GradingNet: Towards Providing Reliable Supervisions for Weakly Supervised Object Detection by Grading the Box Candidates

Qifei Jia[1,2], **Shikui Wei**[1,2*], **Tao Ruan**[1,2], **Yufeng Zhao**[3], **Yao Zhao**[1,2]
[1]Institute of Information Science, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China
[3]China Academy of Chinese Medical Sciences, Beijing, China
[1,2]{18120303,shkwei,16112064,yzhao}@bjtu.edu.cn, [3]snowmanzhao@163.com

Abstract

Weakly-Supervised Object Detection (WSOD) aims at training a model with limited and coarse annotations for precisely locating the regions of objects. Existing works solve the WSOD problem by using a two-stage framework, i.e., generating candidate bounding boxes with weak supervision information and then refining them by directly employing supervised object detection models. However, most of such works focus mainly on the performance-boosting of the first stage, while ignoring the better usage of generated candidate bounding boxes. To address this issue, we propose a new two-stage framework for WSOD, named GradingNet, which can make good use of the generated candidate bounding boxes. Specifically, the proposed GradingNet consists of two modules: Boxes Grading Module (BGM) and Informative Boosting Module (IBM). BGM generates proposals of the bounding boxes by using standard one-stage weakly-supervised methods, then utilizes the Inclusion Principle to pick out highly-reliable boxes and evaluate the grade of each box. With the above boxes and their grade information, an effective anchor generator and a grade-aware loss are carefully designed to train the IBM. Taking the advantages of the grade information, our GradingNet achieves state-of-the-art performance on COCO, VOC 2007, and VOC 2012 benchmarks.

Introduction

Object detection aims at locating and recognizing objects of interest in given images of various scenes. For a long period of time, a number of methods were proposed to solve the challenges of object detection (Lin et al. 2017; Dai et al. 2016; Wang et al. 2019). Although remarkable progress has been achieved, it is time-consuming and labor-intensive to annotate accurate object bounding boxes for a dataset. Therefore, weakly-supervised object detection (WSOD), which only uses image-level labels for training, is considered a promising solution to the problem.

Traditionally, most of the previous methods (Bilen, Pedersoli, and Tuytelaars 2015; Song et al. 2014; Li et al. 2016; Hoffman et al. 2015) attempt to address the WSOD problem by employing Multiple Instance Learning (MIL) network. In particular, they first decompose images into object proposals and then use MIL to iteratively perform proposal selection and classifier estimation. Though many exciting results have been achieved, they are still far from comparable to fully supervised methods (Girshick 2015; Ren et al. 2015; Redmon et al. 2016). This is mainly because fully supervised methods make use of the strong learning ability of CNN to fit the datasets with accurate region-level annotations. Therefore, multiple instance learning networks can only solve part of the WSOD problem, and some methods treat it as a sub-task of WSOD and then carry out a series of follow-up processing to pursue higher performance.

Recently, some works (Tang et al. 2017, 2018; Yang, Li, and Dou 2019) solve the WSOD problem with a two-stage framework, which uses the top-ranked proposal produced by the weakly-supervised method to train a supervised detector. Since the top-ranked proposal only finds one ground-truth for each category, it will lose many informative proposals in complex visual scenes. To handle this problem, W2F (Zhang et al. 2018) reports a PGE algorithm to better find the pseudo ground-truth. However, since some weakly-supervised detectors are unstable, the generated proposals have some randomness. The mechanical processing method, like PGE, sometimes results in worse effects.

In addition to the above shortcomings, these methods also have a common problem, i.e., they focus only on the performance-boosting of the first stage (weakly-supervised) and assume that the pseudo ground-truth produced by it is accurate in the second stage. However, due to the inherent shortcomings of the weakly-supervised detector, the detected proposals are generally incomplete and inaccurate. Using these proposals as pseudo ground-truth to train a fully-supervised detector in the second stage (fully-supervised) will lead to two problems. Firstly, some of the anchors generated by using the pseudo ground-truth will be misclassified, which will greatly mislead the model. Secondly, lots of pseudo ground-truths can only cover a small and discriminative part of objects, while some can cover the whole object regions.

To address the above-mentioned problems, we propose a new solution for the WSOD problem, which divides it into a preliminary WSOD problem and a mixed problem of incomplete and inaccurate supervision (Zhou 2018). Boxes Grading Module (BGM) and Informative Boosting Module (IBM) are carefully designed to solve the two-stage problem separately. BGM evaluates the quality of each propos-

*Corresponding author

al to make Graded-labels. Specifically, we use the stability of the model to find stable boxes, and then utilize the Inclusion Principle to find out high-quality proposals and make them into Graded-Labels. IBM trains a detector by using these Graded-Labels. Different from the traditional two-stage method, we do not see the second stage as a fully-supervised process. Instead, we regard it as an incomplete supervised problem (the bounding boxes in Graded-Labels are incomplete) and an inaccurate supervision problem (the coordinates of bounding boxes are inaccurate). To tackle the first problem, IBM generates anchors in a predictive way. To solve the second one, a grade-aware loss is adopted by IBM. Besides, we also propose Area-balanced Sampling to control the proportion of positive and negative samples.

In brief, there are four main contributions in this paper: 1) A new two-stage framework is proposed for solving the WSOD problem, which divides the WSOD problem into a preliminary WSOD problem and a mixed problem of incomplete and inaccurate supervisions. 2) A Boxes Grading Module is proposed to grade the generated proposals, which provides more reliable bounding boxes for the following supervised stage. 3) An Informative Boosting Module is proposed for using the Graded-labels produced by BGM to train a supervised detector, which further improves detection performance. 4) The proposed method achieves state-of-the-art performance on COCO, VOC 07, and VOC 12 benchmarks.

