Weakly- and Semi-supervised Faster R-CNN with Curriculum Learning

Jiasi Wang, Xinggang Wang, Wenyu Liu
School of Electronic Information and Communications
Huazhong University of Science and Technology
{wangjiasi, xgwang, liuwy}@hust.edu.cn

***Abstract*—Object detection is a core problem in computer vision and pattern recognition. In this paper, we study the problem of learning an effective object detector using weakly-annotated images (i.e., only the image level annotation is given) and a small proportion of fully-annotated images (i.e., bounding box level annotation is given) with curriculum learning. Our method is built upon Faster R-CNN. Different from previous weakly-supervised object detectors which rely on hand-craft object proposals, the proposed method learns a region proposal network using weakly- and semi-supervised training data. The weakly-labeled images are fed into the deep network in the order of from easy to complex; the process is formulated as curriculum learning. We name the Faster R-CNN trained using Weakly- And Semi-Supervised data with Curriculum Learning as WASSCL R-CNN. WASSCL R-CNN is validated on the PASCAL VOC 2007 benchmark, and obtains 90% of a fully-supervised Faster R-CNN's performance (measured using mAP) with only 15% of fully-supervised annotations together with image-level annotations for the rest images. The results show that the proposed learning framework can significantly reduce the labeling efforts for obtaining reliable object detectors.**

I. Introduction

Recently, object detection based on deep convolutional neural networks has obtained great performance gain and has been successfully applied in many problems. However, current deep learning based object detectors rely on a large amount of fully-annotated training images. In this paper, we mainly study how to learn an accurate object detector using small amount of human labeling efforts.

The problem of reducing human labeling efforts in learning object detector is a significant topic. There are many papers [1], [2], [3], [4], [5], [6] work on the weakly-supervised object detection problem, in which, only the image-level annotation is given. The state-of-the-art method [4] uses a multi-branch deep network to infer the category label of every hand-crafted object proposal. Since the number of hand-crafted object proposals is large (typically, 1000-2000 per-image), the efficiency and performance of weakly-supervised object detection is limited. In this paper, we use a portion of fully-annotated training images to remedy this problem. Besides of only using image-level supervision for object detection, there are some other solutions toward reducing supervision in training object detectors. For example, Redmon and Farhadi [7] proposed to train a 9000 classes object detector with some object class having full supervision and the other object classes only containing image-level supervision. However, in [7], it still requires a large number fully-annotated training images for many object classes. The weak supervision of human interaction has also been considered. Papadopoulos et al. [8] proposed to use human verification in the process of weakly-supervised object detector learning, which requires manual inspection in the loop. Further, Papadopoulos et al. [9] also proposed to use click supervision for training object detectors, which requires to click all bounding boxes. However, in this paper, the WASS setting only requires to annotate 15% bounding boxes with 85% image-level labels and obtains better results than [9], [8], which suggests the proposed method is a more effective approach for reducing annotation for training object detectors.

The general idea of the proposed WASSCL R-CNN is illustrated in Fig. 1. The accurate bounding box information of the fully-annotated images can produce a well performance region proposal network (RPN) o Faster R-CNN. In weakly-supervised object detection, there is no bounding box annotations, thus the weakly-supervised object detectors have difficulties on generating object locations. Weakly-supervised object detectors rely on hand-craft object proposals and only focus on classifying object proposals. The RPN remedies this flaw and boosts performance. After RPN is learned, the weakly-supervised detector learning problem can be improved using curriculum learning. The concept of curriculum learning is proposed by Bengio et al. [10], indicating that learning from easy to hard examples can be beneficial. It has been applied to various problems in computer vision [11], [12], [13], [14], where people present different definitions to "easy" examples. Some require human labelers to evaluate the difficulty level of images [12] while others measure the easiness based on labeling time [14]. In our work, we determine the easiness by training a SVM classifier using part of fully annotated data, without requiring additional human labeling.

There are lots of images with only image-level labels in our setting, we use multi-instance learning (MIL) [15] to dig information in them. While MIL algorithms are easily to convergence to poor local optima. To avoid the situation that directly adding all weakly-supervised data may deteriorate the performance of deep detectors, it is natural to combine curriculum learning which can be seen as a general strategy for global optimization of non-convex functions [10] to add the weakly-supervised data for training from "easy" to "hard".

