PreDet: Large-scale weakly supervised pre-training for detection

Vignesh Ramanathan
Facebook AI
vigneshr@fb.com

Rui Wang
Facebook AI
ruiw@fb.com

Dhruv Mahajan
Facebook AI
dhruvm@fb.com

Abstract

State-of-the-art object detection approaches typically rely on pre-trained classification models to achieve better performance and faster convergence. We hypothesize that classification pre-training strives to achieve translation invariance, and consequently ignores the localization aspect of the problem. We propose a new large-scale pre-training strategy for detection, where noisy class labels are available for all images, but not bounding-boxes. In this setting, we augment standard classification pre-training with a new detection-specific pretext task. Motivated by the noise-contrastive learning based self-supervised approaches, we design a task that forces bounding boxes with high-overlap to have similar representations in different views of an image, compared to non-overlapping boxes. We redesign Faster R-CNN modules to perform this task efficiently. Our experimental results show significant improvements over existing weakly-supervised and self-supervised pre-training approaches in both detection accuracy as well as fine-tuning speed.

1. Introduction

We address the problem of large-scale weakly supervised pre-training for detection, where we assume that noisy classification labels are available for images, but localization (bounding-boxes) information is missing. Almost all state-of-the-art approaches use pre-trained classification models and fine-tune them for detection tasks. Fine-tuning mainly yields two significant benefits: (a) improved accuracy and (b) speedup in training for detection. Recently, there has been a lot of work on large-scale [51, 34] pre-training of classification models with noisy labels from the web. However, the benefits are more pronounced for classification tasks compared to detection or instance segmentation [34].

We hypothesize that pre-training for classification tasks overemphasizes translation invariance [26] as shown in Fig. 1. Different crops of an image that share similar content but do not have high overlap are required to be similar to each other. As seen in the figure, this runs contrary to the detection objective and could result in feature representations that are sub-par for the target detection task. In theory, this could be handled by pre-training a detection model on a large dataset from scratch. However, this is impractical due to the huge effort required to annotate bounding boxes for images at that scale. Instead, we propose to supplement the standard classification pre-training task with a novel self-supervised pretext task that is closer to detection.

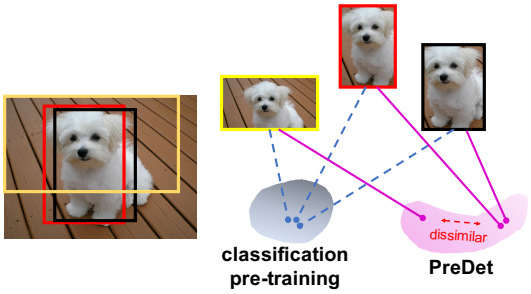

Figure 1. Consider the bounding boxes shown on the left. In classification pre-training, it is desirable for the model to learn similar representations for all of them to incorporate translation invariance. However, this is undesirable for detection, which requires bounding boxes with small or no overlap to be dissimilar. This forms the core of our pre-training method, which uses a pretext task to force the model to distinguish the non-overlapping boxes.

Recent self-supervised approaches such as MoCo[18] have shown that maximizing the agreement between two views (constructed via transformations) of the same image and minimizing it for different images works really well for feature learning. These tasks typically require a dictionary-lookup, wherein one view serves as the query, while the other view is part of a dictionary. In our approach, we extend this idea to detection. We use a bounding box from one-view of an image as a query to retrieve the same bounding box (or a box highly overlapping with it) from another view of the image. We refer to this task as *query-box lookup*. This ensures that boxes with sufficiently high overlap are similar to each other, while non-overlapping boxes have distinct representations.

Ideally, the query-box should be retrieved from the set of all bounding boxes in the image. However, this is too

large to handle. Hence, we restrict the look-up to a smaller, but representative set of "proposal" boxes. These proposals should include high-quality hard-negatives to make the retrieval task useful. Fortunately, this proposal-set construction problem has been solved by Region proposal network (RPN) in the context of object detection for Faster R-CNN. In our approach, we adapt RPN to instead construct query-specific proposals. We refer to it as the contrastive RPN (CRPN). Similarly, we also adapt the Region of Interest (ROI-head) module from Faster R-CNN to carry out the retrieval task, and refer to it as the contrastive ROI-head (CROI-head). This design has the advantage of making our model-architecture similar to Faster R-CNN detection model. We refer to our approach as PreDet.

We show that PreDet pre-trained on a dataset of $50M$ images with noisy hashtags as labels provides two main benefits over existing approaches: higher average precision (AP) and faster fine-tuning. When fine-tuning a ResNeXt-101-32x8d Mask R-CNN model for the standard $90k$ iterations on MS-COCO [31], initializing from PreDet achieves 3.4% and 2.9% absolute improvement in AP^{box} compared to ImageNet pre-trained model and self-supervised SEER [13] model pre-trained on $1B$ images respectively. More impressively, fine-tuning PreDet for just $90k$ iterations also outperforms these models, when they are fine-tuned for longer duration ($6\times$ $-9\times$ more) by 1.3%. We observe similar gains for RetinaNet models and other detection datasets (LVIS-v1 [17] and PASCAL VOC [11]). We also conduct extensive experiments to understand the effect of model capacity and target dataset size, and find PreDet to be particularly impactful for larger models and smaller target datasets.

2. Related work

Pre-training for detection ImageNet pre-training has contributed to the success of many computer vision tasks. In the last few years, several works [2, 34, 23, 51, 63, 24, 32, 36, 16, 68] have shown that pre-training on larger but noisier web-scale data leads to improvements on multiple target tasks. However, these works primarily target classification and provide limited gains for detection as shown in [34].

