Weakly Supervised Learning of Deformable Part-Based Models for Object Detection via Region Proposals
Yuxing Tang, Xiaofang Wang, Emmanuel Dellandréa, Liming Chen

To cite this version:
Abstract—Successful deformable part-based models (DPM) for visual object detection relies on training images with fully annotated object bounding boxes. In the context of lack of object level annotations, our goal is to propose a model enhancing the weakly supervised DPM by emphasizing the importance of location and size of the initial class specific root filter. To adaptively select a discriminative set of candidate bounding boxes as this root filter estimate, first, we explore the generic objectness measurement to combine the most salient regions and “good” region proposals. Second, we propose the learning of the latent class label of each candidate window as a binary classification problem, by training category specific classifiers used to coarsely classify a candidate window into either target object or non-target class. Finally, we design a flexible enlarging-and-shrinking post-processing procedure to modify the output of DPM, which can effectively fit it to the approximative aspect ratio of the object and further improve the final accuracy. Extensive experimental results on the challenging PASCAL Visual Object Class (VOC) 2007 dataset demonstrate that our proposed framework is effective for initializing the DPM root filter. Our method also shows competitive final localization performance with the state-of-the-art weakly supervised object detection methods.

Index Terms—Object detection, deformable part-based models, region proposals, weakly supervised learning.

I. INTRODUCTION

OBJECT detection/localization in images is one of the most widely studied problems in computer vision. This task remains challenging mainly due to scale and viewpoint variation, deformation, occlusion, background clutter, intra-class variations and inter-class similarities for the objects in real world images. For most of the existing methods, a fully supervised learning (FLS) approach is adopted [1], [2], [3], [4], where positive training images are manually annotated with bounding boxes encompassing the objects of interest. This manual annotation of object location for large-scale image database is extremely laborious and unreliable thought extremely valuable. However, it is usually much easier to obtain weakly labeled data, where image level labels (e.g., user generated image tags on Internet) are presented. As a result, in this paper, in contrast to the traditional FSL, we are interested in weakly supervised learning (WSL) for object detection, where the exact object locations in positive training examples are not provided, given only the binary labels indicating the presence or absence of the objects of interest.

A. Related work

In recent years, there has been a substantial amount of work in weakly supervised object detection. From weakly annotated examples, the common practice is to jointly learn an appearance model together with the latent object location. The majority of related work treats WSL for object detection as an MIL (Multiple Instance Learning) [5] problem. In the MIL framework, one has some positive and negative bags. A bag is positive when it has at least one positive instance, while it is negative if all the instances are negative. The objective of MIL is to train a classifier which can correctly classify a test instance as either positive or negative. MIL problems are usually solved by finding a local minimum of non-convex objective function (e.g., MI-SVM [6]). Galleguillos et al. [7] first use the MIL model to recognize and localize objects based on multiple stable segmentations. [8], [9] use variants of MIL to learn object detectors from weakly labeled images and videos. Cinbis et al. [10] use multi-fold training procedure for MIL to avoid rapid convergence to poor local optima. Also in order to get rid of bad local minimum, Song et al. [11] initialize the object locations via a discriminative submodular covering method.

Another main strategy for WSL detection is to utilize a category-independent saliency measure to predict whether a given image region belongs to an object or not. For example, Deselaers et al. [12] propose a fully connected CRF (Conditional Random Field) [13] which aims to select a candidate window with the highest objectness score [14] in each positive training image.

Some work cast the WSL problem as a transfer learning (TL) problem. For example, Shi et al. [15] formulate a TL based on learning to rank, which effectively transfers a model for predicting object location from an auxiliary dataset to a target dataset with completely unrelated object categories. Hoffman et al. [16] propose an algorithm which can learn the difference between the image classifier and the object detector, and transfers this knowledge to classifiers for categories without bounding box annotated data, turning them into detectors. However, for both of these methods, auxiliary object level annotations for part of the dataset are required.

In addition, Pandey et al. [17] modify the fully supervised DPM to a weakly supervised manner without object level
annotations, which learns structural object detectors based on randomly initialized windows in the positive training images. Shi et al. [18] propose a WSL framework based on Bayesian joint topic modelling which localizes object across different classes concurrently. Recently, Wang et al. [19] propose to learn the latent categories using probabilistic Latent Semantic Analysis (pLSA), and select the target object category by evaluating each latent category’s discrimination. Bilen [20] et al. propose to couple a smooth discriminative learning procedure with a convex clustering algorithm, by imposing the similarity among objects of the same class.

B. Motivation and Contribution

Deformable Part-based Models (DPM) [2] and its variants [21], [22], [23] have been dominant in supervised object detection on challenging PASCAL VOC datasets [24] for a long period. The DPM represents an object with a holistic root filter that approximately covers an entire object and several higher resolution part filters that capture smaller local appearances (parts) of the object. It also characterizes the deformations by links connecting different parts. In the standard (fully supervised) DPM framework, the root filter is initialized with the positive ground-truth object bounding box, and it is allowed to move around in its small neighborhood to maximize the filter score. The locations of object parts are always treated as latent information due to the unavailability of object parts annotations upon most occasions. A latent SVM (LSVM) is adopted to learn the deformation of the objects, which can alternate between fixing latent values (part locations) for positive examples and optimizing its objective function.

Pandey et al. [17] modify the fully supervised DPM to a weakly supervised manner without object level annotations, which treats the location of root filter and part filters full latent, and learns structural object detectors based on the entire image. Its root filter’s location is initialized randomly based on a window which has at least 40% overlap with the positive training image, and its aspect ratio is initialized roughly to the average of the aspect ratios of positive training examples. However, the specific size and location of the initial root filter, as well as their aspect ratio are indicated to have a significant impact on the final localization result [1], [2], [17]. By randomly initialization, the object detector tends to learn spurious models of other classes or background regions, leading to lower accuracy during testing. And to our best knowledge, method for initializing the root filter based on theoretical deduction in weakly supervised DPM, as well as the definition of the aspect ratio of the objects, have not been well studied in [17].

To make up the performance gap between weakly and fully supervised DPM, in this paper, we are motivated to propose a model that enhancing the weakly supervised DPM by emphasizing the importance of location and size of the initial class specific root filter. To be precise, our goal is to discover a reliable initial set of image windows that are probably going to contain the target objects in the positive training images with only category level annotations, so as to represent the object instances. Hence, our WSL framework incorporates adaptive window selection from class independent object proposals and training of deformable part-based models. In particular, we explore the “objectness” approaches [14], [25], which generate class independent object proposals with corresponding scores indicate their probabilities of being object instances, then we adaptively select a reliable set of windows from the derived object proposals for each image as initialization, by fusing visual saliency and “objectness” scores. Two different initialization schemes are developed: single region and multiple regions initialization. The former tends to select one relative larger bounding box which may contain the most salient part in the image, while the latter is much more generalized, which selects a small number of object estimations that can also capture smaller and scattered objects. For multiple regions initialization, the labels of the regions are latent information. We learn the latent class label by framing it as a classification problem, which tries to coarsely classify each region into target object class or non-target class by some class specific classifiers. The generated object estimations are treated as the initial root filter estimates for training DPM detector.

