# Label-PEnet: Sequential Label Propagation and Enhancement Networks for Weakly Supervised Instance Segmentation

Weifeng Ge[1,2,3], Sheng Guo[1,2], Weilin Huang[1,2]*, and Matthew R. Scott[1,2]
[1]Malong Technologies, Shenzhen, China
[2]Shenzhen Malong Artificial Intelligence Research Center, Shenzhen, China
[3]The University of Hong Kong
{terrencege,sheng,whuang,mscott}@malong.com

## Abstract

*Weakly-supervised instance segmentation aims to detect and segment object instances precisely, given image-level labels only. Unlike previous methods which are composed of multiple offline stages, we propose Sequential Label Propagation and Enhancement Networks (referred as Label-PEnet) that progressively transforms image-level labels to pixel-wise labels in a coarse-to-fine manner. We design four cascaded modules including multi-label classification, object detection, instance refinement and instance segmentation, which are implemented sequentially by sharing the same backbone. The cascaded pipeline is trained alternatively with a curriculum learning strategy that generalizes labels from high-level images to low-level pixels gradually with increasing accuracy. In addition, we design a proposal calibration module to explore the ability of classification networks to find key pixels that identify object parts, which serves as a post validation strategy running in the inverse order. We evaluate the efficiency of our Label-PEnet in mining instance masks on standard benchmarks: PASCAL VOC 2007 and 2012. Experimental results show that Label-PEnet outperforms the state-of-art algorithms by a clear margin, and obtains comparable performance even with fully supervised approaches.*

## 1. Introduction

Deep convolutional neural networks (CNNs) have made a series of breakthroughs in computer vision by, using large-scale manually-labeled data for training. By designing strong network architectures, CNNs can detect object locations and segment object instances precisely. However, the performance on object detection or segmentation will drop considerably due to lack of strong annotation provided at the object level or pixel level [27, 7, 12, 43], *i.e.* when there are only image-level labels available.

To investigate the ability of CNNs to estimate pixel-wise labels by given image-level supervision only, various weakly-supervised approaches have been developed for object detection or instance segmentation. A number of methods [4, 36, 37] exploit a bottom-up approach to group pixels into proposals, and then evaluate the proposals repetitively in an effort to search exact object locations. Several algorithms dissect the classification process of CNNs in a top-down [41, 24] or bottom-up manner [42], to generate seeds for instance segmentation [43]. There are also some hybrid approaches that combine both bottom-up and top-down cues [32, 12].

Existing weakly-supervised methods can achieve competitive results, but the performance is still significantly lower than that of fully-supervised counterparts. Although we can roughly identify an object using a classification network, it is particularly challenging to precisely infer pixel-wise labels from a classification model, even using multiple post-processing methods. This inspired us to re-think the ability of CNNs for various vision tasks, such as image classification, object detection and instance segmentation. We observed that full supervision with accurate annotations is the key to success. Therefore, the central issue for weakly-supervised detection and segmentation is to transfer image-level supervision to pixel-wise labels gradually and smoothly, in a coarse-to-fine manner by designing multiple cascaded modules.

The 2-D structure of convolutional kernels allows CNNs to grasp local information accurately, and enlarge the size of receptive fields gradually with the increase of convolutional layers, which enable the CNN model to memorize and classify objects accurately. Our goal is to enable CNNs to segment objects by just providing image-level labels. We design CNNs by introducing four new modules: (1) multi-label classification module, (2)

*Weilin Huang is the corresponding author.

object detection module, (3) instance refinement module, and (4) instance segmentation module, which are cascaded and implemented sequentially.

**Multi-Label Classification Module.** In this module, images are first partitioned into pieces, which are then grouped into regions to generate a set of object proposals. We employ unsupervised process, search [38] or edge box [44], where pixels are organized by low-level statistics for generating object candidates. Then a classification branch and a class-wise weight branch are incorporated to perform multi-label classification. We propose a proposal refinement module able to identify exact object locations and mine pixel-wise labels in object proposals.

**Object Detection Module.** The rough object locations generated are used to train a standard object detection with Faster-RCNN [30]. But they can be unstable with online training we implement. Thus we explore object scores generated from the classification module to guide the training of current object detection, and infer object locations with the model during online training. We further perform proposal calibration to improve detection accuracy, and identify pixels that belong to the corresponding objects.

**Instance Refinement Module.** With the generated object locations and instance masks, we perform instance segmentation using a standard Mask-RCNN [17]. However, current supervised information is still not strong enough, and we need to further explore object scores generated from the detection module to guide the training of current instance segmentation. Furthermore, a new instance branch is explored to perform instance segmentation, because the previous instance masks are generated based on individual samples, and the CNNs can summarize and gradually rectify these object masks with increasing accuracy when the masks are used as supervision.

**Instance Segmentation Module.** In this module, we obtain relatively strong supervision from the previous modules, which are used to guide the training of current instance segmentation, and generate final results.

The main contributions of this work are summarized as:

First, we introduce Sequential Label Propagation and Enhancement Networks (Label-PEnet) for weakly-supervised instance segmentation, which can be trained in an end-to-end manner. Our framework is composed of four modules that mine, summarize and rectify the appearance of objects repetitively. It is an important step forward in exploiting the ability of CNNs to recognize objects from image level to pixel level, and thus boost up the performance of weakly-supervised instance segmentation.

Second, we propose a proposal calibration module to uncover the classification process of CNNs, and then mine the pixel-wise labels from image level and object level. In this module, both top-down and bottom up methods are combined to identify object pixels with increased accuracy.

Third, to validate the effectiveness of the proposed Label-PEnet, we conduct experiments on standard benchmarks: PASCAL VOC 2007 and PASCAL VOC 2012. Experimental results show that Label-PEnet outperforms state-of-art approaches by a clear margin, and obtains comparable performance even compared with fully supervised methods.

## 2. Related Work

We briefly review related studies on weakly-supervised object detection and segmentation, along with recent neural attention methods and applications of curriculum learning.

**Weakly-Supervised Object Detection and Segmentation.** Weakly-supervised object detection and segmentation is very challenging but is important to image understanding, since it aims to locate and segment objects using image-level labels only [27, 7]. There are usually three kinds of methods: bottom-up manner, top-down manner, and the combination of two. For example, methods in [27, 10, 9] treat the weakly-supervised object localization as a multi-label classification problem, and locate objects by using specific pooling layers. On the other hand, approaches in [4, 36] extract and select object instances from images using selective search [38] or edge boxes [44], and solve the weakly-supervised detection problem with multi-instance learning [8]. The method in [43] finds peaks in the class activation map, and then propagate the peaks to find the corresponding object proposals generated by MCG [28]. Since there is no sufficient supervision to train an instance segmentation network with image-level labels, in this paper, we decompose instance segmentation task into multiple simpler problems, and utilize the ability of neural networks in identifying object pixels to solve them progressively.