In summarize, we propose a novel solution termed WASSCL R-CNN to the WASS object detection problem. Different from

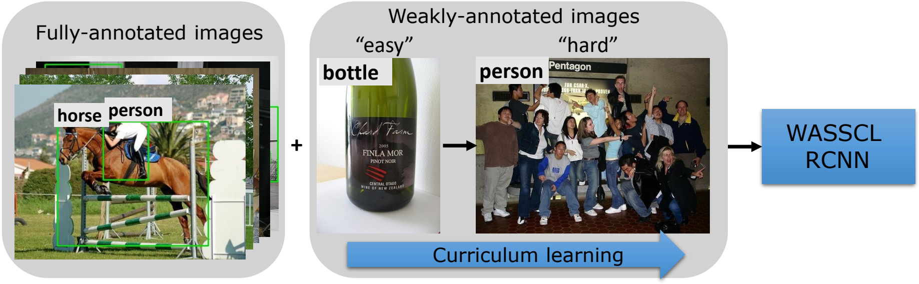

Fig. 1. Illustration of the proposed WASSCL R-CNN. At first, a Faster R-CNN is learned using a portion of randomly selected fully-annotated training images. Then, the weakly-annotated images which have image-level labels are gradually fed into the WASSCL network following the order of from "easy" to "hard" for both classifying region proposals and improving the region proposal network.

previous weakly-supervised learning methods, it gets rid of hand-crafted object proposals. Its all components are integrated in a single deep network, optimized in a end-to-end manner via stochastic gradient decent (SGD) with curriculum learning. WASSCL R-CNN is evaluated on a challenging object detection benchmark (PASCAL VOC 2007). It outperforms the state-of-the-art by a significant margin with very few amount of fully-annotated training images. Besides, its performance gets very closing to the fully-supervised Faster R-CNN and its speed is as fast as Faster R-CNN.

II. Method

Our detector is a derivation of Faster R-CNN [16] and trained in a stage-wise fashion, denoted as WASS R-CNN, combined WASS R-CNN with curriculum learning, denoted as WASSCL R-CNN. In this section, we present the details about the architecture of WASS R-CNN detector, the training strategy and curriculum learning.

A. Architecture

A WASS R-CNN detector is a cascade of Faster R-CNN and K classifiers. There are both fully and weakly annotated images in training dataset, and their supervision information are in different structure. For fully using them, we design different proposal classifier block respectively. The overall architecture of our method is shown in Fig. 2.

1) Data stream for fully annotated images: This part for fully annotated images is exactly similar to the original Faster-RCNN work, the detail is just the same as in [16].

Given an image I with bounding box $\mathbf{B} \in \mathbb{R}^{m\times 4}$ and category of each box $\mathbf{y} \in \mathbb{R}^{(C+1)\times m}$. Feed it into the Faster RCNN network, after the feature extractor and proposal generator (RPN module), we get some region proposals $R = (R_1, R_2, ..., R_n)$.

We sample the proposals to keep the ratio of positive samples to negative samples is approximately $1 : 3$ during a mini-batch, then the proposal feature extractor extracts features for these carefully chosen proposals. The RCNN module performs bounding box regression and classification based on these features, shown by the blue arrow in Fig. 2.

The loss is comprised of two classification losses and two regression losses.

$$L_{fully} = \mathrm{L}_{\mathrm{RPN}} + \mathrm{L}_{\text{Fast R-CNN}} \tag{1}$$

$$L_{RPN} = \frac{1}{N_{cls}} \sum_{i=1} \mathrm{L}_{cls}(\tilde{p_i}, p_i) + \frac{1}{N_{reg}} \sum_{i=1} p_i \mathrm{L}_{reg}(\tilde{B_i}, B_i) \tag{2}$$

$$L_{FastR-CNN} = \frac{1}{N_{cls}} \sum_{i=1} \mathrm{L}_{cls}(\tilde{y_i}, y_i) + \tag{3}$$

$$\frac{1}{N_{reg}} \sum_{i=1} y_i \mathrm{L}_{reg}(\tilde{B_i}, B_i) \tag{4}$$

Here, the L_{cls} is the log loss, the L_{reg} is robust $smooth_{L_1}$ loss. Vector with $\sim$ means prediction values.