Related Work

Weakly Supervised Learning: Common weakly supervised learning methods are Multiple Instance Learning (MIL) and Latent Variable Learning (LVL). MIL splits the image into positive and negative parts, each image is considered as a bag of candidate object instances. Most existing WSOD methods (Gokberk Cinbis, Verbeek, and Schmid 2014; Cinbis, Verbeek, and Schmid 2017; Wang et al. 2015; Hoffman et al. 2015) treat the problem as a MIL problem. However, positive object instances sometimes focus on the most discriminative parts of an object instead of the whole region, which causes the inaccurate object localization of detectors. In addition, since the underlying MIL optimization is non-convex, it is sensitive to positive instance initialization and tends to get trapped in local optima.

Some LVL algorithms are also used to solve the WSOD problem. Clustering methods (Song et al. 2014; Tang et al. 2018) recognize latent objects by finding the most discriminative clusters. Latent SVM (Yu and Joachims 2009; Ye et al. 2017) optimizes the learned object locations by Expectation-Maximization algorithm. Entropy based methods (Miller et al. 2012; Bouchacourt, Nowozin, and Pawan Kumar 2015; Wan et al. 2018) use entropy in LVL to measure the randomness of object localization during the learning process. Unfortunately, these methods often become stuck in a poor local minimum just like MIL.

Two-stage WSOD approaches: Recently, some WSOD methods (Zhang et al. 2018; Tang et al. 2017; Wei et al. 2018; Tang et al. 2018; Yang, Li, and Dou 2019; Bilen and Vedaldi 2016; Kantorov et al. 2016) follow a two-stage procedure which uses the strong regression ability of fully-supervised detector to guide weakly-supervised detection. In the first stage, the MIL network is used, which uses CNN as detectors to activate regions of interest on the feature maps and localizes objects by leveraging spatial distributions and informative patterns captured in the convolutional layers. In the second stage, a fully supervised detector is trained to further refine object location by using the selected boxes of the first phase as supervision. The main functionality of the second stage is to regress the object locations more precisely. In this paper, we design Boxes Grading Module (BGM) to process the information produced by the first stage and help with better training in the second stage.

Fully-supervised detector used in WSOD: Object detection has been widely studied in the last few decades, many methods have been proposed, such as the Fast R-CNN (Girshick 2015), Faster R-CNN (Ren et al. 2015) and other methods (Lin et al. 2017; Dai et al. 2016; Wang et al. 2019) based on them. Specifically, Faster R-CNN improved Fast R-CNN and has achieved breakthrough in speed. Though great progress has been achieved, fully-supervised methods still require accurately bounding-box annotations, which are expensive and time-consuming. In this paper, we design Informative Boosting Module (IBM) based on fully-supervised methods to train detector and use Graded-Labels produced by BGM to replace bounding-box annotations.

Method

The overview of GradingNet is illustrated in Figure 1, which include two main modules: Boxes Grading Module (BGM) and Informative Boosting Module (IBM). The BGM utilizes a novel inclusion principle to select high-quality regions from object proposals and categorizes them into various Graded-Labels. Then, IBM makes use of those Graded-Labels to train a more accurate detector.

Inclusion Principle

For a long period of time, a quantity of WSOD works is proposed to solve the challenges of locating the complete object. Although remarkable progress has been achieved, most of the methods are only able to locate the discriminative parts of objects. To tackle this problem, we first propose a simple Inclusion Principle.

Overview: The idea of the Inclusion Principle comes from the relationship of object parts, i.e., the whole object contains part of object, which contains the most discriminative part. For example, the human body contains upper part of human, and the upper part of human contains head. When the detector detects many parts within an object, we can use this principle to judge which part is the most complete.

Details: Inclusion is an abstract conception and we use the following Eq.(1) to quantify it. For boxes A and B, we define Intersection over Single (IoS) to measure the degree of inclusion,

$$IoS\left(A,B\right)=\frac{A\cap B}{B} \tag{1}$$

Under this definition, if $IoS\ (A,\ B) > T_{IoS}$, T_{IoS} is the overlap threshold, we determine that A include B.

Based on the principle, if we locate the most discriminative part of object, the whole object will be found. Therefore, we first propose a method to find the discriminative part.

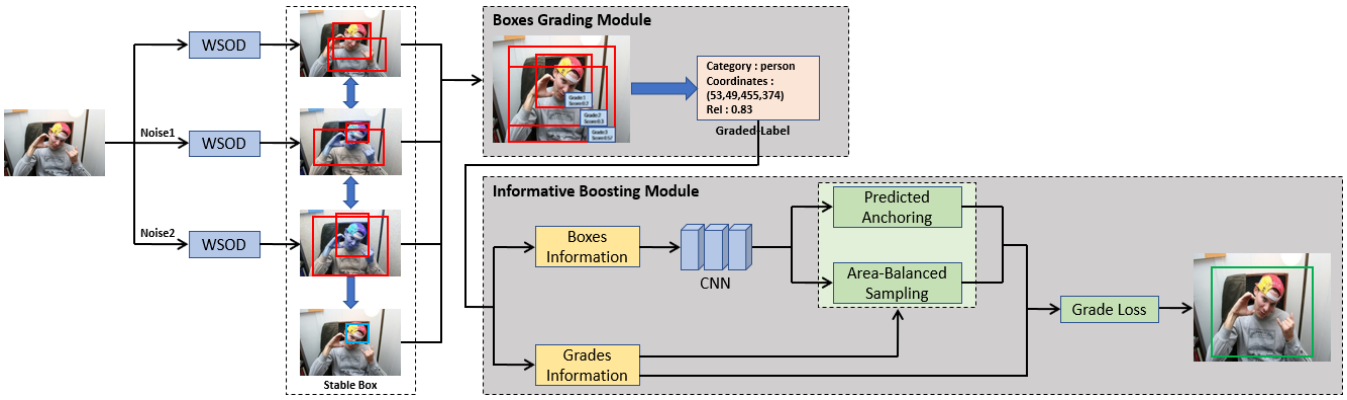


Figure 1: Overview of the proposed GradingNet, WSOD modules can adopt any existing one-stage weakly-supervised methods.