**Neural Attention.** Neural attention aims to understand the classification process of deep neural networks, and learn the relationship between the pixels in the input image and the neural activations in convolutional layers. Recent effort has been made to explain how neural networks work [41, 2, 24]. In [24], Lapuschkin *et al.* extended a layer-wise relevance propagation (LRP) [1] to visualize inherent structured reasoning of deep neural networks. To identify the important regions producing final classification results, Zhang *et al.* [41] proposed a positive neural attention back-propagation scheme, called excitation back-propagation (Excitation BP). Other related methods include Grad-CAM [34] and network dissection [2]. Neural attention obtains pixel-wise class probabilities using image-level

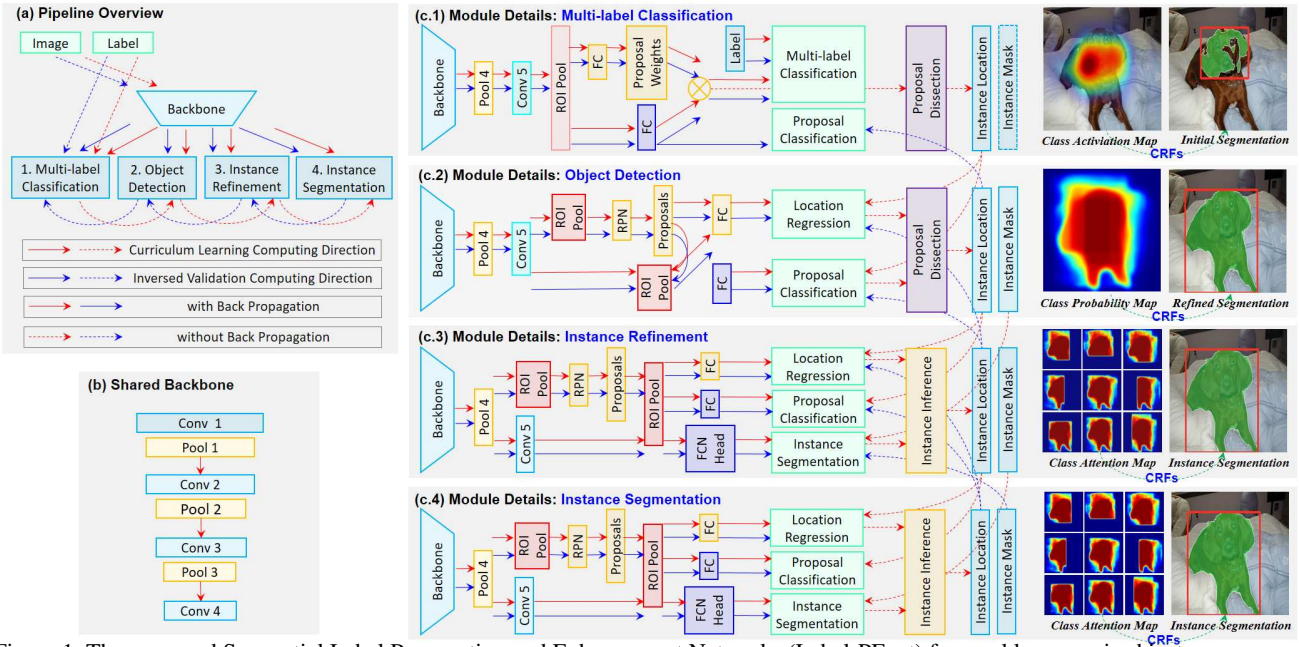



Figure 1. The proposed Sequential Label Propagation and Enhancement Networks (Label-PEnet) for weakly-supervised instance segmentation. (a) Overview: the training pipeline contains two different stages. One is curriculum learning stage which learns from image-level labels to pixel-wise labels. The other one learns in an inverse order to validate the results generated from the previous modules. (b) Shared backbone: the backbone is shared by all modules, and is fixed during the cascaded pre-training and recurrent mixed fine-tuning. (c) The details of different modules for multi-label classification, object detection, instance refinement, and instance segmentation.

labels in a top-down manner on a well trained network. In our pipeline, we propose a forward network that computes the pixel-wise class probability map for each individual proposal online. This allows us to transfer image-level labels to pixel-wise ones, providing richer supervision for subsequent object detection and instance segmentation.

**Curriculum Learning.** Curriculum learning [3] is set of machine learning methods that decompose a complicated learning task into multiple sub-tasks with gradually increasing learning difficulty. In [3], Yoshua *et al.* described the concept of curriculum learning, and used a toy classification problem to show the advantage of decomposing a complex problem into multiple simpler ones. Various machine learning algorithms [35, 14] follow a similar divide-and-conquer strategy in curriculum learning. Recently, Sheng *et al.* [15] proposed CurriculumNet for large-scale weakly-supervised image classification. CurriculumNet is able to learn high-performance CNNs from an image dataset constraining a large amount of noisy images and labels, which were collected rawly from the Internet without any human annotation [26]. In this paper, we adopt this strategy to decompose the instance segmentation problem into multi-label image classification, object detection and instance segmentation sequentially. All the learning tasks in these modules are relatively simple by using the training data with the refined supervision generated from the previous stage.

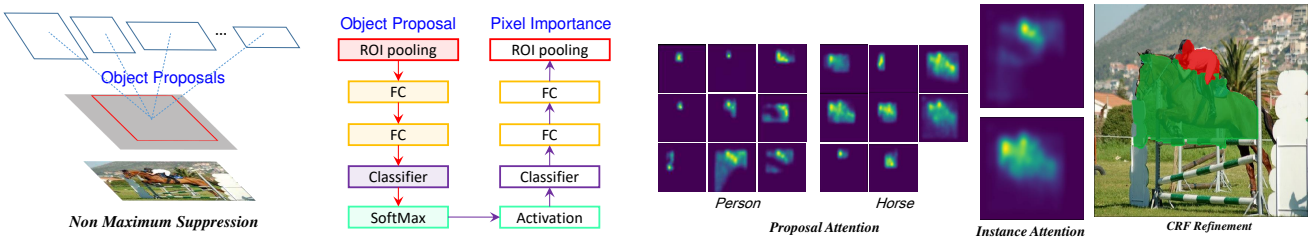
# 3. Label-PEnet: Sequential Label Propagation and Enhancement Networks

## 3.1. Preliminary and Overview

Given an image $\boldsymbol{I}$ associated with an image-level label $\boldsymbol{y}_I = [y^1, y^2, ..., y^{\mathcal{C}}]^T$, our goal is to estimate pixel-wise labels $\boldsymbol{Y}_I = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_P]^T$ for each object instance. $\mathcal{C}$ is the number of object classes, $P$ is the number of pixels in $\boldsymbol{I}$. $y^l$ is a binary value, where $y^l = 1$ means the image $\boldsymbol{I}$ contains the $l$-th object category, and otherwise, $y^l = 0$. The label of a pixel $p$ is denoted by a $\mathcal{C}$-dimensional binary vector $\boldsymbol{y}_p$. In this work, we propose a weakly-supervised learning approach for instance segmentation (Label-PEnet), which is inspired by the divide-and-conquer idea in curriculum learning [3]. This allows us to train our model with increasingly stronger supervision which is learned automatically by propagating object information from image level to pixel level via four cascaded modules: multi-label classification module, object detection module, instance refinement module, and instance segmentation module. The proposed Label-PEnet is described in Fig. 1.