2) Data stream for weakly annotated images: Given an image I with image-level label $\mathbf{y} \in \mathbb{R}^{(C+1)}$ fed into the network, lots of proposals with objectness scores will be generated by RPN module.

While the ratio between object and non-object samples is seriously imbalanced. Since the objectness score indicates the probability that there is object contained in this region proposal, those proposals whose scores are smaller than a threshold (here we set it 0.03) will be removed. After this filter step, many negative proposals have been removed. Thus, the searching space of the objects and the complexity of computation could be dramatically reduced for the next fully connection layers.

Since we set a loose criterion in the previous filter step, the top-ranked proposals are still coarse and may contain many false positive samples. We apply MIL [15] to mine confident candidates for proposal classifier shown by the red arrow in Fig. 2. Treat each image as a bag that contain lots of instances. If the bag is positive, there is at least one positive instance in the bag. On the contrary the bag is negative, all the instances in the bag are negative. For each proposal the classifier produce a $C + 1$ dimension vector ϕ that predicts whether it contains object and which kind of object it contains. Let ϕ^n pass through a Max pooling layer, where n is the number of proposals, we get a $C + 1$ dimension vector that can be supervised by the image-level label $\mathbf{y} = [y_0, y_1, y_2, ..., y_C]^T \in \mathbb{R}^{(C+1)\times 1}$ with a

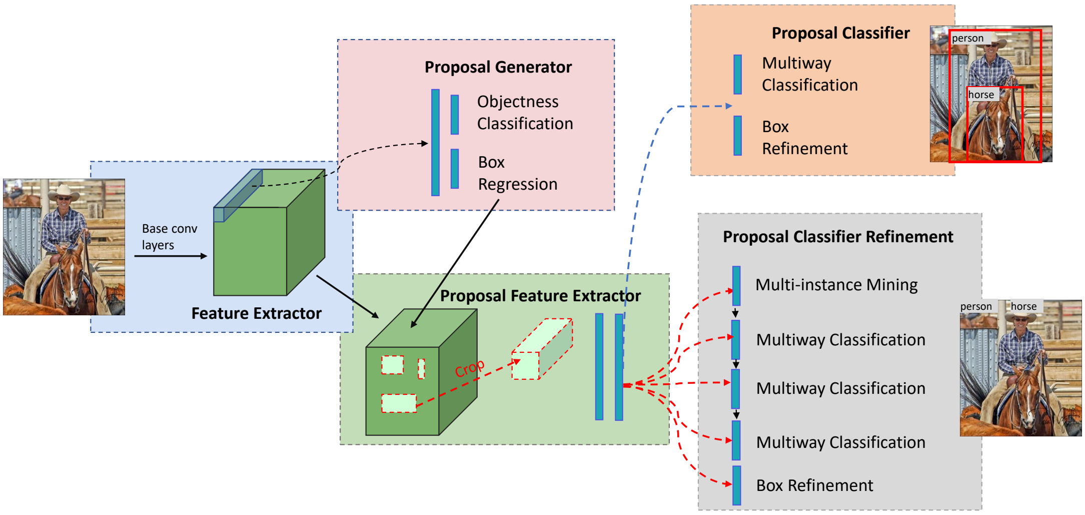

Fig. 2. The network architecture of WASS Faster R-CNN, which is a cascade of Faster R-CNN and K proposal refinement classifiers. WASS R-CNN is designed to jointly use fully and weakly labeled images. If the image is fully-annotated, the proposal classifier is shown as the blue arrow, if the image is weakly-annotated, the proposal classifier is shown as red arrow.

standard multi-class cross entropy loss shown in Eq. (5)

$$\mathrm{L}_{\text{muli\_label}} = -\sum_{c=0}^{C}\{y_c \log \phi_c + (1-y_c)\log(1-\phi_c)\}. \quad (5)$$

However, with only the image-level supervision, the top-ranking proposals may fail to meet the standard object detection requirement (IoU>0.5 between ground truths and predicted boxes). They may only capture objects partially, or capture too much background.