Finding the Stable Boxes

A well-trained detector can localize discriminative parts in an image, even though the image is perturbed by a batch of noises. According to this characteristic, the discriminative part is named stable box and we expect to utilize this kind of stability of a detector to find them. Hence, we first augment the dataset with our carefully designed noises, then feed them into a pre-trained weakly supervised detector and take the predicted boxes as candidates. Specifically, for each image $I^i \in D, i = 1, 2, ..., |D|$ in the dataset D, we generate two kinds of additive noises, δ_1 and δ_2. δ_1 is generated by randomly changing 0.5% of pixels while the other is changing 1% of pixels. We add these noises on the original image to get two perturbed images, i.e., $I^i_{\delta_1} = I^i + \delta^i_1$ and $I^i_{\delta_2} = I^i + \delta^i_2$. The original images together with the noisy images from the augmented dataset D_+:

$$\begin{aligned} D_+ &= D \cup D_{\delta_1} \cup D_{\delta_2} \\ D_{\delta_1} &= \{I^i_{\delta_1} = I^i + \delta^i_1 | i = 1, 2, ..., |D|\} \\ D_{\delta_2} &= \{I^i_{\delta_2} = I^i + \delta^i_2 | i = 1, 2, ..., |D|\} \end{aligned} \tag{2}$$

A weakly supervised detector takes the images in D_+ as the input and outputs predicted box proposals. The design of such a detector is beyond the scope of this paper, therefore, we apply an existing method (e.g., OICR (Tang et al. 2017)) to execute the above process. For each image I in D_+, a collection P_+ of the box candidates with confidence scores is generated:

$$P_+ = \{(b^i_+, s^i_+) | b^i_+ \in R^4, s^i_+ \in R, i = 1, ..., |P|\} \tag{3}$$

where b^i_+ represents a box candidate, and s^i_+ is the corresponding confidence score within $[0, 1]$. To coarsely purify the predictions, we perform NMS with a loose threshold T_{nms1} on each P_+.

Then, we use the stability of weakly-supervised method to find stable boxes. Considering the collections P, P_{δ_1} and P_{δ_2} on the same image in D, D_{δ_1} and D_{δ_2}:

$$\begin{aligned} P &= \{b_i | b_i \in R^4, i = 1, ..., |P|\} \\ P_{\delta_1} &= \{b^i_{\delta_1} | b^i_{\delta_1} \in R^4, i = 1, ..., |P|\} \\ P_{\delta_2} &= \{b^i_{\delta_2} | b^i_{\delta_2} \in R^4, i = 1, ..., |P|\} \end{aligned} \tag{4}$$

For a certain box A in P, we find the boxes B and C closest to it from P_{δ_1} and P_{δ_2} according to the IoU. And the stability (ST) of box A can be calculated as:

$$ST_A = \lambda_1 IoU(A, B) + \lambda_2 IoU(A, C) \tag{5}$$

where λ_1 and λ_2 are parameters that weight the importance of B and C, respectively. The process is repeated until the ST of all boxes in P, P_{δ_1} and P_{δ_2} are calculated and in which NMS operates on it, only the boxes whose ST larger than a pre-defined threshold T_{st} constitutes the stable boxes collection P_{st}. The collection P_{ot} which consists of all boxes except stable boxes is confirmed in the meantime.

$$\begin{aligned} P_{st} &= \{b^i_{st} | b^i_{st} \in R^4, i = 1, ..., |P|\} \\ P_{ot} &= \{b^i_{ot} | b^i_{ot} \in R^4, i = 1, ..., |P|\} \end{aligned} \tag{6}$$

Boxes Grading Module

The BGM relying on the Inclusion Principle use stable boxes to select high-quality regions from object proposals. Given a set of proposals $P = \{p_1,...,p_N\}$ in image X, the corresponding confidence scores output by the one-stage weakly-supervised detector is unreliable. Therefore, we first initialize a set of scores $S = \{s_1,...,s_N\}$ and a set of grades $G = \{g_1,...,g_N\}$. The scores are used as a standard to identify a high-quality subset P_g of boxes. And the grades further evaluate the quality of selected boxes.

We initialize the sets of S and G according to the Inclusion Principle. Each box is represented as a node in the oriented graph (Figure 2), and two nodes are defined as inclusion if the IoS (Eq.(1)) of their corresponding boxes is below a threshold (solid line in Figure 2). The BGM provides a method to find the high-quality region (node 1 in Figure 2): To encourage selecting regions containing complete object, we define a way to update the scores of parent nodes (nodes 1 and 2 in Figure 2) using their children nodes.

$$Score_A = Score_A + Score_B IoS(A, B) \tag{7}$$

Based on the Inclusion Principle, if regions contain more complete object boxes, their corresponding nodes will have more children nodes. For this reason, the score S of a more complete object box is updated more times and gets a larger

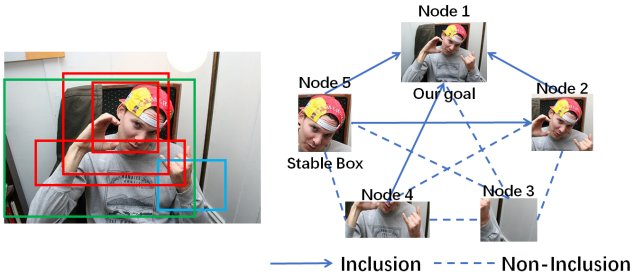


Figure 2: All the detected proposals (left) and oriented graph representation of them (right). Our goal is to select the green boxes and we achieve this by analyzing the inclusion relationship between proposals.

value. And its grade G is directly assigned to a higher value than its children node.