The main contributions in this work are four-fold:

1) We propose a selection model based on generic “objectness” and visual saliency to adaptively select a discriminative set of candidate windows which tend to represent the object instances in the image.
2) We frame the learning of the latent class label of each candidate window as a binary classification problem, by training category specific classifiers which tries to coarsely classify a candidate window into either target object or non-target class.
3) We propose to use a flexible enlarging-and-shrinking post-processing procedure to modify the predicted output of DPM detector, which can effectively generate more accurate bounding box by better conserving foreground and cropping out plain background regions, to approximatively fit for the aspect ratio of the object.
4) Extensive experiments are carried out on two subsets and the whole set of the challenging PASCAL VOC 2007 database [24] with different criteria, namely annotation accuracy in terms of correct localization on training set, and detection accuracy in terms of average precision on test set. Experimental results demonstrate that our proposed framework is effective for initialization of DPM root filter, and shows competitive final localization performance with the state-of-the-art weakly supervised object detection methods.

A preliminary version of this work appeared in [26], which fuses the generic “objectness” with deformable part-based models for WSL detection. This paper includes that work but significantly extends it in the following ways. Firstly, we explore a much more generalized model M-WDPM (multiple regions initialization for weakly supervised deformable part-based models) which tries to select multiple regions, and we learn the latent label information of these regions in an effective way. This model shows its superiority in discovering not
only salient objects but also smaller and scattered objects to SWDPM (multiple regions initialization for weakly supervised DPM) in [26]. Secondly, we experiment with advanced region proposals generated by Selective Search [25], and we also adopt the latest deep features to represent the image content.

Thirdly, we evaluate our framework on the entire PASCAL VOC 2007 dataset, and compare it with state-of-the-arts. We also analyze the types of error that our detection framework inclines to make.

C. The Organization of the Paper

The rest of the paper is organized as follows: we present our weakly supervised DPM framework in details in Section II, and in Section III we present our experimental results and the comparison with other methods on PASCAL VOC 2007 datasets. In Section IV, we conclude our work.

II. Fusing Generic Objectness and Deformable Part-Based Models for Weakly Supervised Object Detection

In this section, we detail our approach of the weakly supervised DPM for object detection. Firstly, we introduce our approach to adaptively select the representative and discriminative regions from the category-independent object proposals. Secondly, we elaborate how to learn the latent class information when multiple regions are selected. Then we briefly describe the weakly supervised learning procedures using the selected regions with DPM and detection rescoring algorithm for testing. Finally we propose our new post-processing method to further refine the predicted object bounding box obtained by weak DPM detector, so as to cover the object more precisely.

A. Object Estimations: Initialization

In the weakly supervised DPM training procedure, a good initialization of the root filter is crucial. Hence, our goal is to discover a reliable initial set of image windows that are probably going to contain the target objects in the positive training images with only category level annotations, in order to represent the object instances.

1) Region extraction: Two general approaches have been proposed for generating class-independent object proposals in recent years: window scoring methods such as Objectness [14], BING [27], EdgeBoxes [28] and grouping methods such as Selective Search [25], Constrained Parametric Min-Cuts (CPMC) [29], Multiscale Combinatorial Grouping (MCG) [30]. We use Selective Search since it has been used as the proposal generating method by state-of-the-art supervised R-CNN detector [4]. We also report results using objectness method [14] to make comparison with prior detection work [14], [26].

Given an input image \( I \) (shown in Fig.1(a)), we first select top \( n \) scored windows \( W = \{w_1, w_2, \ldots, w_n\} \) and corresponding scores, denoted as \( S = \{s_1, s_2, \ldots, s_n\} \), which indicate the probabilities to cover objects within them, generated by Selective Search (shown in Fig.1 (b)). To balance a high recall (i.e., covering more objects) and computation efficiency (i.e., small number of region proposals), we set \( n = \min(1000, N) \) according to [31], where \( N \) is the number of proposals generated by Selective Search.

Based on the fact that the region proposal method is designed to capture all possible objects within an image, we assume that it has the reliability for providing a set of good candidate windows \( W^\prime \subseteq W \) which covers the object of interest. However, the windows with the higher scores are not always the effective choices [15], which usually encompass other noisy background, or locate poorly on object targets (e.g., they
may cover only the object parts). To extract a reliable set of object estimations from the pool of \( n \) windows, we design a recursive selection scheme shown in Fig.1 (c)-(g).

2) Salient reference region: For weakly supervised learning, it is obvious the initialization of DPM root filter is significant. It will hurt the detector greatly if it shoots on the background region. Consequently, starting from visually meaningful regions (foreground objects) is imperatively necessary. Inspired by the success of visual saliency applied in salient object recognition, we compute the reference region \( R \) (shown in Fig.1 (d)) by thresholding and merging the saliency map (or heat map) \( M \) (shown in Fig.1 (e)). The value of saliency map \( M \) at pixel \( I(i,j) \) is obtained by summing up the scores of the windows that cover this pixel:

\[
M(i,j) = \sum_{k=1}^{n} M_k(i,j)
\]

where,

\[
M_k(i,j) = \begin{cases} s_k, & \text{if } I(i,j) \in w_k, \forall w_k \in W, \\ 0, & \text{otherwise.} \end{cases}
\]

The reference region \( R \) can be one connected (continuous) region or several discrete regions in the image according to the score range and threshold value.

3) Coarse candidate windows pool: It is known that the score given by Selective Search (i.e., objectness score) corresponds with the probability to have target object inside its window to some extent. To take advantage of this auxiliary information, we concurrently select 200 out of \( n \) windows that with higher scores as candidates, according to the histogram of \( n \) scored windows (shown in Fig.1(e)). In order to avoid near duplicate candidate windows, we further perform non-maximum suppression (NMS) to get a finer set of candidates. Contrary to the common practice, which starts the suppression procedure from highest scored window, we randomly choose one, because we observed that the window with the highest score is not necessarily the best. Fig.1 (f) illustrates the derived smaller set of \( l \) confident candidates \( \hat{W} = \{ \hat{w}_1, \hat{w}_2, \ldots, \hat{w}_l \} \), and their corresponding scores denoted as \( \hat{S} = \{ \hat{s}_1, \hat{s}_2, \ldots, \hat{s}_l \} \).

