DAP: Detection-Aware Pre-training with Weak Supervision

Yuanyi Zhong[1], Jianfeng Wang[2], Lijuan Wang[2], Jian Peng[1], Yu-Xiong Wang[1], Lei Zhang[2]

[1] University of Illinois at Urbana-Champaign {yuanyiz2, jianpeng, yxw}@illinois.edu
[2] Microsoft {jianfw, lijuanw, leizhang}@microsoft.com

Abstract

This paper presents a detection-aware pre-training (DAP) approach, which leverages only weakly-labeled classification-style datasets (e.g., ImageNet) for pre-training, but is specifically tailored to benefit object detection tasks. In contrast to the widely used image classification-based pre-training (e.g., on ImageNet), which does not include any location-related training tasks, we transform a classification dataset into a detection dataset through a weakly supervised object localization method based on Class Activation Maps to directly pre-train a detector, making the pre-trained model location-aware and capable of predicting bounding boxes. We show that DAP can outperform the traditional classification pre-training in terms of both sample efficiency and convergence speed in downstream detection tasks including VOC and COCO. In particular, DAP boosts the detection accuracy by a large margin when the number of examples in the downstream task is small.

1. Introduction

Pre-training and fine-tuning have been a dominant paradigm for deep learning-based object recognition in computer vision [14, 10, 29, 17]. In such a paradigm, neural network weights are typically pre-trained on a large dataset (*e.g.*, through ImageNet [8] classification training), and then transferred to initialize models in downstream tasks. Pre-training can presumably help improve downstream tasks in multiple ways. The low-level convolutional filters, such as edge, shape, and texture filters, are already well-learned in pre-training [42]. The pre-trained network is also capable of providing meaningful semantic representations. For example, in the case of ImageNet classification pre-training, since the number of categories is large (1000 classes), the downstream object categories might be related to a subset of the pre-training categories and can reuse the pre-trained feature representations. Pre-training may also help the optimizer avoid bad local minima by providing a better initial-
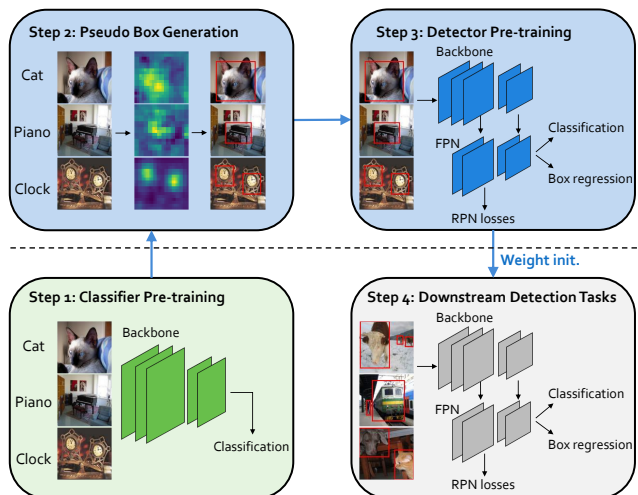

Figure 1. The DAP workflow. It consists of 4 steps: (1) Classifier pre-training on a weak supervision dataset, (2) Pseudo box generation by WSOL (*e.g.*, through CAM as illustrated), (3) Detector pre-training with the generated pseudo boxes, (4) Downstream detection tasks. The traditional classification pre-training and fine-tuning directly go from Step (1) to (4) at the bottom, while DAP inserts the additional Steps (2) and (3) at the top. In both cases, the pre-trained weights are used to initialize the downstream models. DAP gives the model a chance to learn how to perform explicit localization, and is able to pre-train detection-related components while classification pre-training cannot, such as the FPN, RPN, and box regressor in a Faster RCNN detector.

ization point than a completely random initialization [12]. Therefore, fine-tuning would only require a relatively small number of gradient steps to achieve competitive accuracy.

However, the empirical gain for object detection brought by classification pre-training is diminishing with successively larger pre-training datasets, ranging from ImageNet-1M, ImageNet-5k [17], to ImageNet-21k (14M), JFT-300M [36], and billion-scale Instagram images [25]. Meanwhile, [16] shows that training from random initialization (*i.e.*, from scratch) can work equally well with sufficiently large data (COCO [24]) and a sufficiently long training time, making the effect of classification pre-training questionable.

We conjecture that the diminishing gain of classifica-

tion pre-training for object detection is due to several mismatches between the pre-training and the fine-tuning tasks. Firstly, the task objectives of classification and detection are different. Existing classification pre-training is typically unaware of downstream detection tasks. The pre-training adopts a single whole-image classification loss which encourages translation and scale-invariant features, while the detection fine-tuning involves several different classification and regression losses which are sensitive to object locations and scales. Secondly, the data distributions are misaligned. The localization information required by detection is not explicitly made available in classification pre-training. Thirdly, the architectures are misaligned. The network used in pre-training is a bare backbone network such as a ResNet model [18] followed by an average pooling and a linear classification layer. In contrast, the network in an object detector contains various additional architectural components such as the Region Proposal Network (RPN) [29], the Feature Pyramid Network (FPN) [22], the ROI classification heads and the bounding box regression heads [29], *etc*. These unique architectural components in detectors are not pre-trained and are instead randomly initialized in detection fine-tuning, which could be sub-optimal.