## 3.2. Multiple Cascaded Modules

**Multi-Label Classification Module.** This module aims to generate a set of rough object proposals with corresponding class confident values and proposal weights, by just us-




(a) Candidate Object Proposals (b) Feed Forward Classification Dissection (c) Instance Attention Generation

Figure 2. The proposal calibration module. From left to right: (a) Candidate object proposals: All candidate object proposals suppressed by NMS are taken to generate the instance attention map. (b) Feed forward classification dissection module: The excitation back-propagation process is inversed into a feed forward manner to get online. (c) Instance attention generation: The attention of individual proposals are added to get the instance attention. CRF [23] is used to get the final segmentation results.

ing image-level category labels. To identify rough regions of objects, we exploit selective search [38] to generate a set of object proposals $\boldsymbol{R} = (R_1, R_2, ..., R_n)$. These object candidates are then used as input to our multi-label classification module for collecting more confident candidates, and learning to identify pixels which paly the key role in the classification task.

For an image $\boldsymbol{I}$ of $W \times H$, given a deep neural network $\phi_d(\cdot, \cdot; \theta)$ with a convolutional stride of $\lambda_s$, we have convolutional feature maps with a spatial size of $H/\lambda_s \times W/\lambda_s$ in the last convolutional layer. Then ROI pooling [13] is performed on the convolutional feature maps to compute the features for each object proposals in $\boldsymbol{R}$, resulting in $|\boldsymbol{R}|$ regional features for image $\boldsymbol{I}$. Two fully-connected layers are applied separately to the computed regional features, generating classification results, $\boldsymbol{x}^{c,1} \in \mathbb{R}^{|\boldsymbol{R}| \times \mathcal{C}}$, and weight vectors, $\boldsymbol{x}^{p,1} \in \mathbb{R}^{|\boldsymbol{R}| \times \mathcal{C}}$, for the $|\boldsymbol{R}|$ object proposals. The proposal weights indicate the contribution of each proposal to the $\mathcal{C}$ categories in image-level multi-label classification. A softmax function is applied to normalize the weights as,

$$\boldsymbol{w}_{ij}^1 = \frac{e^{\boldsymbol{x}_{ij}^{p,1}}}{\sum_{i=1}^{|\boldsymbol{R}|} e^{\boldsymbol{x}_{ij}^{p,1}}}. \tag{1}$$

where $\boldsymbol{w}_{ij}^{p,1}$ stands for the weight of the $i$-th proposal on the $j$-th class. We can have a normalized weight matrix $\boldsymbol{w}^1 \in \mathbb{R}^{|\boldsymbol{R}| \times \mathcal{C}}$. Then the final score for each proposal on different classes is calculated by taking an element-wise product, $\boldsymbol{x}^1 = \boldsymbol{x}^{c,1} \odot \boldsymbol{w}^{p,1}$, and the final image-level multi-label classification results are computed by summing over all the proposals associated to each class, $s_c^1 = \sum_{i=1}^{|\boldsymbol{R}|} \boldsymbol{x}_{ic}^1$, generating in a final score vector for the input image $\boldsymbol{I}$, $\boldsymbol{s}^1 = \left[s_1^1, s_2^1, ..., s_{\mathcal{C}}^1\right]$, indicating a confident value for each class. A probability vector $\hat{\boldsymbol{p}}^1 = \left[\hat{p}_1^1, \hat{p}_2^1, ..., \hat{p}_{\mathcal{C}}^1\right]$ can be computed by applying a softmax function to $\boldsymbol{s}^1$, and the loss function for image-level multi-label classification is,

$$\mathcal{L}_1(\boldsymbol{I}, \boldsymbol{y}_I) = -\sum_{k=1}^{\mathcal{C}} y^k \log \hat{p}_k^1. \tag{2}$$

**Proposal Calibration.** The generated object proposals, with their classification scores, $\boldsymbol{x}^{c,1}$, are further processed by proposal calibration which is a proposal refinement submodule that refines the generated proposals. The goal is to improve the prediction accuracy on object bounding boxes and segmentation masks, resulting in stronger and more accurate supervision for next modules.

Recent work of [41] introduces a new Excitation Back-Propagation (Excitation BP) able to generate a discriminative object-based attention map by using the predicted image-level class labels, which inspired us to compute an attention map for each proposal. For proposal calibration, we explore a same network architecture as the classification module. Specifically, given a proposal $R_i$, we apply a softmax function on its class prediction $\boldsymbol{x}_i^{c,1} \in \mathbb{R}^{\mathcal{C}}$ to have a normalized vector, $\boldsymbol{w}^{c,1}$, and predict an object class $c_i$ by using the highest value. Then we get a class activation vector, $\boldsymbol{a}_i^{c,1} \in \mathbb{R}^{\mathcal{C}}$, by setting all other elements to 0, except for the $c_i$-th one in $\boldsymbol{w}^{c,1}$. We perform the Excitation BP [41] in a feed forward manner from the classification layer to the ROI pooling layer by using the activation vector, and generate an attention map, $\boldsymbol{A}_i$, for each proposal, as shown in Fig. 2. Then for the label $c$ in the ground truth of image $\boldsymbol{I}$, we perform non-maximum suppression (NMS), and generate an object candidate $R^c$ with the highest confidence. For those proposals which are suppressed by $R^c$, we add their proposal attention maps to the corresponding locations in the image, and generate a class-specific attention map $\boldsymbol{A}^c$. Finally, we can compute a set of object attention maps of $\boldsymbol{A} = [\boldsymbol{A}^1, \boldsymbol{A}^2, ..., \boldsymbol{A}^{\mathcal{C}}] \in \mathbb{R}^{\mathcal{C} \times H \times W}$, with a background map, $\boldsymbol{A}_0 = max(0, 1 - \Sigma_{l=1}^{\mathcal{C}} y^l \boldsymbol{A}_l)$. We further perform a conditional random field (CRF [23]) to segment object regions more accurately from the corresponding attentional maps, resulting in a set of segmentation masks, $\boldsymbol{S}^1 \in \mathbb{R}^{\mathcal{K} \times H \times W}$, with corresponding object bounding boxes, $\boldsymbol{B}^1 \in \mathbb{R}^{\mathcal{K} \times 4}$. Meanwhile, for each pair of bounding box and segmentation mask, we simply use the classification score in $\boldsymbol{w}^{c,1}$ as a weight $\boldsymbol{W}^1 \in \mathbb{R}^{\mathcal{K}}$ to guide the training of next object detection module.