4) Object invariant estimations: Given the reference region \( R \) which implies the most salient region (or regions) within an image, and confident candidate windows \( \hat{W} \) with scores \( \hat{S} \), the overlap between them provides valuable information to find the locations of target objects. We will propose two different schemes to fuse the salient region(s) with the extracted candidate windows.

a) Single region initialization: In [17], the root filter of the DPM is randomly initialized from a single window which covers at least 40% overlap with the original image. To demonstrate our selection scheme is superior than the randomly chosen window, we also filter out only one single window \( w^* \) from the candidates pool \( \hat{W} \). Intuitively, we expect this window estimation to cover as much as the salient reference region \( R \) and to have a relative higher objectness score as well. Hence forth, the estimation of the initial object bounding box with objectness score \( (w^*,s^*) \) (Fig.1(g), upper image) can be determined by optimizing the following function:

\[
(w^*,s^*) = \arg\max_{\hat{s}_i} \left[ \alpha \hat{s}_i + (1 - \alpha) \frac{\text{area}(R \cap \hat{w}_i)}{\text{area}(R \cup \hat{w}_i)}, \right], \quad i \in [1,l]
\]

where \( \alpha \) is a parameter used to control the influence of the objectness score \( s_i \). In practice, \( \alpha = 0.2 \), was selected by a grid search over \{0.1,0.2,0.3,0.4\} on a validation set, for the purpose of emphasizing the priority of the intersection overlap union (IoU) overlap between the candidate window and merged salient reference region.

The single region initialization scheme prefer to select a relative large region which may contain the most salient part in the image. It can produce good DPM object detectors in a weakly supervised manner, when very few objects gathering together in an image. For example, by adopting the single region scheme, the blue window in Fig.1(g) upper image, is used as a positive training example (i.e. DPM root filter initialization) for both horse and person category.

b) Multiple regions initialization: In fact, multiple objects (e.g., 2.5 objects in average for PASCAL VOC2007 trainval dataset) can be scattered anywhere in an image. We can therefore further improve DPM detectors by providing more object estimations as root filter initializations, instead of training the object detectors with a single window for each image. For each image, we are motivated to select a small number of object estimations that can also capture smaller and scattered objects, which can better represent the original image. We adopt the similar criteria as the score function Eq. (3). To alleviate the influence of the area of \( R \), we set \( \alpha \) to be 0.3. Instead only selecting the maximal scoring window in Eq. (3), we pick out top \( Q \) scored windows \( W^* \) for each image.

After generating several object estimations from each image, the next step is to approximately identify the class label of each estimation given only the labels of the whole image. For example, in Fig. 1(g) bottom image, the color windows with solid lines are associated with the horse and person labels. However, so far we have no idea which object(s) (or even background) is/are inside each bounding box. We will commit ourselves to solve this problem in the next subsection.

B. Learning Latent Object Classes via Region Classification

For each positive training image, we have generated \( Q \) object invariant estimations with the multiple regions initialization scheme. Consider an object category, e.g., horse, which has \( P \) positive training images, we can totally obtain \( z = P \times Q \) object estimations. Obviously, some of these object estimations come from other categories (e.g., person, sheep, object parts or the background regions as well), where the class labels are latent information. In this paper, we frame the latent class learning problem as a classification problem by coarsely classifying these object estimations into either target object category or non-target category (i.e., other classes, object parts or background).

1) Region representation: We use the deep convolutional neural networks (CNN) features to represent the regions (object estimations). Firstly, we pre-train an eight-layer (five convolutional layers and three fully-connected layers) Alex-Net
Fig. 2. Illustration of our latent class learning framework for the horse category. For each object category, we train a linear SVM classifier with the CNN features (output of CNN's fc6 layer). Object estimations from the positive training images of this category are scored by its SVM. We select the regions with higher scores by thresholding as the representative objects of this category (horse vs. non horse for this example).

<table>
<thead>
<tr>
<th>Positive horse images</th>
<th>Feature vectors</th>
<th>Initial object estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Positive horse images" /></td>
<td><img src="image" alt="Feature vectors" /></td>
<td><img src="image" alt="Initial object estimation" /></td>
</tr>
</tbody>
</table>

Table: Feature vectors and initial object estimation for horse and non-horse categories.

<table>
<thead>
<tr>
<th>Horse estimation</th>
<th>Non Horse estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.91</td>
<td>0.80</td>
</tr>
<tr>
<td>0.37</td>
<td>0.61</td>
</tr>
<tr>
<td>-0.58</td>
<td>-0.06</td>
</tr>
<tr>
<td>-0.82</td>
<td>-0.61</td>
</tr>
</tbody>
</table>

CNN with caffe implementation [33] on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 classification dataset [34], which contains 1.2 million images of 1000 categories. Then we warp each region into a required fixed pixel size of 227 × 227, and subtract it with the mean RGB image, and forward propagate it through the network. Finally, we take the output of the fist fully-connected layer (i.e., fc6 layer) to represent the input region. The output of fc6 layer is a 4096-dimensional feature vector. This feature extraction process is similar to R-CNN [4], but it is worth noticing that we do not fine-tune the pre-trained CNN on target dataset, because the object level annotations are assumed not to be available in the weakly annotated data. And we do not pad the region with additional image context around it, as our region estimation is already expected to have a significant coverage of the context information due to our selection schemes in Section II-A3.

2) Region classification: Consider training a horse detector. For all the P positive training images in the horse category, we generate z object invariant estimations. Intuitively, only part of these z regions contain the target horse object, others may have person, sheep, dog or even background. We learn the latent categories in these regions via region classification.

We first train a horse linear SVM [35] classifier using the images labeled with horse as positive training examples and the ones without horses as negative examples. We compute the similar 4096-dimensional CNN feature vector as in Section II-B1 on whole images. We then run the trained horse classifier on the z object invariant estimations in the positive training images. By thresholding the SVM scores, finally we obtain a subset z′ regions from z estimations (z′ < z). These z′ regions are assumed to represent the target horse category, which can be treated as positive training examples of the horse detector.

Suppose we have K categories we want to detect. We train one binary SVM classifier on positive and negative images of each category, and run these K classifiers on their corresponding object estimations. We select high scoring regions for each target category so as to represent the objects of interest. Fig. 2 shows the latent class learning framework using SVM classification on the horse category.

Note that the above latent class learning process is only applied to multiple regions initialization, since for single region initialization, the unique generated window is used toinitialize the DPM root filter for any categories appeared in the image.

