Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning

Weifeng Ge Sibei Yang Yizhou Yu
Department of Computer Science, The University of Hong Kong

Abstract

Supervised object detection and semantic segmentation require object or even pixel level annotations. When there exist image level labels only, it is challenging for weakly supervised algorithms to achieve accurate predictions. The accuracy achieved by top weakly supervised algorithms is still significantly lower than their fully supervised counterparts. In this paper, we propose a novel weakly supervised curriculum learning pipeline for multi-label object recognition, detection and semantic segmentation. In this pipeline, we first obtain intermediate object localization and pixel labeling results for the training images, and then use such results to train task-specific deep networks in a fully supervised manner. The entire process consists of four stages, including object localization in the training images, filtering and fusing object instances, pixel labeling for the training images, and task-specific network training. To obtain clean object instances in the training images, we propose a novel algorithm for filtering, fusing and classifying object instances collected from multiple solution mechanisms. In this algorithm, we incorporate both metric learning and density-based clustering to filter detected object instances. Experiments show that our weakly supervised pipeline achieves state-of-the-art results in multi-label image classification as well as weakly supervised object detection and very competitive results in weakly supervised semantic segmentation on MS-COCO, PASCAL VOC 2007 and PASCAL VOC 2012.

1. Introduction

Deep neural networks give rise to many breakthroughs in computer vision by usinging huge amounts of labeled training data. Supervised object detection and semantic segmentation require object or even pixel level annotations, which are much more labor-intensive to obtain than image level labels. On the other hand, when there exist image level labels only, due to incomplete annotations, it is very challenging to predict accurate object locations, pixel-wise labels, or even image level labels in multi-label image classification.

Given image level supervision only, researchers have proposed many weakly supervised algorithms for detecting objects and labeling pixels. These algorithms employ different mechanisms, including bottom-up, top-down [44, 23] and hybrid approaches [32], to dig out useful information. In bottom-up algorithms, pixels are usually grouped into many object proposals, which are further classified, and the classification results are merged to match groundtruth image labels. In top-down algorithms, images first go through a forward pass of a deep neural network, and the result is then propagated backward to discover which pixels actually contribute to the final result [44, 23]. There are also hybrid algorithms [32] that consider both bottom-up and top-down cues in their pipeline.

Although there exist many weakly supervised algorithms, the accuracy achieved by top weakly supervised algorithms is still significantly lower than their fully supervised counterparts. This is reflected in both the precision and recall of their results. In terms of precision, results from weakly supervised algorithms contain much more noise and outliers due to indirect and incomplete supervision. Likewise, such algorithms also achieve much lower recall because there is insufficient labeled information for them to learn comprehensive feature representations of target object categories. However, different types of weakly supervised algorithms may return different but complementary subsets of the ground truth.

These observations motivate an approach that first collect as many evidences and results as possible from multiple types of solution mechanisms, put them together, and then remove noise and outliers from the fused results using powerful filtering techniques. This is in contrast to deep neural networks trained from end to end. Although this approach needs to collect results from multiple separately trained networks, the filtered and fused evidences are eventually used for training a single network used for the testing stage. Therefore, the running time of the final network during the testing stage is still comparable to that of state-of-the-art end-to-end networks.

According to the above observations, we propose a weakly supervised curriculum learning pipeline for object recognition, detection and segmentation. At a high level, we obtain object localization and pixelwise semantic labeling results for the training images first using their image level labels, and then use such intermediate results to train object detection, semantic segmentation, and multi-label image classification networks in a fully supervised manner.

Since image level, object level and pixel level analysis has mutual dependencies, they are not performed independently but organized into a single pipeline with four stages. In the first stage, we collect object localization results in the training images from both bottom-up and top-down weakly supervised object detection algorithms. In the second stage, we incorporate both metric learning and density-based clustering to filter detected object instances. In this way, we obtain a relatively clean and complete set of object instances. Given these object instances, we further train a single-label object classifier, which is applied to all object instances to obtain their final class labels. Third, to obtain a relatively clean pixel-wise probability map for every class and every training image, we fuse the image level attention map, object level attention maps and an object detection heat map. The pixel-wise probability maps are used for training a fully convolutional network, which is applied to all training images to obtain their final pixel-wise label maps. Finally, the obtained object instances and pixel-wise label maps for all the training images are used for training deep networks for object detection and semantic segmentation respectively. To make pixel-wise label maps of the training images help multi-label image classification, we perform multi-task learning by training a single deep network with two branches, one for multi-label image classification and the other for pixel labeling. Experiments show that our weakly supervised curriculum learning system is capable of achieving state-of-the-art results in multi-label image classification as well as weakly supervised object detection and very competitive results in weakly supervised semantic segmentation on MS-COCO [26], PASCAL VOC 2007 and PASCAL VOC 2012 [12].

In summary, this paper has the following contributions.

- We introduce a novel weakly supervised pipeline for multi-label object recognition, detection and semantic segmentation. In this pipeline, we first obtain intermediate labeling results for the training images, and then use such results to train task-specific networks in a fully supervised manner.
- To localize object instances relatively accurately in the training images, we propose a novel algorithm for filtering, fusing and classifying object instances collected from multiple solution mechanisms. In this algorithm, we incorporate both metric learning and density-based clustering to filter detected object instances.
- To obtain a relatively clean pixel-wise probability map for every class and every training image, we propose an algorithm for fusing image level and object level attention maps with an object detection heat map. The fused maps are used for training a fully convolutional network for pixel labeling.