**Object Detection Module.** With the generated proposal bounding boxes $\boldsymbol{B}^1 \in \mathbb{R}^{\mathcal{K}\times 4}$ and the corresponding weights $\boldsymbol{W}^1 \in \mathbb{R}^{\mathcal{K}}$, we train a standard object detection model by using them as ground truth. The main difference is that we provide a learned weight for each generated proposal during training. By following Faster-RCNN [31], we sample positive and negative proposals around a ground truth bounding box, and each proposal sampled has a same weight with the corresponding ground truth. Then the optimization objective of region proposal network (RPN) is,

$$\begin{aligned} L\left(w_i, t_i\right)_{rpn} =& \frac{1}{N_{rpn}} \sum_i L_{obj}(w_i, w_i^*) \\ &+ \lambda \frac{1}{N_{rpn}} \sum_i w_i^* L_{reg}(t_i, t_i^*), \end{aligned} \tag{3}$$

where $N_{rpn}$ is the number of candidate proposals, $w_i$ is the predicted object score, $t_i$ is the predicted location offset, $w_i^*$ is the proposal weight, $t_i^*$ is the pseudo object location, $\lambda$ is a constant value. $L_{obj}$, $L_{cls}$ and $L_{reg}$ are the object or non-object loss, classification loss, and bounding boxes regression loss respectively. For the RCNN part, the optimization objective is computed as,

$$\begin{aligned} L\left(p_i, t_i\right)_{rcnn} =& \frac{1}{N_{rcnn}} \sum_i w_i^* L_{cls}(p_i, p_i^*) \\ &+ \lambda \frac{1}{N_{rcnn}} \sum_i w_i^* L_{reg}(t_i, t_i^*). \end{aligned} \tag{4}$$

where $p_i$ is the classification score, and $p_i^*$ indicates the object class. $N_{rcnn}$ is the number of proposals generated by RPN, and $L_{cls}$ is the classification loss. On the head of Faster-RCNN architecture, we perform proposal calibration sub-module to refine the object proposals, which is similar to that of multi-label classification module. This enables the model to generate dense proposal attention maps. In inference, multiple object candidates can be generated for multiple labels, which are different from the proposal calibration in classification module that outputs multiple candidates for each label. Finally, we can obtain multiple instance marks, $\boldsymbol{S}^2$, with corresponding object bounding boxes, $\boldsymbol{T}^2$, and weights, $\boldsymbol{W}^2 \in \mathbb{R}^{\mathcal{J}}$, where $\mathcal{J}$ is the number of object instances.

**Instance Refinement Module.** With the generated instance masks $\boldsymbol{S}^2$ and object bounding boxes $\boldsymbol{T}^2$, we can train an instance segmentation task having a joint detection branch and mask branch similar to that of Mask R-CNN [17]. In this module, we implement instance inference for dense pixel-wise prediction rather than proposal calibration, by following the feed forward inference as [17]. Object instances are learnt and modeled in the module by collecting part of the information hidden in the results generated from previous modules. We perform object instance segmentation with the learned weights $\boldsymbol{W}^2$, and our training process follows that of Mask-RCNN [17]. As in the proposal calibration, object masks affiliated with the predicted object location are summed together to generate an instance attention map. Similarly, we perform CRF [23] to obtain more accurate results of instance segmentation.

**Instance Segmentation Module.** In this module, image-level labels have been successfully transferred into dense pixel-wise labels. Then we perform standard instance segmentation in a fully supervised manner, by simply following the training strategies as implemented in the instance refinement module. Final results can be generated during inference.

### 3.3. Training with Label Propagation

To better train multiple sequential models and avoid local minima, we initialize the backbone network with an ImageNet pre-trained model. The training is implemented sequentially by using the output of previous module, with gradually enhanced supervision. We develop a two-stage training process containing cascaded pre-training and forward-backward learning with curriculum.

**Cascaded Pre-Training.** The parameters of backbone network are fixed during cascaded pre-training stage. We pre-train the four cascaded modules in sequence, from multi-label classification to instance segmentation. When the training of current module is converged, the model outputs are regularized and refined, and then are used as supervision of the next module. By this way, we decompose a weakly-supervised instance segmentation task into four sequential sub-tasks where image-level supervision is propagated gradually and efficiently.

**Forward-Backward Learning with Curriculum** Training four sequential models is difficult. The network will get into local minima easily with sequential label propagation. To overcome this problem, we propose a forward-backward learning method by leveraging curriculum learning. It contains two sub-stages: a forward curriculum learning phase and backward validation phase, as shown in Fig. 1. In the forward curriculum learning, the four modules are trained in one direction where the supervised information is enhanced gradually. While in the backward validation, training is performed in an inverse order. The backward validation starts from instance segmentation module, where we just perform inference at the module, and generate object locations and instance masks for instance refinement module. Then the instance refinement module is trained in a fully supervised manner, and provide object locations for object detection module. In multi-label classification module, we set the proposals, which have an overlap of $> \beta$ (= 0.5) with the objects detected by the detection module, with a label of