There has also been extensive analysis [28, 19, 49] on the transferability of such pre-trained networks to detection. In particular, [19, 12, 48, 67] showed that models trained from scratch achieve comparable performance with ImageNet-pretrained models. However, they require significantly longer training schedules. On the other hand, Object365 [45] showed that pre-training a detection model on a larger detection dataset leads to performance improvements for detection. However, it requires a huge effort to annotate bounding boxes for the pre-training dataset.

Another line of work [25, 64] explores ways to transfer pre-training weights more effectively for detection tasks. BiT [24] showed that good weight normalization can lead to better transferability. These are complimentary to our approach and can be used with our method for better gains.

Learning detection and semantic segmentation models [9, 8, 53, 57, 21, 46, 47] directly from web-scale data in a weakly-supervised manner has also shown some promise. However, the performance gap between weakly-supervised and fully-supervised approaches is still large. We show that web-scale data can be better leveraged through a new detection-specific pre-training approach to further improve the performance of fully-supervised detection models.

Self-supervised pre-training There has been extensive research in unsupervised feature learning [1, 4, 41, 56, 3, 10, 59] for classification tasks. Notably, contrastive learning based approaches [5, 6, 15, 35, 27] have made huge strides. Recent approaches like MoCo [18, 7], Swav [5, 13] and InfoMin [54] achieve better performance than fully supervised pre-training even for detection. InfoMin [54] showed the importance of selecting the right augmentation strategy to construct these different views of an image and is complementary to the pretext task introduced in our work. In contrast, we introduce a self-supervised task that is much closer to detection and show the benefits of combining self-supervised learning with classification pre-training.

Semi-supervised learning and Self-training Semi-supervised and self-training methods [50, 62, 22, 39, 29, 70, 60, 61] train a model jointly with a specific target dataset and the provided large-scale dataset, and have shown benefits for detection. However, they require knowledge of the target task and dataset ahead of time, as well as longer fine-tuning schedules on the combined dataset. These approaches can be combined with pre-training to realize complimentary benefits, particularly for smaller target datasets, as shown in [70].

3. Approach

Similar to [34], we assume that each image is accompanied by multiple noisy class labels such as hashtags. At high level, our approach has two loss components: classification pre-training loss $\mathcal{L}_{cls}$, and detection specific self-supervised loss $\mathcal{L}_{det}$. We refer to our approach as *PreDet*.

3.1. Classification pre-training loss

We start with a classification model (shown in blue in Fig. 2). Since each image contain multiple labels, we use multi-label cross-entropy loss [34]. We use a CNN backbone with a Feature Pyramid Network (FPN) and a classification head added to each pyramid level. These heads are identical with shared parameters. The classification loss $\mathcal{L}_{cls}$ is the average of the losses from all pyramid levels.

3.2. Adding detection self-supervision

Pre-training only with a classification loss could make the model learn similar representations for different bound-

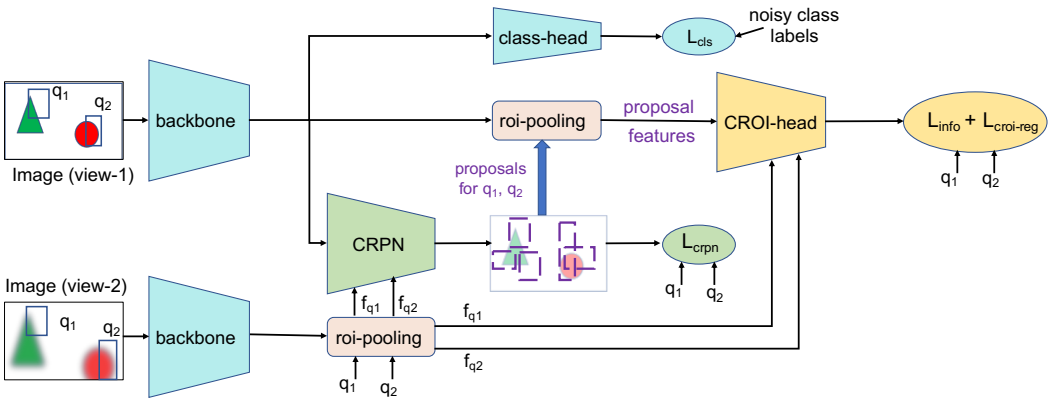

Figure 2. The pre-training approach for PreDet is visualized here. Two-views of the same image along with the same set of queries for each view are seen by the model. The blue modules correspond to typical classification-only pre-training method. The CRPN and CROI modules (shown in green and yellow respectively) add new self-supervision losses which are detection-specific.

ing boxes in the image, even if they don't have sufficient overlap. This is not beneficial for object detection. Hence, we add a self-supervised task that forces the model to learn distinct representations for them.

Given an image I, we first construct a set of Q *queries*, where each query q is simply a local region defined by the bounding box co-ordinates. These query boxes need not correspond to actual objects in the image or the class labels and are simply patches obtained around informative regions in the image. In our work, we sample them from bounding boxes generated by an unsupervised proposal generation method: EdgeBox [69].