2. Related Work

Weakly Supervised Object Detection and Segmentation. Weakly supervised object detection and segmentation respectively locates and segments objects with image-level labels only [28, 7]. They are important for two reasons: first, learning complex visual concepts from image level labels is one of the key components in image understanding; second, fully supervised deep learning is too data hungry.

Methods in [28, 10, 9] treat the weakly supervised localization problem as an image classification problem, and obtain object locations in specific pooling layers of their networks. Methods in [4, 38] extract object instances from images using selective search [40] or edge boxes [48], convert the weakly supervised detection problem into a multi-instance learning problem [8]. The method in [8] at first learns object masks as in [10, 9], and then uses the E-M algorithm to force the network to learn object segmentation masks obtained at previous stages. Since it is very hard for a network to directly learn object locations and pixel labels without sufficient supervision, in this paper, we decompose object detection and pixel labeling into multiple easier problems, and solve them progressively in multiple stages.

Neural Attention. Many efforts [44, 2, 23] have been made to explain how neural networks work. The method in [23] extends layer-wise relevance propagation (LRP) [1] to comprehend inherent structured reasoning of deep neural networks. To further ignore the cluttered background, a positive neural attention back-propagation scheme, called excitation back-propagation (Excitation BP), is introduced in [44]. The method in [2] locates top activations in each convolutional map, and maps these top activation areas into the input image using bilinear interpolation.

In our pipeline, we adopt the excitation BP [44] to calculate pixel-wise class probabilities. However for images with multiple category labels, a deep neural network could fuse the activations of different categories in the same neurons. To solve this problem, we train a single-label object instance classification network and perform excitation BP in this network to obtain more accurate pixel level class probabilities.

Curriculum Learning. Curriculum learning [3] is part of the broad family of machine learning methods that starts with easier subtasks and gradually increases the difficulty level of the tasks. In [3], Yoshua *et al.* describe the concept of curriculum learning, and use a toy classification problem to show the advantage of decomposing a complex problem into several easier ones. In fact, the idea behind curriculum learning has been widely used before [3]. Hinton *et al.* [17] trained a deep neural network layer by layer using a restricted Boltzmann machine [36] to avoid the local min-

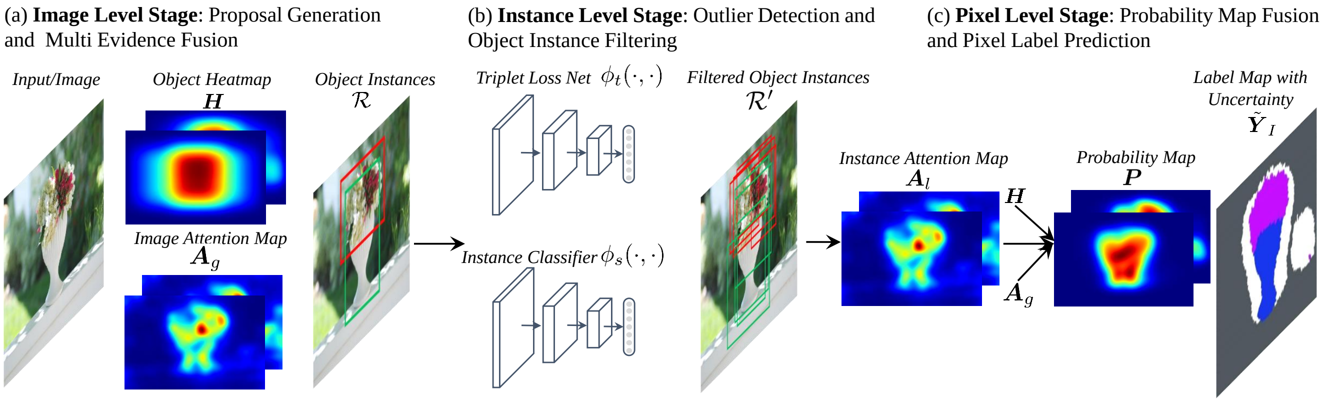

(a) **Image Level Stage**: Proposal Generation and Multi Evidence Fusion
(b) **Instance Level Stage**: Outlier Detection and Object Instance Filtering
(c) **Pixel Level Stage**: Probability Map Fusion and Pixel Label Prediction
Input/Image
Object Heatmap $\boldsymbol{H}$
Object Instances $\mathcal{R}$
Image Attention Map $\boldsymbol{A}_g$
Triplet Loss Net $\phi_t(\cdot,\cdot)$
Instance Classifier $\phi_s(\cdot,\cdot)$
Filtered Object Instances $\mathcal{R}'$
Instance Attention Map $\boldsymbol{A}_l$
$\boldsymbol{H}$
$\boldsymbol{A}_g$
Probability Map $\boldsymbol{P}$
Label Map with Uncertainty $\tilde{\boldsymbol{Y}}_I$

Figure 1. The proposed weakly supervised pipeline. From left to right: (a) Image level stage: fuse the object heatmaps $\boldsymbol{H}$ and the image attention map $\boldsymbol{A}_g$ to generate object instances $\mathcal{R}$ for the instance level stage, and provide these two maps for information fusion at the pixel level stage. (b) Instance level stage: perform triplet loss based metric learning and density based clustering for outlier detection, and train a single label instance classifier $\phi_s(\cdot,\cdot)$ for instance filtering. (c) Pixel level stage: integrate the object heatmaps $\boldsymbol{H}$, instance attention map $\boldsymbol{A}_l$, and image attention map $\boldsymbol{A}_g$ for pixel labeling with uncertainty.

ima in deep neural networks. Many machine learning algorithms [37, 14] follow a similar divide-and-conquer strategy in curriculum learning.