Aiming at bridging the gap between pre-training with classification data and detection fine-tuning, we introduce a Detection-Aware Pre-training (DAP) procedure as shown in Figure 1. There are two desired properties that are necessary to pre-train a detector: (1) Classification should be done *locally* rather than globally; (2) Features should be *capable* of predicting bounding boxes and can be easily adapted to any desired object categories after fine-tuning. With the desired properties in mind, DAP starts from pre-training a classifier on the classification data, and extracts the localization information with existing tools developed in Weakly Supervised Object Localization (WSOL) based on Class Activation Maps (CAM) [47]. The next step is to treat the localized instances as pseudo bounding boxes to pre-train a detection model. Finally, the pre-trained weights are used for model initialization in downstream detection tasks such as VOC [13] and COCO [24]. DAP enables the pre-training of (almost) the entire detector architecture and offers the model the opportunity to adapt its representation to perform localization explicitly. Our problem setting focuses on leveraging the weak image-level supervision in classification-style data for pre-training (ImageNet-1M and ImageNet-14M) [8], therefore makes a head-to-head comparison to the traditional classification pre-training. Note that our setting is different from unsupervised pre-training [15, 4, 5] which is only based on unlabeled images, and is different from fully-supervised detection pre-training [32] which is hard to scale.

Comprehensive experiments demonstrate that adding the simple lightweight DAP steps in-between the traditional classification pre-training and fine-tuning stages yields consistent gains across different downstream detection tasks. The improvement is especially significant in the low-data regime. This is particularly useful in practice to save the annotation effort. In the full-data setting, DAP leads to faster convergence than classification pre-training and also improves the final detection accuracy by a decent margin. Our work suggests that a carefully designed detection-specific pre-training strategy with classification-style data can still benefit object detection. We believe that this work makes the first attempt towards detection-aware pre-training with weak supervision.

2. Related Work

Pre-training and fine-tuning paradigm. Pre-training contributed to many breakthroughs in applying CNN for object recognition [14, 10, 29, 17]. A common strategy, for example, is to pre-train the networks through supervised learning on the ImageNet classification dataset [8, 30] and then fine-tune the weights in downstream tasks. Zeiler *et al.* visualize the convolutional filters in a pre-trained network, and find that intermediate layers can capture universal local patterns, such as edges and corners that can be generalizable to other vision tasks [42]. Pre-training may ease up the difficult optimization problem of fitting deep neural nets via first-order methods [12]. Recently, the limit of supervised pre-training has been pushed by scaling up the datasets. In Big Transfer (BiT), the authors show that surprisingly high transfer performance can be achieved across 20 downstream tasks by classification pre-training on a dataset of 300M noisy-labeled images (JFT-300M) [5]. Notably, pre-training on JFT-300M drastically improves the performance with small data. Similarly, Mahajan *et al.* explore the limits of (weakly) supervised pre-training with noisy hashtags on billions of social media (Instagram) images [25]. The traditional ImageNet-1M becomes a small dataset compared to the Instagram data. A gain of 5.6% can be achieved on ImageNet-1M classification accuracy by pre-training on the billion-scale data. As for related work in other deep learning fields, pre-training is also a dominant strategy in natural language processing (NLP) and speech processing [31, 41]. For example, BERT [9] and GPT-3 [3] show that language models pre-trained on massive corpora can generalize well to various NLP tasks.

Pre-training and object detection. However, the story of how and to what extent classification pre-training is helping object detection is up for debate. On one hand, it is observed that pre-training is important when downstream data is limited [1, 16]. On the other hand, there is a line of work reporting competitive accuracy when training modern object detectors from scratch [37, 33, 49, 16]. The gain brought by classification pre-training on larger datasets seems dimin-

ishing [20, 25, 16]. Classification pre-training may sometimes even harm localization when the downstream data is abundant while benefit classification [25]. Shinya *et al.* try to understand the impact of ImageNet classification pre-training on detection and discover that the pre-trained model generates narrower eigenspectrum than the from-scratch model [34]. Recent work proposes a cheaper Montage pre-training for detection on the *target detection* data and obtains an on-par or better performance than ImageNet classification pre-training [48]. Our work aims at improving the usefulness of pre-training with *classification-style* data (*e.g.*, ImageNet) for detection, by resolving the misalignment between pre-training and fine-tuning tasks through the Detection-Aware Pre-training procedure. Leveraging weak supervision is encouraging as the pre-training dataset can be easily scaled up. This is different from pre-training on a fully-supervised detection data [32, 21], which requires expensive annotation cost.

Weakly Supervised Object Localization (WSOL). We leverage WSOL in DAP to locate bounding boxes. WSOL refers to a class of object localization methods that rely on weak supervision (image-level labels) [27, 47, 35, 44, 45, 7, 6], which is exactly what we need for the pre-training data. Many of those methods are based on Class Activation Maps (CAMs) [27, 47, 45]. CAMs highlight the strongest activation regions for a given class thus can roughly locate objects. CAM-style methods remain among the most competitive approaches for WSOL to date [6]. Weakly Supervised Object Detection (WSOD) [2, 19, 38, 40, 43, 46] is a highly related area to WSOL. WSOD tends to focus on detecting possibly multiple objects in multi-labeled images, while WSOL focuses on localizing one object instance. Comparably, WSOD requires more computational cost, and thus we focus on WSOL for large-scale pre-training data.

Self-supervised learning. Self-supervised (*e.g.*, the contrastive learning approaches [15, 4, 5, 11]) pre-training utilizes raw images to pre-train a network without any annotation. While this is an emerging area, the task is challenging due to the lack of annotations, especially for object detection. For example, the backbone in these works still shares the same backbone with the classification task, and ignores detection-related components, *e.g.*, feature pyramid network. Meanwhile, the goal of these works is different from ours. We target at leveraging *classification*-style data *specifically* for detection, while they focus on learning general visual representation from unlabeled data.