Besides, to suppress selecting regions containing many objects. The score of their corresponding parent node can only be updated by a single child node with the same grade G. If a parent node has many children nodes with $G = i$, its score is updated only once. However, if it also has child node with $G = j$ ($i \neq j$), its score can be updated normally. The BGM algorithm is summarized in Algorithm 1.

We obtain the boxes P_g required by Graded-Labels and the grade G_g of it. Although G_g can evaluate the quality of P_g, we still require a better parameter to avoid the influence of different grade distribution on different category. So we count the total number m of boxes in each category (all images) and use the parameter m and G (each box) with category = n to calculate the reliability (Rel) using Eq.(8). The parameter Rel replace G_g are included in Graded-Labels to measure the quality of a box.

$$Rel_A = \frac{G_A}{G_{avg}} = \frac{m * G_A}{\sum_{cls=n} G} \tag{8}$$

Informative Boosting Module

IBM trains a detector by making use of CNN to fit the Graded-Labels and further improve detection performance. All components will be detailed in the following.

Predicted Anchoring: Since the bounding boxes in labels are incomplete, if we generate anchors on the whole feature map, some of the anchors will be misclassified. Inspired by GA-RPN (Wang et al. 2019), we predict the position of the anchor's center point to generate anchors around the Graded-Labels. Specifically, we perform a 1×1 convolution on the feature map, dividing the output into two signal-channel maps. The element-wise sigmoid function is applied on those two maps to get two probability maps FC1 and FC2. Each value in FC1 (FC2) illustrates how likely this position is a center point of a positive (negative) anchor.

To train the center point probability matrix of positive/negative anchors separately, we use two binary label maps where 1 represents a valid location to place the center point of anchor and 0 represents other regions. We first map

Algorithm 1 Boxes Grading Module

Input: Image X in Datasets D_+; Stable boxes collection P_{st}; Other boxes collection P_{ot}; T_{nms2}; T_{IoS}
Output: Boxes P_g

$S_{st} \leftarrow \{0.2\}^N$; $S_{ot} \leftarrow \{0.1\}^N$; $S = S_{st} \cup S_{ot}$
$G_{st} \leftarrow \{1\}^N$; $G_{ot} \leftarrow \{0\}^N$; $G = G_{st} \cup G_{ot}$
$k \leftarrow 0$; $i \leftarrow 1$
while $k \neq 2$ **do**
  for $A \in P_{ot}$ and $B \in P_{st} \cup P_{ot}$ **do**
    if $G_B = i$ and $IoS(A, B) > T_{IoS}$ **then**
      $G \leftarrow G_A \leftarrow i + 1$
      $S \leftarrow S_A \leftarrow$ Compute S using Eq.(7)
      if $k \geqslant 0$ **then**
        $k \leftarrow k$ - 1
      end if
    else
      $i \leftarrow i + 1$
      $k \leftarrow k + 1$
    end if
  end for
end while
$P_g = nms(P_{st} \cup P_{ot}, S, T_{nms2})$

the bounding box (x_g, y_g, w_g, h_g) which represents a box with w_g width, h_g and (x_g, y_g) as the center point to the corresponding feature map scale, and obtain (x'_g, y'_g, w'_g, h'_g). For positive anchor, we generate center box (x'_g, y'_g, Kw'_g, Kh'_g), the center point generated in it. For negative anchors, we generate outbox1 (x'_g, y'_g, $(K+1)w'_g$, $(K+1)h'_g$) and outbox2 (x'_g, y'_g, Lw'_g, Lh'_g), the center point generated in the region which included by outbox1 but excluded by outbox2.

After the center point is determined, we generate 3 scales with box areas of 128/256/512 squares pixels, and 3 aspect ratios of 1:1, 1:2, and 2:1 in each point of the positive/negative center points region. For training, in the region of positive/negative center points, each point chooses one proposal which has the largest/smallest IoU with Graded-Labels in 9 proposals. For testing, we choose all the 9 proposals as anchors. The whole process is shown in Figure 3.

Area-balanced Sampling: For the problem of unbalanced positive and negative samples in Graded-Labels, we propose a novel solution. Because of the Predicted Anchoring, the number of positive/negative anchors is directly determined by the area of center points region. We can control the sample proportion by dynamically adjusting parameters K and L mentioned in Predicted Anchoring. Since the bounding boxes with higher Rel (Eq.(8)) are more accurate and the anchor generated near it has higher quality, we set $K = Rel$. According to experience, the ratio of positive and negative samples is 1:3, and we can get the Eq.(9),

$$\frac{K^2}{(K+1)^2 - \mathrm{L}^2} = \frac{1}{3} \tag{9}$$

then parameter L is determined by the following Eq.(10)

$$L = \sqrt{(K+1)^2 - 3K^2} \tag{10}$$

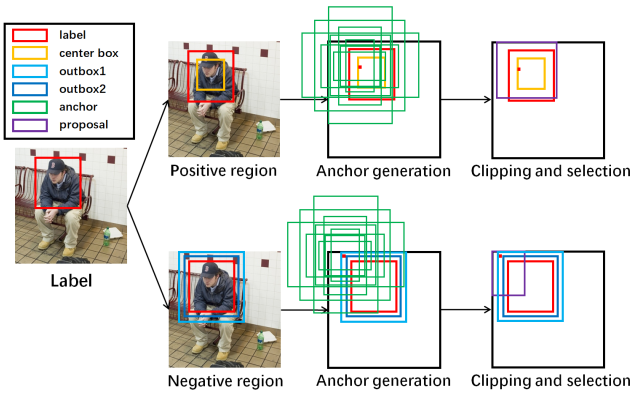


Figure 3: Overview of the predicted anchoring. The center point of positive anchor is included by the center box and negative anchor is included by the outbox1 but excluded by the outbox2. We generate 9 size anchors on all the center point and choose 1 proposal which has the largest/smallest IoU with Graded-Labels as positive/negative sample.