Motivated by contrastive learning approaches [18], we construct two different *views* of the same image, by applying different random transformations. Note that these transformations also include cropping and scaling. As a result, query q will have different bounding box co-ordinates in different views, even though it represents the same region. We then pass view-2 of I through the backbone and do ROI-pooling [43] to construct a query feature vector f_q for query q. We also pass view-1 of I through the backbone to obtain image feature maps f_I. The backbone feature extraction from view-1 is exactly the same as done in Faster R-CNN [43, 20] models before the Region Proposal Network (RPN) stage. Constructing the query features and backbone features from two different views make the task harder and hence learning more effective. We now formally define our *query-box lookup* task as follows:

query-box lookup: Given a query feature vector f_q from view-2 corresponding to a query-box q, and backbone feature map f_I from view-1, retrieve the bounding boxes having high-overlap with q in view-1 of the image.

It is desirable to consider all possible bounding boxes in view-1 to perform query-box lookup. However, it is impractical to construct feature representations for all boxes in an image. Instead, we adopt the proposal generation method proposed in Faster R-CNN, to first extract a subset of "proposal" boxes from view-1 and lookup only within this set. The proposals are obtained by a variant of the Region Proposal Network (RPN) that is trained to select the most likely set of boxes in view-1 that are visually similar to the query-boxes from view-2.

3.2.1 Contrastive Region Proposal Network (CRPN)

We design a region proposal network (RPN) similar to Faster R-CNN, but to generate proposal boxes for the input queries, instead of objects. We refer to this module as the contrastive RPN (CRPN), visualized in detail in Fig. 3. The proposals from CRPN would include potential matches for the query-boxes, as well as hard-negative examples that do not overlap with the query-boxes.

Similar to the RPN in Faster R-CNN, we consider A anchor boxes of different sizes and aspect-ratios, centered at each anchor in the feature-map f_I[1]. For each anchor box and query pair, we generate a classification score that indicates if the anchor box is a match for the query, i.e., if it has high overlap with the query. Unlike Faster R-CNN, the proposals cover image-specific queries and not a set of objects that are common to all images. Hence, we use image-specific input query feature f_q (from view-2) as additional contextual information to generate these classification scores for each query q. To achieve this, we first obtain a feature map f_I^{cls} from f_I (as shown in Fig. 3). This feature map has D-dimensional features for all A anchor boxes at each anchor. In order to generate query-specific scores, we also construct A feature vectors f_q^{cls} from the query feature f_q and compute dot-product with anchor-box features

[1]In the case of FPN, we will have multiple feature maps and separate anchors for each feature map

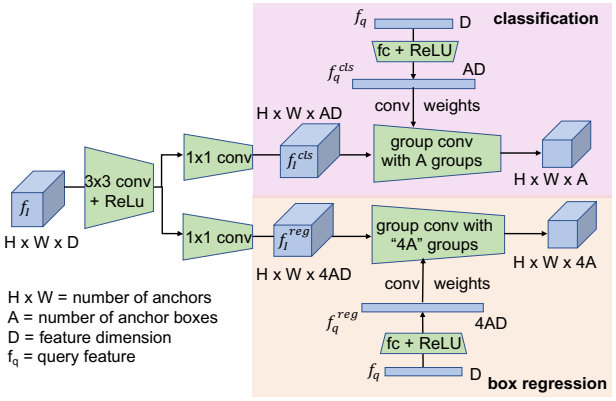

Figure 3. CRPN takes an input feature of dimension $H \times W \times D$ and a query feature f_q to produce query scores of dimension $H \times W \times A$ and regression co-ordinates of dimension $H \times W \times 4A$ to regress the A anchor boxes at each anchor to the query. This is run in parallel for all Q input queries.

in f_I^{cls}. This results in A scores for query q at each anchor. In our implementation, this dot-product is computed simultaneously for all the queries and anchor-boxes as a 1×1 group-convolution with A groups. Please refer to supplementary for details.

CRPN also produces bounding box regression values for each {anchor box, query} combination, that will regress the anchor box to the query-box (in view-1's co-ordinate system). We use the regression parametrization of Faster R-CNN (Eq. 2 in [43]). We follow the same design used to obtain query scores, and generate $4A$ regressions co-ordinates, corresponding to A anchor boxes for each query q at each anchor (shown in the bounding box regression part of Fig. 3).

CRPN losses: Following the typical RPN, we sample 128 positive and 128 negative anchor boxes to train CRPN as well. For every anchor box, we identify the query box that has the maximum overlap with it (or random query if overlap with all query boxes is zero). We denote the anchor box as a positive sample if this overlap is greater than 0.7 and as a negative sample if it is less than 0.3. We use the score predicted by CRPN for this {query, anchor box} combination as the anchor box's classification score to define a proposal classification loss[2], similar to the one defined for Faster R-CNN. Also for the positive samples, we use the regression co-ordinates predicted by the model for the corresponding query (having maximum overlap with the anchor box) to train an RPN regression loss, similar to Faster R-CNN as well. We refer to the combined RPN classification and regression losses as $\mathcal{L}_{crpn}$.

[2]Note that it is possible that a positive proposal box has high overlap with multiple queries. In practice, this rarely happens due to sparse number of queries considered per image and NMS applied to Edge Box proposals for queries selection.

Proposal construction: For each anchor box, we choose the maximum score predicted by CRPN across all queries as the proposal score for the anchor box. We also use the regression co-ordinates predicted by CRPN corresponding to this highest-scoring query to regress the anchor box and obtain a proposal bounding box. We choose the top K proposals based on their proposal score, followed by non-maximal suppression to select a subset of K_{nms} proposal boxes to pass to the next module, which performs query-box lookup from this set.