In this paper, we adopt this strategy to decompose the pixel labeling problem into image level learning, object instance level learning and pixel level learning. All the learning tasks in these three stages are relatively simple using the training data in the current stage and the output from the previous stage.

3. Weakly Supervised Curriculum Learning

3.1. Overview

Given an image $\boldsymbol{I}$ associated with an image level label vector $\boldsymbol{y}_I = [y^1, y^2, ..., y^{\mathcal{C}}]^T$, our weakly supervised curriculum learning aims to obtain pixel-wise labels $\boldsymbol{Y}_I = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_P]^T$, and then use these labels to assist weakly supervised object detection, semantic segmentation and multi-label image classification. Here $\mathcal{C}$ is the total number of object classes, P is the total number of pixels in $\boldsymbol{I}$, and y^l is binary. $y^l = 1$ means the l-th object class exists in $\boldsymbol{I}$, and $y^l = 0$ otherwise. The label of a pixel p is denoted by a $\mathcal{C}$-dimensional binary vector $\boldsymbol{y}_p$. The number of object classes existing in $\boldsymbol{I}$, which is the same as the number of positive components of $\boldsymbol{y}_I$ is denoted by K. Following the divide-and-conquer idea in curriculum learning [3], we decompose the pixel labeling task into three stages: the image level stage, the instance level stage and the pixel level stage.

3.2. Image Level Stage

The image level stage not only decomposes multi-label image classification into a set of single-label object instance classifications, but also provides an initial set of pixel-wise probability maps for the pixel level stage.

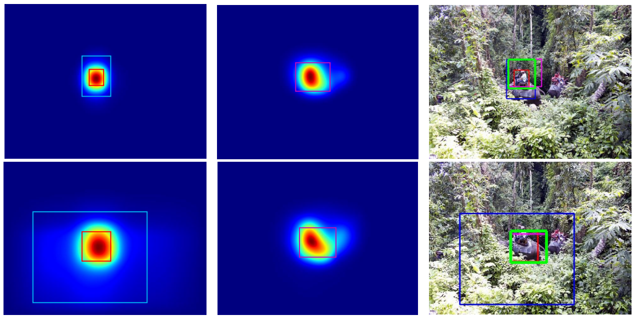

(a) Heatmap Proposals (b) Attention Proposals (c) Fused Proposals

Figure 2. (a) Proposals $\boldsymbol{R}^h$ and $\boldsymbol{R}^l$ generated from an object heatmap, (b) proposals generated from an attention map, (c) filtered proposals (green), heatmap proposals (red and blue), and attention proposals (purple).

Object Heatmaps. Unlike the fully supervised case, weakly supervised object detection produces object instances with higher uncertainty and also misses a higher percentage of true objects. To reduce the number of missing detections, we propose to compute an object heatmap $\boldsymbol{H}$ for every object class existing in the image.

For an image $\boldsymbol{I}$ with width W and height H, a dense set of object proposals $\boldsymbol{R} = (R_1, R_2, ..., R_n)$ are generated using sliding anchor windows. And the feature stride λ_s is set to 8. The number of locations in the input image where we can place anchor windows is $H/\lambda_s \times W/\lambda_s$. Denote the short side of image $\boldsymbol{I}$ by $L\rho$. Following the setting used for RPN [29], we let the anchor windows at a single location have four scales $[L\rho/8, L\rho/4, L\rho/2, L\rho]$ and three aspect ratios [0.5, 1, 2]. After proposals out of image borders have been removed, there are usually 12000 remaining proposals per image. Here we define a stack of object heatmaps $\boldsymbol{H} = [\boldsymbol{H}^1, \boldsymbol{H}^2, ..., \boldsymbol{H}^{\mathcal{C}}]$ as a $\mathcal{C} \times H \times W$ matrix, and all values are set to zero initially. The object detection and classification network $\phi_d(\cdot, \cdot)$ used here is the weakly supervised object testing net VGG-16 from [38]. For every proposal R_i in $\boldsymbol{R}$, its object class probability vector $\phi_d(\boldsymbol{I}, R_i)$ is added to all the pixels in the corresponding window in the heatmaps. Then every heatmap is normalized to [0, 1] as follows,

$$\boldsymbol{H}^c = (\boldsymbol{H}^c - min(\boldsymbol{H}^c))/max(\boldsymbol{H}^c),$$

where $\boldsymbol{H}^c$ is the heatmap for the c-th object class. Note that only the heatmaps for object classes existing in $\boldsymbol{I}$ are normalized. All the other heatmaps are ignored and set to zeros.

Multiple Evidence Fusion. The object heatmaps highlight the regions that may contain objects even when the level of supervision is very weak. However, since they are generated using sliding anchor windows at multiple scales and aspect ratios, they tend to highlight pixels near but outside true objects, as shown in Fig 2. Given an image classification network trained using the image level labels (here we use GoogleNet V1 [44]), neural attention calculates the contribution of every pixel to the final classification result. It tends to focus on the most influential regions but not necessarily the entire objects. Note that false positive regions may occur during excitation BP [44]. To obtain more accurate object instances, we integrate the top-down attention maps $\boldsymbol{A}_g = [\boldsymbol{A}_g^1, \boldsymbol{A}_g^2, ..., \boldsymbol{A}_g^C]$ with the object heatmaps $\boldsymbol{H} = [\boldsymbol{H}^1, \boldsymbol{H}^2, ..., \boldsymbol{H}^C]$.