Grade Loss: Our IBM is trained with the following multi-task loss:

$$L_{multi} = \lambda_c L_{cls} + \lambda_r L_{reg} + \lambda_p L_{pa} \tag{11}$$

where L_{cls} is the commonly used classification loss in detection tasks (Ren et al. 2015), L_{reg} is the regression loss including grade information and L_{pa} is an additional loss for the predicted anchor localization.

For L_{reg}, since using the bounding box with low Rel to train a model will lead to inaccurate location. A proper way is to link regression loss with Rel to adjust the proportion of loss provided by labels with different Rel, as follows:

$$L_{reg} = \sum_i Rel_i p_i^* L_r\left(t_i\ ,\ t_i^*\right) \tag{12}$$

where t_i is a vector representing the 4 coordinates of the predicted bounding box, and t_i^* is that of the box in Graded-Labels associated with a positive anchor. L_r represents smooth L_1 loss. Such loss makes the box with lower reliability have less impact on the regression loss.

For L_{pa}, we use the sum of two log losses over two classes (positive vs Non-positive samples, negative vs Non-negative samples).

$$L_{pa} = \sum_i L_{log}\left(P_i\ ,\ P_i^*\right) + \sum_j L_{log}\left(N_j\ ,\ N_j^*\right) \tag{13}$$

Experiments

Datasets and Evaluation Metrics

The training datasets we use in all experiments are three challenging benchmarks in object detection: PASCAL VOC 2007 (5011 images for training, 4952 images for testing), PASCAL VOC 2012 (Everingham et al. 2015) (11540 images for training, 10991 images for testing) datasets which are widely used as benchmarks for WSOD. And MS COCO 2014 (Lin et al. 2014) dataset (about 80K images for training, 40K images for validation) which is the popular dataset used for supervised object detection but rarely used in W-SOD. In all experiments, we only use image-level labels.

For evaluation on VOC 2007 and 2012, we use two kinds of measurements: 1) Average Precision (AP) and the mean of AP (mAP) on the test set. 2) CorLoc on the trainval set. All metrics are based on the PASCAL criterion. For evaluation on MS COCO, we use two main metrics AP and AP_{50} which are the standard MS COCO criterion.

Implementation Details

All the experiments are implemented based on PyTorch on 4 NVIDIA GeForce GTX 2080Ti. All the settings of our four baselines (OICR, PCL, MELM and C-MIL) are kept identical to (Tang et al. 2017; Tang et al. 2020; Wan et al. 2018, 2019) For BGM, both λ_1 and λ_2 are set to 0.5. The threshold T_{nms1} and T_{st} for NMS are set to 0.8 and 0.7 respectively. The thresholds T_{nms2} and T_{IoS} for NMS is set to 0.7 and 0.6 respectively. For IBM, VGG16 model is adopted as our backbone network. We set λ_c and λ_r to 1 and λ_p to 0.5. The batch size is set to 16, initialize the learning rate as 1×10^{-3}, and then decrease it to 1×10^{-4} and 1×10^{-5} at 3 epochs and 6 epochs, eventually stop at 7 epochs.

Ablation Studies

We conduct some ablation experiments on the COCO 2014 and VOC 2007 datasets, which include the influence of each part in GradingNet and the individual influence of IBM.

Influence of the GradingNet: We present the GradingNet as a combination of Head, Neck and Body part. Each part contains some alternative approaches. The experimental results are shown in Table 1. We can observe that even using the simple Neck and Body (TS and Fast R-CNN), the results are improved compared to only use that original weakly-supervised method, i.e., 0.6% AP improvement in OICR and 0.9% AP improvement in PCL, 0.5% AP_{50} improvement in OICR and 0.3% AP_{50} improvement in PCL. We attribute this to the selective ability of the Neck and regression ability of the Body.

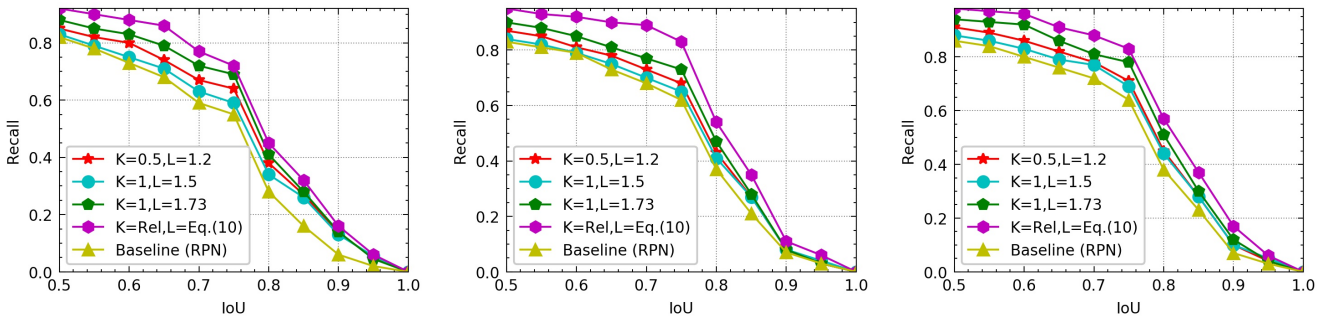
1) The choice of Neck: When we use the same Body (Fast R-CNN, Faster R-CNN or IBM), our BGM performs better than TS and PGE in most evaluation metrics. For example, we use OICR as Head and Fast R-CNN as Body, combining with the BGM designed by us, the framework achieves 8.3% AP, which is 0.8% higher than the framework that uses TS as Neck and 0.6% higher than the framework that uses PGE as Neck respectively. We can observe that only our BGM has significant improvement in all weakly-supervised methods, while TS and PGE perform poorly in MELM. This proves that the BGM which use the instability of weakly-supervised detector is more general.