3.2.2 Contrastive Region of Interest (CROI) head

We modify the ROI-head from Faster R-CNN to retrieve boxes that have high-overlap with the query boxes, from the proposals returned by CRPN. We refer to this module as contrastive ROI-head (CROI-head). The ROI-pooled features from all the proposals selected by CRPN and the query features are provided as input to the CROI head. The similarity between the proposal features and query features is used to define a retrieval loss which enforces high similarity, only if a proposal has high overlap with the query box.

Query-box lookup: The ROI-pooled feature for proposal p is first passed through an MLP (box-head similar to Faster R-CNN) to obtain feature b_p. Similarly the query feature f_q is passed through the same MLP to obtain b_q. The query-proposal score s^{pq} corresponding to each proposal and query pair (p, q) is computed by passing b_p and b_q through a FC layer, and then computing the cosine similarity between them (Fig. 4). The predicted scores are used in an InfoNCE [37] loss to ensure that a proposal having good overlap with a query has a higher score corresponding to the query, compared to other proposal boxes that do not overlap with it. To enforce this, we first construct a set $\mathcal{P}_{neg}$ of proposals that have IoU overlap of less than 0.5 with all the queries for the image. Then, for each query q, we sample one positive proposal p_q^+ that has an IoU overlap greater than 0.5 and define the InfoNCE loss below:

$$\mathcal{L}_{info} = -\frac{1}{Q}\sum_{q}\log\frac{\exp\left(s^{p_q^+ q}/\tau\right)}{\exp\left(s^{p_q^+ q}/\tau\right) + \sum_{p^- \in \mathcal{P}_{neg}}\exp\left(s^{p^- q}/\tau\right)},$$

where τ is the temperature hyper-parameter in [37]. Note that this loss is similar to the contrastive loss defined in self-supervised classification work [18] but applied to the proposal boxes in the same image. We set τ to 0.07 in all our experiments.

Additional regression loss: The positive proposal p_q^+ may not exactly overlap with the query q. Taking inspiration from Faster R-CNN, we add an additional component to CROI-head that regresses proposals to the query boxes (in view-1's co-ordinate system) they overlap with. This helps

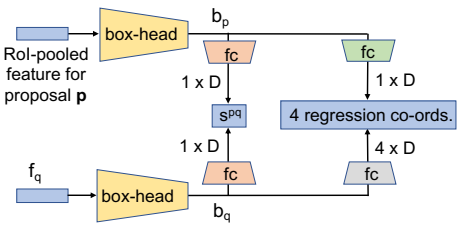

Figure 4. The CROI-head takes roi-pooled feature from a proposal p and a query feature f_q to produce query scores for the proposal s^{pq} and regression co-ordinates to regress p to query box q. The modules shown in the same color share the same parameters.

the model incorporate spatial information in the features as well.

Each proposal p needs to be regressed to a query box q. We use the query-specific feature vector b_q as contextual information to generate regression co-ordinates from the proposal feature b_p. We project b_q to a $4 \times D$ feature and b_p to $1 \times D$ feature using FC layers (Fig. 4). The dot-product between these two features provides the 4 regression co-ordinates. We use a smoothed-L1 loss as per Faster R-CNN to measure deviation of predicted regression co-ordinates from the target co-ordinates of the queries in view-1. We refer to the average of this loss over all Q queries as $\mathcal{L}_{croi-reg}$.

Overall Detection Loss: The overall detection loss is the sum of losses from the CRPN and CROI modules:

$$\mathcal{L}_{det} = \mathcal{L}_{crpn} + \mathcal{L}_{info} + \mathcal{L}_{croi-reg}. \quad (1)$$

3.3. Training

Our model is trained end-to-end to minimize the total loss, which is the sum of classification loss from Section 3.1 and detection loss (Eq. 1).

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{det}. \quad (2)$$

Every training batch of size N contains two-views of $N/2$ distinct images. ROI-pooling is done at the appropriate FPN-level corresponding to the query box-size.

4. Implementation details

4.1. Pretraining

Pre-training dataset. We start with the 1230 object classes defined in LVIS-v0.5 [17]. We expand each class label (eg: "dog") to obtain multiple words that refer to it and convert them to hashtags (eg: "#dogs", "#canine"). We download publicly available images, tagged with at least one of these hashtags. We use the original LVIS labels associated with the hashtags as class labels for these images. We retain only the object classes that have atleast 5000 unique images, resulting in a slightly reduced set of 1209 classes and $49.93M$ images. We refer to this dataset as *IG-50M*. We also show results of using ImageNet-1k [44] for pre-training.

PreDet model. We implement our model using detectron2 [58] and the default Faster R-CNN hyper-parameters in [58] wherever possible. We use the same augmentations as MoCo-v2[18] to construct different "views". However, our model input has a higher resolution of 480×480. We randomly sample 16 EdgeBoxes per image to construct queries, and set the size of $\mathcal{P}_{neg}$ in CROI-head to 512. We train PreDet models with a batch-size of 1024 over 128 GPUs for $540k$ iterations, summing to 8.3 days of training on 128 GPUs. We use an initial learning rate of 0.2 and reduce it by a factor of 0.1 in four uniform steps.

4.2. Fine-tuning

Over the years, various models [30, 38, 42, 20, 33, 55, 65, 66, 52] have been proposed for object detection and instance segmentation. Our pre-training approach can be applied to all these models in practice. In this work, we pick a popular two-stage model (Mask R-CNN [20]) and a popular single-stage model (RetinaNet [30]) to demonstrate the benefits of PreDet. We consider the following datasets.

