InsCLR: Improving Instance Retrieval with Self-Supervision

Zelu Deng[1]*, **Yujie Zhong**[2]*, **Sheng Guo**[3], **Weilin Huang**[4]†

[1] Dmall
[2] Meituan Inc.
[3] MY Bank, Ant Group
[4] Alibaba Group
zelu.deng@dmall.com, jaszhong@hotmail.com, {guosheng.guosheng, weilin.hwl}@alibaba-inc.com

Abstract

This work aims at improving instance retrieval with self-supervision. We find that fine-tuning using the recently developed self-supervised learning (SSL) methods, such as SimCLR and MoCo, fails to improve the performance of instance retrieval. In this work, we identify that the learnt representations for instance retrieval should be invariant to large variations in viewpoint and background etc., whereas self-augmented positives applied by the current SSL methods can not provide strong enough signals for learning robust instance-level representations. To overcome this problem, we propose InsCLR, a new SSL method that builds on the *instance-level* contrast, to learn the intra-class invariance by dynamically mining meaningful pseudo positive samples from both mini-batches and a memory bank during training. Extensive experiments demonstrate that InsCLR achieves similar or even better performance than the state-of-the-art SSL methods on instance retrieval. Code is available at https://github.com/zeludeng/insclr.

Introduction

Large-scale instance-level image retrieval (also known as particular object retrieval) has been studied for over two decades. Given an image with a query object, the goal is to retrieve all the images containing the query object in a large corpus of images. A key challenge is to design or learn image-level descriptors for the accurate search. Recently, several works show that fine-tuning the pretrained network on large-scale instance-retrieval datasets can significantly improve the performance (Gordo et al. 2016; Noh et al. 2017; Weyand et al. 2020). However, annotating large-scale data is time-consuming and requires huge human labour. Alternatively, image labels can be generated by using the reconstructed 3D models obtained from a traditional retrieval system (Radenović, Tolias, and Chum 2016, 2018), and the advanced Structure-from-Motion (SfM) pipeline (Schonberger and Frahm 2016). It can be considered as training networks without human annotation, but using supervision information generated from other computer vision systems which are also expensive to implement.

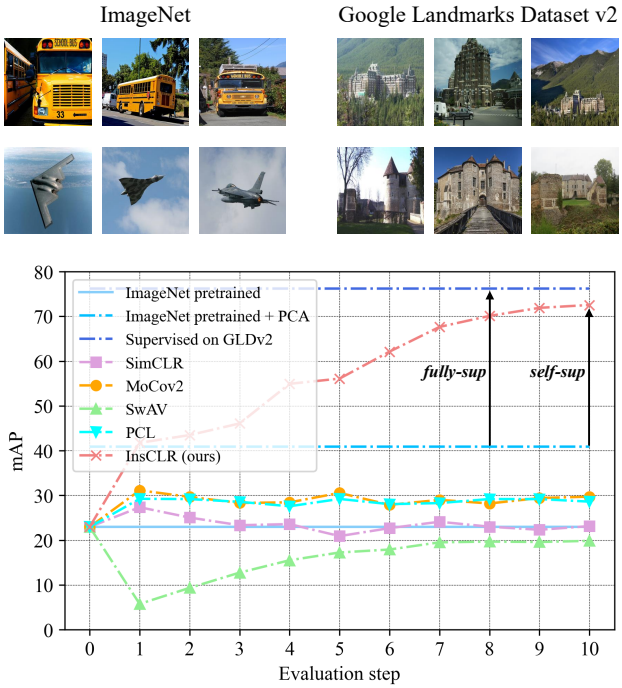

Figure 1: Top: example images from ImageNet and Google Landmarks Dataset v2 (GLDv2). In GLDv2, images of the same class (i.e. those in the same row) have much larger variations in viewpoint and background. Bottom: the performance of instance retrieval on $\mathcal{R}$Oxf (medium protocol), with a number of self-supervised learning methods, including SimCLR, MoCov2, PCL, SwAV and our InsCLR. ResNet-101 pretrained on ImageNet is used for all methods, and then self-supervised learning is performed on GLDv2.