To address this problem, we take the online classifier refinement strategy proposed by [4]. The motivation is that the proposal with top score may failed to contain just the whole object well, the proposals who have high overlaps with it may do or at least have a greater IOU with the ground truths. The weakly annotated images is similar to the original work of [4] with an important difference: the RPN module is jointly trained with the cascade of classifier. For more details of the classification subnet, please refer to the original OICR paper [4]

After this classifiers refinement stage, we use the pseudo ground truths produced by the last detector the guide the RPN network to learn to regress proposals, which is the same as in [7] Since the RPN and the Fast R-CNN detectors share the base convolutional layers. Without refining RPN, the performance will degrade heavily.

The loss for weakly annotated data is shown as Eq. (6)

$$\mathrm{L}_{\text{weakly}} = \mathrm{L}_{RPN} + \mathrm{L}_{\text{muli\_label}} + \sum_{k=0}^{K}\mathrm{L}_{\text{cls}}^{k} \quad (6)$$

B. *Training strategy*

As we mentioned before that for weakly annotated images, we use RPN to generate proposal and the objectness scores to filer negative proposals. The premise is that the RPN and detector are accurate enough to do "self-training". So first, we use all the fully annotated images to train a Faster R-CNN detector. We use it to initialize WASSCL R-CNN. Then we refine the WASSCL jointly using both strong and weakly annotated images.

During the refining stage, that the proportion of weakly annotated images is relatively large must be taken into consideration. We apply the basic detector on the weakly labeled data, choose proposals with top scores for classes that present in the image. Add these extra boxes to pseudo ground truth produced by the last classifier to guide RPN module training, which can be regarded as a format of knowledge transfer between teacher model (basic detector) and student model (WASS R-CNN).

During observing the detection results that obtained by applying the basic detector on weakly labeled images, we found that for some of them the detector can predict as precise as the ground truth. That means for the detector trained with these strong annotated data, these images are "easy". While for some others, the performance is far from good, these images are "hard". It is natural to combine curriculum learning which is a kind of learning paradigm that is inspired by the learning process of humans and animals in the "self-training" refinement stage. The samples are not learned randomly but organized in a meaningful order which illustrates from easy to gradually more complex examples. First, we use all the fully labeled images and "easy" weakly labeled images to fine tune basic detector, after some iterations; add these a little more complex examples, training for some iterations; finally use all the training images for fine tuning, then we get the WASSCL R-CNN detector.

C. *Learning the "easy" vs. "hard" classifier for curriculum learning*

So how to evaluate the complexity of image with only the image-level label? It can not be judged by weakly labels arbitrarily. If the image-level is a one-hot vector, that is to say there is only one class presents in this image. If there are many objects of this class and many occlusions, it is still “hard”. We think the complexity is related to the strong annotated training data as well. The image that has the similar object class and space distribution with training data is “easy”.

Inspired by [17], we use the mAPI (mean average precision per image) shown as Eq. (7) to evaluate the complexity.

$$\text{mAPI} = -\frac{1}{|C|}\sum_{c=0}^{C} AP_i \tag{7}$$

To get the complexity of all the weakly labeled data, we first split the strong labeled data into two parts, the ratio of them is about 5:3, use the larger part to train a detector. Then apply the detector on the smaller part, we can obtain the detection results $\{V_C, V_S, V_B\}$, where V_C denotes the category probability over C classes, V_S denotes the confidence scores of proposals, V_B denotes the coordinates. We extract features based on the top N proposals with highest confidence scores to determine whether this image is “easy” or “hard” using a binary linear support vector machine (SVM) classifier. Suppose we have a set of training examples $(x_i, y_i), 1 \leq i \leq n,\ y_i \in \{-1, 1\}$ (hard, easy) is the label of image i, a linear classifier $sgn(w^\top x + b)$ is learned by solving the following SVM problem

$$\min_{w,b} \frac{1}{2} w^\top w + C\sum_{i=1}^{n} \xi_i \tag{8}$$

$$s.t. \quad y_i(w^\top w + b) \geq 1 - \xi_i, \xi_i \geq 0, 0 \leq i \leq 1 \tag{9}$$

In which C is a hyperparameter that balances between large margin and small empirical error, and ξ is the cost associated with the $i-th$ image x_i [17].