MS-COCO and LVIS-v1. We use MS-COCO-2017 [31] train and validation datasets for fine-tuning and evaluation respectively. We also experiment with LVIS-v1 [17] which is an extension of MS-COCO with 1203 classes.

We fine-tune the models with standard $1\times, 2\times, 3\times, 6\times, 9\times$ learning rate schedules as suggested in [19] with a batch size of 16. All models are fine-tuned without freezing any layers. We use grid-search to find the optimal learning-rate and decay factor for pre-trained backbones from PreDet as well as the baselines. See the supplementary material for details.

We noticed that for instance segmentation with Mask R-CNN, the mask-head parameters benefited from a larger lr when using pre-trained models like PreDet, SEER and InfoMin. This is due to the mask-head having multiple convolution layers that are initialized from scratch unlike other modules. We performed a grid-search and found that scaling the lr by a factor of 4 worked best for all methods.

The shortest side of the image is resized to a value randomly sampled from $\{640, 672, 704, 736, 768, 800\}$ during training with random crop and flip augmentations. At test-time the shortest side is resized to 800.

PASCAL VOC. The PASCAL [11] dataset contains 20 classes with bounding box annotations. We train with the PASCAL VOC-`trainval07+12` split and evaluate on PASCAL VOC-`test07` split. We train the models for $24k$ iterations with a batch size of 16, starting from a lr of 0.02 and dropping the lr by 0.1 twice at $18k$ and $22k$ iterations.

Evaluation. We report Average Precision for detection AP^{box} and segmentation AP^{mask}, averaged over IoU thresholds from 0.5 to 0.9 as per MS-COCO definitions.

L_{cls}	L_{crpn}	L_{info}	$L_{croi-reg}$	AP^{box}
✓				44.6
	✓	✓	✓	43.7
✓		✓	✓	46.3
✓		✓		45.9
✓	✓	✓	✓	**47.1**

Table 1. Results of fine-tuning Mask R-CNN on MS-COCO with ResNeXt-101-32x8d + FPN pre-trained with different loss components from PreDet.

5. Experiments

We use IG-50M as the default pre-training dataset and ResNeXt-101-32x8d with FPN as the default backbone.

5.1. PreDet design choices

We study the importance of different losses in PreDet model: classification loss (L_{cls}), CRPN losses (L_{cls}) and CROI-head losses (L_{info}, $L_{croi-reg}$). We pre-train ResNeXt-101-32x8d with different combinations of these losses and then fine-tune on MS-COCO for $1\times$ schedule. Tab. 1 shows the results. Note that the first row in which we enable L_{cls} refers to the standard classification setting but with the addition of FPN. Also, while training a model without CRPN losses, we randomly sampled positive and negative boxes for each query box to train the CROI-head.

We see that without classification loss, performance drops significantly by 3.4% (43.7 vs. 47.1). Adding CROI-head losses to this model improves the performance by 1.7% (44.6 vs. 46.3). Within the CROI-head, we observe that the regression loss contributes 0.4% (45.9 vs. 46.3). Finally, the inclusion of CRPN contributes an additional 0.8%. This shows that all components are important and we achieve the best result by training with all the losses.

5.2. Detection results

We report results for models initialized with different pre-training methods: (a) *from scratch*: training from scratch without any pre-training, (b) *cls-imagenet*: pre-training on ImageNet with standard classification loss[3], (c) *cls-IG50M*: pre-training on IG-50M with classification loss only without self-supervised detection losses[4], (d) *InfoMin*: pre-training with the recent self-supervised approach [54] [5], (e) *SEER*: pre-training with self-supervised SWAV [5] method on IG-1B dataset by SEER [13] [6], (f) *PreDet-ImageNet*: PreDet model trained with ImageNet, and (g) *PreDet-IG50M*: PreDet model trained with IG-50M dataset.

[3]We tried both 224×224 and 480×480 input resolutions without any noticeable difference; results are reported for 224×224 resolution.

[4]We tried different hyper-parameters but found the values used for PreDet to be optimal even for this setting.

[5]We used the model released on the project web-page.

[6]we used the model obtained from the authors.

5.2.1 MS-COCO dataset

Tab. 2 shows the performance of Mask R-CNN and RetinaNet fine-tuned on MS-COCO for different pre-training approaches with the $1\times$ schedule. In addition, we also report the best fine-tuning schedule for each approach and the performance at that schedule. More detailed results are shown in the supplementary.

Mask R-CNN with ResNeXt-101-32x8d. For $1\times$ schedule, we notice that both PreDet-ImageNet and PreDet-IG50M achieve better performance than other approaches. Compared to InfoMin, the second best approach, PreDet-IG50M achieves an AP^{box} improvement of 2.3% (44.8 vs. 47.1) and AP^{mask} improvement of 1.5% (40.2 vs. 41.7%). We also observe that classification-only pre-trained models (cls-IG50M and cls-ImageNet) have significantly lower performance compared to their PreDet counterparts. Also, while PreDet-ImageNet outperforms other baselines, it falls short of PreDet-IG50M (45.8 vs 47.1). This indicates that pre-training on larger noisy dataset has significant benefit over smaller fully-supervised dataset. Finally, model trained from scratch performs worse than other approaches.

Comparing the best performance among all the schedules, we observe that PreDet-IG50M outperforms training from scratch, the second best approach, by 1.3% (45.8 vs. 47.1) in AP^{box} and by 1.2% (40.7 vs. 41.7) in AP^{mask}. More importantly, while all other approaches achieve their best performance at $3\times$ or higher schedule, PreDet does so at $1\times$ schedule. Thus, not only does PreDet achieve significantly better performance, it also converges faster.