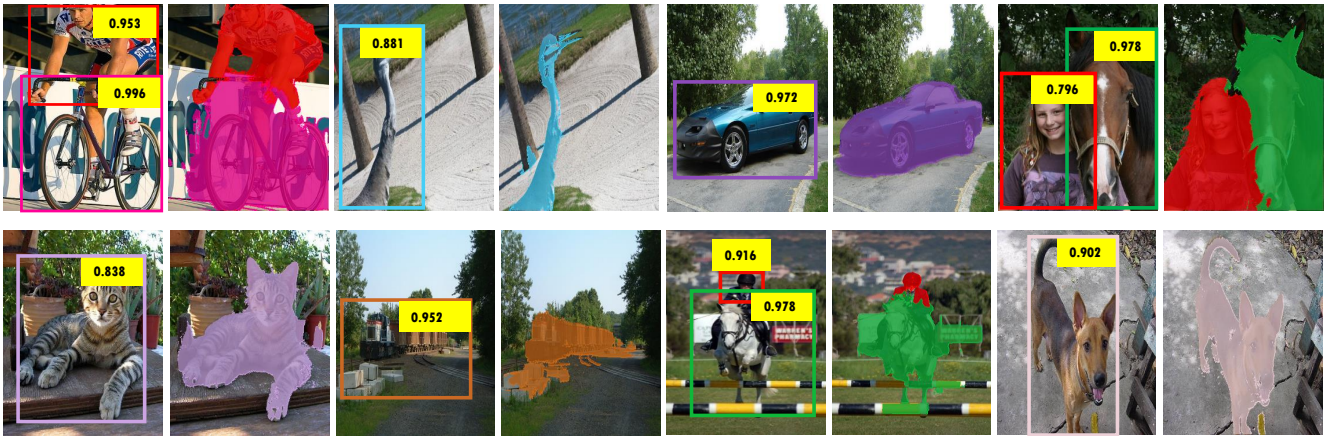



Figure 3. Instance detection and segmentation results on Pascal VOC 2012 (the first row) and Pascal VOC 2007 (the second row). The proposals with the highest confidence are selected and visualized. The segmentation results are post-processed by CRF [23].

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OM+MIL+FRCNN[25] | 54.5 | 47.4 | 41.3 | 20.8 | 17.7 | 51.9 | 63.5 | 46.1 | 21.8 | 57.1 | 22.1 | 34.4 | 50.5 | 61.8 | 16.2 | 29.9 | 40.7 | 15.9 | 55.3 | 40.2 | 39.5 |
| HCP+DSD+OSSH3[20] | 54.2 | 52.0 | 35.2 | 25.9 | 15.0 | 59.6 | 67.9 | **58.7** | 10.1 | **67.4** | 27.3 | 37.8 | 54.8 | **67.3** | 5.1 | 19.7 | 52.6 | 43.5 | 56.9 | 62.5 | 43.7 |
| OICR-Ens+FRCNN[36] | **65.5** | 67.2 | 47.2 | 21.6 | 22.1 | 68.0 | **68.5** | 35.9 | 5.7 | 63.1 | **49.5** | 30.3 | 64.7 | 66.1 | 13.0 | 25.6 | 50.0 | 57.1 | 60.2 | 59.0 | 47.0 |
| MEFF+FRCNN[12] | 64.3 | 68.0 | **56.2** | **36.4** | 23.1 | 68.5 | 67.2 | 64.9 | 7.1 | 54.1 | 47.0 | **57.0** | 69.3 | 65.4 | **20.8** | 23.2 | 50.7 | 59.6 | **65.2** | 57.0 | 51.2 |
| Multi-label Cls Module† | 41.2 | 42.0 | 6.5 | 17.1 | 7.1 | 54.1 | 40.5 | 8.5 | 17.3 | 33.0 | 13.2 | 10.3 | 24.4 | 54.0 | 5.5 | 7.5 | 20.0 | 39.2 | 49.9 | 47.3 | 26.9 |
| Object Det Module† | 49.1 | 61.3 | 24.8 | 15.9 | 46.9 | 58.9 | 25.3 | 17.7 | 23.3 | 41.8 | 28.9 | 42.4 | 67.1 | 25.3 | 6.7 | 50.4 | 40.9 | 62.4 | 50.4 | 42.3 | 39.1 |
| Instance Ref Module† | 62.3 | 68.3 | 47.2 | 27.9 | 53.8 | 69.1 | 39.9 | 41.9 | 25.9 | 56.5 | 40.1 | 53.0 | 70.0 | 44.9 | 13.3 | 53.5 | 51.1 | 68.6 | 60.9 | 45.2 | 49.7 |
| Instance Seg Module† | 63.8 | 69.0 | 47.9 | 35.3 | **56.1** | 68.9 | 41.5 | 42.7 | 25.9 | 58.3 | 44.3 | 52.5 | 70.3 | 44.4 | 13.8 | **56.9** | 52.9 | 70.0 | 62.3 | 49.9 | 51.3 |
| Multi-label Cls Module‡ | 42.4 | 43.8 | 8.9 | 18.7 | 6.5 | 55.7 | 42.0 | 10.0 | 18.3 | 34.3 | 14.5 | 11.4 | 24.8 | 56.2 | 3.7 | 9.1 | 22.1 | 40.5 | 51.1 | 46.5 | 28.0 |
| Object Det Module‡ | 51.2 | 63.0 | 28.8 | 17.5 | 51.1 | 60.3 | 28.9 | 20.7 | 25.9 | 41.0 | 31.2 | 46.4 | 68.1 | 27.1 | 6.0 | 50.9 | 43.6 | 65.8 | 50.6 | 40.3 | 40.3 |
| Instance Ref Module‡ | 63.2 | 67.5 | 48.3 | 29.8 | 54.8 | 70.4 | 40.9 | 42.6 | **27.9** | 55.0 | 41.5 | 54.3 | 70.0 | 43.2 | 15.3 | 55.4 | 52.4 | 69.0 | 62.2 | 46.8 | 50.5 |
| Instance Seg Module‡ | 65.7 | **69.4** | 50.6 | 35.8 | 55.5 | **71.9** | 43.6 | 45.3 | 27.5 | 58.5 | 45.4 | 55.4 | **71.7** | 45.8 | 18.2 | 56.6 | **56.1** | **72.0** | 64.6 | 51.4 | **53.1** |

Table 1. Average precision (in %) of weakly-supervised methods on PASCAL VOC 2007 *detection test* set. † stands for the results of the cascaded pre-training. ‡ stands for the results of the recurrent mixed fine-tuning.

the corresponding objects or background. Then we perform single-label classification on these proposals, and at the same time, keep training multi-label classification task.

## 4. Experimental Results

Our methods were implemented using Caffe [19] and run on an NVIDIA TITAN RTX GPU with 24GB memory. The parameters of object detection and instance segmentation modules are the same with Faster R-CNN [30] and Mask R-CNN [17]. Several examples are illustrated in Fig. 3.

### 4.1. Network Structures

**Backbone Network.** The backbone network is based on VGG-16, where the layers after $relu4\_3$ are removed. As shown in Fig 1, only the first four convolutional stages are preserved. All the parameters are initialized from an ImageNet pre-trained model.

**Multi-label Classification Module.** Following the backbone network, the fifth convolution stage contains $conv5\_1$, $conv5\_2$, and $conv5\_3$. Dilations in these three layers are set to 2. The feature stride $\lambda_s$ at layer $relu5\_3$ is 8. A ROI pooling [13] is added to generate a set of $512 \times 7 \times 7$ feature volumes. Then there are $fc6$ and $fc7$ layers followed. The classification branch and the proposal weight branch are initialized randomly using a Gaussian initializer as in [18].