For object classes existing in image $\boldsymbol{I}$, their corresponding heatmaps $\boldsymbol{H}$ and attention maps $\boldsymbol{A}_g$ are thresholded by distinct values. The heatmaps $\boldsymbol{H}$ are too smooth to indicate accurate object boundaries, but they provide important spatial priors to constrain object instances obtained from the attention maps. We assume that regions with a sufficiently high value in the object heatmaps should at least include parts of objects, and regions with sufficiently low values everywhere do not contain any objects. Following this assumption, we threshold the heatmaps with two values 0.65 and 0.1 to identify highly confident object proposals $\boldsymbol{R}^h = (R_1^h, R_2^h, ..., R_{N_h}^h)$ and relatively low confident object proposals $\boldsymbol{R}^l = (R_1^l, R_2^l, ..., R_{N_l}^l)$ after connected component extraction. Then the attention maps are thresholded by 0.5 to attention proposals $\boldsymbol{R}^a = (R_1^a, R_2^a, ..., R_{N_a}^a)$ as shown in Fig 2. N_h, N_l and N_a are the proposal numbers of $\boldsymbol{R}^h$, $\boldsymbol{R}^l$ and $\boldsymbol{R}^a$. All these object proposals have corresponding class labels. During the fusion, for each object class, the attention proposals $\boldsymbol{R}^a$ which cover more than 0.5 of any proposals in $\boldsymbol{R}^h$ are preserved. We denote these proposals by $\mathcal{R}$, each of which is modified slightly to completely enclose the corresponding proposal in $\boldsymbol{R}^h$ meanwhile be completely contained inside the corresponding proposal in $\boldsymbol{R}^l$ (Fig 2).

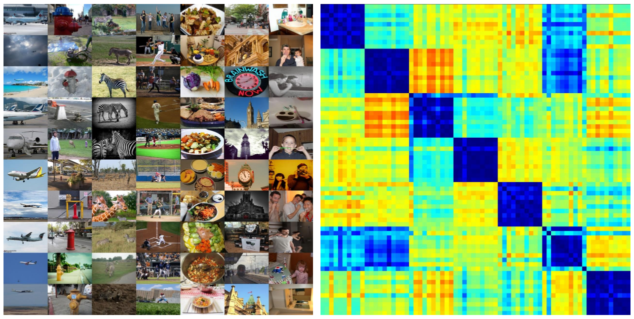

(a) Input Proposals (b) Distance Map

Figure 3. (a) Input proposals of the triplet-loss network, (b) distance map computed using features from the triplet-loss network.

3.3. Instance Level Stage

Since multiple object categories present in the same image make it hard for neural attention to obtain an accurate pixel-wise attention map for each class, we train a single-label object instance classification network and compute attention maps in this network to obtain more accurate pixel level class probabilities. The fused object instances from the image level stage are further filtered by metric learning and density-based clustering. The remaining labeled object proposals are used for training this object instance classifier, which can also be used to further remove remaining false positive object instances.

Metric Learning for Feature Embedding. Metric learning is popular in face recognition [34], person re-identification and object tracking [34, 46, 39]. It embeds an image $\boldsymbol{X}$ into a multi-dimensional feature space by associating this image with a fixed size vector, $\phi_t(\boldsymbol{X}, \cdot)$, in the feature space. This embedding makes similar images close to each other and dissimilar images apart in the feature space. Thus the similarity between two images can be measured by their distance in this space. The triplet-loss network $\phi_t(\cdot, \cdot)$ proposed in [34] has the additional property that it can well separate classes even when intra-class distances have large variations. When there exist training samples associated with incorrect class labels, the loss stays at a high value and the distances between correctly labeled and mislabeled samples remain very large even after the training process has run for a long time. Now let $\mathcal{R} = [R_1, R_2, ..., R_O]^T$ denote the fused object instances from all training images in the image level stage, and $\mathcal{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_O]^T$ are their labels. Here O is the total number of fused instances, and $\boldsymbol{y}_l$ is the label vector of instance R_l. We train a triplet-loss network $\phi_t(\cdot, \cdot)$ using GoogleNet V2 with BatchNorm as in [34]. Each mini-batch first chooses b object classes randomly, and then chooses a instances from these classes randomly. These instances are cropped out from the training images and fed into $\phi_t(\cdot, \cdot)$. Fig. 3 visualizes a mini-batch composition and the corresponding pairwise distances among instances.

Clustering for Outlier Removal. Clustering aims to remove outliers that are less similar to other object instances in the same class. Specifically, we perform density based clustering [31] to form a single cluster of normal instances within each object class independently, and instances outside this cluster are considered outliers. This is different from that in [31]. Let $\mathcal{R}^c$ denote instances in $\mathcal{R}$ with class label c, and N_c is the number of instances in $\mathcal{R}^c$. Calculate the pairwise distances $d(\cdot, \cdot)$ among these instances, and obtain the N_c by N_c distance matrix $\boldsymbol{D}^c$. For an instance $\mathcal{R}_n^c$, if its distance from another instance is less than λ_d (= 0.8), its density d_n^c is increased by 1. Rank these instances by their densities in a descending order, and choose the instance ranked at the top as the seed of the cluster. Then add instances to the cluster following the descending order if their distance to any element in the cluster is less than λ_d and their density is higher than $N_c/4$.

Instance Classifier for Re-labeling. Since metric learning and clustering screen object instances in an aggressive way and may heavily decrease their recall, we use the normal instances surviving the previous clustering step to train an instance classifier, which is in turn used to re-label all object proposals generated in the image level stage again. This is a single-label classification problem as each object instance is allowed a single label. GoogleNet V1 with the SoftMax loss serves as the classifier $\phi_s(\cdot, \cdot)$, and it is fine-tuned from the image level classifier. For every object proposal generated in the previous image level stage, if its label predicted by the instance classifier does not match its original label, it is labeled as an outlier and permanently discarded.