2) The choice of Body: Similarly, when we use the same Neck (TS, PGE or BGM), our IBM also performs better than Fast R-CNN and Faster R-CNN in most evaluation metrics. This is mainly because our IBM has a better fit ability to the Graded-Labels.

Influence of the IBM: To study the influence of IBM, we compute the proposal Recall at different IoU thresholds with

Head		Neck			Body			Metric					
OICR	PCL	TS	PGE	BGM (ours)	Fast	Faster	IBM(ours)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓								6.9	16.5	6.1	3.1	7.8	12.9
✓		✓			✓			7.5	17.0	6.3	3.2	8.4	13.1
✓		✓				✓		7.3	17.1	5.8	3.1	8.6	13.5
✓		✓					✓	7.8	18.0	7.0	3.1	8.9	14.1
✓			✓		✓			7.7	18.5	7.3	3.0	9.0	14.7
✓			✓			✓		7.2	17.4	7.0	3.1	8.5	13.6
✓			✓				✓	7.9	18.6	8.5	3.0	9.2	15.3
✓				✓	✓			8.3	18.9	8.1	**3.3**	9.4	16.0
✓				✓		✓		8.0	19.0	8.2	3.2	9.1	15.4
✓				✓			✓	**8.7**	**20.1**	**8.5**	**3.3**	**9.8**	**16.9**
								+1.8	**+3.6**	**+2.4**	**+0.2**	**+2.0**	**+4.0**
	✓							8.3	19.1	7.6	3.3	9.5	15.3
	✓	✓			✓			9.2	19.4	8.1	3.3	10.2	16.2
	✓	✓				✓		8.9	19.1	8.0	3.4	10.5	16.0
	✓	✓					✓	9.5	19.3	8.2	3.2	10.5	16.8
	✓		✓		✓			9.4	18.6	8.3	3.1	10.7	15.8
	✓		✓			✓		9.3	18.9	8.3	3.2	10.4	16.5
	✓		✓				✓	9.5	19.0	8.1	3.4	10.9	16.5
	✓			✓	✓			10.1	22.6	8.2	**3.6**	10.6	17.0
	✓			✓		✓		9.9	21.4	7.8	3.5	10.4	16.3
	✓			✓			✓	**10.8**	**22.9**	**9.0**	3.5	**11.1**	**17.3**
								+2.5	**+3.8**	**+1.4**	**+0.2**	**+1.6**	**+2.0**

Table 1: Ablation study on COCO 2014 validation set. TS represents only using top-scoring proposals. PGE is proposed in W2F (Zhang et al. 2018) and we reproduce it according to the paper. Fast/Faster represent Fast/Faster R-CNN.



(a) 300 proposals (b) 1000 proposals (c) 2000 proposals

Figure 4: IoU-Recall curve for different number of proposal (300, 1000 and 2000) methods on the VOC 2007 test set. Both IBM and RPN are trained by Graded-Labels.

ground-truth and draw the IoU-Recall curve. As shown in Figure 4, our method obtains higher Recall than the RPN (Ren et al. 2015). We also report the influence of different parameters K and L in IBM on Recall. It shows that the values of K and L which strictly follow our Method section are optimal. We attribute this to the Predicted Anchoring and Area-balanced Sampling in IBM.

Comparison with State-of-the-Art

PASCAL VOC: For fairly compare with most state of the art WSOD works, we evaluate our method on the PASCAL VOC datasets. Table 2 shows the mAP on VOC 07 test set, we only show the AP of 10 object categories, but mAP is calculated based on all the 20 object categories. We present the performance of our GradingNet on the four baselines. Added the GradingNet, OICR, PCL, MELM and C-MIL achieve 48.1%, 49.4%, 52.5% and 54.3% mAP, respectively, which are significantly higher than the baselines and the baselines re-trained by fully-supervised detector. In addition, Our GradingNet-C-MIL results (54.3% mAP) surpass all the previous methods. Table 3 shows the Corloc on VOC 07 trainval set. Similarly, added the GradingNet, OICR, PCL, MELM and C-MIL achieve 68.6%, 69.1%, 63.2% and 72.1% Corloc, respectively, which are also higher than the baselines and the baselines re-trained by fully-supervised detector. GradingNet-C-MIL achieves the highest result (72.1%) and surpass all the previous methods.

Table 3 also shows our performance in terms of mAP and Corloc on the VOC 12 test and trainval sets. The results also show the huge improvement between our methods and baselines. And our GradingNet-C-MIL also achieves the highest mAP (50.5%) and the second highest CorLoc (71.9%).