We also compare AP^{box}_{50}, AP^{box}_{75} [7] numbers in Tab. 3 for the top approaches for Mask R-CNN with best schedules. Note that AP^{box}_{75} is a stricter metric for localization compared to AP^{box}_{50}. We observe that InfoMin improves AP^{box}_{50} compared to training from scratch, but suffers a decrease in AP^{box}_{75}. This shows that self-supervised approaches improve image-level classification which helps improve coarser object detection reflected by AP^{box}_{50}, but not precise localization reflected by AP^{box}_{75}. PreDet-IG50M improves both AP^{box}_{50}, AP^{box}_{75} by $2.6\%, 1.4\%$ respectively, compared to the model trained from scratch.

RetinaNet with ResNeXt-101-32x8d. A significant improvement is observed for the RetinaNet model in Tab. 2, where PreDet-IG50M trained for only $1\times$ schedule improves AP^{box} by 2.1% (43.0% vs. 45.1%) compared to all other baselines trained for longer schedules. Also, all approaches outperform training from scratch.

Mask R-CNN with other backbones. In order to compare with the directly published results in other works, we also train a ResNet-50 model and a larger RegNet64 [40] Mask R-CNN model and show results in Tab. 4. For ResNet-50, we compare with self-supervised methods like MoCo-

[7]AP^{box}_{50}, AP^{box}_{75} are AP at IoU thresholds 0.50, 0.75 respectively.

pre-training	Mask R-CNN					RetinaNet		
	1× sched.		best sched.			1× sched.	best sched.	
	AP^{box}	AP^{mask}	sched.	AP^{box}	AP^{mask}	AP^{box}	sched.	AP^{box}
from scratch	33.9	31.0	9×	45.8	40.7	27.6	9×	40.7
cls-ImageNet	43.8	39.0	6×	44.9	39.9	41.4	1×	41.4
cls-IG50M	44.4	39.4	3×	44.6	39.5	41.8	1×	41.8
InfoMin [54]	44.8	40.2	3×	45.6	40.5	43.0	1×	43.0
SEER-IG1B [13]	44.3	39.9	3×	45.1	40.1	40.3	6×	41.7
PreDet-ImageNet	45.8	40.8	1×	45.8	40.8	43.1	1×	43.1
PreDet-IG50M	**47.1**	**41.7**	**1×**	**47.1**	**41.7**	**45.1**	**1×**	**45.1**

Table 2. Results on MS-COCO for Mask R-CNN and RetinaNet, with ResNeXt-101-32x8d + FPN backbone when pre-trained with different approaches. We report results for 1× fine-tuning for all approaches. We also report the best fine-tuning schedule and the performance at this schedule for each approach.

pre-training	AP^{box}	AP^{box}_{50}	AP^{box}_{75}
from scratch	45.8	65.6	50.2
InfoMin	45.6	65.9	49.9
PreDet-IG50M	**47.1**	**68.2**	**51.6**

Table 3. AP^{box} , AP^{box}_{50} , AP^{box}_{75} for Mask R-CNN initialized from scratch, Infomin and PreDet-IG50M, fine-tuned on MS-COCO.

v2 [18] and InfoMin [54]. Unlike the ResNeXt models in Tab. 2, we train Mask R-CNN with a deeper RoI-head (4 convolution and 1 fully-connected layers) as per MoCo-v2 and InfoMin settings for fair comparison. Even for a smaller ResNet-50 model, our approach shows significant improvement for 1× schedule. In the same table, we also compare results for RegNet64 with the SEER [13] model. PreDet achieves better performance compared to other pre-training methods. Note that SEER is pre-trained on $1B$ images while PreDet training is performed on $50M$ images. Since RegNet64 is a very high capacity model, we expect PreDet performance to improve further with a larger dataset.

5.2.2 LVIS-v1 dataset

Tab. 5 shows the AP^{box} , AP^{mask} results for the LVIS-v1 dataset, averaged over 3 runs. We see slightly different trends compared to MS-COCO. All models converge faster compared to MS-COCO. We observe that the best AP^{box} results for a model initialized with SEER (28.6) is better than the one for a model trained from scratch (28.1). Also, SEER performs better (28.2) than InfoMin (27.3) unlike on COCO, where InfoMin outperformed SEER. However, PreDet-IG50M trained for just 1× schedule achieves the best AP^{box} of 30.1 compared to all other models. This shows that PreDet can consistently transfer well to different target datasets. Similar trends are observed for AP^{mask} .

5.2.3 PASCAL VOC dataset

Tab. 6 shows AP^{box} , AP^{box}_{50} and AP^{box}_{75} results for Faster R-CNN with ResNeXt-101-32x8d model, averaged over 3 runs. Our PreDet models significantly outperform other pre-training approaches. It improves AP^{box} by 3.9% compared to the next best model, InfoMin (62.5% vs 58.6%). Interestingly InfoMin outperforms SEER by 1.2% (66.0 vs 64.8) in AP^{box}_{75} but suffers a drop in AP^{box}_{50} of 1.7% (83.1 vs 84.8), while PreDet improves both numbers.

5.3. Effect of fine-tuning dataset size

We study the effect of PreDet as we vary the size of the target dataset during fine-tuning. Similar to [19], we sample $1k, 5k, 10k, 35k$ images at random from MS-COCO-train dataset to create smaller datasets. For each of them, we use grid-search to choose the best learning rate schedule (detailed in the supp. document). We also use a larger training-time scale augmentation range of $[512, 800]$ as per [19]. We compare the AP^{box} for ResNeXt-101-32x8d Mask R-CNN models initialized from scratch, ImageNet pre-trained model and PreDet-IG50M in Tab. 7.