We organize the feature with $\{V_C, V_S, V_B, Y\}$, Y denotes the image-level labels. The reason why we use $\{V_C, V_S, V_B, Y\}$ is that the difference between V_C and Y can reflect that whether the detector makes a right classification, the V_S reflects the confidence and V_B shows the space relationship of each proposal.

With the bounding box annotations, we can get the mAPI of each image in the small part of fully annotated images and evaluate the complexity of each image with it: the image is “easy”, if mAPI is greater than 0.9, “hard” with the mAPI smaller than 0.1, normal otherwise. Next, we use the organized features and complexity labels to train a SVM classifier, so we can get the complexity information of the all weakly labeled data by this classifier.

III. Experiments

In this section, we will present the results and detailed analysis of the proposed WASSCL R-CNN.

A. Datasets and Evaluation Measures

We evaluate our method on PASCAL VOC 2007 dataset [22] , which is a very challenging and commonly used benchmark in object detection. The dataset contains 2501 training images, 2510 validation images and 4952 test images in 20 object class. For each image, bounding box annotations are given in the task of object detection. In the experiments, for all object detection method, the train and val sets are used for training and the test set is used for testing. Thus, it has 5011 images for training and 4952 images for testing. For evaluation, we use the standard mean average precision (mAP) metric.

B. Implementation Details

As described in Method Section, the proposed WASSCL R-CNN contains two stages: The Faster R-CNN detector training stage, the weakly-supervised training stage with curriculum learning. Even without combining curriculum learning, the WASS R-CNN detector can accomplish the weakly- and semi-supervised object detection task. WASS R-CNN is much stronger than the state-of-the-art weakly supervised method. And WASSCL R-CNN is stronger than WASS R-CNN. In the experiments, we use two different base networks: VGG_M and VGG_16 pretrained on the ImageNet dataset [23].

For training Faster R-CNN using fully-supervised training images, we randomly select 15% of the images in the trainval set (811 images) and use SGD to optimize the network. The mini-batch size is set to 1. For the first 30K iterations, the learning rate is set to 0.001; for the next 20K, it decades to 0.0001. The momentum and weight decay are set to 0.9 and 0.0005 respectively. And the online refinement classifier numbers is 3 for all the experiments (i.e., $K = 3$). We denote this detector as Faster_15

In the training of WASS R-CNN, we jointly use both fully-supervised and weakly-supervised data as well as the pseudo labels produced by Faster_15 to guide RPN at the same time. For taking full advantage of the fully labeled training images, if the bounding box annotations of the image are provided, we replace the pseudo labels (produced by Faster_15) with the well-annotated labels. The mini-batch size, optimizer, momentum and weight decay are the same as the previous stage.

For training WASSCL R-CNN, we randomly select 511 images (about 10% percent of the whole training dataset) from the 811 fully-supervised images to train a Faster R-CNN detector, denoted as Faster_10, and apply it on the rest 300 images. Using the detection results and their bounding box annotations, we compute mAPI and train SVM classifier with extracted features. We find that the feature vector organized in the format “20+20+(conf+4s)5 (The first 20 denotes the image-level labels and the latter denotes a 20-dim histogram formed by predict class labels. 5 denotes we use top 5 proposals) can get the best performance SVM classifier, shown as Table II.

Since the training data for Faster_15 detector is the whole fully labeled data while we split it for training SVM classifier, which may change the distribution of classes, and produce some false hard examples. To reduce the impact of this situation, we make a 10-fold multi cross validation and take the voting results of the 10 SVM classifiers.

With the information of complexity, we first feed the fully-supervised and easy data into network, after 17k iterations, add the images with normal complexity, then training for 25k

TABLE I
AVERAGE PRECISION (IN %) FOR DIFFERENT METHODS ON VOC 2007 TEST SET. THE FIRST PART SHOWS THE RESULTS OF TWO EFFECTIVE WEAKLY-SUPERVISED OBJECT DETECTORS (WSSDN AND OICR). THE SECOND PART SHOWS THE RESULTS OF THE FULLY-SUPERVISED FAST R-CNN DETECTOR. THE THIRD PART SHOWS THE RESULTS OF WASS AND WASSCL.