Self-training. Self-training, which refers to the technique of iterative pseudo-labeling and re-training in semi-supervised learning, can also improve detection performance [28, 50]. Self-training [50] revisits a large auxiliary dataset multiple times, while we assume that the pre-training dataset is not available in downstream tasks. In ad-
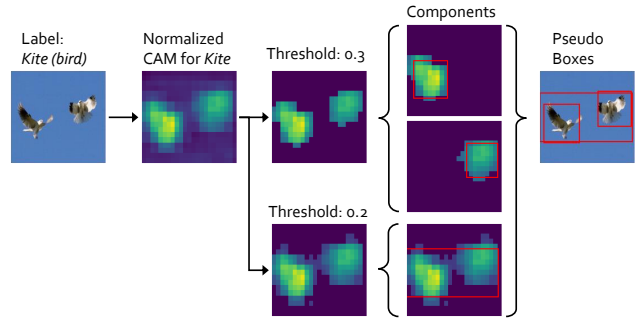

Figure 2. Pseudo box generation procedure. We threshold the CAM with different values and fit a box for each large connected component in each thresholded CAM. The results are merged with NMS. The reason to use different thresholds is to increase the recall. In this example, a too low threshold would fail to discern the two kites. In other cases, a too high threshold might fail to capture the object's whole extent. We find these noisy pseudo labels are sufficient to pre-train the detector to achieve noticeable gains.

dition, self-training is complementary to an improved pre-training approach, which has been verified in speech recognition [41] and will also be demonstrated in Sec. 4.

3. Detection-Aware Pre-training

3.1. Overview of workflow

Figure 1 illustrates the workflow of Detection-Aware Pre-training (DAP) with image-level annotations for object detection. We describe each step in detail below.

Step 1: Classifier Pre-training. The foremost step is to train a deep CNN classifier on the pre-training dataset. Deep CNN classifier usually connects a CNN backbone with an average pooling layer and a linear classification layer [18]. The network is typically trained with a cross-entropy loss on the image and image-level label pairs [18].

The traditional classification pre-training approach directly transfers the network weights of the backbone into the downstream detection fine-tuning tasks. DAP adds the pseudo box generation and the detector pre-training steps in between. In both pre-training approaches, the neural network weights are the only medium of knowledge transfer.

Step 2: Pseudo Box Generation with CAM. From a trained CNN classifier, the Class Activation Map (CAM) of a ground-truth labeled class can be extracted by converting the final classification layer of that class directly into a 1×1 convolution on the last feature map with the average pooling layer removed (and without the activation function) [27, 47]. To improve quality, we can average the CAMs obtained from images with different transformations, *e.g.*, left-right flip and multi scales.

We develop a simple procedure inspired by existing WSOL literature [27, 47, 45, 6] to infer bounding boxes from a CAM, as illustrated in Figure 2. First, the CAM is

normalized to range $[0, 1]$ via an affine transformation based on the extreme values. Here x, y are the horizontal and vertical coordinates:

$$\text{CAM}(x,y) = \frac{\text{CAM}(x,y) - \min \text{CAM}(x,y)}{\max \text{CAM}(x,y) - \min \text{CAM}(x,y)}. \quad (1)$$

Then we threshold the CAM with a hyper-parameter τ and an indicator function $\mathbb{1}\{\cdot\}$:

$$M(x,y) = \text{CAM}(x,y) \times \mathbb{1}\{\text{CAM}(x,y) > \tau\}. \quad (2)$$

Several object instances of the same category could present in a single image, *e.g.*, the two kites in Figure 2. Hence we find connected components on the thresholded CAM M and filter out the components if the area is less than half of the largest component's area. This could remove noisy and small components. Then, we calculate the bounding box coordinates for each component. Denote Ω as the point set of one component. The bounding box (x_c, y_c, w, h) covering Ω is constructed by matching the first and second moments (mean and variance) with a rectangle through the following equations:

$$x_c = \frac{\sum_{(x,y)\in\Omega} M(x,y)x}{\sum_{(x,y)\in\Omega} M(x,y)}, \quad (3)$$

$$y_c = \frac{\sum_{(x,y)\in\Omega} M(x,y)y}{\sum_{(x,y)\in\Omega} M(x,y)}, \quad (4)$$

$$w = \sqrt{12 \frac{\sum_{(x,y)\in\Omega} M(x,y)(x - x_c)^2}{\sum_{(x,y)\in\Omega} M(x,y)}}, \quad (5)$$

$$h = \sqrt{12 \frac{\sum_{(x,y)\in\Omega} M(x,y)(y - y_c)^2}{\sum_{(x,y)\in\Omega} M(x,y)}}. \quad (6)$$

To increase the recall rate of pseudo boxes, we repeat the above procedure multiple times with different threshold values τ. The final results are merged with Non-Maximum Suppression (NMS) based on the Intersection over Union (IoU) between boxes. The boxes are assigned the ground-truth image-level labels as class labels.

Step 3: Detector Pre-training. The pseudo box generation procedure effectively transforms a classification dataset into a detection dataset to be readily used in a standard detection training algorithm. We initialize the backbone in this step with the classification model, and initialize the detector-specific components such as FPN, RPN, and detection heads randomly. Note that we intentionally simplify the pre-training step by treating the detector as a black box. This has two advantages: (1) The approach can be easily generalized to other detector architectures; (2) The approach can leverage existing knowledge about how to train those architectures well and requires minimal code change.

Step 4: Downstream Detector Fine-tuning. When fine-tuning the downstream detection tasks, the pre-trained detector weights are used to initialize a new model, except for the last layers which depend on the number of categories. Our approach is able to initialize more network layers than the traditional classification pre-training.

3.2. Discussion

In Step 2, we adopt a straight-forward CAM-based WSOL approach for its simplicity. An alternative design choice is to obtain the localization information through WSOD [2, 19, 38, 40, 43, 46]. However, WSOD is computationally expensive for large-scale datasets, as it typically needs to extract hundreds or thousands of proposals (*e.g.*, through Selective Search [39]) and learn a multi-instance classifier. Handling cluttered scenes by WSOD is in general a hard problem that warrants further study. In contrast, our approach takes advantage of simple scenes (in, *e.g.*, ImageNet) and only needs to quickly scan each image in an inference mode without extra training, which can be easily scaled up to larger-scale datasets.