We notice that the effect of pre-training is more pronounced when the dataset size is smaller. ImageNet pre-trained models outperform the from-scratch training when the dataset size is $10k$ or smaller. PreDet models outperform all other models in every setting, with improvement ranging from 4.8% (11.3% vs. 16.1%) at size $1k$ to 1.3% (45.8% vs. 47.1%) at size $118k$. This demonstrates the importance of good pre-training in the low-shot settings.

5.4. Effect of model capacity

We evaluate the effect of our pre-training approach on backbones of different sizes. In Fig. 5(a), we plot AP^{box} when fine-tuning Mask R-CNN on MS-COCO for ResNeXt models with varying number of parameters with 1× schedule. We observe that compared to ImageNet pre-training, improvement is larger for higher capacity models (3.6% for 101-32x16d) than for smaller capacity (2.3% for 50-32x4d) models. This is consistent with the observations for classification pre-training in previous works [34].

5.5. Analyzing features from PreDet

We worked with the hypothesis that classification pre-training incorporates translation invariance, which causes bounding boxes in an image with limited overlap to have

Model	Pre-training method	Schedule	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
ResNet-50 + FPN	cls-ImageNet	1×	39.7	59.5	43.3	35.9	56.6	38.6
	MoCo-v2 [18]		40.1	59.8	44.1	36.3	56.9	39.1
	InfoMin [54]		40.6	60.6	44.6	36.7	57.7	39.4
	PreDet-IG50M		**42.1**	**62.5**	**46.0**	**37.4**	**59.1**	**39.9**
ResNet-50 + FPN	cls-ImageNet	2×	41.6	61.7	45.3	37.6	58.7	40.4
	MoCo-v2 [18]		41.7	61.6	45.6	37.6	58.7	40.5
	InfoMin [54]		42.5	62.7	46.8	38.4	59.7	41.4
	PreDet-IG50M		**43.3**	**63.3**	**47.7**	**38.7**	**60.5**	**41.5**
RegNet-64 + FPN	cls-ImageNet	1×	45.9	67.8	50.9	41.0	65.3	44.0
	SEER [14]		48.1	**70.5**	52.9	43.2	**67.6**	46.4
	PreDet-IG50M		**49.1**	**70.5**	**53.9**	**43.3**	67.4	**46.9**

Table 4. Results for Mask R-CNN on MS-COCO with ResNet-50 and RegNet64 backbones when pre-trained with different approaches.

pre-training	1× sched.		best sched.		
	AP^{box}	AP^{mask}	sched.	AP^{box}	AP^{mask}
from scratch	15.1	14.9	6×	28.0	26.9
cls-ImageNet	24.5	24.3	3×	25.6	24.8
cls-IG50M	24.0	23.7	3×	25.6	24.7
InfoMin [54]	25.3	24.7	2×	27.3	26.3
SEER-IG1B [13]	28.2	27.7	2×	28.6	27.8
PreDet-ImageNet	26.1	25.6	1×	26.1	25.6
PreDet-IG50M	**30.1**	**29.2**	1×	**30.1**	**29.2**

Table 5. Results on LVIS-v1 for Mask R-CNN with ResNeXt-101-32x8d + FPN backbone when pre-trained with different approaches. We report results for 1× fine-tuning and the best fine-tuning schedule for each model.

pre-training	AP^{box}	AP^{box}_{50}	AP^{box}_{75}
from scratch	36.4	62.4	36.8
cls-ImageNet	56.8	82.7	63.8
InfoMin [54]	58.6	83.1	66.0
SEER-IG1B [13]	58.5	84.8	64.8
PreDet-ImageNet	61.1	84.8	68
PreDet-IG50M	**62.5**	**85.6**	**69.8**

Table 6. Results on PASCAL VOC for Faster R-CNN with ResNeXt-101-32x8d + FPN when pre-trained with different approaches. We report results on PASCAL VOC-`test07` after fine-tuning on PASCAL VOC-`trainval07+12` for $24k$ iterations.

Dataset size	From scratch	cls-ImageNet	PreDet-IG50M
1k	4.2	11.3	16.1
5k	12.5	22.3	26.7
10k	25.3	26.5	30.3
35k	37.8	37.6	40.1
118k	45.8	44.9	47.1

Table 7. Results for Mask R-CNN with ResNeXt-101-32x8d + FPN trained on MS-COCO datasets of varying sizes.

similar representations. This should be rectified by PreDet. In other words, similarity between bounding boxes in an image, measured using PreDet features should be more correlated with the overlap between the boxes. We test this hypothesis through an analysis experiment.

We sampled multiple pairs of bounding boxes from an image, such that their Intersection over Union (IoU) is uniformly distributed from $0.05-0.95$. For each bounding box pair, we measured the cosine-similarity between the ROI-pooled features of the boxes, extracted with a pre-trained model. We repeat this for different images. We plot the

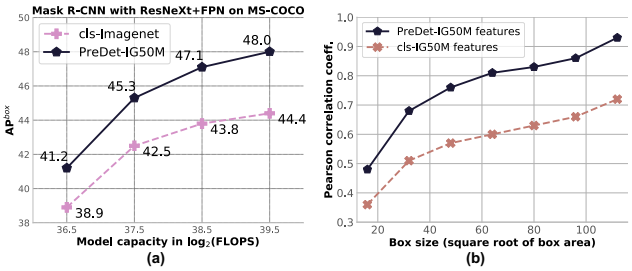

Figure 5. (a) AP^{box} for ImageNet pre-trained and PreDet pre-trained models of varying capacities are shown. The four models shown here are ResNeXt-50x4d, 101-32x4d, 101-32x8d and 101-32x16d, arranged in increasing order of compute. (b) Correlation between IoU overlap of bounding boxes and their visual similarity measured using different pre-trained features. We plot this correlation for bounding boxes of different sizes.