C. Weakly Supervised DPM Training and Testing Details

We design two different kinds of deformable part based models for weakly supervised object detection according to different initialization schemes in Section II-A.

1) Single region initialization for weak DPM (S-WDPM) detection: Similarly to [2], each root filter hypothesis in a positive training image is initialized with the corresponding derived bounding box from the single region initialization scheme. The size and aspect ratio of the DPM root filter are decided by the average size and aspect ratio of the object estimation boxes (ground-truth bounding box and aspect ratio are used in [2]). The root filter hypothesis is allowed to move around in a small neighborhood to maximize the filter score to compensate for imprecise bounding box estimation from Section II-A4a. As in [17], we represent an image by a multiscale HOG feature pyramid [1] of 16 levels. For this DPM model, we use only a single component, since the multiple components is used for detecting objects with different views. We set the number of parts in this DPM to 8 as [2]). And for negative training examples, we use random negatives from other object categories. For testing, sliding window approach is adopted. This single region initialized weakly supervised DPM detection model is denoted as S-WDPM. We refer the reader to [2] for more details concerning
the DPM training and detection procedures.

2) Multiple regions initialization for weak DPM (M-WDPM) detection: For the M-WDPM (multiple regions initialized weakly supervised DPM), we make it much “deeper” with the DeepPyramid feature [36], for the reason that the HOG feature is suboptimal compared to deep features computed by CNN [3, 37, 4, 38, 19]. The feature map is computed by the fifth convolutional layer (conv5), which has 256 feature channels. We represent each image (or region) with a feature pyramid of 7 levels (256 feature channels. We represent each image (or region)

For the M-WDPM (multiple regions ini-
the DPM training and detection procedures.

conv

HOG feature is suboptimal compared to deep features com-
puted by CNN [3, 37, 4, 38, 19]. The feature map is computed by the fifth convolutional layer (conv5), which has 256 feature channels. We represent each image (or region) with a feature pyramid of 7 levels (256 feature channels. We represent each image (or region)

pyramid). For training, the selected object estimations from

Section II-B2 are treated as positive training examples, and
the random windows from negative images are defined as

negative examples. We use a DPM with 3 components and
8 parts per component according to [36]. The training and
testing procedures are similar to S-WDPM above, but we add
a simple bounding box resoring stage with the help of a front-
to-end CNN padded with a softmax classifier.

The resoring function is defined as:

\[
s_{det}^i = \kappa s_{M-WDPM}^i + (1 - \kappa)s_{cls}^i, \ i \in [1, K]
\] (4)

where, \(0 \leq s_{M-WDPM} \leq 1\) is the normalized DPM detection
score on a sub-window of the \(i\)-th detector, and \(0 \leq s_{cls} \leq 1\) is the softmax classification score of the corresponding \(i\)-th
category on this sub-window. \(\kappa\) is a hyperparameter used to
leverage the two scores, which ranges from 0.6 to 0.9. The
final predicted windows are obtained by thresholding the \(s_{det}^i\)
in Eq. (4).

In order to train this front-to-end CNN classifier described
above, we fine-tune the pre-trained CNN with image level
annotations on our training data. We implement it by removing
the last 1000-way softmax layer while keeping all the other
parameters and adding a new randomly initialized \(K\)-way
softmax classification layer, and we then fine-tune the entire
network based on the image-level labels.

In [2], contextual information is exploited to resore the
bounding boxes. However, it needs object-level annotations
to extract the contextual information. Our detection resoring
method does not require the object level annotations, and it
leads to a remarkable improvement in the average precision
on several classes in the PASCAL VOC 2007 datasets (see
Section III-B). An example of our bounding box resoring
procedure is shown in Fig. 3.

D. Bounding Box Post-processing

In many cases, the bounding boxes generated by DPM
detectors are too large (resp. small) when detecting very small
(resp. large) objects due to the restrictions of the size of
the root filter and the scale of the feature pyramid. To improve
the localization and to obtain a more precise prediction of the
bounding box aspect ratio, we post-process each bounding box
by enlarging or shrinking (ES post-processing) it to cover the
object as much as possible. This is done using an improved
version of the method proposed in [39] which measures the
amount of area that the edge energy occupies. In brief, we first
augment the original bounding box \(w = (x_{\min}, y_{\min}, x_{\max}, y_{\max})\)
to 120% of the original width and height (i.e., 144% in total

area, denoted as \(w^\text{aug} = (x_{\min}^\text{aug}, y_{\min}^\text{aug}, x_{\max}^\text{aug}, y_{\max}^\text{aug})\). Expanding from
the centroid if applicable. Otherwise, stop when reaching the
border of the image.), and calculate the absolute values of the
gradients \(L_{\text{edge}}\) by applying a 3 \times 3 Laplacian filter with \(\gamma = 0.2\)
over the augmented bounding box. To easily calculate the
edge spatial distribution, we then resize the gradient magnitude
image size to 100 \times 100 and normalize the image sum to 1,
i.e., \(L_{\text{edge}}^\text{aug}\). And we set the values which are less than 10% of
the maximum \(L_{\max}\) to 0. Finally, we expand the bounding box
in four directions from the current centroid \((x_c^\prime, y_c^\prime)\) and stop
until it contains 98% of the total gradient magnitude (edge
energy) in the augmented box. Detailed algorithm is listed in
Algorithm 1.