Method	aero	bike	bird	boat	bott	bus	car	cat	chair	cow	table	dog	horse	mbike	perso	plant	sheep	sofa	train	tv	mAP
WSDDN VGG_M [3]	43.6	50.4	32.2	26.0	9.8	58.5	50.4	30.9	7.9	36.1	18.2	31.7	41.4	52.6	8.8	14.0	37.8	46.9	53.4	47.9	34.9
WSDDN VGG_16 [3]	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
OICR VGG_M [4]	53.1	57.1	32.4	12.3	15.8	58.2	56.7	39.6	0.9	44.8	39.9	31.0	54.0	62.4	4.5	20.6	39.2	38.1	48.9	48.6	37.9
OICR VGG_16 [4]	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	41.2
WCCN VGG_16 [18]	49.5	60.6	38.6	29.2	16.2	70.8	56.9	42.5	10.9	44.1	29.9	42.2	47.9	64.1	13.8	23.5	45.9	54.1	60.8	54.5	42.8
Jie VGG_16 [19]	54.2	52.0	35.2	25.9	15.0	59.6	67.9	58.7	10.1	67.4	27.3	37.8	54.8	67.3	5.1	19.7	52.6	43.5	56.9	62.5	43.7
DPM [20]	32.8	56.8	2.5	16.8	28.5	39.7	51.6	21.3	17.9	18.5	25.9	8.8	49.2	41.2	36.8	14.6	16.2	24.4	39.2	39.1	30.2
Fast R-CNN VGG_M [21]	70.9	70.9	62.5	46.7	28.0	70.9	72.7	77.4	33.7	66.6	61.6	70.3	74.8	69.8	62.2	30.1	59.6	62.1	70.0	65.4	61.3
Faster R-CNN VGG_M [a]	66.8	70.3	58.4	44.9	34.5	67.9	75.9	68.7	41.3	62.9	57.1	64.4	77.1	71.0	67.0	33.4	58.5	57.0	68.1	62.8	60.4
Fast R-CNN VGG_16 [21]	77.6	78.6	71.0	61.3	39.6	78.6	78.2	83.5	43.7	74.4	67.7	82.0	81.4	75.5	67.8	32.2	68.0	69.1	78.6	70.2	69.0
Faster R-CNN VGG_16 [16]	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6	69.9
Faster R-CNN (15%) VGG_M	51.1	56.2	39.2	28.6	20.4	51.8	64.2	56.5	21.6	47.6	40.7	50.5	67.2	60.3	60	20.2	41.6	36.6	56.9	49.2	45.8
Faster R-CNN (15%) VGG_16	51.9	67.9	49.8	33.5	31.4	61.9	69.6	68.7	28.8	62.3	46.4	64.5	73.8	63	66.6	22.4	52.2	50	62.5	56.6	54.2
WASS R-CNN VGG_M	55	64.2	44.7	27.9	23.8	58.2	65.4	63.3	23.4	52.6	49.4	54.8	70.9	65.1	57.4	26.3	49.6	45.7	67.2	58.1	51.2
WASS R-CNN VGG_16	61	73.7	57.1	42	37	72.5	70.5	75.6	36.4	67	60	73	77.5	68.1	66.3	32.3	57.2	59.8	71.8	65.5	61.2
WASSCL R-CNN VGG_M	57.2	62.6	45	32.8	25.6	61.1	68.5	64.7	29.5	56.3	53.3	59.3	74.2	65.3	58.9	26.7	47.9	47.5	63.4	58.8	52.9
WASSCL R-CNN VGG_16	58.2	75.9	56.6	45.2	39.6	73.2	75.8	77.2	38.4	65.7	61	72.3	78.6	67.3	68.1	33	61.5	61.1	72.1	66.7	62.4

[a] We get the result by run the code provided by https://github.com/rbgirshick/py-faster-rcnn.

iterations, last use all the data, training for 30k iterations. The source code in the experiments will be released on publication.