3.4. Pixel Level Stage

In previous stages, we have already built an image classifier, a weakly supervised object detector, and an object instance classifier. Each of these deep networks produces its own inference result from the input image. For example, the image classifier generates a global attention map, and the object detector generates the object heatmaps. In the pixel level stage, we still perform multi-evidence filtering and fusion to integrate the inference results from all these component networks to obtain the pixelwise probability map indicating potential object categories at every pixel. The global attention map $\boldsymbol{A}_g$ from the image classifier has a full knowledge about the objects in an image but sometimes only focuses on the most important object parts. The object instance classifier has a local view of each individual object. With the help of object-specific local attention maps generated from the instance classifier, we can avoid missing small objects.

Instance Attention Map. Here we define the instance attention map $\boldsymbol{A}_l$ as a $\mathcal{C} \times H \times W$ matrix, and all values are zero initially. For every surviving object instance from the instance level stage, the object instance classifier $\phi_s(\cdot, \cdot)$ is used to extract its local attention map, and add it to the corresponding region in the instance attention map $\boldsymbol{A}_l$. Normalize the range of $\boldsymbol{A}_l$ to [0, 1] as we did for object heatmaps.

Probability Map Integration. The final attention map $\boldsymbol{A}$ is obtained by calculating the element-wise maximum between the image attention map $\boldsymbol{A}_g$ and the instance attention map $\boldsymbol{A}_l$. That is, $\boldsymbol{A} = max(\boldsymbol{A}_l, \boldsymbol{A}_g)$. For both the heatmap $\boldsymbol{H}$ and the attention map $\boldsymbol{A}$, only the classes existing in the image are considered. The background maps of $\boldsymbol{A}$ and $\boldsymbol{H}$ are defined as follows,

$$\begin{aligned}\boldsymbol{A}_0 &= max(0, 1 - \Sigma_{l=1}^{\mathcal{C}} y^l \boldsymbol{A}_l),\\ \boldsymbol{H}_0 &= max(0, 1 - \Sigma_{l=1}^{\mathcal{C}} y^l \boldsymbol{H}_l).\end{aligned}$$

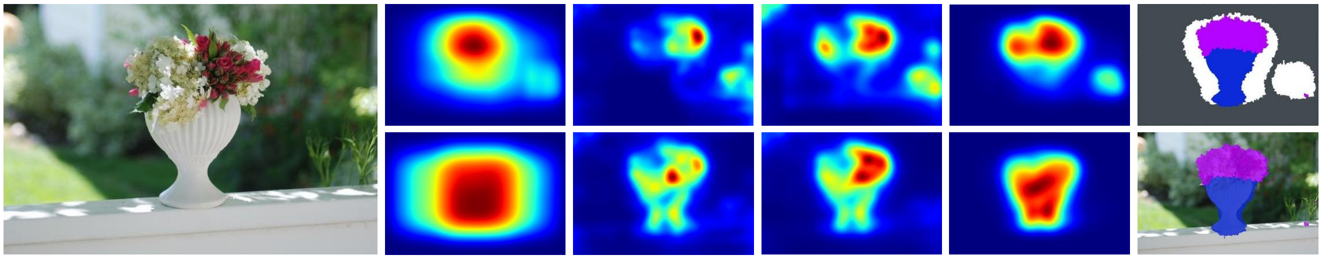
Now both $\boldsymbol{A}$ and $\boldsymbol{H}$ become $(\mathcal{C}+1) \times H \times W$ matrices. For the l-th channel, if $y^l = 0$, $\boldsymbol{A}_l = 0$ and $\boldsymbol{H}_l = 0$. Then we perform softmax on both maps along the channel dimension independently. The final probability map $\boldsymbol{P}$ is defined as the result of applying softmax to the element-wise product between $\boldsymbol{A}$ and $\boldsymbol{H}$ by treating $\boldsymbol{H}$ as a filter. That is, $\boldsymbol{P} = \text{softmax}(\boldsymbol{H} \odot \boldsymbol{A})$.

Pixel Labeling with Uncertainty. Pixel labels $\boldsymbol{Y}_I$ are initialized with the probability map $\boldsymbol{P}$. For every pixel p, if the maximum element in its label vector $\boldsymbol{y}_p$ is larger than a threshold (=0.8), we simply set the maximum element to 1 and other elements to 0; otherwise, the class label at p is uncertain. To inspect these uncertain pixels more carefully, we obtain additional evidence by computing their saliency scores $\boldsymbol{S}$ (normalized into $[0, 1]$) using an existing state-of-the-art salient object detection algorithm [25]. Given an uncertain pixel q with a high saliency score ($\boldsymbol{S}_q \geqslant 0.3$), if the maximum element in its label vector $\boldsymbol{y}_q$ is larger than a threshold (=0.6) and this element does not correspond to the background, we set the maximum element to 1 and other elements to 0. Given another uncertain pixel o with a low saliency score ($\boldsymbol{S}_o < 0.3$), if the maximum element in its label vector $\boldsymbol{y}_o$ corresponds to the background, we set the background element to 1 and other elements to 0.

4. Object Recognition, Detection and Segmentation

4.1. Semantic Segmentation

Given pixel-wise labels generated at the end of the pixel level stage for all training images, we train a fully convolutional network (FCN) similar to the network in [27] to perform semantic segmentation. Note that all pixels with uncertain class labels are excluded during training. In the prediction part, we adopt atrous spatial pyramid pooling as in [5]. The resulting trained network can be used for labeling all pixels in any testing image as well as pixels with uncertain labels in all training images.