As the bounding boxes are not verified by a human judge, the pseudo annotation could be noisy, *e.g.*, incomplete boxes, incorrect localization. However, the pseudo annotation is only used for pre-training, and the fine-tuning process can compensate for the noisy labels to a certain extent. While a more sophisticated treatment might produce more accurate pseudo boxes, we find in the experiments that the pseudo boxes generated from our simple approach can yield substantial improvement in downstream detection tasks through detection-aware pre-training.

4. Experiment

4.1. Settings

Pre-training Datasets. We use ImageNet-1M and ImageNet-14M [8, 30] as the pre-training datasets. ImageNet-1M contains 1.28 million images of $1K$ categories. ImageNet-14M is a larger variant of ImageNet which contains 14 million images of $22K$ categories.

Detection Datasets. For the detection fine-tuning tasks, we leverage the widely-used Pascal VOC [13], Common Objects in Context (COCO) [24] datasets. The Pascal VOC dataset has different versions of each year's competition. Our first setting is based on the VOC 2007 version, where the training set is the trainval2007 ($5,011$ images) and the evaluation set is test2007 ($4,952$ images). The other setting, which we refer to as VOC 07+12, is to merge the trainval2007 and trainval2012 as the training set ($11,540$ images in total), and evaluate on the test2007 set. This is a widely-used protocol in the literature [16]. The VOC dataset has 20 object categories. For the COCO dataset, we

adopt the COCO 2017 train/val split where the train set contains $118K$ valid images and the val set has 5000 images. The COCO dataset has 80 object categories. On top of the aforementioned settings, we also simulate the corresponding low-data settings by varying the number of randomly sampled per-class images (5, 10, 20, 50, 100 images per class), to compare the fine-tuning sample efficiency of different pre-training strategies.

Architecture. Our approach is independent of the detector framework. Here, we use Faster RCNN [29] with Feature Pyramid Networks (FPN) [22] and ResNet-50 [18] as the testbed. In ablation studies, we also include other variants, *e.g.*, RetinaNet [23] and ResNet-101 backbone [18].

Hyper-parameters. In the first stage of classifier pre-training, we use the torchvision[1] pre-trained model for ImageNet-1M experiments. For ImageNet-14M, the classifier is trained with batch size as 8192 for 50 epochs on 64 GPUs. The initial learning rate is 3.2 and decayed with a cosine scheduler.

In the second stage of pseudo box generation, we average the CAMs obtained from two image scales, *i.e.*, short side length as 288 or 576, and from the original and the left-right flipped images. On the normalized ($[0, 1]$) CAMs, we use 4 different thresholds, *i.e.*, $\tau = 0.2, 0.3, 0.4, 0.5$, to generate boxes of various sizes to improve the recall rate. In the end, the mined boxes are merged by NMS with IoU threshold 0.8. The τ and the NMS IoU threshold are further studied in the supplementary material. With ResNet-50, this stage takes less than 13 min on ImageNet-1M and about 2.3 hours on ImageNet-14M with 64 GPUs.

In the third stage of detector pre-training, the model is trained with batch size 32 on 16 GPUs for $40,038$ iterations on ImageNet-1M or $443,658$ iterations on ImageNet-14M. We enable multi-scale augmentation, *i.e.*, the short edge is randomly drawn from $(96, 160, 320, 640)$. The smallest scale is as small as 96 because ImageNet tends to contain large central objects, while we expect the pre-trained detector to be able to handle diverse object scales. This stage takes roughly 1.8 hours on ImageNet-1M or 17.6 hours on ImageNet-14M with Faster RCNN FPN ResNet-50, which is only a small extra cost on top of classification pre-training. As a reference, 90 epochs of ImageNet-1M ResNet-50 classifier training takes 7 hours on 16 GPUs.

In the final stage of fine-tuning, we perform experiments on Pascal VOC [13] and COCO [24]. On COCO, the model is fine-tuned with $90K$ steps (1x) with batch size 16. The initial learning rate is 0.02 and reduced by 0.1 times at $60K$ and $80K$ steps. The image's short side is 800. On VOC 07 and VOC 07+12, the model is trained for 14 epochs ($4.5K$ steps). The initial learning rate is 0.01 and reduced to 0.001 at the 10th epoch. The input image's short side is 640.

---

[1]https://pytorch.org/docs/stable/torchvision/index.html

Table 1. COCO full-data detection results. CLS and DAP refer to the baseline classification and our pre-training strategies. The improvement of DAP over CLS is marked in Δ row. IN-1M and IN-14M correspond to using ImageNet-1M or ImageNet-14M as pre-training set. We report the $AP_{.5:.95}$: the mean of average precisions, $AP_{.5}$, $AP_{.75}$: AP at IoU 0.5 and 0.75, $AP_{\{s,m,l\}}$: AP for small, medium, large objects, calculated on COCO 2017 val.

Pre-train	$AP_{.5:.95}$	$AP_{.5}$	$AP_{.75}$	AP_s	AP_m	AP_l
IN-1M CLS	36.73	58.04	39.72	20.57	39.56	48.51
IN-1M DAP	37.25	58.98	40.46	21.71	40.64	48.34
Δ	**+0.52**	**+0.94**	**+0.74**	**+1.14**	**+1.08**	**+0.83**
IN-14M CLS	38.87	61.87	42.41	23.79	42.15	49.89
IN-14M DAP	39.57	63.05	43.02	24.03	42.96	51.15
Δ	**+0.70**	**+1.18**	**+0.61**	**+0.24**	**+0.81**	**+1.26**

Table 2. VOC 07 and 07+12 full-data detection results. CLS and DAP refer to the baseline classification and our pre-training strategies. IN-1M and IN-14M correspond to using ImageNet-1M or ImageNet-14M for pre-training. We report $AP_{.5}$ which is the area under the precision-recall curve at IoU threshold 0.5, and $AP_{.5, 07metric}$ which is the 11-point metric at IoU 0.5 defined in Pascal VOC 2007 challenge [13], caculated on VOC 2007 test.

