CoDo: Contrastive Learning with Downstream Background Invariance for Detection

Bing Zhao Jun Li Hong Zhu
Department of AI and HPC Inspur Electronic Information Industry Co., Ltd
Beijing, China
{zhaobing01,lijun09,zhuHongbj}@inspur.com

Abstract

The prior self-supervised learning researches mainly select image-level instance discrimination as pretext task. It achieves a fantastic classification performance that is comparable to supervised learning methods. However, with degraded transfer performance on downstream tasks such as object detection. To bridge the performance gap, we propose a novel object-level self-supervised learning method, called Contrastive learning with Downstream background invariance (CoDo). The pretext task is converted to focus on instance location modeling for various backgrounds, especially for downstream datasets. The ability of background invariance is considered vital for object detection. Firstly, a data augmentation strategy is proposed to paste the instances onto background images, and then jitter the bounding box to involve background information. Secondly, we implement architecture alignment between our pretraining network and the mainstream detection pipelines. Thirdly, hierarchical and multi views contrastive learning is designed to improve performance of visual representation learning. Experiments on MSCOCO demonstrate that the proposed CoDo with common backbones, ResNet50-FPN, yields strong transfer learning results for object detection.

1. Introduction

The paradigm of supervised pretraining and finetuning has been dominant in computer vision for a period of time. Typically, the pretraining is optimized on large-scale labeled datasets, and then regarded as initialized weights to finetune for various downstream tasks [12] [14] [32]. Instead, self-supervised learning (SSL) aims to learn the generic pretraining representations, independent of manual labels [18]. Recently, SSL has achieved a performance comparable to supervised pretraining in image classification [10] [2] [1] [9]. However, it suffers a performance degradation when applied to downstream tasks, such as object detection. It indicates that the existing SSL approaches mainly focus on image classification, without considering the location modeling ability for object detection. The performance gap of mainstream SSL methods in image classification and object detection is as suggested in Figure 1. It can be observed that the linear classification accuracy of the relevant SSL methods on ImageNet dataset is increasing from 67.5% to 75%. While compared with MOCO [10], which is proposed in 2020, the detection performance (mAP) of recent approaches is reduced when finetuning on MSCOCO dataset.

Existing works that contribute to bridge the gap usually focus on pretext task and architectural alignment. Firstly, instance discrimination, the typical SSL pretext task, usually assumes that different data-augmentations (views) of the same image should be similar but discriminable from other images. Generally speaking, researchers tend to be-
Figure 2. The overview of CoDo. For a pretraining image, we utilize selective search to generate a series of proposals. Only one proposal is randomly selected, then pasted onto various downstream background images.

2. Related work

Self-supervised learning refers to learn visual features from unlabeled data without human annotations [13]. A mainstream solution is to utilize various pretext tasks to generate pseudo labels in order to obtain the generalizable representations, such as Rotation [8], Colorization [30], Inpainting [19], Jigsaw Puzzle [6] and so on. Recently, contrastive learning becomes the most popular image-level pretext task for self-supervised learning [4]. The optimization goal is to maximize the representation of image instances and their corresponding views, while minimizing the remaining other image instances [25]. The further contrastive learning approaches focus on a better construction of negative sample and a simplification of network structure. For instance, Momentum Contrast (MoCo) [10] builds a moving-averaged encoder and maintain a negative sample queue to minimize a contrastive loss. SimCLR [2] verifies the effectiveness of data augmentation strategy and increases the batch size of training samples for contrastive learning. Bootstrap Your Own Latent (BYOL) [9] minimizes a similarity loss between online and target networks without using negative pairs. Barlow Twins [29] introduces a cross-correlation matrix to avoid trivial constant solutions, the simplified model dose not need predictor network, momentum encoder and stop-gradients any more.