Method	aero	boat	bottle	car	chair	dog	horse	mbike	person	plant	mAP
OICR (Tang et al. 2017)	58.5	16.9	17.4	60.8	8.2	31.3	51.9	64.8	13.6	23.1	42.0
PCL (Tang et al. 2020)	57.1	16.9	18.8	63.7	17.0	33.2	54.4	68.3	16.8	25.7	45.8
MELM (Wan et al. 2018)	55.6	29.1	16.4	68.1	25.0	53.2	49.6	68.6	2.0	25.4	47.3
C-MIL (Wan et al. 2019)	62.5	32.1	19.8	66.1	20.0	53.5	57.4	68.9	8.4	24.6	50.5
WSOD2 (Zeng et al. 2019)	65.1	**39.2**	24.3	66.2	29.8	60.1	71.2	70.7	21.9	28.1	53.6
C-MIDN (Gao et al. 2019)	53.3	26.1	20.3	69.9	28.7	64.6	58.0	**71.2**	20.0	27.5	52.6
OIM (Lin et al. 2020)	55.6	27.9	21.1	68.3	21.3	54.5	56.5	70.1	12.5	25.0	50.1
SLV (Chen et al. 2020)	**65.6**	37.1	24.6	70.3	30.8	61.4	65.3	68.4	12.4	29.9	53.5
OICR+FRCNN (Tang et al. 2017)	65.5	21.6	22.1	68.5	5.7	30.3	64.7	66.1	13.0	25.6	47.0
PCL+FRCNN (Tang et al. 2020)	63.2	22.6	27.3	69.1	12.0	37.3	63.3	63.9	15.8	23.6	48.8
C-MIL+FRCNN (Wan et al. 2019)	61.8	28.9	18.9	69.6	18.5	66.9	65.9	65.7	13.8	22.9	53.1
C-MIDN+FRCNN (Gao et al. 2019)	54.1	26.4	22.2	68.9	25.2	70.3	66.3	67.5	21.6	24.4	53.6
OIM+FRCNN (Lin et al. 2020)	53.4	26.0	27.7	69.7	21.4	63.7	63.7	67.4	10.9	25.3	52.6
SLV+FRCNN (Chen et al. 2020)	62.1	34.5	25.6	67.4	24.2	**71.6**	**72.0**	67.2	12.1	24.6	53.9
GradingNet-OICR (ours)	63.2	23.4	23.2	68.3	10.6	41.3	60.1	68.2	20.2	22.9	48.1
GradingNet-PCL (ours)	60.3	25.2	21.1	64.6	15.0	54.0	58.8	65.4	22.7	20.1	49.4
GradingNet-MELM (ours)	57.3	31.6	31.0	71.8	29.0	65.6	71.0	68.7	**29.2**	24.1	52.5
GradingNet-C-MIL (ours)	61.8	**39.2**	**31.6**	**75.0**	**33.9**	61.9	71.3	64.4	28.2	**33.2**	**54.3**

Table 2: Average precision (%) on the PASCAL VOC 2007 test set.

Method	VOC 07	VOC 12	
	CorLoc(%)	mAP	CorLoc(%)
OICR (Tang et al. 2017)	61.2	38.2	63.5
PCL (Tang et al. 2020)	63.0	41.6	65.0
MELM (Wan et al. 2018)	61.4	42.4	-
C-MIL (Wan et al. 2019)	65.0	46.7	67.4
WSOD2 (Zeng et al. 2019)	69.5	47.2	71.9
C-MIDN (Gao et al. 2019)	68.7	50.2	71.2
OIM (Lin et al. 2020)	67.2	45.3	67.1
SLV (Chen et al. 2020)	71.0	49.2	69.2
OICR+FRCNN (Tang et al. 2017)	64.3	42.5	65.6
PCL+FRCNN (Tang et al. 2020)	66.6	44.2	68.0
C-MIDN+FRCNN (Gao et al. 2019)	71.9	50.3	**73.3**
OIM+FRCNN (Lin et al. 2020)	68.8	46.4	69.5
GradingNet-OICR (ours)	68.6	44.3	68.9
GradingNet-PCL (ours)	69.1	47.0	69.3
GradingNet-MELM (ours)	63.2	48.6	62.8
GradingNet-C-MIL (ours)	**72.1**	**50.5**	71.9

Table 3: CorLoc (%) on the PASCAL VOC 2007 trainval set, mAP (%) and CorLoc (%) on the PASCAL VOC 2012 test and trainval sets.

Method	AP	AP_{50}
PCL (Tang et al. 2020)	8.5	19.4
C-MIDN (Gao et al. 2019)	9.6	21.4
WSOD2 (Zeng et al. 2019)	10.8	22.7
OICR+FRCNN (Tang et al. 2017)	7.7	17.4
PCL+FRCNN (Tang et al. 2020)	9.2	19.6
GradingNet-OICR (ours)	8.7	20.1
GradingNet-PCL (ours)	10.8	22.9
GradingNet-MELM (ours)	11.0	22.6
GradingNet-C-MIL (ours)	**11.6**	**25.0**

Table 4: AP and AP_{50} on the COCO 2014 validation set.

MS COCO: We also report the results on COCO 2014 validation set in Table 4. It is worth mentioning that a few methods report results on COCO dataset. Nevertheless, we present the performance of our GradingNet on four baselines. All the four methods we present achieve high performance and our GradingNet-C-MIL achieve 11.6% AP and 25.0% AP_{50}, which create a new state-of-the-art.

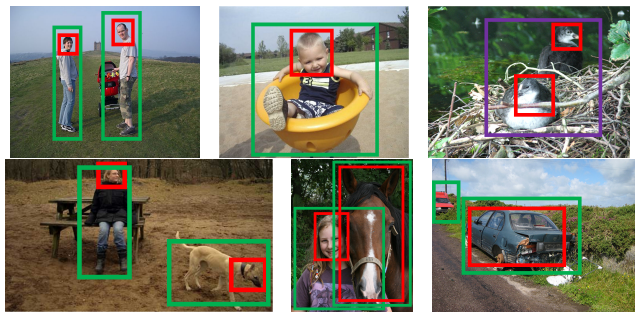


Figure 5: Example results by GradingNet-C-MIL and C-MIL. Green/purple boxes indicate correct/failure cases by GradingNet-C-MIL, and red ones indicate cases by C-MIL.

Qualitative Results and Discussion

Figure 5 exhibits some cases of GradingNet. We can obverse that GradingNet can well contain the whole object, while there remains a challenge to solve the detection problem in close or overlapping objects. It is worth mentioning that for the "person" class, our detection performance is better than most weakly-supervised object detectors. The reason is that the Inclusion Principle we proposed caters to the structure of the human body. In addition, to the best of our knowledge, our GradingNet is a rare framework that focuses on the better usage of candidate bounding boxes generated by standard one-stage weakly-supervised methods. This field that traditional WSOD works ignore is worth studying.