Train set	Pre-train	$AP_{.5}$	$AP_{.5, \text{ 07metric}}$
07 trainval	IN-1M CLS	77.36	75.00
	IN-1M DAP	79.93 (**+2.57**)	77.57 (**+2.57**)
	IN-14M CLS	80.74	78.29
	IN-14M DAP	84.24 (**+3.50**)	81.54 (**+3.25**)
07+12 trainval	IN-1M CLS	83.77	80.97
	IN-1M DAP	84.49 (**+0.72**)	82.00 (**+1.03**)
	IN-14M CLS	86.91	83.56
	IN-14M DAP	87.84 (**+0.93**)	84.53 (**+0.97**)

For the low data settings, training with the same number of iterations as the full data setting is sub-optimal. Early stop is needed. Following [16], we tune the number of iterations. As in [17, 26], we use fixed BatchNorm and freeze the the first conv block of ResNet in all fine-tuning experiments. The weight decay coefficient is set to 1e-4 in ImageNet-1M experiments and 1e-5 in ImageNet-14M experiments. We do not use test time multi-scale augmentation.

Evaluation metrics. On COCO, we report the standard AP metrics [24], *i.e.*, $AP_{.5:.95}$, the mean of average precisions (AP) evaluated at IoU thresholds $0.5, 0.55, \cdots, 0.95$. $AP_{.5}$ and $AP_{.75}$ are also reported for AP at IoU 0.5 and 0.75. $AP_{\{s,m,l\}}$ are for small ($< 32^2$ pixels), medium, and large ($\geq 96^2$ pixels) objects, determined by the area of a bounding box. For VOC, we report $AP_{.5}$ and the 11-point version $AP_{.5,07metric}$ defined by the VOC 2007 challenge [13].

4.2. Main results

We denote 1N-1M CLS and 1N-14 CLS as the shorthands for the traditional classification pre-training strategy on ImageNet-1M and ImageNet-14M, respectively. Simi-
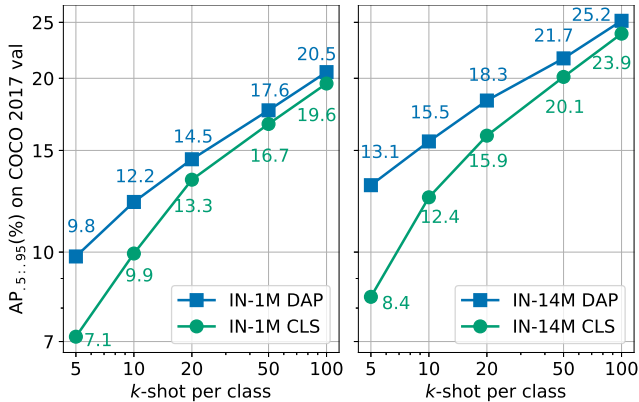
Figure 3. COCO k-shot low-data detection results. CLS and DAP refer to the baseline classification and our pre-training strategies. IN-1M (left) and IN-14M (right) correspond to using ImageNet-1M or ImageNet-14M as pre-training set. In the horizontal direction, we vary the number of images per class, and in the vertical direction, we report the $AP_{.5:.95}$ on COCO 2017 val. There are 80 classes in COCO, so 5-shot corresponds to 400 images in total.
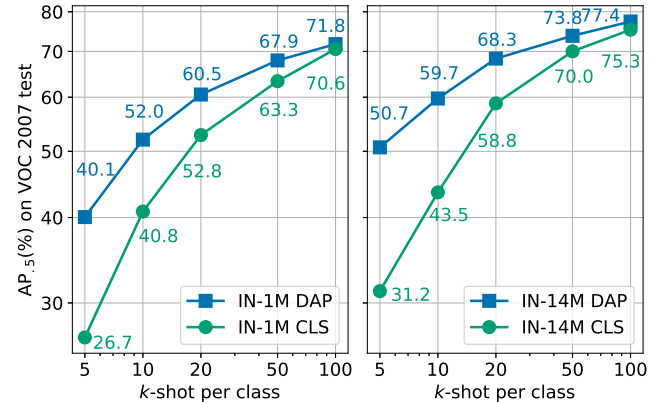

Figure 4. VOC k-shot low-data detection results. IN-1M (left) and IN-14M (right) refer to using ImageNet-1M or ImageNet-14M as pre-training set. In the horizontal direction, we vary the number of images per class, and in the vertical direction, we report the $AP_{.5}$ on VOC 2007 test. We sample the training images from the combined VOC 07+12 trainval set. There are 20 classes in VOC, so 5-shot corresponds to 100 images in total.

larly, our DAP strategy is denoted as IN-1M DAP and IN-14M DAP on the two ImageNet dataset variants. In DAP, 4.1M pseudo boxes are mined in ImageNet-1M and 47M boxes in ImageNet-14M. The results are summarized in Tables 1, 2 for the full-data setting and Figures 3, 4 for the low-data setting, and observations are as follows.

DAP is more effective than classification pre-training in the full-data setting. The full-data setting results in Tables 1 and 2 tell that DAP performs consistently better than classification pre-training (CLS) across all metrics. The gain is especially significant for the VOC dataset, reaching a ≥ 2.5 $AP_{.5}$ increase with 07 trainval and a roughly $+1$ $AP_{.5}$ increase with 07+12 trainval. And the gain on COCO $AP_{.5:.95}$ is $+0.52$ with ImageNet-1M and $+0.7$ with ImageNet-14M. The results suggest that DAP makes better use of the ImageNet dataset to pre-train the network than CLS pre-training.