In this work, we aim to explore a more generic setting: can we improve the retrieval quality by using images only, without human annotations or other computer vision systems? Formally, the goal is to close the performance gap between a network trained on a general-purpose dataset (i.e. *ImageNet*) and a network fine-tuned with *full-supervision* on instance-retrieval datasets, by using unlabeled images only. Notably, this work focus on the fine-tuning stage instead of

*These authors contributed equally.
†Corresponding author.

training from scratch. A recent approach described in (Iscen et al. 2018) works with such assumptions. It constructs positive and negative samples using a manifold similarity (Iscen et al. 2017) computed from a similarity graph which is built on Euclidean distance. However, it focuses on designing an offline preprocessing of the images to generate the corresponding image labels, and the training procedure remains the same as the standard supervised training. In this work, we go beyond such offline preprocessing, and provide a more dynamic solution that generates self-supervised learning signals in an online manner during training, with minimal offline computation.

Recent self-supervised learning (SSL) methods, such as MoCo (He et al. 2020) and SimCLR (Chen et al. 2020a), train networks by learning instance discrimination between self-image (or self-augmented image) and other images. However, for the task considered in this work, *we find that simply applying these state-of-the-art SSL methods to learn image-level representations for instance retrieval is far from ideal.* As Figure 1 shows, performing self-supervised learning with SimCLR or MoCov2 (Chen et al. 2020b) with an ImageNet-pretrained network on the recently released Google Landmarks Dataset v2 (Weyand et al. 2020) (without using ground-truth labels) can not obtain the expected performance on the public benchmark for instance retrieval: revisited Oxford (Radenović et al. 2018).

As Figure 1 (top) shows, different from ImageNet, objects in instance retrieval may have large variance in viewpoint, background clutter, occlusion and illumination conditions etc.. Therefore, an important capability for instance retrieval is to learn strong object representations that are robust to the large intra-class variation, and to focus on discriminating object instances rather than images. However, existing image-level SSL methods (such as MoCov2 and SimCLR) can not fully explore the intra-class information which is particularly useful to instance retrieval in datasets like GLDv2.

Consequently, we introduce an *instance-level* SSL method that learns to capture the abovementioned properties from instance-retrieval datasets. The proposed method explores contrastive learning signals explicitly from intra-class pairs by mining cross-image pseudo positives from both the mini-batches and a memory bank along the training. It encourages the model to pull the images of the same class but having different viewpoints or backgrounds closer in the feature space. The mined positives provide much more meaningful learning signals than the self-augmented image pairs, particularly on learning robust intra-class representations. The proposed method is code-named InsCLR. We make the following contributions:

– We identify the limitation of state-of-the-art image-level SSL methods such as MoCov2 and SimCLR, and propose InsCLR for *instance-level SSL* which is able to learn strong instance representations robust to large intra-class variance.

– To build meaningful instance-level contrastive information, we propose new algorithms to dynamically mine pseudo positives from both mini-batches and the memory bank in the contrastive learning framework.

– Extensive experiments across three public benchmarks (revisited Oxford, Paris and INSTRE) demonstrate that the proposed InsCLR surpasses all other self-supervised methods, and even outperforms many recent supervised methods. In particular, as Figure 1 shows, our InsCLR achieves 73.1 mAP on the revisited Oxford (medium), *significantly* closing the gap between the unsupervised fine-tuning and the best-performing supervised counterpart (Weyand et al. 2020) (76.2 mAP) on instance retrieval.

Related Work

Image representations for instance-level image retrieval. In image retrieval, representing images with image-level (i.e. global) features is particularly favoured in practice due to its run-time efficiency. In the era of deep learning, global features can be generated by aggregating CNNs features (Babenko and Lempitsky 2015; Tolias, Sicre, and Jégou 2015; Arandjelovic et al. 2016; Gordo et al. 2016; Radenović, Tolias, and Chum 2016; Tolias, Avrithis, and Jégou 2016; Noh et al. 2017; Radenović, Tolias, and Chum 2018). Apart from global features, local features are also used to perform spatial verification (Philbin et al. 2007; Noh et al. 2017; Cao, Araujo, and Sim 2020), which incorporates the geometric information of objects and results in a more reliable matching. In this work, we focus on learning global image descriptors with self-supervision due to its simplicity, and leave local descriptors for future work.