C. Results

The main results of WASSCL R-CNN are given in Table I. Some of the detection results are visualized in Fig 3. In the table, we compared the proposed WASS R-CNN and WASSCL R-CNN to some weakly-supervised object detectors, including WSSDN [3] and OICR [4], and the fully-supervised Faster R-CNN detector. When using VGG_M as the base network, the mAPs over 20 classes are 37.9%, 51.2% 52.9% and 60.4% for OICR, WASS R-CNN, WASSCL R-CNN and Faster R-CNN, respectively. When using VGG_16 as the base network, the mAPs are 41.2%, 61.2% and 62.4% 69.9% for OICR, WASS R-CNN, WASSCL R-CNN and Faster R-CNN, respectively.

From the results, we can observe that: 1) By using the 15% fully-supervised training images, WASS R-CNN has 13.3% mAP and 20% mAP improvement over OICR using VGG_M and VGG_16 as base network respectively; WASSCL has 15% mAP and 21.2% mAP improvement over OICR using VGG_M and VGG_16 as base network respectively; 2) WASS R-CNN has a great improvement in some difficult classes for weakly-supervised detectors such as dog, cat and person; 3) After using the semi-supervised training data, the performance of WASS R-CNN has been significantly improved, combined with curriculum learning the performance can be improved greater, the WASSCL reaches 90% of fully supervised Faster R-CNN's mAP when using VGG_16 as the base network and 88% of the fully supervised Faster R-CNN's mAP when using VGG_M as the base network; The results are very impressive, since they are getting more and more closing the fully-supervised Faster R-CNN detectors.

TABLE II
THE ACCURACY OF SVM CLASSIFIER(TAKE THE AVERAGE OF 10 CLASSIFIER).

Model	easy sample	normal sample	hard sample
VGG_M	0.73	0.80	0.85
VGG_16	0.72	0.79	082

Since the 15% fully-supervised training images are randomly selected from the trainval set, it is necessary to check of the influence of the random selection. The mean standard deviations over the 20 classes are very small, close to 0.2% or 0.3%, which shows WASS R-CNNs are robust the randomly selection of the fully-supervised training images.

D. Training and testing speed

Here we give the training and testing speed of the proposed WASSCL R-CNN shown in Table III. WASSCL R-CNN is built upon Faster R-CNN, so benefit from the shared convolutional features, our WASSCL R-CNN is also almost proposal cost free and the speed during training and testing is faster as well, thanks to the fewer proposals. But for weakly labeled images, without bounding box information, WASSCL R-CNN set a loose criterion to filter proposals. The number of proposals after ROI pooling is more than it in Faster R-CNN, so there is more region wise computation in WASSCL. That is the main reason why the training and testing speed of WASSCL R-CNN is slower than Faster R-CNN. But they are close, especially in testing stage.

TABLE III
THE TRAINING AND TESTING RATE OF WASSCL R-CNN ON A TITAN X GPU WITH VGG_16.

Method	training	testing
Faster R-CNN	3.5 fps	5.9 fps
WASSCL	2.6 fps	5.7 fps

IV. CONCLUSION

We proposed an end-to-end deep framework for general object detection with weakly- and semi-supervised setting. It