Conclusion

In this paper, we propose a novel framework GradingNet, which regards the classical WSOD problem as a preliminary WSOD problem and a mixed problem of incomplete and inaccurate supervision. We deal with the former problem through the Boxes Grading Module (BGM) and tackle the latter problem through the Informative Boosting Module (IBM). The proposed GradingNet achieves state-of-the-art performance on COCO, VOC 07 and VOC 12 benchmarks.

Acknowledgments

This work is supported in part by National Key Research and Development of China (2017YFC1703503), in part by National Natural Science Foundation of China (61972022, 61532005, U1936212), and in part by the Fundamental Research Funds for the Central Universities(2018JBZ001) .

References

Bilen, H.; Pedersoli, M.; and Tuytelaars, T. 2015. Weakly supervised object detection with convex clustering. In *CVPR*, 1081–1089.

Bilen, H.; and Vedaldi, A. 2016. Weakly supervised deep detection networks. In *CVPR*, 2846–2854.

Bouchacourt, D.; Nowozin, S.; and Pawan Kumar, M. 2015. Entropy-based latent structured output prediction. In *ICCV*, 2920–2928.

Chen, Z.; Fu, Z.; Jiang, R.; Chen, Y.; and Hua, X.-S. 2020. SLV: Spatial Likelihood Voting for Weakly Supervised Object Detection. In *CVPR*, 12995–13004.

Cinbis, R. G.; Verbeek, J.; and Schmid, C. 2017. Weakly Supervised Object Localization with Multi-Fold Multiple Instance Learning. *T-PAMI* 39(1): 189–203.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 379–387.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *IJCV* 111(1): 98–136.

Gao, Y.; Liu, B.; Guo, N.; Ye, X.; Wan, F.; You, H.; and Fan, D. 2019. C-MIDN: Coupled Multiple Instance Detection Network With Segmentation Guidance for Weakly Supervised Object Detection. In *ICCV*, 9834–9843.

Girshick, R. 2015. Fast r-cnn. In *ICCV*, 1440–1448.

Gokberk Cinbis, R.; Verbeek, J.; and Schmid, C. 2014. Multi-fold mil training for weakly supervised object localization. In *CVPR*, 2409–2416.

Hoffman, J.; Pathak, D.; Darrell, T.; and Saenko, K. 2015. Detector discovery in the wild: Joint multiple instance and representation learning. In *CVPR*, 2883–2891.

Kantorov, V.; Oquab, M.; Cho, M.; and Laptev, I. 2016. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 350–365. Springer.

Li, D.; Huang, J.-B.; Li, Y.; Wang, S.; and Yang, M.-H. 2016. Weakly supervised object localization with progressive domain adaptation. In *CVPR*, 3512–3520.

Lin, C.; Wang, S.; Xu, D.; Lu, Y.; and Zhang, W. 2020. Object Instance Mining for Weakly Supervised Object Detection. In *AAAI*, 11482–11489.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*, 2117–2125.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Miller, K.; Kumar, M. P.; Packer, B.; Goodman, D.; and Koller, D. 2012. Max-margin min-entropy models. In *AISTATS*, 779–787.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *CVPR*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 91–99.

Song, H. O.; Lee, Y. J.; Jegelka, S.; and Darrell, T. 2014. Weakly-supervised discovery of visual pattern configurations. In *NIPS*, 1637–1645.

Tang, P.; Wang, X.; Bai, S.; Shen, W.; Bai, X.; Liu, W.; and Yuille, A. 2020. PCL: Proposal Cluster Learning for Weakly Supervised Object Detection. *T-PAMI* 42(1): 176–191.

Tang, P.; Wang, X.; Bai, X.; and Liu, W. 2017. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2843–2851.

Tang, P.; Wang, X.; Wang, A.; Yan, Y.; Liu, W.; Huang, J.; and Yuille, A. 2018. Weakly supervised region proposal network and object detection. In *ECCV*, 352–368.

Wan, F.; Liu, C.; Ke, W.; Ji, X.; Jiao, J.; and Ye, Q. 2019. C-MIL: Continuation multiple instance learning for weakly supervised object detection. In *CVPR*, 2199–2208.

Wan, F.; Wei, P.; Jiao, J.; Han, Z.; and Ye, Q. 2018. Min-entropy latent model for weakly supervised object detection. In *CVPR*, 1297–1306.

Wang, J.; Chen, K.; Yang, S.; Loy, C. C.; and Lin, D. 2019. Region proposal by guided anchoring. In *CVPR*, 2965–2974.

Wang, X.; Zhu, Z.; Yao, C.; and Bai, X. 2015. Relaxed multiple-instance SVM with application to object discovery. In *ICCV*, 1224–1232.

Wei, Y.; Shen, Z.; Cheng, B.; Shi, H.; Xiong, J.; Feng, J.; and Huang, T. 2018. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In *ECCV*, 434–450.

Yang, K.; Li, D.; and Dou, Y. 2019. Towards precise end-to-end weakly supervised object detection network. In *ICCV*, 8372–8381.

Ye, Q.; Zhang, T.; Ke, W.; Qiu, Q.; Chen, J.; Sapiro, G.; and Zhang, B. 2017. Self-learning scene-specific pedestrian detectors using a progressive latent model. In *CVPR*, 509–518.

Yu, C.-N. J.; and Joachims, T. 2009. Learning structural svms with latent variables. In *ICML*, 1169–1176.

Zeng, Z.; Liu, B.; Fu, J.; Chao, H.; and Zhang, L. 2019. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *ICCV*, 8292–8300.

Zhang, Y.; Bai, Y.; Ding, M.; Li, Y.; and Ghanem, B. 2018. W2f: A weakly-supervised to fully-supervised framework for object detection. In *CVPR*, 928–936.

Zhou, Z.-H. 2018. A brief introduction to weakly supervised learning. *National Science Review* 5(1): 44–53.