This post-processing technique is not only able to crop out
plain background regions, but also can expand to cover the
foreground regions which are not encompassed by the original
box. However, the cropping method in [17] is probably to fail
with the latter. Fig. 4 shows a few examples of our bounding
box post-processing results. It is also worth noticing that this
post-processing technique works efficiently for the objects
with a unique or plain background, but has limited help for

```
Algorithm 1 Bounding box post-processing pipeline.

Input:
Original bounding box: \(w = (x_{\min}, y_{\min}, x_{\max}, y_{\max})\);
original image width: \(w_o\); original image height: \(h_o\);
maximal expanding rate: \(\beta = 1.2\);
Laplacian filter shape: \(\gamma = 0.2\).
Output:
Cropped bounding box: \(w' = (x_{\min}', y_{\min}', x_{\max}', y_{\max}')\).

1: centroid: \((x_c, y_c) = (\frac{x_{\min} + x_{\max}}{2}, \frac{y_{\min} + y_{\max}}{2})\);
2: augmented width: \(a = \beta \ast (x_{\max} - x_{\min})\);
3: augmented height: \(b = \beta \ast (y_{\max} - y_{\min})\);
4: if \(x_c - \frac{a}{2} > 0\) then
5: \(x_{\min}' = \text{ceil}(x_c - \frac{a}{2})\);
6: else
7: \(x_{\min}' = 1\);
8: end if
9: if \(x_c + \frac{a}{2} < w_o\) then
10: \(x_{\max}' = \text{floor}(x_c + \frac{a}{2})\);
11: else
12: \(x_{\max}' = w_o\);
13: end if
14: similar for \(y_{\min}'\) and \(y_{\max}'\);
15: \(y_{\min}' = (y_{\min} + y_{\max} + a_{\text{aug}})\);
16: \(L_{\text{edge}} = \text{filter(image}(w^\text{aug}), \text{laplacian})\); \(\gamma\);
17: \(L_{\text{edge}} = \text{norm}(\text{resize}(\text{abs}(L_{\text{edge}}), [100, 100]), 1)\);
18: \(L_{\text{edge}} = \text{max}(L_{\text{edge}})\);
19: for \(i = 1, 2, . . . , 100\) do
20: \(j = 1, 2, . . . , 100\) do
21: if \(L_{\text{edge}}(i, j) < 0.1 \ast L_{\text{edge}}\) then
22: \(L_{\text{edge}}(i, j) = 0\);
23: end if
24: end for
25: end for
26: current centroid: \((x_c', y_c') \leftarrow \text{average energy point of } L_{\text{edge}}\); \(\text{current centroid: } (x_c', y_c') \leftarrow \text{average energy point of } L_{\text{edge}}\);
27: while energy in \(w'' < 0.98 \ast \sum(L_{\text{edge}})\) do
28: \(w'' = (x_{\min}'' y_{\min}'' x_{\max}'' y_{\max}'') \leftarrow \text{update by expanding bounding box in four directions } (-x, -y, x, y)\) from the current centroid \((x_c', y_c')\);
29: end while
30: project \(w''\) into original image: \(w' = (x_{\min}', y_{\min}', x_{\max}', y_{\max}') \leftarrow w''(x_{\min}'', y_{\min}'' x_{\max}'' y_{\max}'')\).
```
Fig. 3. Illustration of detection rescoring using M-WDPM and CNN softmax classifier. For a testing image, $K$ (number of classes in target dataset) class-specific M-WDPMs are applied on it in sliding window manner. For each sub-window detected by M-WDPM, the normalized detection score is rescored by the softmax classifier of the detected category. In this example, the wrongly detected car and bicycle are finally discarded by the detector after the rescoring stage.

Fig. 4. Examples of bounding box enlarging and shrinking. Boxes before (resp. after) post-processing are shown in red (resp. yellow).

those with cluttered or textured background.

III. EXPERIMENTAL EVALUATION

In this section, we present the experimental results of our proposed framework with two different initialization schemes (i.e., S-WDPM using single region initialization and M-WDPM using multiple regions initialization) on the challenging PASCAL VOC 2007 dataset [24].

A. Experiments with S-WDPM

1) Datasets: Following the protocol of previous works [17], [12], [40], we evaluate the performance of our proposed S-WDPM (single region initialized weak DPM) framework on two subsets from the training and validation set (trainval) of the PASCAL VOC 2007 dataset (VOC07) [24]: VOC07-6×2 and VOC07-14. The VOC07-6×2 subset contains 6 classes (aeroplane, bicycle, boat, bus, horse and motorbike) with Left and Right views (aspects) of each class, resulting in a total of 12 separating classes. The VOC07-14 subset (same with PASCAL07-all defined in [17]) consists of 42 class/view combinations covering 14 classes and 5 views (Left, Right, Frontal, Rear and Unspecified). Similar to [17], [12], [40], we remove all the images annotated as difficult or truncated in both training and evaluation steps.

2) Evaluation protocol: To make fair comparisons, we only choose the detection window with highest score per image, although our method can detect multiple instances appeared in the image using sliding window approach. We also report both results for initial and refined localization as [17], [40]. A refined localization is obtained by an iteratively trained DPM detector for one/several iteration(s) to refine the initial detection using the previous annotations as ground truth. Performance is evaluated with the percentage of training (train + val) images in which an object is correctly covered by the window (i.e. CorLoc [12]), if the strict PASCAL-overlap criterion is satisfied (intersection-over-union > 0.5).

3) Experimental evaluation: We compare our S-WDPM with Weak DPM [17], Weak objectness [12] and Joint topic model [18]. For the Weak objectness approach [12], the region proposal with the highest “objectness” score is selected as the
predicted window. As Table I shows, our method outperforms [12] and our baseline approach [17] on both datasets. Both [17] and our S-WDPM use the same HOG feature pyramid for the DPM. We present our results using two kinds of object proposal generating methods: objeness (obj) and Selective Search (SS). For obj, our average performance of initial detection before post-processing the bounding boxes on the VOC07-6×2 and VOC07-14 subsets is 38.74% and 21.73% respectively, versus 37.22% and 19.98% for [17]. These improvements are due to the initial object estimate of our method described in Section II-A4a, which gives a better initialization of the root filter of DPM detectors. We can also observe that both the post-processing method of cropping [17] (i.e., S-WDPM(crop) in Table I) and our enlarging-or-shrinking (i.e., S-WDPM(ES)) post-processing method steadily improve the average localization accuracy. In particular, our ES method is superior to the cropping method of [17], as our cropped bounding box is not only able to shrink to crop out the background regions, but also capable of enlarging to cover the whole foreground object resulted by incomplete coverage of the original window. An example is shown in the last row of Fig. 5, where the target object (motorbike) is only partially localized by the initial detector (shown in red rectangles in the middle and right images) for both [17] and our method. However, in the final detection (shown in yellow) after post-processing, our method is able to enlarge the bounding box to approximately include the whole object, while [17] tends to crop out both foreground and background regions.

Additionally, the rows start with “Refinement” in Table I indicate that localization accuracy can benefit from the iterative refinement process. It is worth mentioning that with a better initialization, our models converge to a steady level of performance after one less round of costly re-training (i.e., 2 iterations for obj vs. 3 iterations) than [17], and achieve slightly better results in the mean time.

The detailed comparisons for our S-WDPM using obj with the state-of-the-arts on the VOC07-6×2 dataset are listed in Table II. The results show that our method outperforms [17] for most of the categories. Especially, our method achieves the state-of-the-art results in some classes where the target object possesses the most salient regions in that category (e.g., aeroplane, bus, horse). Interestingly, even without refinement process, the accuracy for our method with certain category (e.g., aeroplane left) is superior to the competitors with the time-consuming refinement procedure. Fig. 5 visually compares some of our results with those of [17].