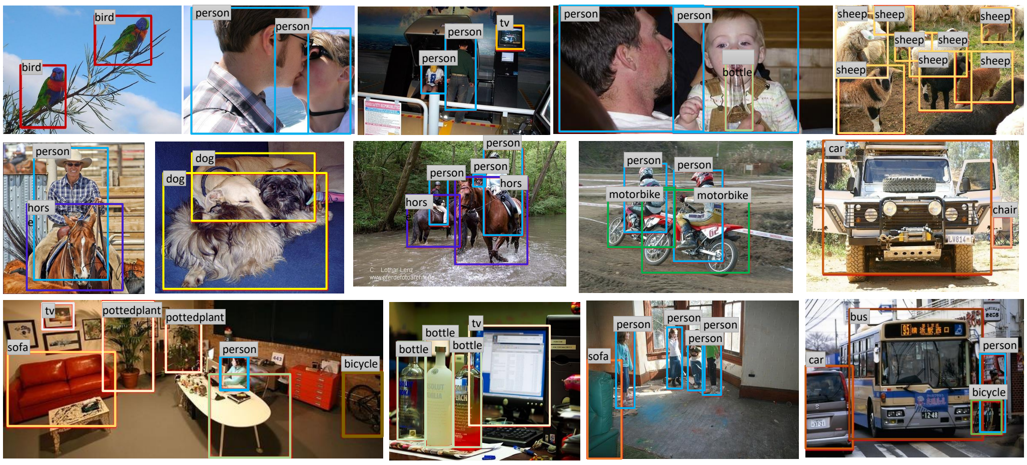

Fig. 3. Example detection results on PASCAL VOC 2007 test set of WASSCL R-CNN with VGG_M trained on 2007 trainval (52.9% mAP) using 15% fully-supervised training images. The score threshold 0.3 is used to display these images.

is easy to combine with curriculum learning and has achieved a very impressive detection accuracy on the challenging PASCAL VOC 2007 dataset. The results show that the proposed WASSCL R-CNN bridges the performance gap between weakly supervised object detectors and fully supervised ones. In future, we will focus on investing the power of transfer learning on the problem of training object detection with less human annotations.

V. Acknowledgement

This work was partly supported by NSFC (No.61733007, No.61503145) and the fund of HUST-Horizon Computer Vision Research Center. Xinggang Wang was sponsored by CCF-Tencent Open Research Fund, the Program for HUST Academic Frontier Youth Team, and Young Elite Sponsorship Program by CAST, No. YESS 20150077.

References

[1] P. Viola, J. C. Platt, C. Zhang *et al.*, “Multiple instance boosting for object detection,” in *NIPS*, vol. 2, 2005, p. 5.

[2] J.-Y. Zhu, J. Wu, Y. Xu, E. Chang, and Z. Tu, “Unsupervised object class discovery via saliency-guided multiple class learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 4, pp. 862–875, 2015.

[3] H. Bilen and A. Vedaldi, “Weakly supervised deep detection networks,” in *CVPR*, 2016, pp. 2846–2854.

[4] P. Tang, X. Wang, X. Bai, and W. Liu, “Multiple instance detection network with online instance classifier refinement,” in *CVPR*, 2017.

[5] M. Gao, A. Li, R. Yu, V. I. Morariu, and L. S. Davis, “C-wsl: Count-guided weakly supervised localization,” *arXiv preprint arXiv:1711.05282*, 2017.

[6] R. G. Cinbis, J. Verbeek, and C. Schmid, “Weakly supervised object localization with multi-fold multiple instance learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 189–203, 2017.

[7] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in *CVPR*, 2017.

[8] D. P. Papadopoulos, J. Uijlings, F. Keller, and V. Ferrari, “Training object class detectors with click supervision,” 2017.

[9] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, “We don’t need no bounding-boxes: Training object class detectors using only human verification,” in *CVPR*, 2016, pp. 854–863.

[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[11] Y. J. Lee and K. Grauman, “Learning the easy things first: Self-paced visual category discovery,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1721–1728.

[12] A. Pentina, V. Sharmanska, and C. H. Lampert, “Curriculum learning of multiple tasks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5492–5500.

[13] M. Shi and V. Ferrari, “Weakly supervised object localization using size estimates,” in *European Conference on Computer Vision*. Springer, 2016, pp. 105–121.

[14] R. Tudor Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, and V. Ferrari, “How hard can it be? estimating the difficulty of visual search in an image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2157–2166.

[15] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *Artificial Intelligence*, vol. 89, no. 1, pp. 31–71, 1997.

[16] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.

[17] H.-Y. Z. B.-B. Gao and J. Wu, “Adaptive feeding: Achieving fast and accurate detections by adaptively combining object detectors,” 2017.

[18] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, “Weakly supervised cascaded convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 914–922.

[19] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu, “Deep self-taught learning for weakly supervised object localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1377–1385.

[20] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[21] R. Girshick, “Fast r-cnn,” in *ICCV*, 2015, pp. 1440–1448.

[22] M. Everingham, A. Zisserman, C. K. Williams, L. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó *et al.*, “The pascal visual object classes challenge 2007 (voc2007) results,” 2007.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012, pp. 1097–1105.