-supervised model retraining [46, 47]. Unlike previous methods [50, 47, 42], JTSM is end-to-end trainable. When ResNet18-WS is used as backbone, JTSM achieves comparable performance with previous state-of-the-art methods.

4.4. Weakly Supervised Object Detection

We evaluate the detection performance of the proposed JTSM on all three datasets, in which we only used image-level labels of thing categories. We also remove the segmentation refinement branch, as panoptic mining branch already outputs detection results. Comparisons with recent state-of-the-art methods are listed in Tab. 8. With ResNet18-WS backbone, JTSM reaches the state-of-the-art mAP of 53.4% and 51.5% on VOC 2007 and VOC 2012. JTSM produces 9.4% mAP and 21.3% mAP0.5 on MS-COCO.

Although JTSM is not specially designed for object detection, it shows surprising results and achieves state-of-the-art performance for many metrics. We attribute the performance gains to MoIPool, which enables to extract pixel-accurate feature maps of arbitrary-shape regions.

5. Conclusion

In this paper, we propose a Joint Thing-and-Stuff Mining (JTSM) framework to learn panoptic segmentation with only image-level labels for the first time. To achieve this goal, a novel mask of interest pooling (MoIPool) is proposed to extract pixel-accurate feature maps of arbitrary-shape regions, which outputs fix-size feature maps for all semantic categories with the same representational power. We further integrate the mined masks with bottom-up object evidence to improve spatial coherence and contour localization. Finally, additional instance and semantic segmentation are learned via self-train to refine panoptic segmentation. Experimental results on PASCAL VOC and MS COCO demonstrate the effectiveness of JTSM compared to strong baselines. As a by-product, JTSM achieves competitive results for weakly supervised object detection and instance segmentation.

6. Acknowledgment

This work is supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No.U1705262, No.62072386, No.62072387, No.62072389, No.62002305, No.61772443, No.61802324 and No.61702136), Guangdong Basic and Applied Basic Research Foundation (No.2019B1515120049) and Beijing Nova Program under Grant Z201100006820023.
References


[38] Linpu Fang, Hang Xu, Zhili Liu, Sarah Parisot, and Zhenguo Li. EHSOD: CAM-Guided End-to-end Hybrid-Supervised Object Detection with Cascade Refinement. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2