We find that the best detection result using the Selective Search (63.31%) is 3.49% better than the objectness (59.82%) within the same S-WDPM detection model without post-processing, and is 3.22% better (67.13% vs. 63.91%) with post-processing, on the VOC07-6×2 dataset. This is in accord

| Table I | Average localization accuracy (in %) of our S-WDPM (Single region initialized weak DPM with HOG features) compared with state-of-the-art competitors on the two variations of the PASCAL VOC 2007 datasets. "crop" and "ES" denote the cropping method from [17] and our enlarging & shrinking post-processing. "obj" and "SS" denote the objeness and Selective Search region proposal generating method. "S" and "G" denote the sampling and Gaussian strategy from [18]. |
|--------|--------|--------|--------|--------|--------|--------|
|        | no post-processing | with post-processing |        |        |
|        | [17] | S-WDPM | [17]-crop | S-WDPM(crop) | S-WDPM(ES) | [18] |
|        |        | obj SS | obj SS | obj SS | obj SS | S G |
| Dataset | VOC07-6×2 |        |        |        |        |     |
| Initialization | 37.22 | 38.74 | 41.52 | 44.62 | 47.85 | 48.40 | 48.59 | 50.01 | 50.8 | 51.5 |
| Refinement 1 | 51.63 | 55.85 | 63.31 | 53.11 | 56.78 | 64.25 | 58.02 | 67.13 | 65.5 | 66.1 |
| Refinement 2 | 56.99 | 59.82 | — | 59.31 | 63.31 | — | 63.91 | — | — | — |
| Refinement 3 | 59.32 | — | — | 61.05 | — | — | — | — | — | — |
| Result from [12] |        |        |        |        |        |     |
|        | 19.98 | 21.73 | 24.87 | 23.00 | 24.20 | 26.30 | 25.12 | 31.84 | 32.2 | 30.5 |
| Refinement 1 | 25.11 | 27.46 | 31.15 | 26.38 | 28.21 | 33.10 | 28.94 | 34.91 | 33.8 | 32.5 |
| Refinement 2 | 27.69 | 28.95 | — | 29.39 | 32.87 | — | 32.82 | — | — | — |
| Refinement 3 | 28.98 | — | — | 30.31 | — | — | — | — | — | — |
| Result from [12] |        |        |        |        |        |     |
|        |        |        |        |        |        |     |
| Dataset | VOC07-14 |        |        |        |        |     |
| Initialization | 19.98 | 21.73 | 24.87 | 23.00 | 24.20 | 26.30 | 25.12 | 31.84 | 32.2 | 30.5 |
| Refinement 1 | 25.11 | 27.46 | 31.15 | 26.38 | 28.21 | 33.10 | 28.94 | 34.91 | 33.8 | 32.5 |
| Refinement 2 | 27.69 | 28.95 | — | 29.39 | 32.87 | — | 32.82 | — | — | — |
| Refinement 3 | 28.98 | — | — | 30.31 | — | — | — | — | — | — |
| Result from [12] |        |        |        |        |        |     |
|        |        |        |        |        |        |     |
| Table II | Class level localization accuracy (in %) for the VOC07-6×2 dataset for our S-WDPM(ES) using objectness proposals vs. [17], [12], [40]. |
|        | Initialization | Refined by detector |        |        |
|        | ours [17] | [40] | ours [17] | [12] |
| aero left | 65.1 | 55.8 | 39.1 | 69.7 | 65.1 | 58.0 |
| aero right | 64.1 | 61.5 | 50.0 | 84.6 | 82.1 | 59.0 |
| bike left | 31.3 | 31.3 | 28.4 | 85.4 | 87.5 | 46.0 |
| bike right | 42.0 | 44.0 | 30.6 | 54.0 | 68.0 | 40.0 |
| boat left | 9.1 | 4.6 | 15.1 | 13.6 | 2.3 | 9.0 |
| boat right | 9.3 | 9.3 | 20.7 | 14.0 | 7.0 | 16.0 |
| bus left | 23.8 | 23.8 | 31.0 | 42.9 | 28.6 | 38.0 |
| bus right | 65.2 | 52.2 | 35.1 | 69.6 | 47.8 | 74.0 |
| horse left | 64.6 | 60.4 | 48.5 | 87.5 | 83.3 | 58.0 |
| horse right | 73.9 | 67.4 | 45.2 | 76.1 | 80.4 | 52.0 |
| mbike left | 64.1 | 48.7 | 46.3 | 87.2 | 92.3 | 67.0 |
| mbike right | 70.6 | 76.5 | 55.3 | 82.4 | 88.2 | 76.0 |
| average | 48.6 | 44.6 | 37.1 | 63.9 | 61.1 | 50.0 |
with the conclusion in [31]. Moreover, it achieves comparable or slightly better results than the sophisticated joint topic learning models in [18] with running DPM refinement only once. As shown in Table I, the SS also outperforms obj on the VOC07-14 dataset. Consequently, we entirely adopt the Selective Search method (‘fast’ option) for our next experiments.

For the DeepPyramid feature extraction, we use a single NVIDIA GeForce GTX 780 GPU with 3GB memory. And we reduce the resized image resolution from 1713 × 1713 in [36] to 1505 × 1505 to avoid running out of memory.

2) Parameter selection: As discussed in Section II-A4b, we can generate \( Q \) region estimations for each image. \( Q \) is a parameter which impacts the quality of the positive training examples. If it is too large, there would be an enormous number of noisy samples for latent class learning. If it is set to be very small, the instances in the original image would not able to be comprehensively represented. Therefore, we experimentally vary \( Q = \{3, 5, 10, 15, 20, 30\} \) to see which one performs best on the PASCAL VOC 2007 validation set. We implement this by directly measuring the average annotation accuracy for all the classes, on the generated bounding boxes (\( Q \) per image) with the Pascal-overlap criterion. Fig. 6 shows the annotation accuracy for different \( Q \). We find that \( Q = 10 \) obtains the best result (34.5% average accuracy). When it is very small (e.g., 3), the performance drops dramatically to 27.0%. This is because some of the “good” region proposals are not selected due to very small \( Q \), while some selected “bad” regions may harm the model. When it goes up from 10 to 30, the performance declines gradually. One explanation for this might be that many object parts or background regions would be included when \( Q \) is large. Hence, we set \( Q = 10 \) in all of our experiments. Fig. 7 shows three example images and their 10 selected regions. The \( \kappa \) in Eq. (4) which leverages the classification and detection scores is set to 0.7 according to cross-validation on a subset of the validation data.

2. B. Experiments with M-WDPM

1) Dataset and settings: We evaluate our generalized model: M-WDPM (multiple regions initialized weak DPM) on the much more challenging dataset: the whole PASCAL VOC 2007 dataset. It contains totally 9963 images of 20 object categories, which is split into training (2501), validation (2510) and test (4952) sets. This dataset is challenging because it has large inter-class similarities, intra-class variances, cluttered backgrounds, and scale changes. We only use the image level category labels for this task. And images labeled as “difficult” ones are discarded as common practice in previous studies. For the M-WDPM testing, we only run the DPM once for efficiency, although the iterative detector refinement can steadily improve the final performance. The annotation accuracy on the trainval (training + validation) set and average precision (AP) for detection on the test set are reported.