DAP benefits more from larger pre-training dataset. Comparing ImageNet-1M and ImageNet-14M, our DAP scales up better to ImageNet-14M. Improvement on ImageNet-14M is larger than on ImageNet-1M: $+0.7$ vs $+0.52$ on $\Delta AP_{.5:.95}$ with COCO in Table 1, $+3.5$ vs $+2.16$ with VOC 07 and $+0.93$ vs $+0.72$ with VOC 07+12 on $\Delta AP_{.5}$ in Table 2. The training process and pseudo box generation hyper-parameters are shared between the 1M and 14M results. The only difference is the size of the pre-training datasets. Therefore, this observation suggests that DAP benefits more from the larger ImageNet dataset.

DAP improves low-data performance. The low-data setting is of great practical value to reduce the annotation cost. In Figure 3, 4, we mimic this low-data regime by downsam-
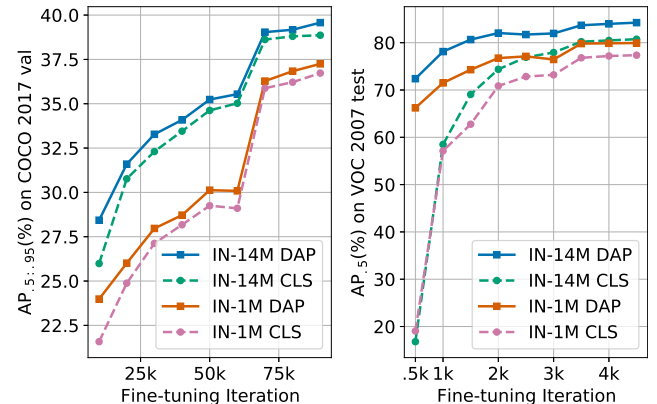

Figure 5. Learning curves of Faster RCNN FPN ResNet-50 during COCO and VOC 07 training, with different pre-trained models as initialization. DAP is able to provide a "head-start" for fine-tuning (i.e., faster convergence), and is almost always leading the CLS pre-trained counterparts. The abrupt increase in COCO AP is caused by learning rate reduction at 60k-step.

pling the COCO and VOC datasets. Compared with CLS, we observe that fine-tuning from DAP benefit much more in the low-data setting than the full-data setting. For example, in the 5-shot case in Figure 3, IN-1M DAP outperforms IN-1M CLS pre-training by 2.6 $AP_{.5:.95}$ (left), and IN-14M DAP surpasses IN-14M CLS by a significant 4.7 $AP_{.5:.95}$ (right). Similarly, in Figure 4, the VOC $\Delta AP_{.5}$ is as much as $+13.4$ (IN-1M) and $+19.5$ (IN-14M) in the 5-shot case, compared to $+0.72$ and $+0.93$ in the full-data setting.

4.3. Analysis

Faster convergence with DAP than classification pre-training. As our DAP approach provides greater accuracy improvement, we study the convergence behavior by plot-
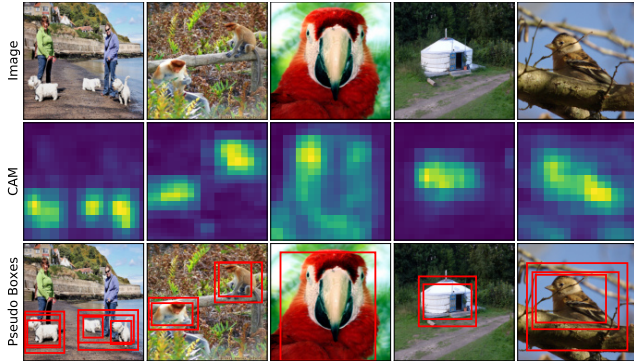
Figure 6. Examples of ImageNet images, CAMs & pseudo boxes.

Table 3. Comparison to the variant that freezes ResNet-50 backbone and only pre-trains the additional layers in Faster RCNN.

Pre-train	COCO AP$_{.5:.95}$	VOC07 AP$_{.5}$	VOC07+12 AP$_{.5}$
IN-1M CLS	36.73	77.36	83.77
IN-1M DAP	37.25	79.93	84.49
IN-1M DAP (Freeze)	36.89	78.86	83.91

Table 4. ResNet-101 Faster RCNN FPN results on COCO.

Pre-train	AP$_{.5:.95}$	AP$_{.5}$	AP$_{.75}$	AP$_s$	AP$_m$	AP$_l$
IN-1M CLS	39.11	61.06	42.59	22.98	42.35	50.50
IN-1M DAP	39.28	61.32	42.81	23.45	42.70	51.47
Δ	**+0.17**	**+0.26**	**+0.22**	**+0.47**	**+0.35**	**+0.97**
IN-14M CLS	43.18	66.76	47.31	26.29	47.21	55.55
IN-14M DAP	43.92	67.16	48.39	27.41	48.18	56.37
Δ	**+0.74**	**+0.40**	**+1.08**	**+1.12**	**+0.97**	**+0.82**

Table 5. ResNet-101 Faster RCNN FPN results on VOC.

Train set	Pre-train	AP$_{.5}$	AP$_{.5, 07metric}$
07 trainval	IN-1M CLS	78.02	75.73
	IN-1M DAP	80.95 (**+2.93**)	78.03 (**+2.30**)
	IN-14M CLS	84.58	81.88
	IN-14M DAP	86.63 (**+2.05**)	83.71 (**+1.83**)
07+12 trainval	IN-1M CLS	84.91	81.81
	IN-1M DAP	85.49 (**+0.58**)	82.43 (**+0.62**)
	IN-14M CLS	89.41	85.92
	IN-14M DAP	90.69 (**+1.28**)	86.55 (**+0.63**)

Table 6. RetinaNet (ResNet-50 FPN) results on COCO 2017 val.

Pre-train	AP$_{.5:.95}$	AP$_{.5}$	AP$_{.75}$	AP$_s$	AP$_m$	AP$_l$
IN-1M CLS	36.22	55.11	38.56	19.72	39.56	48.74
IN-1M DAP	36.96	55.99	39.43	20.41	40.20	50.61
Δ	**+0.74**	**+0.88**	**+0.87**	**+0.69**	**+0.64**	**+1.87**

Table 7. RetinaNet (ResNet-50 FPN) results on VOC 2007 test.