(a) Input/Image (b) Object Heatmap (c) Image Attention (d) Instance Attention (e) Probability (e) Segmentation

Figure 4. The pixel labeling process in the pixel level stage. White pixels in the last column indicate pixels with uncertain labels.

4.2. Object Detection

Once all pixels with uncertain labels in the training images have been re-labeled using the above network for semantic segmentation, we generate object instances in these images by computing bounding boxes of connected pixels sharing the same semantic label. As in [38] and [24], we train fast RCNN [13] using these bounding boxes and their associated labels. Since the bounding boxes generated from the semantic label maps may contain noise, we filter them using our object instance classifier as in Section 3.3. VGG-16 is still the base network of our object detector, which is trained with five scales and flip as in [38].

4.3. Multi-label Classification

The main component in our multi-label classification network is the structure of ResNet-101 [16]. There are two branches after layer $res4b22\_relu$ of the main component, one branch for classification and the other for semantic segmentation. Both branches share the same structure after layer $res4b22\_relu$. Here we adopt multi-task learning to train both branches. The idea is using the training data for the segmentation branch to make the convolutional kernels in the main component more discriminative and powerful. This network architecture is shown in the supplemental materials. Layer $pool5$ of ResNet-101 in the classification branch is removed, and the output $\boldsymbol{X}(\in \mathbb{R}^{14\times14\times2048})$ of layer $res5c$ is a $14 \times 14 \times 2048$ matrix. $\boldsymbol{X}$ is directly fed into a $2048 \times 1 \times 1 \times C$ convolutional layer, and a classification map $\hat{\boldsymbol{Y}}_{cls}(\in \mathbb{R}^{14\times14\times\mathcal{C}})$ is obtained. We let the semantic label map $\hat{\boldsymbol{Y}}_{seg}(\in \mathbb{R}^{14\times14\times\mathcal{C}})$ play the role of an attention map $\hat{\boldsymbol{Y}}_{att}$ after the summation over each channel of the semantic label map is normalized to 1. The final image level probability vector $\hat{\boldsymbol{y}}$ is the result of spatial average pooling over the element-wise product between $\hat{\boldsymbol{Y}}_{cls}$ and $\hat{\boldsymbol{Y}}_{att}$. Here $\hat{\boldsymbol{Y}}_{att}$ is used to identify important image regions and assign them larger weights. At the end, the probability vector $\hat{\boldsymbol{y}}$ is fully connected to an output layer, which performs binary classification for each of the $\mathcal{C}$ classes. The cross-entropy loss is used for training the multi-label classification network. The segmentation branch uses atrous spatial pyramid pooling to perform semantic segmentation, and softmax is applied to enforce a single label per pixel.

5. Experimental Results

All our experiments are implemented using Caffe [18] and run on an NVIDIA TITAN X(Maxwell) GPU with 12GB memory. The hyper-parameters in Section 3 are set according to common sense and confirmed after we visually verify that the segmentation results on a few training samples are valid. The same parameter setting is used for all datasets and has not been tuned on any validation sets.

5.1. Semantic Segmentation

Datasets and performance measures. The Pascal VOC 2012 dataset [11] serves as a benchmark in most existing work on weakly-supervised semantic segmentation. It has 21 classes and 10582 training images (the VOC 2012 training set and additional data annotated in [15]), 1449 for validation and 1456 for testing. Only image tags are used as training data in our experiments. We report results on both the validation (supplemental materials) and test sets.

Implementation details. Our network is based on VGG-16. The layers after $relu5\_3$ and layer $pool4$ are removed. Dilations in layers $conv5\_1$, $conv5\_2$, and $conv5\_3$ are set to 2. The feature stride λ_s at layer $relu5\_3$ is 8. We add the atrous spatial pyramid pooling as in DeepLab V3 [5] after layer $relu5\_3$. The dilations in our atrous spatial pyramid pooling layers are $[1, 2, 4, 6]$. This FCN is implemented in py-faster-rcnn [30]. 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 1200. During testing, the original size of an input image is preserved. The network is fine-tuned from the pre-trained model for ImageNet in [35]. The learning rate γ is set to 0.001 in the first 20k iterations, and 0.0001 in the next 20k iterations. The weight decay is 0.0005, and the mini-batch size is 1. Post-processing using CRF [22] is added during testing.

Result comparison. We compare our method with existing state-of-the-art algorithms. Table 1 lists the results of weakly supervised semantic segmentation on Pascal VOC 2012. The proposed method achieves 55.6% mean IoU, comparable to the state of the art (AE-SPL [43]). Recent