**Object Detection Module.** As in multi-label classification module, dilations in $conv5\_1$, $conv5\_2$, and $conv5\_3$ are set to 2. The RPN [30] contains three convolutional layers which are all initialized with Gaussian distributions with 0 mean and standard deviation of 0.01. It generates proposals where ROI pooling [13] is conducted on the feature maps $relu5\_3$. A proposal classification branch and a bounding box regression branch are presented by following two fully-connected layers $fc6$ and $fc7$.

**Instance Refinement Module and Instance Segmentation Module.** The two module have the same network architecture. They contain an object detection part and an instance segmentation part. The object detection part is similar to that in object detection module. The only difference is that the RPN and the subsequent ROI pooling take the feature maps of the layer $pool4$ as input (not $relu5\_3$). For the instance segmentation part, we adopt the atrous spatial pyramid pooling as that of DeepLab V3 [5] after layer $relu5\_3$. The dilations in our atrous spatial pyramid pooling layers are $[1, 2, 4, 6]$.

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OICR-VGG16[36] | 67.7 | 61.2 | 41.5 | 25.6 | 22.2 | 54.6 | 49.7 | 25.4 | 19.9 | 47.0 | 18.1 | 26.0 | 38.9 | 67.7 | 2.0 | 22.6 | 41.1 | 34.3 | 37.9 | 55.3 | 37.9 |
| WSDDN+context[7] | 64.0 | 54.9 | 36.4 | 8.1 | 12.6 | 53.1 | 40.5 | 28.4 | 6.6 | 35.3 | 34.4 | 49.1 | 42.6 | 62.4 | **19.8** | 15.2 | 27.0 | 33.1 | 33.0 | 50.0 | 35.3 |
| HCP+DSD+OSSH3+NR[20] | 60.8 | 54.2 | 34.1 | 14.9 | 13.1 | 54.3 | 53.4 | 58.6 | 3.7 | 53.1 | 8.3 | 43.4 | 49.8 | 69.2 | 4.1 | 17.5 | 43.8 | 25.6 | 55.0 | 50.1 | 38.3 |
| OICR-Ens+FRCNN[36] | **71.4** | **69.4** | 55.1 | 29.8 | 28.1 | 55.0 | 57.9 | 24.4 | 17.2 | **59.1** | 21.8 | 26.6 | 57.8 | 71.3 | 1.0 | 23.1 | **52.7** | 37.5 | 33.5 | **56.6** | 42.5 |
| MEFF+FRCNN[12] | 71.0 | 66.9 | **55.9** | **33.8** | 24.0 | 57.6 | **58.0** | **61.4** | 22.5 | 58.4 | 19.2 | **58.7** | 61.9 | **75.0** | 11.2 | 23.9 | 50.3 | 44.9 | 41.3 | 54.3 | 47.5 |
| Multi-label Cls Module‡ | 37.1 | 40.0 | 5.9 | 11.7 | 5.5 | 48.3 | 40.5 | 7.0 | 16.3 | 29.2 | 9.9 | 8.3 | 19.3 | 51.1 | 3.0 | 6.1 | 17.0 | 36.3 | 46.4 | 39.1 | 23.9 |
| Object Det Module‡ | 49.2 | 57.0 | 25.1 | 13.9 | 49.5 | 53.3 | 25.3 | 15.9 | 20.0 | 36.5 | 29.1 | 42.1 | 60.9 | 22.9 | 5.5 | 43.5 | 37.8 | 63.4 | 48.7 | 35.8 | 36.8 |
| Instance Ref Module‡ | 57.9 | 65.5 | 43.9 | 26.9 | 50.9 | 64.7 | 35.9 | 38.7 | 22.8 | 50.9 | 38.9 | 50.9 | 65.5 | 39.5 | 13.6 | 52.9 | 48.9 | 65.7 | 57.9 | 41.9 | 46.7 |
| Instance Seg Module‡ | 60.8 | 65.4 | 46.2 | 31.4 | **50.3** | **68.3** | 40.7 | 39.9 | **25.3** | 52.8 | **43.4** | 53.9 | **68.2** | 40.8 | 15.9 | **53.1** | 50.0 | **68.1** | **59.8** | 49.0 | **49.2** |

Table 2. Average precision (in %) of weakly-supervised methods on PASCAL VOC 2012 *detection test* set.

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mCorLoc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OICR-VGG16[36] | 81.7 | 80.4 | 48.7 | 49.5 | 32.8 | 81.7 | 85.4 | 40.1 | **40.6** | 79.5 | 35.7 | 33.7 | 60.5 | 88.8 | 21.8 | 57.9 | **76.3** | 59.9 | 75.3 | **81.4** | 60.6 |
| WSDDN-Ens[7] | 68.9 | 68.7 | 65.2 | 42.5 | 40.6 | 72.6 | 75.2 | 53.7 | 29.7 | 68.1 | 33.5 | 45.6 | 65.9 | 86.1 | 27.5 | 44.9 | 76.0 | 62.4 | 66.3 | 66.8 | 58.0 |
| OM+MIL+FRCNN[25] | 78.2 | 67.1 | 61.8 | 38.1 | 36.1 | 61.8 | 78.8 | 55.2 | 28.5 | 68.8 | 18.5 | 49.2 | 64.1 | 73.5 | 21.4 | 47.4 | 64.6 | 22.3 | 60.9 | 52.3 | 52.4 |
| HCP+DSD+OSSH3[20] | 72.2 | 55.3 | 53.0 | 27.8 | 35.2 | 68.6 | 81.9 | 60.7 | 11.6 | 71.6 | 29.7 | 54.3 | 64.3 | 88.2 | 22.2 | 53.7 | 72.2 | 52.6 | 68.9 | 74.4 | 54.9 |
| OICR-Ens+FRCNN[36] | 85.8 | **82.7** | 62.8 | 45.2 | **43.5** | **84.8** | **87.0** | 46.8 | 15.7 | 82.2 | **51.0** | 45.6 | 83.7 | **91.2** | 22.2 | **59.7** | 75.3 | 65.1 | 76.8 | 78.1 | 64.3 |
| MEFF+FRCNN[12] | 88.3 | 77.6 | 74.8 | 63.3 | 37.8 | 78.2 | 83.6 | 72.7 | 19.4 | 79.5 | 46.4 | **78.1** | 84.7 | 90.4 | 28.6 | 43.6 | 76.3 | 68.3 | **77.9** | 70.6 | 67.0 |
| Label-PEnet | **89.8** | 82.6 | **75.3** | **65.7** | 39.2 | 80.2 | 81.6 | **77.7** | 18.4 | **82.7** | 49.3 | 75.0 | **86.9** | 85.9 | **30.7** | 49.6 | 75.3 | **71.5** | 76.1 | 70.6 | **68.2** |

Table 3. CorLoc (in %) of weakly-supervised methods on PASCAL VOC 2007 *detection trainval* set.