Pearson correlation coefficient between IoU overlap and visual similarity, for box-pairs of different sizes in Fig. 5(b). We measured visual similarity using PreDet-IG50M as well as cls-IG50M model. Across all box-sizes, the correlation is significantly higher with PreDet. This shows that PreDet features preserve spatial overlap between different regions in an image, making them better suited for detection.

6. Conclusion

Pre-training a model with a classification task has long been used to speed-up training and improve the performance of target detection tasks. However, as we shown analytically, classification pre-training focuses on translation invariance, which causes the model to learn similar representations for non-overlapping bounding boxes in an image. This can be detrimental for detection. We proposed a strategy to get around this issue, by augmenting classification pre-training with novel self-supervised detection losses. This approach referred to as PreDet achieves state-of-the-art detection performance with significant speed-ups in fine-tuning time for detection on multiple datasets: MS-COCO, LVIS-v1 and PASCAL VOC. We studied the effect of PreDet on models of different capacities and showed its effectiveness in low-shot training regimes as well.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 2

[2] Alessandro Bergamo and Lorenzo Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. *Advances in neural information processing systems*, 23:181–189, 2010. 2

[3] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pages 517–526. PMLR, 2017. 2

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 2, 6

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[8] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1431–1439, 2015. 2

[9] Santosh K Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3270–3277, 2014. 2

[10] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015. 2

[11] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 2, 5

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *arXiv preprint arXiv:1810.12890*, 2018. 2

[13] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021. 2, 6, 7, 8

[14] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6391–6400, 2019. 8

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2

[16] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018. 2

[17] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 2, 5

[18] 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. 1, 2, 3, 4, 5, 7, 8

[19] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 2, 5, 7

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 3, 5

[21] Seunghoon Hong, Donghun Yeo, Suha Kwak, Honglak Lee, and Bohyung Han. Weakly supervised semantic segmentation using web-crawled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7322–7330, 2017. 2

[22] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. 2019. 2

[23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021. 2

[24] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6(2):8, 2019. 2

[25] Jason Kuen, Federico Perazzi, Zhe Lin, Jianming Zhang, and Yap-Peng Tan. Scaling object detection by transferring classification weights. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6044–6053, 2019. 2

[26] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015. 1

[27] Chunyuan Li, Xiujun Li, Lei Zhang, Baolin Peng, Mingyuan Zhou, and Jianfeng Gao. Self-supervised pre-training with

hard examples improves visual representations. *arXiv preprint arXiv:2012.13493*, 2020. 2

[28] 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. 2

[29] Yandong Li, Di Huang, Danfeng Qin, Liqiang Wang, and Boqing Gong. Improving object detection with selective self-supervised self-training. In *European Conference on Computer Vision*, pages 589–607. Springer, 2020. 2

[30] 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

[31] 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. 2, 5

[32] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. 2

[33] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019. 5

[34] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 1, 2, 7

[35] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 2

[36] Li Niu, Qingtao Tang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Learning from noisy web data with category-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7689–7698, 2018. 2

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4

[38] Wanli Ouyang, Kun Wang, Xin Zhu, and Xiaogang Wang. Chained cascade network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1938–1946, 2017. 5

[39] 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. 2

[40] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 6

[41] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 2

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 5

[43] 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. 3, 4

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[45] 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/CVF International Conference on Computer Vision*, pages 8430–8439, 2019. 2

[46] Tong Shen, Guosheng Lin, Chunhua Shen, and Ian Reid. Bootstrapping the performance of webly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1363–1371, 2018. 2

[47] Yunhang Shen, Rongrong Ji, Zhiwei Chen, Yongjian Wu, and Feiyue Huang. Uwsod: Toward fully-supervised-level capacity weakly supervised object detection. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[48] 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

[49] Yosuke Shinya, Edgar Simo-Serra, and Taiji Suzuki. Understanding the effects of pre-training for object detectors via eigenspectrum. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[50] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2

[51] 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, 2

[52] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 5

[53] Qingyi Tao, Hao Yang, and Jianfei Cai. Zero-annotation object detection with web knowledge transfer. In *Proceedings*

of the European Conference on Computer Vision (ECCV), pages 369–384, 2018. 2

[54] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 2, 6, 7, 8

[55] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019. 5

[56] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[57] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2314–2320, 2016. 2

[58] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[59] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2

[60] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 2

[61] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 2

[62] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019. 2

[63] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 2

[64] Dongzhan Zhou, Xinchi Zhou, Hongwen Zhang, Shuai Yi, and Wanli Ouyang. Cheaper pre-training lunch: An efficient paradigm for object detection. In *European Conference on Computer Vision*, pages 258–274. Springer, 2020. 2

[65] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 5

[66] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2019. 5

[67] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 2268–2277, 2019. 2

[68] Bohan Zhuang, Lingqiao Liu, Yao Li, Chunhua Shen, and Ian Reid. Attend in groups: a weakly-supervised deep learning framework for learning from web data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1878–1887, 2017. 2

[69] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. 3

[70] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020. 2