For the M-WDPM-HOG baseline which computes the HOG features and does not make use of auxiliary training data from ILSVRC 2012 classification task [34] as in Section III-A2 on the PASCAL VOC 2007 trainval set. Table III reports our experimental results compared with the state-of-the-art WSL methods for object detection.

3) Annotation evaluation: We evaluate the same CorLoc [12] as in Section III-A2 on the PASCAL VOC 2007 trainval set. Table III reports our experimental results compared with the state-of-the-art WSL methods for object detection.
training the DPM root filters.

It is also observed that, with the help of auxiliary training data and recently popular deep features, the average accuracy of our M-WDPM-Deep model increases by 4% over the M-WDPM-HOG model. And our detection rescoring method (i.e., M-WDPM-rescore) further improves the performance for most of the categories. The average improvement for detection rescoring on all 20 classes is 2.7% (43.5% vs. 40.8%). Our M-WDPM-rescore method is comparable with the newly invented convex clustering approach [20], but it is worse than the LCL method [19] on average. Though [19] achieves the state-of-the-art performance on many classes, it depends on more sophisticated Super-Vector Coding [44] of the deep CNN features that tragically increases the feature dimensionality. And it fails on some categories such as boat and table. However, our W-SDPM-rescore exhibits steady agreeable performance on all the categories. Especially, our W-SDPM-rescore works well on categories where target objects are relatively salient. (e.g., aeroplane, boat, bus, cat and table.) Moreover, it achieves the best results for the classes such as aeroplane, boat, and cat.


### Table III

Comparisons of weakly supervised object detectors on PASCAL VOC 2007 trainval set in terms of correct localization (CORLOC [12], in %) on positive training images. († indicates methods using auxiliary training data from ILSVRC 2012.)

<table>
<thead>
<tr>
<th>method / class</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>our M-WDPM-HOG</td>
<td>67.6</td>
<td>51.7</td>
<td>32.2</td>
<td>20.1</td>
<td>14.7</td>
<td>41.6</td>
<td>58.8</td>
<td>57.1</td>
<td>9.0</td>
<td>41.9</td>
</tr>
<tr>
<td>our M-WDPM-deep†</td>
<td>71.2</td>
<td>58.1</td>
<td>37.0</td>
<td>22.4</td>
<td>17.1</td>
<td>44.6</td>
<td>62.4</td>
<td>60.2</td>
<td>17.3</td>
<td>48.8</td>
</tr>
<tr>
<td>our M-WDPM-rescore†</td>
<td>82.1</td>
<td>55.1</td>
<td>42.8</td>
<td>35.3</td>
<td>14.8</td>
<td>57.9</td>
<td>66.2</td>
<td>69.8</td>
<td>17.5</td>
<td>51.6</td>
</tr>
<tr>
<td>Joint Learning [8]</td>
<td>30.7</td>
<td>16.5</td>
<td>23.0</td>
<td>14.9</td>
<td>4.9</td>
<td>29.6</td>
<td>35.3</td>
<td>7.2</td>
<td>23.4</td>
<td></td>
</tr>
<tr>
<td>MI-SVM [41]</td>
<td>37.8</td>
<td>17.7</td>
<td>26.7</td>
<td>13.8</td>
<td>4.9</td>
<td>34.4</td>
<td>46.6</td>
<td>5.4</td>
<td>29.8</td>
<td></td>
</tr>
<tr>
<td>Model Drift [9]</td>
<td>42.4</td>
<td>46.5</td>
<td>18.2</td>
<td>8.8</td>
<td>2.9</td>
<td>40.9</td>
<td>73.2</td>
<td>5.4</td>
<td>30.5</td>
<td></td>
</tr>
<tr>
<td>MIL-Negative [40]</td>
<td>45.8</td>
<td>21.8</td>
<td>30.9</td>
<td>20.4</td>
<td>5.3</td>
<td>37.6</td>
<td>40.8</td>
<td>7.3</td>
<td>29.8</td>
<td></td>
</tr>
<tr>
<td>Transfer Learning [15]</td>
<td>54.7</td>
<td>22.7</td>
<td>33.7</td>
<td>24.5</td>
<td>4.6</td>
<td>33.9</td>
<td>57.0</td>
<td>7.3</td>
<td>39.1</td>
<td></td>
</tr>
<tr>
<td>Joint Topic [18]</td>
<td>67.3</td>
<td>54.4</td>
<td>34.3</td>
<td>17.8</td>
<td>1.3</td>
<td>46.6</td>
<td>50.7</td>
<td>6.7</td>
<td>32.4</td>
<td></td>
</tr>
<tr>
<td>Convex Clustering†</td>
<td>66.4</td>
<td>59.3</td>
<td>42.7</td>
<td>20.4</td>
<td>21.3</td>
<td>63.4</td>
<td>74.3</td>
<td>59.6</td>
<td>21.1</td>
<td>58.2</td>
</tr>
<tr>
<td>LCL-pLSA† [19]</td>
<td>80.1</td>
<td>63.9</td>
<td>51.5</td>
<td>45.2</td>
<td>40.8</td>
<td>52.1</td>
<td>58.8</td>
<td>43.5</td>
<td>26.2</td>
<td>53.4</td>
</tr>
</tbody>
</table>