Self-supervised representation learning. Recently, prominent performance in image classification is achieved by contrastive learning (Oord, Li, and Vinyals 2018; He et al. 2020; Chen et al. 2020a,b; Caron et al. 2020). In particular, MoCo (He et al. 2020; Chen et al. 2020b), SimCLR (Chen et al. 2020a) and SwAV (Caron et al. 2020) further reduce the performance gap between self-supervised networks and fully-supervised networks. Different from MoCo and SimCLR that learn by the image discrimination, SCAN (Van Gansbeke et al. 2020) and InterCLR (Xie et al. 2020) reveal that more important semantic information can be explored across images. Recent works, such as PCL (Li et al. 2020) and SwAV (Caron et al. 2020), were developed to learn intra-class information implicitly by assigning similar images to same prototypes/clusters. In this work, we show that the intra-class information can be better explored by explicitly finding cross-image positive pairs with a pairwise learning paradigm.

Feature memory bank. In contrastive learning framework, memory banks can be used in both supervised (Li et al. 2019; Wang et al. 2020) and unsupervised learning (He et al. 2020; Chen et al. 2020b) with different motivations. Different from them, we propose to mine both positive and negative samples in the memory for unsupervised learning, which has never been explored before.

The Proposed InsCLR

In this section, we present details of the proposed InsCLR which is able to learn strong image representations for instance retrieval by mining pseudo positives and negatives in a self-supervised manner. We start from an overview of the

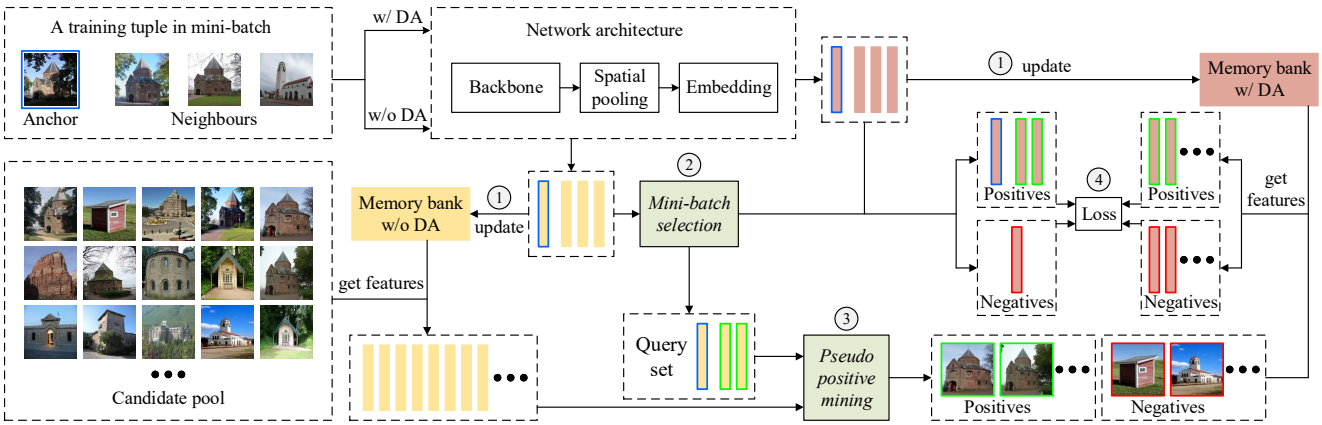

Figure 2: Overview of InsCLR. During training, each image is fed to the network in two forms: with and without random data augmentations (DA). The former (red) contributes to the back-propagation, while the latter (yellow) is used for the robust positive mining. Step 1: the images are encoded by the network, and the output features are first used to update those in the corresponding memory banks. Step 2: positives are selected within the mini-batch. The selected positive features without DA form a query set. Step 3: with the query set, pseudo positives are mined from the memory features corresponding to the candidate pool of the anchor. The features that are not selected as positives become negatives. Step 4: a contrastive loss is computed based on the anchor, the mined positives and negatives (with DA).

method, including a formal definition of the task, the network architecture and the setup for training samples. Then, we describe details of mining positives in mini-batches and the memory bank, and the training loss.