### 4.2. Implementation Details

**Cascaded Pre-Training.** In the cascaded pre-training stage, we train the multi-label classification, object detection, instance refinement and instance segmentation in forward order but keep the parameters in the backbone network fixed. For data augmentation, we use five image scales (480, 576, 688, 864, 1024) (the shorter side is resized to one of these scales) and horizontal flip, and cap the longer side at 1,200. The mini-batch size for SGD is set to 2, and the learning rate is set to 0.001 in the first 40K iterations and then decrease to 0.0001 in the following 10K iterations. The weight decay is 0.0005, and the momenta is 0.9. These settings are used in all the four modules. We start training the next module only when the training of previous module is finished. Selective Search (SS) [38] is adopted in the multi-label classification module to generate about 1,600 object proposals per image. For the RPN in the object detection module and instance segmentation module, we follow [30] to use 3 scales and 3 aspect ratios, yielding $k = 9$ anchors at each sliding position. The sizes of the convolutional feature maps after ROI pooling in the detection branch and segmentation branch are $7 \times 7$ and $14 \times 14$, respectively.

**Forward-Backward Learning with Curriculum.** As shown in Fig 1, there are two training graphs: a curriculum learning graph and an inverse validation graph. In the recurrent mixed fine-tuning stage, we perform the forward curriculum training and backward validation training alternatively at each iteration. All layers with learnable parameters are trained in an end-to-end manner. The training starts from the cascaded pre-trained model. The learning rate are kept at 0.0001 in the following 80K iterations. During testing, we use the original size of input image.

### 4.3. Weakly Supervised Object Detection

**Datasets and Performance Measurements.** We evaluate the performance of the object detector in different modules in Section 3.2 on Pascal VOC 2007 and Pascal VOC 2012 [11]. Each of the two datasets is divided into train, val and test sets. The trainval sets (5011 images for 2007 and 11540 images for 2012) are used for training, and only image tags are used. In our experiments, only image-level labels are used, without any bounding boxes information or pixel-wise annotation. We test our model with two measurements: mAP and CorLoc. Following the standard Pascal VOC protocol, the mAP is used for testing our models on the test sets, and the correct localization (CorLoc) is used for measuring the object localization accuracy [6] on the trainval sets whose image tags are already used as training data.

**Result Comparison.** Object detection results measured by mAP on Pascal VOC 2007 test set (Table 1) and Pascal VOC 2012 test set (Table 2) are reported. Object localization results measured by CorLoc on Pascal VOC 2007 trainval set and Pascal VOC 2012 trainval set are presented in Table 3 and Table 4. On Pascal VOC 2007 test set, our method achieves the highest mAP (53.1%), with at least 1.9% higher than the latest state-of-the-art algorithms including MEFF [12], OICR[36] and HCP+DSD+OSSH3[20]. Our trained model also achieves the highest mAP (49.2%) among all weakly-supervised algorithms on Pascal VOC 2012 test set, with 1.7% higher than the latest result from [12]. When compare the object localization accuracy (CorLoc), our results are also very competitive among the state-of-art results. Our trained models on Pascal VOC 2007 trainval set and Pascal VOC 2012 trainval set achieve 68.2% and 71.3% respectively, which are 1.2% and 1.9% higher than the previous best results.

| method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mCorLoc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OICR-VGG16[36] | 86.2 | 84.2 | 68.7 | 55.4 | 46.5 | 82.8 | 74.9 | 32.2 | 46.7 | 82.8 | 42.9 | 41.0 | 68.1 | 89.6 | 9.2 | 53.9 | 81.0 | 52.9 | 59.5 | **83.2** | 62.1 |
| WSDDN+context[7] | 78.3 | 70.8 | 52.5 | 34.7 | 36.6 | 80.0 | 58.7 | 38.6 | 27.7 | 71.2 | 32.3 | 48.7 | 76.2 | 77.4 | 16.0 | 48.4 | 69.9 | 47.5 | 66.9 | 62.9 | 54.8 |
| HCP+DSD+OSSH3+NR[20] | 82.4 | 68.1 | 54.5 | 38.9 | 35.9 | 84.7 | 73.1 | 64.8 | 17.1 | 78.3 | 22.5 | 57.0 | 70.8 | 86.6 | 18.7 | 49.7 | 80.7 | 45.3 | **70.1** | 77.3 | 58.8 |
| OICR-Ens+FRCNN[36] | **89.3** | **86.3** | 75.2 | 57.9 | **53.5** | 84.0 | **79.5** | 35.2 | **47.2** | **87.4** | 43.4 | 43.8 | 77.0 | 91.0 | 10.4 | **60.7** | **86.8** | 55.7 | 62.0 | 84.7 | 65.6 |
| MEFF+FRCNN[12] | 88.0 | 81.6 | 75.8 | 60.9 | 46.2 | 85.3 | 75.3 | 76.5 | 47.2 | 85.4 | 47.7 | 74.3 | 87.8 | **91.4** | 21.6 | 55.3 | 77.9 | **68.8** | 64.9 | 75.0 | 69.4 |
| Label-PEnet | 89.1 | 84.3 | **78.8** | **63.2** | 47.9 | **88.7** | 76.8 | **77.2** | 46.3 | 87.2 | **50.4** | **78.9** | **91.8** | 90.1 | **25.7** | 56.3 | 78.5 | 66.3 | 69.9 | 78.3 | **71.3** |

Table 4. CorLoc (in %) of weakly-supervised methods on PASCAL VOC 2012 *detection trainval* set.

| method | bg | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SEC[22] | 83.5 | 56.4 | 28.5 | 64.1 | 23.6 | 46.5 | 70.6 | 58.5 | 71.3 | **23.2** | 54.0 | 28.0 | 68.1 | 62.1 | 70.0 | 55.0 | 38.4 | 58.0 | 39.9 | 38.4 | 48.3 | 51.7 |
| FCL[32] | 85.7 | 58.8 | 30.5 | 67.6 | 24.7 | 44.7 | 74.8 | 61.8 | 73.7 | 22.9 | 57.4 | 27.5 | 71.3 | 64.8 | **72.4** | 57.3 | 37.0 | 60.4 | 42.8 | 42.2 | **50.6** | 53.7 |
| TP-BM[21] | 83.4 | 62.2 | 26.4 | **71.8** | 18.2 | 49.5 | 66.5 | 63.8 | 73.4 | 19.0 | 56.6 | 35.7 | 69.3 | 61.3 | 71.7 | **69.2** | 39.1 | 66.3 | **44.8** | 35.9 | 45.5 | 53.8 |
| AE-PSL[40] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 55.7 |
| MEFF[12] | 86.6 | 72.0 | 30.6 | 68.0 | **44.8** | 46.2 | 73.4 | 56.6 | 73.0 | 18.9 | 63.3 | 32.0 | 70.1 | 72.2 | 68.2 | 56.1 | 34.5 | **67.5** | 29.6 | 60.2 | 43.6 | 55.6 |
| MCOF-VGG16[39] | 85.8 | **74.1** | 23.6 | 66.4 | 36.6 | **62.0** | **75.5** | **68.5** | **78.2** | 18.8 | **64.6** | 29.6 | **72.5** | 61.6 | 63.1 | 55.5 | 37.7 | 65.8 | 32.4 | **68.4** | 39.9 | 56.2 |
| Label-PEnet | **88.3** | 73.5 | **33.2** | 70.1 | 43.2 | 49.1 | 74.7 | 57.0 | 75.9 | 20.7 | 64.3 | 32.3 | 72.3 | **74.1** | 71.1 | 57.4 | 38.3 | 69.1 | 31.2 | 61.1 | 45.2 | **57.2** |