The further goal of SSL is to learn general representations which can be transferred for downstream tasks. Some modified approaches have been proposed to bridge the gap between pretraining and downstream tasks. Detco [26] and Insloc [28] separately design a detection-friendly pretext task. Detco contrasts the global image and local image patches to improve object detection. Insloc constructs a new view by randomly pasting foreground objects in different background images for contrastive learning. OLR [27] and Soco [23] propose object-level unsupervised representation learning framework for object detection respectively. Self-
EMD [17] performs the Earth Mover’s Distance as a metric to measure the spatial similarity between two image representations in order to learn visual spatial representations for object detection. DiLo [31] selects saliency estimation to localize the foreground object in a data-driven approach. Self-EMD [17] directly pre-trains on MSCOCO and achieves a higher detection performance, instead of commonly used ImageNet.

3. Method

We propose a new object-level contrastive learning framework as shown in Figure 2, which introduces the unlabeled downstream datasets information during pre-training. The proposals of the pretraining images are pasted to the downstream images, the bonding box of proposals is deformed to generate new views with downstream information. It enables localization of the foreground object during pretraining, and can be regarded as a simulation of object detection during pretrained. Considering that no supervised information of downstream dataset is introduced into the pre-training, so there is no risk of data leakage. Meanwhile, we introduce architectural alignment to pretrain the essential properties of object detector, such as FPN, RoiAlign and R-CNN head. Our proposed method is detailed below.

3.1. Copy, paste and jitter (CPJ)

We design a data augmentation method for proposals to realize location modeling, terms as CPJ. In order to realize object-level contrastive learning, we select selective search to generate proposal in unsupervised way. Considering ImageNet is usually regarded as a single-centric-object dataset, most proposals are similar. So we only randomly select one proposal for each image, those proposals with too large or too small aspect ratios (>3 or ≤1/3) are ignored. In addition to ImageNet, we add two downstream object detection datasets (MSCOCO and Pasco VOC [7]) as alternative background images. For the same pretraining image, depending on the number of views involved in the contrastive learning, we randomly select the corresponding number of background images to generate the pasted images.

Background invariance is important for object detection, namely, a robust detector can recognize the foreground objects on various backgrounds. We paste the proposals at arbitrary aspect ratios and scales onto various downstream background images. In this process, translation and scale invariance are also considered. The pasted position is treated as the bounding box. Then the bounding box is jittered to contain background images. The jittered boxes are filtered by a proper IoU threshold, we set it to be greater than 0.6. The whole image I and transformed bounding box bb are regarded as inputs of Query network and Key network to conduct contrastive learning.

\[ I_q, bb_q = CPJ(I_p, I_{dq}). \]  
\[ I_{k_i}, bb_{k_i} = CPJ(I_p, I_{dk_i}). \]

Where \( I_p \) is the proposal of pretraining images, \( I_{dq} \) is the downstream background image for Query network, and \( I_{dk_i} \) is the i-th downstream background image for Key network.

To be consistent with existing SSL methods, the default version of our proposed method selects two views for Query network and Key network separately. Actually, multi views help to increase diversity during contrastive learning. We also design a 4 views version, which view 2 to view 4 are inputs of Key network.

3.2. Hierarchical contrastive learning

In our approach, the pipeline of MOCO-V2 [3] is adopted as baseline for learning contrastive representations. We will describe how to achieve structural alignment for MOCO V2 in this section. We select the Resnet50 with FPN as Query network \( f^q \) and Key network \( f^k \). FPN is a common component in the object detector, which can fuse different feature maps to reach a better detection performance. To further align with Mask R-CNN, RoiAlign is introduced to extract feature of bounding box from the output of FPN. For Query network, the object-level feature representation \( v_q \) is extracted from an image \( I_q \) and the corresponding bounding box \( bb_q \) as follows. The computation for Key network is similar.

\[ v_q = \text{RoIAlign} \left(f^q(I_q), bb_q\right). \]
\[ v_{k_i} = \text{RoIAlign} \left(f^k(I_{k_i}), bb_{k_i}\right). \]