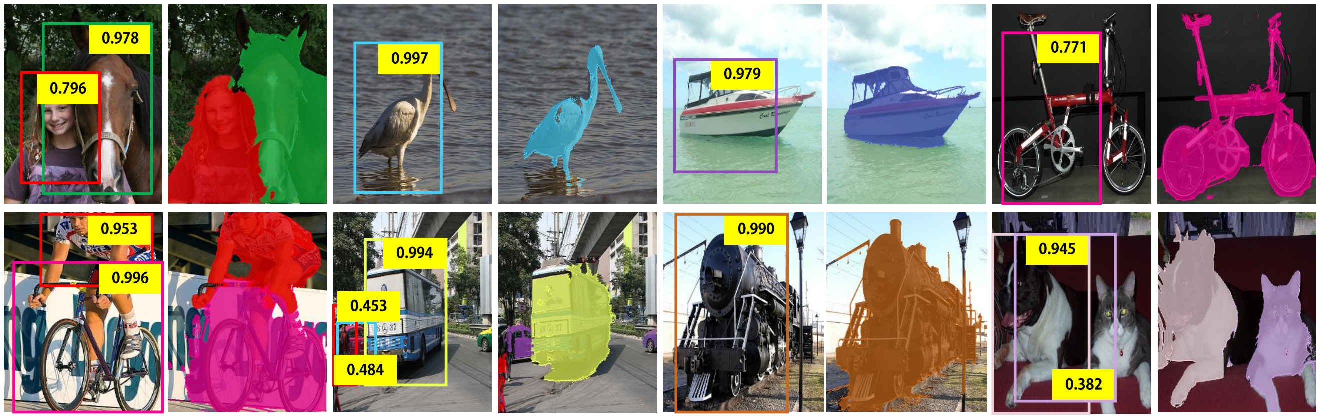


Figure 5. The detection and semantic segmentation results on Pascal VOC 2012 test set (the first row) and Pascal VOC 2007 test set (the second row). The detection results are gotten by select proposals with the highest confidence of every class. The semantic segmentation results are post-processed by CRF [22].

method	bg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
SEC[21]	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[20]	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[43]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	55.7
Ours+CRF	**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**

Table 1. Comparison among weakly supervised semantic segmentation methods on PASCAL VOC 2012 *segmentation test* set.

algorithms, including AE-PSL[**?**], F-B [33], FCL [32], and SEC [21], all conduct end-to-end training to learn object score maps. Our method demonstrates that if we filter and integrate multiple types of intermediate evidences at different granularities during weakly supervised training, the results become equally competitive or even better.

5.2. Object Detection

Datasets and performance measures. The performance of our object detector in Section 4.2 is evaluated on the popular Pascal VOC 2007 and Pascal VOC 2012 datasets [11]. Each of these 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. Two measures are used to test our model: mAP and CorLoc. According to the standard Pascal VOC protocol, the mean average precision (mAP) is used for testing our trained 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.

Implementation details. We use the code for py-faster-rcnn [30] to implement fast R-CNN [13]. The network is still VGG-16. The learning rate is set to 0.001 in the first 30k iterations, and 0.0001 in the next 10k iterations. The momentum and weight decay are set to 0.9 and 0.0005 respectively. We follow the same data augmentation setting in [38], use five image scales (480, 576, 688, 864, 1200) and horizontal flip, and cap the longer image side at 2000.

Result comparison. Object detection results on Pascal VOC 2007 test set (Table 2) and Pascal VOC 2012 test set (supplemental materials) are reported. Object localization results on Pascal VOC 2007 trainval set and Pascal VOC 2012 trainval set are also reported (supplemental material). On Pascal VOC 2012 test set, our algorithm achieves the highest mAP (47.5%), at least 5.0% higher than the latest state-of-the-art algorithms including OICR [38] and HCP+DSD+OSSH3[19]. Our trained model also achieves the highest mAP (51.2%) among all weakly supervised algorithms on Pascal VOC 2007 test set, 4.2% higher than the latest result from [38]. The object localization accuracy (CorLoc) of our trained model on Pascal VOC 2007 trainval set and Pascal VOC 2012 trainval set are respectively 67% and 69.4%, which are 2.7% and 3.8% higher than the previous best.

5.3. Multi-Label Classification

Dataset and performance measures. Microsoft COCO [26] is the most popular dataset in multi-label classification. MS-COCO was primarily built for object recognition tasks in the context of scene understanding. The training set is composed of 82081 images in 80 classes, on average 2.9 object labels per image. Since the groundtruth labels of the test set is not available, performance evaluation is conducted on the validation set with 40504 images. We train our models on the training set and test them on the validation set.

Performance measures for multi-label classification is

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[24]	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[19]	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[38]	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
Ours+FRCNN w/o clustering	66.7	61.8	55.3	41.8	6.7	61.2	62.5	**72.8**	12.7	46.2	40.9	**71.0**	67.3	64.7	**30.9**	16.7	42.6	56.0	65.0	26.5	48.5
Ours+FRCNN w/o uncertainty	66.8	63.4	54.5	42.2	5.8	60.5	58.3	67.8	7.8	46.1	40.3	71.0	68.2	62.6	30.7	16.5	41.1	55.2	66.8	25.2	47.5
Ours+FRCNN w/o instances	**67.7**	62.9	53.1	**44.4**	11.2	62.4	58.5	71.2	8.3	45.7	41.5	71.0	68.0	59.2	30.3	15.0	42.4	56.0	**67.2**	26.8	48.1
Ours+FRCNN	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**

Table 2. Average precision (in %) of weakly supervised methods on PASCAL VOC 2007 *detection test* set.

method	F1-C	P-C	R-C	F1-O	P-O	R-O	F1-C/top3	P-C/top3	R-C/top3	F1-O/top3	P-O/top3	R-O/top3
CNN-RNN[41]	-	-	-	-	-	-	60.4	66.0	55.6	67.8	69.2	66.4
RLSD[45]	-	-	-	-	-	-	62.0	67.6	57.2	66.5	70.1	63.4
RNN-Attention[42]	-	-	-	-	-	-	67.4	79.1	58.7	72.0	84.0	63.0
ResNet101-SRN[47]	70.0	**81.2**	63.3	75.0	84.1	67.7	66.3	**85.8**	57.5	72.1	88.1	61.1
ResNet101(448 × 448)(baseline)	72.8	73.8	**72.9**	76.3	77.5	**75.1**	69.5	78.3	**63.7**	73.1	83.8	**64.9**
Ours	**74.9**	80.4	70.2	**78.4**	**85.2**	72.5	**70.6**	84.5	62.2	**74.7**	**89.1**	64.3

Table 3. Performance comparison among multi-label classification methods on Microsoft COCO 2014 *validation* set.

quite different from those for single-label classification. Following [47, 42], we employ macro/micro precision, macro/micro recall, and macro/micro F1-measure to evaluate our trained models. For precision, recall and F1-measure, labels with confidence higher than 0.5 are considered positive. "P-C", "R-C" and "F1-C" represent the average per-class precision, recall and F1-measure while "P-O", "R-O" and "F1-O" represent the average overall precision, recall and F1-measure. These measures do not require a fixed number of labels per image. To compare with existing state-of-the-art algorithms, we also report the results of top-3 labels with confidence higher than 0.5 as in [42].