Overview

Problem definition. We follow the line of work that represents images by global features (Babenko and Lempitsky 2015; Tolias, Sicre, and Jégou 2015; Gordo et al. 2016; Cao, Araujo, and Sim 2020) extracted from CNNs. Image retrieval is then performed by computing a cosine similarity between a query image and a set of gallery images in the feature space: $S_{ij} = \cos(f_i, f_j)$, where f denotes the image feature extracted by the CNN. In this case, the retrieval quality entirely depends on image-level representations computed from CNNs. Given an ImageNet-pretrained network and an instance-retrieval-oriented dataset, the objective of this work is to learn strong image representations for instance retrieval by using the dataset in a self-supervised learning (SSL) manner. As discussed previously, the state-of-the-art SSL methods are mainly designed for image classification, and fail to capture the large intra-class invariance (such as viewpoint, background etc.) for instance retrieval. In this work, we propose InsCLR to close this gap by mining informative cross-image positives during training, and manage to match the performance of fully-supervised methods.

The learning framework of InsCLR. As illustrated in Figure 2, InsCLR mainly consists of a network to encode images, memory banks to store image features, and the proposed methods to mine pseudo positives from both mini-batches and the memory bank. For each anchor image, the mined positives as well as the negatives are used to compute a contrastive loss for training the network. We briefly describe the adopted network architecture and our training sample configuration in the following.

Network architecture. To make a fair comparison, we adopt s simple network architecture to produce image-level features. As shown in Figure 2 (top-middle), it consists of three components: a backbone network, a spatial pooling layer and an embedding module.

Memory bank. We leverage memory banks to store a large amount of sample representations during training, which provide more diverse yet meaningful hard samples apart from those in mini-batches. Different from previous image-level SSL methods (Wu et al. 2018; He et al. 2020) that regard all the features in the memory bank as negative samples, we propose to mine pseudo positive samples from the memory bank. Although an additional momentum encoder (He et al. 2020) can be used to alleviate the problem of inconsistency between the features in the memory, we only maintain one encoder (similar to (Wang et al. 2020)) due to its efficiency and simplicity.

Setup for training samples. SCAN (Van Gansbeke et al. 2020) simply collects positives for each mini-batch by computing the nearest neighbours of an anchor image, using a pre-trained model. The nearest neighbours are computed offline, and are fixed during the whole training process. This inspired us to first compute the nearest neighbours for each image using a pre-trained model, which can be used in the subsequent step as prior knowledge for dynamically selecting positives during training. Specifically, we construct a pre-computed candidate pool for each image, which contains the potential positives as well as the hard negatives (with a high similarity). The candidate pool for each image is obtained by computing its P nearest neighbours in the whole training set using only the global features extracted by ImageNet-pretrained networks in an offline manner. A

training tuple is formed by an anchor image with its N_b top-ranked images from its candidate pool. Anchors are randomly selected from the whole training set. A mini-batch is then constructed by multiple such tuples, e.g. 16 training tuples with $N_b = 3$ form a mini-batch of 64.

Positive Selection in Mini-Batches

To identify more informative positives during training, we investigate various approaches to dynamically collect positives from each tuple, which are described as follows.

We consider taking all the N_b images in the tuple as positives, referred as *nn* (similar to (Van Gansbeke et al. 2020)), as a baseline. We then investigate four threshold-based strategies to select the positives from the N_b images. Given a threshold T_b, the four methods are defined as follows.

(1) **Augmented similarity.** We adopt a threshold to select the positives, i.e. computing feature similarities between the anchor and N_b neighboring images in the training tuple, and only considering the images with a similarity over the predefined threshold as pseudo positives: $S_{ij}^{DA} > T_b$. However, this similarity is highly unreliable since it is computed using the images after random augmentations. (2) **Unaugmented similarity.** To overcome this limitation, the second strategy is to feed the original images without any data augmentation, and apply the threshold on their similarities: $S_{ij}^{w/o\,DA} > T_b$. Note that the unaugmented version is only used for similarity computation, and does not contribute to the training loss. (3) **Sample-relative similarity.** A universal threshold may not work reliably to all anchor images. Some classes may have smaller intra-class variance, and thus require larger thresholds. We further develop the third strategy that selects the positives by using sample-relative similarities. Namely, the similarities are scaled by dividing by the largest similarity in each training tuple: $S_{ij}^{rel} > T_b$. (4) **Multi-scale similarity.** Lastly, based on the second strategy, we intend to improve the similarity by feeding the unaugmented images with multiple scales, which is the fourth strategy: $S_{ij}^{ms} > T_b$.