And R-CNN head \( f^{R-H} \) is built to obtain embeddings for contrastive learning. The latent embeddings \( e_q \) and \( e_{k_i} \) for Query network and Key network are as follows:

\[ e_q = f^{R-H}(v_q). \]
\[ e_{k_i} = f^{R-H}(v_{k_i}). \]

SSL usually selects contrastive loss, i.e. InfoNCE to compute the similarity between views. For Query view and i-th Key view, the loss function is as follows. Notably, The calculation is performed in a hierarchical manner. Because \( e_q \) and \( e_{k_i} \) can be divided corresponding to the output of FPN (P2, P3, P4, P5). Contrastive learning can be carried out at a finer level.

\[ L_{q-k_i} = - \log \frac{\exp(e_q \cdot e_{k_i} / \tau)}{\sum_{i=0}^{N} \exp(e_q \cdot e_{k_i} / \tau)} \]
Table 1. Downstream task performance on COCO by using Mask R-CNN with R50-FPN.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Epoch</th>
<th>AP&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sup&gt;mk&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;&lt;sup&gt;mk&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;&lt;sup&gt;mk&lt;/sup&gt;</th>
<th>AP&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sup&gt;mk&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;&lt;sup&gt;mk&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;&lt;sup&gt;mk&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sepurvised</td>
<td>90</td>
<td>38.9</td>
<td>59.6</td>
<td>42.7</td>
<td>35.4</td>
<td>56.5</td>
<td>38.1</td>
<td>41.3</td>
<td>61.3</td>
<td>45.0</td>
<td>37.3</td>
<td>58.3</td>
<td>40.3</td>
</tr>
<tr>
<td>MOCO</td>
<td>200</td>
<td>38.5</td>
<td>58.9</td>
<td>42.0</td>
<td>35.1</td>
<td>55.9</td>
<td>37.7</td>
<td>40.8</td>
<td>61.6</td>
<td>44.7</td>
<td>36.9</td>
<td>58.4</td>
<td>39.7</td>
</tr>
<tr>
<td>MOCO v2</td>
<td>200</td>
<td>40.4</td>
<td>60.2</td>
<td>44.2</td>
<td>36.4</td>
<td>57.2</td>
<td>38.9</td>
<td>41.7</td>
<td>61.6</td>
<td>45.6</td>
<td>37.6</td>
<td>58.7</td>
<td>40.5</td>
</tr>
<tr>
<td>SimCLR</td>
<td>200</td>
<td>39.6</td>
<td>59.1</td>
<td>42.9</td>
<td>34.6</td>
<td>55.9</td>
<td>37.1</td>
<td>40.8</td>
<td>60.6</td>
<td>44.4</td>
<td>36.9</td>
<td>57.8</td>
<td>39.8</td>
</tr>
<tr>
<td>InfoMin</td>
<td>200</td>
<td>40.6</td>
<td>60.6</td>
<td>44.6</td>
<td>36.7</td>
<td>57.7</td>
<td>39.4</td>
<td>42.5</td>
<td>62.7</td>
<td>46.8</td>
<td>38.4</td>
<td>59.7</td>
<td>41.4</td>
</tr>
<tr>
<td>InfoMin</td>
<td>800</td>
<td>41.2</td>
<td>61.2</td>
<td>44.8</td>
<td>35.9</td>
<td>57.9</td>
<td>38.4</td>
<td>42.1</td>
<td>62.3</td>
<td>46.2</td>
<td>38.0</td>
<td>59.5</td>
<td>40.8</td>
</tr>
<tr>
<td>BYOL</td>
<td>300</td>
<td>40.4</td>
<td>61.6</td>
<td>44.1</td>
<td>37.2</td>
<td>58.8</td>
<td>39.8</td>
<td>42.3</td>
<td>62.6</td>
<td>46.2</td>
<td>38.3</td>
<td>59.6</td>
<td>41.1</td>
</tr>
<tr>
<td>SAWAY</td>
<td>400</td>
<td>39.6</td>
<td>60.1</td>
<td>42.9</td>
<td>34.7</td>
<td>56.6</td>
<td>36.6</td>
<td>42.3</td>
<td>62.8</td>
<td>46.3</td>
<td>38.2</td>
<td>60.0</td>
<td>41.0</td>
</tr>
<tr>
<td>DenseCL</td>
<td>200</td>
<td>40.3</td>
<td>59.9</td>
<td>44.3</td>
<td>36.4</td>
<td>57.0</td>
<td>39.2</td>
<td>41.2</td>
<td>61.9</td>
<td>45.1</td>
<td>37.3</td>
<td>58.9</td>
<td>40.1</td>
</tr>
<tr>
<td>InsLoc</td>
<td>200</td>
<td>41.4</td>
<td>61.7</td>
<td>45.0</td>
<td>37.1</td>
<td>58.5</td>
<td>39.6</td>
<td>42.3</td>
<td>63.5</td>
<td>47.5</td>
<td>38.7</td>
<td>60.5</td>
<td>41.9</td>
</tr>
<tr>
<td>CoDo</td>
<td>200</td>
<td>41.2</td>
<td>61.3</td>
<td>45.1</td>
<td>36.9</td>
<td>58.2</td>
<td>39.4</td>
<td>42.6</td>
<td>66.0</td>
<td>46.7</td>
<td>38.2</td>
<td>59.7</td>
<td>41.0</td>
</tr>
<tr>
<td>CoDo</td>
<td>400</td>
<td>41.9</td>
<td>61.8</td>
<td>45.8</td>
<td>37.4</td>
<td>58.8</td>
<td>40.3</td>
<td>43.1</td>
<td>63.4</td>
<td>47.1</td>
<td>38.8</td>
<td>60.6</td>
<td>41.4</td>
</tr>
<tr>
<td>CoDo&lt;sub&gt;m&lt;/sub&gt;</td>
<td>400</td>
<td>43.1</td>
<td>63.3</td>
<td>47.1</td>
<td>38.3</td>
<td>60.3</td>
<td>41.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2. The influence of background datasets on COCO by using Mask R-CNN with R50-FPN. Q means Query Network and K means Key Network.