Implementation details. Our main network for multi-label classification is ResNet-101 as described earlier. The resolution of the input images is at 448×448. We first train a network with the classification branch only. As a common practice, a pre-trained model for ImageNet is fine-tuned with the learning rate γ set to 0.001 in the first 20k iterations, and 0.0001 in the next 20k iterations. The weight decay is 0.0005. Then we add the segmentation branch and train this new branch only by fixing all the layers before layer $res4b22\_relu$ and the classification branch. The learning rate is set to 0.001 in the frist 20k iterations, and 0.0001 in the next 20k iterations. At last, we train the entire network with both branches using the cross-entropy loss for multi-label classification for 30k iterations with a learning rate 0.0001 while still fixing the layers before layer $res4b22\_relu$.

Result comparison. In addition to our two-branch network, we also train a ResNet-101 classification network as our baseline. The multi-label classification performance of both networks on MS-COCO is reported in Table 3. Since the input resolution of our baseline is 448×448, in comparison to the latest work (ResNet101-SRN) [47], the performance of our baseline is slightly better. Specifically, the F1-C of our baseline is 72.8%, which is 2.8% higher than the F1-C of ResNet101-SRN. In comparison to the baseline, our two-branch network further achieves overall better performance. Specifically, the P-C of our two-branch network is 6.6% higher than the baseline, the R-C is 2.7% lower, and the F1-C is 2.1% higher. All F1-measures (F1-C, F1-O, F1-C/top3 and F1-O/top3) of our two-branch network are the highest among all state-of-the-art algorithms.

5.4. Ablation Study

We perform an ablation study on Pascal VOC 2007 detection test set by replacing or removing a single component in our pipeline every time. First, to verify the importance of object instances, we remove all steps related to object instances, including the entire instance level stage and the operations related to the instance attention map in the pixel level stage. The mAP is decreased by 3.1% as shown in Table 2. Second, the clustering and outlier detection step in the instance level stage is removed. We directly train an instance classifier using the object proposals from the image level stage. The mAP is decreased by 2.7%. Third, instead of labeling a subset of pixels only in the pixel level stage, we assign a unique label to every pixel even in the case of low confidence. The mAP drops to 47.5%, 3.7% lower than the performance of the original pipeline.

6. Conclusions

In this paper, we have presented a new pipeline for weakly supervised object recognition, detection and segmentation. Different from previous algorithms, we fuse and filter object instances from different techniques and perform pixel labeling with uncertainty. We use the resulting pixel-wise labels to generate groundtruth bounding boxes for object detection and attention maps for multi-label classification. Our pipeline has achieved clearly better performance in all of these tasks. Nevertheless, how to simplify the steps in our pipeline deserves further investigation.

References

[1] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 2

[2] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009. 2, 3

[4] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 2

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5, 6

[6] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *International journal of computer vision*, 100(3):275–293, 2012. 7

[7] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool. Weakly supervised cascaded convolutional networks. *arXiv preprint arXiv:1611.08258*, 2016. 2

[8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997. 2

[9] T. Durand, T. Mordan, N. Thome, and M. Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017. 2

[10] T. Durand, N. Thome, and M. 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. 2

[11] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 6, 7

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 2

[13] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 6, 7

[14] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017. 3

[15] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 991–998. IEEE, 2011. 6

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[17] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 2

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 6

[19] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu. Deep self-taught learning for weakly supervised object localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 7, 8

[20] D. Kim, D. Cho, D. Yoo, and I. So Kweon. Two-phase learning for weakly supervised object localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 7

[21] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711. Springer, 2016. 7

[22] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 6, 7

[23] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Muller, and W. 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. 1, 2

[24] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. 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. 6, 8

[25] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 478–487, 2016. 5

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 7

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 5

[28] M. Oquab, L. Bottou, I. Laptev, and J. 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. 2

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In

Advances in neural information processing systems, pages 91–99, 2015. 3

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 6, 7

[31] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. 5

[32] A. Roy and S. 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. 1, 7

[33] F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez. Built-in foreground/background prior for weakly-supervised semantic segmentation. In *European Conference on Computer Vision*, pages 413–432. Springer, 2016. 7

[34] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 4

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[36] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986. 2

[37] L. Sun, Q. Huo, W. Jia, and K. Chen. A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906–2920, 2015. 3

[38] P. Tang, X. Wang, X. Bai, and W. Liu. Multiple instance detection network with online instance classifier refinement. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 4, 6, 7, 8

[39] G. Tsagkatakis and A. Savakis. Online distance metric learning for object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(12):1810–1821, 2011. 4

[40] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2

[41] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2285–2294, 2016. 8

[42] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 464–472, 2017. 8

[43] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE CVPR*, 2017. 6, 7

[44] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision*, pages 543–559. Springer, 2016. 1, 2, 4

[45] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu. Multi-label image classification with regional latent semantic dependencies. *arXiv preprint arXiv:1612.01082*, 2016. 8

[46] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016. 4

[47] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 8

[48] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. 2