Table 5. Comparisons of weakly-supervised semantic segmentation methods on PASCAL VOC 2012 *segmentation test* set.

| method | $mAP^r_{0.25}$ | $mAP^r_{0.5}$ | $mAP^r_{0.75}$ | ABO |
|---|---|---|---|---|
| PRM-VGG16 [43] | - | 22.0 | - | - |
| PRM-ResNet50 [43] | 44.3 | 26.8 | 9.0 | 37.6 |
| Label-PEnet | **49.1** | **30.2** | **12.9** | **41.4** |

Table 6. Comparisons of weakly-supervised instance segmentation methods on Pascal VOC 2012 *validation* set.

### 4.4. Weakly-Supervised Semantic Segmentation

**Datasets and Performance Measurements.** The Pascal VOC 2012 dataset [11] serves as the most popular benchmark on weakly-supervised semantic segmentation. The dataset is consisted of 21 classes with 10,582 training images (the VOC 2012 training set and additional data annotated in [16]), 1,449 for validation and 1,456 for testing. Only image tags are used as training data in our experiments. We do not use any additional data annotated in[16]. We report results on the test sets in Table 5.

**Result Comparisons.** Table 5 lists the results of weakly-supervised semantic segmentation on Pascal VOC 2012. Our method achieves 57.2% mean IoU, and outperforms the previous state-of-art AE-SPL[40] and MCOF [39] by 1.6% and 1% respectively. Compared with the previous state-of-art algorithms, including AE-SPL[40], F-B [33], FCL [32], and SEC [22], our method decomposes the semantic segmentation problem into three different simpler tasks which allows us to propagate high-level image labels to pixel-wise labels gradually with enhanced accuracy.

### 4.5. Weakly-Supervised Instance Segmentation

**Datasets and Performance Measurements.** We follow the settings in [43] by using Pascal VOC 2012 dataset [11] as benchmark for weakly-supervised instance segmentation. Experimental results are evaluated with $mAP^r$ at IoU threshold 0.25, 0.5 and 0.75, and the Average Best Overlap (ABO) [29]. We report results on the test sets in Table 6.

**Result Comparisons.** We use the VGG16 network as the backbone, and report the performance on $mAP^r_{0.25}$, $mAP^r_{0.5}$, $mAP^r_{0.75}$ and ABO. While for the previous state-of-art, they report all the four metrics only with ResNet50. For VGG16, they report the $mAP^r_{0.5}$ with 22.0%. Our method outperforms PRM-VGG16 by 8.2% on $mAP^r_{0.5}$. Compared even with PRM-ResNet50, our method obtains better results at $mAP^r_{0.25}$ by 4.8%, $mAP^r_{0.5}$ by 3.4%, $mAP^r_{0.75}$ by 3.9% and ABO by 3.8%.

### 4.6. Effectiveness of Different Modules

In Table 2, we compare the performance of different modules on Pascal VOC 2007 detection test set. In the cascaded pre-training, the mAP of multi-label classification is only 26.9%. When we refine the object locations with the proposal dissection module, the object detection module gets 39.1% mAP. In the instance module, the object detection results outperform the object detection module by 10.6%. With the output of the instance refinement as the ground truth to guide the training, the final instance segmentation module achieves 51.3% mAP. This indicates that with the better detection results as the guidance, the object detection results can be improved significantly. When we perform the recurrent mixed fine-tuning, the detection results have 51.3% mAP which is 1.8% higher than that of the cascaded pre-training, and surpass previous methods.

## 5. Conclusions

We have presented a new pipeline for weakly-supervised instance segmentation by introducing four new modules implemented sequentially. We analyzed the classification process of CNNs, and infer object locations and pixel labels progressively. The performance of our method on object detection, semantic segmentation, and instance segmentation was evaluated on the PASCAL VOC 2007 and PASCAL VOC 2012 benchmarks, where our method outperformed current state-of-art methods in each task. In future work, we will pursue a simplification of the training steps to improve the efficiency of learning.

## References


[1] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6541–6549, 2017.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[4] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *International journal of computer vision*, 100(3):275–293, 2012.

[7] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. *arXiv preprint arXiv:1611.08258*, 2016.

[8] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997.

[9] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[10] Thibaut Durand, Nicolas Thome, and Matthieu Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4743–4752, 2016.

[11] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[12] Weifeng Ge, Sibei Yang, and Yizhou Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1277–1286, 2018.

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[14] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017.

[15] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[16] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 991–998. IEEE, 2011.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[20] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Two-phase learning for weakly supervised object localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[22] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711. Springer, 2016.

[23] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[24] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Muller, and Wojciech Samek. Analyzing classifiers: Fisher vectors and deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2912–2920, 2016.

[25] Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly supervised object localization with progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3520, 2016.

[26] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[27] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning

with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694, 2015.

[28] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):128–140, 2017.

[29] Jordi Pont-Tuset and Luc Van Gool. Boosting object proposals: From pascal to coco. In *Proceedings of the IEEE international conference on computer vision*, pages 1546–1554, 2015.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[32] Anirban Roy and Sinisa Todorovic. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3529–3538, 2017.

[33] Fatemehsadat Saleh, Mohammad Sadegh Ali Akbarian, Mathieu Salzmann, Lars Petersson, Stephen Gould, and Jose M Alvarez. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *European Conference on Computer Vision*, pages 413–432. Springer, 2016.

[34] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[35] Lei Sun, Qiang Huo, Wei Jia, and Kai Chen. A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906–2920, 2015.

[36] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[37] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018.

[38] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[39] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1354–1362, 2018.

[40] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[41] Jianming Zhang, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pages 543–559. Springer, 2016.

[42] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[43] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3800, 2018.

[44] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.