Train set	Pre-train	AP$_{.5}$	AP$_{.5, 07metric}$
07 trainval	IN-1M CLS	75.12	72.91
	IN-1M DAP	77.95 (**+2.83**)	75.80 (**+2.89**)
07+12 trainval	IN-1M CLS	81.48	78.69
	IN-1M DAP	84.18 (**+2.70**)	81.15 (**+2.46**)

ting the learning curves of COCO and VOC training with different pre-trained ResNet-50 models in Figure 5. From the figure, we notice that DAP can give a significant initial accuracy boost compared to CLS. For example, in the right part of Figure 5, 500 fine-tuning iterations from DAP already achieve $\geq$ 65 AP$_{.5}$, while the corresponding CLS numbers are lower than 20. This demonstrates that a better pre-trained model can provide faster convergence speed, which is consistent with [25, 20, 16, 34].

Visualization of CAM pseudo boxes. Figure 6 visualizes the CAMs and the mined pseudo boxes in ImageNet-1M images. In all examples, our pseudo box generation procedure can successfully find the rough locations of the objects. The per-component multi-threshold approach is able to recover multiple objects in the first two columns. We notice that the pseudo boxes are noisy, i.e., containing inaccurate boxes such as the loose one around the bird in the last column. Despite the noise, pre-training can still benefit from leveraging a large amount of data. The network can pick up useful learning signals from the noisy pseudo labels.

What is learned in pre-training? To study whether the better performance of DAP is only due to being able to pre-train the additional components of detectors, we conduct an ablation study by freezing the whole ResNet-50 backbone and only pre-training the FPN, RPN and ROI heads in Faster RCNN. The result is in Table 3. The COCO AP$_{.5:.95}$ in this setting is 36.89, and AP$_{.5}$ is 78.86 when trained on VOC 07 and 83.91 trained on VOC 07+12. The downstream task performance is better than CLS pre-training but worse than full DAP, suggesting that DAP not only pre-trains the new layers, but also adapts the feature representations of the entire network more towards detection.

Varying network backbone We change the backbone network from ResNet-50 to the larger ResNet-101. The results are in Table 4, 5 for both COCO and VOC. ResNet-101 delivers higher absolute accuracy than Resnet-50, but DAP again performs consistently better than CLS pre-training.

Varying detector architecture. Our DAP requires no knowledge of the internal mechanism of a detector. We show that our DAP approach generalizes to the RetinaNet detector architecture [23]. RetinaNet is a one-stage detector as opposed to the two-stage detector of Faster RCNN. We pre-train a RetinaNet (ResNet-50) detector on IN-1M for 80, 072 steps with batch size 32 and learning rate 0.005. We can see in Table 6, 7 that the same pipeline works well with RetinaNet and DAP consistently outperforms CLS pre-training on both COCO and VOC.

Accuracy on ImageNet. To study the effect of DAP on the original ImageNet classification task, we evaluate the IN-1M DAP pre-trained Faster RCNN (ResNet-50) as a classifier. The class score is taken as the sum of the con-

Table 8. ImageNet-200 DET reference results trained on VOC 07 trainval and 07+12 trainval, evaluated on VOC 2007 test.

Pre-train	#Image	#Class	#Box	VOC07 $AP_{.5}$	VOC07+12 $AP_{.5}$
IN-1M CLS	1.28M	1K	0	77.36	83.77
IN-1M DAP	1.28M	1K	4.11M	79.93	84.49
IN-14M CLS	14.2M	22K	0	80.74	86.91
IN-14M DAP	14.2M	22K	47.3M	84.24	87.84
IN-200 DET	333K	200	479K	80.53	84.01

Table 9. Faster RCNN FPN ResNet-50 results of training on VOC 07 trainval + self-training on 2012 trainval, evaluated on 07 test.

Pre-train	+Self-train	$AP_{.5}$	$AP_{.5,\ 07metric}$
IN-1M CLS		77.36	75.00
IN-1M DAP		79.52 (**+2.16**)	76.82 (**+1.82**)
IN-1M CLS	✓	77.98	75.71
IN-1M DAP	✓	80.50 (**+2.52**)	78.02 (**+2.31**)

fidence scores of all detected objects of this class. The DAP pre-trained detector achieves 62.73% Top-1 accuracy and 85.99% Top-5 accuracy. These numbers are lower than those of the bare ResNet-50 backbone (76.15% Top-1 and 92.87% Top-5). However, as our experiments have shown, the DAP pre-trained network is better at detection fine-tuning, suggesting that the drop in whole-image classification accuracy is likely traded for better localization and regional classification ability.

Reference with IN-200 DET. ImageNet challenge provides a detection subset of 200 classes, which is referred as IN-200 DET [8, 30]. We present a reference result on using this dataset in fully-supervised detection pre-training in Table 8. We train on IN-200 DET for 5 epochs with scale augmentation $(96, 160, 320, 640)$ and transfer the detector weights. The VOC07 $AP_{.5}$ is improved to 80.53 (vs CLS pre-training 77.36) when trained on VOC 07 and 84.01 (vs CLS 83.77) trained on VOC 07+12. Our DAP achieves even higher accuracy except for the slight drop with IN-1M VOC07. This suggests that DAP may benefit from the substantially more pseudo boxes mined from a larger-scale dataset than the human-labeled boxes in IN-200 DET.