<table>
<thead>
<tr>
<th>Background Datasets</th>
<th>Epoch</th>
<th>Schedule</th>
<th>AP&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;&lt;sup&gt;bb&lt;/sup&gt;</th>
<th>AP&lt;sup&gt;mk&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;&lt;sup&gt;mk&lt;/sup&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;&lt;sup&gt;mk&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q: ImageNet K: COCO</td>
<td>200</td>
<td>1x</td>
<td>40.7</td>
<td>-</td>
<td>60.7</td>
<td>44.5</td>
<td>36.5</td>
<td>57.6</td>
</tr>
<tr>
<td>Q/K: ImageNet+COCO</td>
<td>200</td>
<td>1x</td>
<td>40.8</td>
<td>&lt;0.1</td>
<td>60.7</td>
<td>44.7</td>
<td>36.6</td>
<td>57.6</td>
</tr>
<tr>
<td>Q/K: ImageNet+COCO+VOC</td>
<td>200</td>
<td>1x</td>
<td>41.2</td>
<td>&lt;0.5</td>
<td>61.3</td>
<td>45.1</td>
<td>36.9</td>
<td>58.2</td>
</tr>
<tr>
<td>Q: ImageNet K: ImageNet+COCO+VOC</td>
<td>400</td>
<td>1x</td>
<td>41.4</td>
<td>&lt;0.5</td>
<td>61.7</td>
<td>45.1</td>
<td>37.3</td>
<td>58.6</td>
</tr>
<tr>
<td>Q/K: ImageNet+COCO+VOC</td>
<td>400</td>
<td>1x</td>
<td>41.9</td>
<td>&lt;0.5</td>
<td>61.8</td>
<td>45.8</td>
<td>37.4</td>
<td>58.8</td>
</tr>
</tbody>
</table>

Where \( \tau \) and \( N \) are the temperature and the number of negative samples, respectively.