### Table IV


<table>
<thead>
<tr>
<th>method / class</th>
<th>dog</th>
<th>horse</th>
<th>nbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>our M-WDPM-HOG</td>
<td>13.5</td>
<td>37.2</td>
<td>45.6</td>
<td>57.2</td>
<td>18.7</td>
<td>19.9</td>
<td>36.0</td>
<td>43.7</td>
<td>46.2</td>
<td>38.6</td>
</tr>
<tr>
<td>our M-WDPM-deep†</td>
<td>17.2</td>
<td>37.4</td>
<td>52.4</td>
<td>60.5</td>
<td>20.7</td>
<td>28.8</td>
<td>36.9</td>
<td>50.7</td>
<td>48.6</td>
<td>40.8</td>
</tr>
<tr>
<td>our M-WDPM-rescore†</td>
<td>24.4</td>
<td>39.4</td>
<td>55.9</td>
<td>51.0</td>
<td>24.2</td>
<td>22.9</td>
<td>44.2</td>
<td>52.1</td>
<td>45.2</td>
<td>43.5</td>
</tr>
<tr>
<td>Joint Learning [8]</td>
<td>20.5</td>
<td>32.1</td>
<td>24.4</td>
<td>33.1</td>
<td>17.2</td>
<td>12.2</td>
<td>20.8</td>
<td>28.8</td>
<td>40.6</td>
<td>22.4</td>
</tr>
<tr>
<td>MI-SVM [41]</td>
<td>14.5</td>
<td>32.8</td>
<td>34.8</td>
<td>41.6</td>
<td>19.9</td>
<td>11.4</td>
<td>25.0</td>
<td>36.5</td>
<td>45.2</td>
<td>8.6</td>
</tr>
<tr>
<td>Model Drift [9]</td>
<td>19.0</td>
<td>34.0</td>
<td>48.8</td>
<td>65.3</td>
<td>8.2</td>
<td>10.6</td>
<td>16.7</td>
<td>32.3</td>
<td>54.8</td>
<td>5.5</td>
</tr>
<tr>
<td>MIL-Negative [40]</td>
<td>27.5</td>
<td>41.3</td>
<td>41.8</td>
<td>47.3</td>
<td>24.1</td>
<td>12.2</td>
<td>28.1</td>
<td>32.8</td>
<td>48.7</td>
<td>9.4</td>
</tr>
<tr>
<td>Transfer Learning [15]</td>
<td>24.1</td>
<td>43.3</td>
<td>41.3</td>
<td>51.5</td>
<td>25.3</td>
<td>13.3</td>
<td>28.0</td>
<td>29.5</td>
<td>54.6</td>
<td>11.8</td>
</tr>
<tr>
<td>Joint Topic [18]</td>
<td>16.2</td>
<td>58.9</td>
<td>51.5</td>
<td>64.6</td>
<td>18.2</td>
<td>3.1</td>
<td>20.9</td>
<td>34.7</td>
<td>63.4</td>
<td>5.9</td>
</tr>
<tr>
<td>Convex Clustering†</td>
<td>14.0</td>
<td>38.5</td>
<td>49.5</td>
<td>60.0</td>
<td>19.8</td>
<td>39.2</td>
<td>41.7</td>
<td>30.1</td>
<td>50.2</td>
<td>44.1</td>
</tr>
<tr>
<td>LCL-pLSA† [19]</td>
<td>16.3</td>
<td>56.7</td>
<td>58.3</td>
<td>69.5</td>
<td>14.1</td>
<td>38.3</td>
<td>58.8</td>
<td>47.2</td>
<td>49.1</td>
<td>60.9</td>
</tr>
</tbody>
</table>

![Fig. 7. Three example images and their 10 selected regions (resized to same squared size for regularity).](image-url)
to represent the discovered object windows, its performance (22.7%) is more or less the same with our HOG based M-WDPM, which proves the stronger discrimination of our windows selection method. When using the deep features with additional training data from ImageNet [34], our M-WDPM-deep can achieve an mAP of 24.3%. The boost (1.7%) is not as much as that of the annotation task (4%, see Section III-B3), but still it suffers from the Sim and BG error (as shown in Fig. 8(c)), which validates that our rescoring method M-WDPM-rescore has less error caused by the distribution of TP and each kind of FP in Fig. 8(a). We can see that the majority errors are due to poor localizations (Loc) and confusion with background regions (BG). When adopting the deep features, our M-WDPM-deep encounters less Loc and confusion with similar objects (Sim) and background regions (BG). Our detection rescoring method M-WDPM-rescore continues to improve the average precision (mAP = 27.4%) for nearly all the classes except for the motorbike class. It shows a better performance when compared with [42], [43], and it has a competitive performance when compared with [20]. The performance gap (3.5%) between ours and that of [19] might be partly caused by the use different deep feature representations as discussed in III-B3. We achieve the best detection results for the boat, cat, horse and train classes for this dataset.

In addition, we provide the results obtained by popular supervised object detection methods [2], [36], [4] in the bottom lines of Table IV. One can see that there is still a gap between the weakly supervised detection framework and supervised ones, although our weakly supervised DPM yields better results for some classes (e.g., aeroplane, bird, cat, dog, etc.) to the supervised DPM 5.0 [2].

5) Error analysis: We present an analysis of the types of errors that our M-WDPMs make on the PASCAL VOC 2007 test set in Fig. 8. We use the diagnosis tool of [48] and consider four types of false positive (FP) errors: Loc (poor localizations), Sim (confusion with similar objects), Oth (confusion with other objects, e.g., correctly localize an object but classifying it to a wrong class) and BG (confusion with background or unlabeled objects). Cor indicates correctly located true positives (TP). We visually show the fraction of correct detections (TP) and errors of each kind (FP) among the top ranking $T$ windows in Fig. 8, where $T$ is the number of ground-truth object windows in the test set of PASCAL VOC 2007.

We consider the M-WDPM-HOG as our baseline and show the distribution of TP and each kind of FP in Fig. 8(a). We can see that the majority errors are due to poor localizations (Loc) and confusion with background regions (BG). When adopting the deep features, our M-WDPM-deep encounters less Loc and Oth, but still it suffers from the Sim and BG error (as shown in Fig. 8(b)). In contrast, after detection rescoring, our best performing method M-WDPM-rescore has less error caused by Loc, BG and Oth (Fig. 8(c)), which validates that our rescoring
approach is very efficient in excluding the background regions and avoiding the misclassification. Fig. 8(d) shows the error distribution of the state-of-the-art supervised object detection framework NoC (Networks on Convolutional feature maps) [49]. NoC adopt even deeper VGG-16 [50] nets with bounding box fine-tuning on PASCAL VOC 2007+2012 trainval. The comparison between NoC and our M-WDPM indicates that: (1) deeper network helps increasing the Cor substantially; (2) fine-tuning and supervised training with ground-truth bounding boxes yield far less Sim and Oth errors.

IV. CONCLUSION

In this paper, we proposed a model enhancing the weakly supervised learning by emphasizing the importance of location and size of the initial class specific root filter of deformable part-based models. We follow the general setup of [17] and introduce several substantial improvements to the weakly supervised DPM. The main contributions included a new selection model based on generic “objectness” (region proposals) and visual saliency to adaptively select a reliable set of candidate windows which tend to represent the object instances in the image, and a latent class learning process by coarsely classifying a candidate window into either target object or non-target class. Furthermore we designed a flexible enlarging-and-shrinking post-processing procedure to modify the output bounding boxes of DPM, which can effectively further improve the final accuracy. Experimental results on the challenging PASCAL VOC 2007 database according to various criteria demonstrate that our proposed framework is efficient and competitive with the state-of-the-arts.

REFERENCES