Complementary to self-training. Self-training is another direction of sample-efficient learning which re-trains the model with unlabeled data and pseudo labels [28, 50, 41]. We believe self-training and pre-training are complementary. Intuitively, better pre-training may give the model a head-start in learning downstream tasks, therefore produce better pseudo labels for the subsequent self-training. We consider a VOC semi-supervised setting and verify that DAP can still improve the accuracy under self-training settings. The detector is trained on VOC 2007 trainval initialized with our DAP or with the conventional CLS. Then, We keep as pseudo labels the confident predictions, *i.e.*, score ≥ 0.6 or is the max in the image on the VOC 2012 trainval

Table 10. Comparing CLS and DAP with longer fine-tuning time.

	voc07 5shot 1x	3x	coco 5shot 1x	3x	coco full 1x	3x
IN-1M CLS	26.7	27.3	7.13	7.03	36.73	37.24
IN-1M DAP	40.1	40.9	9.88	9.20	37.25	37.45

(ground-truth labels are removed to have a semi-supervised setting) with test-time flip augmentation, and finally tune the detector for 2 more epochs on all data. The result is shown in Table 9. Self-training improves AP across all pre-training strategies. Notably, in this particular setting, DAP leads to even larger gain than CLS. That is, without self-training, our improvement is $+2.16$ $AP_{.5}$ and with self-training, the gain is $+2.52$.

Longer Training Schedule. In Table 10, we ran 3 settings (Faster RCNN ResNet-50) for $3\times$ longer to study whether the gain of DAP persists with longer fune-tuning time. We observe the difference between DAP and CLS on COCO full gets smaller. However, DAP still brings noticeable gains in the 5-shot settings that are prone to overfitting.

5. Discussion and Conclusion

Implication for future work. This work may open up many future directions. We adopt a straightforward WSOL method in this paper. A more sophisticated WSOL or WSOD method [2, 19, 38, 40, 43] could potentially produce higher-quality pseudo boxes, which may improve pre-training in return. For example, it may require handling the missing label problem in ImageNet. However, we want to emphasize that the simplicity of our pseudo box generation method is also a blessing by being scalable to millions of images such as ImageNet-14M. Another sensible next step is to leverage mixed-labeled data in pre-training, e.g., using semi-supervised WSOD as the pre-training procedure [46]. DAP might also benefit from moving to a more diverse multi-labeled dataset. This would make object localization more challenging, but the network may benefit from seeing more complex contexts. Finally, broadening the approach to mask detection (or instance segmentation) [17] aware pre-training is worth exploring.

Conclusions. In this paper, we have proposed a Detection-Aware Pre-training (DAP) strategy under weak supervision. Specifically designed for object detection downstream tasks, DAP makes better use of a classification-style dataset by transforming it into a detection dataset through a pseudo box generation procedure. The generation is based on a simple yet effective approach built on CAM. DAP reduces the discrepancies in the objective function, the localization information, and the network structure between the pre-training and the fine-tuning tasks. As a consequence, DAP leads to faster and better fine-tuning than classification pre-training. Besides, DAP leads to much higher accuracy in low-data settings and is complementary to advances in self-training [50].

References

[1] Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. In *European conference on computer vision*, pages 329–344. Springer, 2014. 2

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 3, 4, 8

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning*, 2020. 2, 3

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 2020. 2, 3

[6] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3133–3142, 2020. 3

[7] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 2, 4, 8

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019. 2

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 1, 2

[11] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208, 2010. 3

[12] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised pretraining. In *Artificial Intelligence and Statistics*, pages 153–160, 2009. 1, 2

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 2, 4, 5

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 2

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2, 3

[16] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*, pages 4918–4927, 2019. 1, 2, 3, 4, 5, 7

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 5, 8

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3, 5

[19] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *European Conference on Computer Vision*, pages 350–365. Springer, 2016. 3, 4, 8

[20] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *ECCV*, 2020. 3, 7

[21] Hengduo Li, Bharat Singh, Mahyar Najibi, Zuxuan Wu, and Larry S Davis. An analysis of pre-training on object detection. *arXiv preprint arXiv:1904.05871*, 2019. 3

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2, 5

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5, 7

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 4, 5

[25] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 1, 2, 3, 7

[26] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: 11/2020. 5

[27] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 685–694, 2015. 3

[28] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128, 2018. 3, 8

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 5

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 4, 8

[31] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *Proc. Interspeech 2019*, pages 3465–3469, 2019. 2

[32] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 8430–8439, 2019. 2, 3

[33] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. Dsod: Learning deeply supervised object detectors from scratch. In *Proceedings of the IEEE international conference on computer vision*, pages 1919–1927, 2017. 2

[34] Yosuke Shinya, Edgar Simo-Serra, and Taiji Suzuki. Understanding the effects of pre-training for object detectors via eigenspectrum. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3, 7

[35] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE international conference on computer vision (ICCV)*, pages 3544–3553. IEEE, 2017. 3

[36] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 1

[37] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in neural information processing systems*, pages 2553–2561, 2013. 2

[38] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017. 3, 4, 8

[39] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for ob-

ject recognition. *International journal of computer vision*, 104(2):154–171, 2013. 4

[40] Yunchao Wei, Zhiqiang Shen, Bowen Cheng, Honghui Shi, Jinjun Xiong, Jiashi Feng, and Thomas Huang. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 434–450, 2018. 3, 4, 8

[41] Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. Self-training and pre-training are complementary for speech recognition. *arXiv preprint arXiv:2010.11430*, 2020. 2, 3, 8

[42] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1, 2

[43] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8292–8300, 2019. 3, 4, 8

[44] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, 2018. 3

[45] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 597–613, 2018. 3

[46] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. *European conference on computer vision*, 2020. 3, 4, 8

[47] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2, 3

[48] Dongzhan Zhou, Xinchi Zhou, Hongwen Zhang, Shuai Yi, and Wanli Ouyang. Cheaper pre-training lunch: An efficient paradigm for object detection. *European Conference on Computer Vision*, 2020. 3

[49] Rui Zhu, Shifeng Zhang, Xiaobo Wang, Longyin Wen, Hailin Shi, Liefeng Bo, and Tao Mei. Scratchdet: Training single-shot object detectors from scratch. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2268–2277, 2019. 2

[50] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 2020. 3, 8