For multi-view version of our proposed method, we can calculate InfoNCE between view q and all view k. The total loss function is as follows.

\[
L_{q-k} = \sum_{j=0}^{N} (L_{q-k})_j \tag{8}
\]

Where \( N \) is the number of view k.

4. Experiment

4.1. Dataset

The widely used ImageNe-1K with 1.28 million images is adopted as dataset for self-supervised pretraining. MSCOCO is used for finetuning to evaluate the generalization performance on downstream tasks. Significantly, in the procession of data processing, the training sets of ImageNe-1K, PASCAL VOC0712 and MSCOCO are involved as the background images.

4.2. Setting for pretraining and finetuning

During pretraining, we mainly follow the hyperparameters setting of Moco-v2, the total batch size is set to 1024 over 8 Nvidia A100 GPUs, and the initial learning rate is set to 0.06. The optimization takes 200 and 400 epochs for the evaluation of downstream tasks, respectively. During pretraining, we employ the data augmentation pipeline of Moco-v2 for pretraining proposals and the background images.

For finetuning, we validate the performance of pretraining representation on downstream tasks based on Detectron2 [24]. On COCO, we adopt the Mask R-CNN with the R50-FPN. The performances of object detection and instance segmentation under 1x and 2x schedules are reported. The batch size is set to 128 over 8 GPUs. The fine tuning iteration step is set to 90000 and 180000 on the 1x schedule and 2x schedule, respectively. The initial learning rate is 0.02. Finally, on the 1x schedule and 2x schedule, the results of \( AP_{bb} \), \( AP_{50}^{bb} \) and \( AP_{75}^{bb} \) for object detection, \( AP_{mk} \), \( AP_{50}^{mk} \), and \( AP_{75}^{mk} \) for instance segmentation are compared with the state-of-the-art methods. R50-FPN (ResNet-50 with FPN) is the common backbone network for Mask R-CNN and Faster R-CNN to evaluate transfer performance.

4.3. Results

We report the experimental results on object detection and instance segmentation with state-of-the-art approaches. Some of these sota methods are designed to focus on classification, such as SimCLR [2], MoCo [10], MoCo v2 [3], BYOL [9], InfoMin [20] and SwAV [1]. While others are specifically suited to object detection, such as...
Mask R-CNN on MSCOCO. Table 1 shows the results for Mask R-CNN with R50-FPN. The finetuning follows with the COCO 1x and 2x schedules. We compare the proposed method under 200 and 400 epochs of pretraining. The results show that our method exceeds the above two kinds of methods. On 1x schedule, Our method outperforms the baseline MoCo-v2 by +0.8 AP for the R50-FPN. On 2x schedule, our method exceeds MoCo-v2 by +0.8. The multi-view version of CoDo, further boosts the performance and reaches a 43.1 AP.

The influence of Background datasets. We analyze the background datasets selection strategy for Query Network and Key Network. Table 2 shows the influence of background datasets on object detection. Firstly, considering the volume gap between ImageNet and object detection datasets, if we only select the object detection datasets as background may cause homogenization. A proper solution is to gather ImageNet and object detection datasets together as the source of the background images. And the diversity of background dataset sources helps to improve performance. Under 200 epochs, this setting contributes a +0.5AP. Secondly, the background datasets selection for Query and Key Network should be same to reduce Disequilibrium. Under 200 epochs, this role also contributes a +0.5AP.

5. Conclusion
In this paper, we noticed that most existing self-supervised learning methods ignore the function of downstream datasets, especially the location ability of foreground objects in downstream scenes. So we propose a new contrastive learning method, CoDo, to achieve the downstream background invariance of the foreground objects. It is implemented by pasting foreground proposals onto various downstream images for contrastive learning. CoDo achieves a strong results on transfer performance for object detection on MSCOCO. The experimental results demonstrate that transfer performance for object detection can be strengthened by considering downstream background invariance.

References