With these mining approaches, we can select pseudo positives dynamically within each mini-batch. For example, a training tuple with $N_b = 7$ may have 3 images selected as positives, based on one of the proposed methods, and the other 4 are then regarded as negatives. We empirically compare the four strategies in ablation study.

Mining Positives from Memory Bank

Benefits. Apart from learning the discrimination between positives and negatives in mini-batches, we also wish to collect more positives from the memory bank. The benefit of finding positives from the memory bank is two-fold. First, mining more positives from the memory bank will encourage the model to pull potential positives closer, instead of pushing them away by default (i.e. considering them as negatives in the memory). This sets our method apart from image-level SSL like MoCo (He et al. 2020) or SimCLR (Chen et al. 2020a). Second, by excluding the selected positives, the rest of the images in the candidate pool are considered as hard negatives, since they often have high offline similarities with the anchor image (comparing to other images in the dataset).

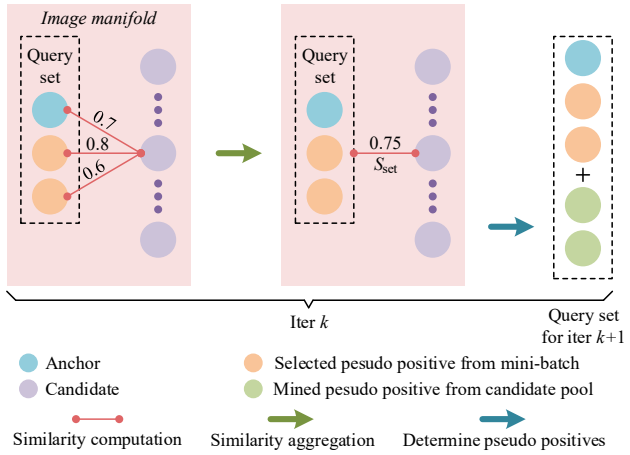

Figure 3: Mining positives in the memory bank. The similarities S_{set} between each image in the candidate pool and the whole query set are computed by pairwise similarity computation followed by aggregation. The positive selection is based on S_{set} and the mined images become part of the new query set.

Mining with query sets. The selected positives within the mini-batch are assumed to be of the same class as the anchor, with a high confidence. Therefore, we can consider the anchor image and its selected positives from the mini-batch as a query set, and then cast the task of mining positives from the memory bank into a retrieval problem with *a set of* query images, instead of a single query image. To this end, we propose a new algorithm that can effectively explore the underlying image relation on-the-fly during training. This procedure is presented in the bottom part of Figure 2. The whole mining process should be performed on the image features extracted without any random augmentation, while the augmented images are used for representation learning. To be specific, our method consists of two steps: similarity computation and aggregation, which are performed on every training tuple in a mini-batch. The method is shown in Figure 3.

Step 1: similarity computation. To make use of every query at hand, we first compute the similarity between the features of each query image and that of memory images. Each image in the pool now has N_p similarity scores, where N_p denotes the size of the query set Q. As an optional step, we can disregard the similarity scores below a threshold. Intuitively, it is possible to have an image which may not look similar to all the images of the same class, e.g. even though images contain the same object, they may not have a low global similarity due to different viewpoint and background clutter etc. Mathematically, we have:

$$S(i, Q) = \phi(S(i, q_1), S(i, q_2), \ldots, S(i, q_{N_p})), \quad (1)$$

where ϕ is the optional discarding step and $q \in Q$.

Step 2: similarity aggregation. For each image in the candidate pool, we measure its similarity to the whole query set by S_{set}. S_{set} is obtained by aggregating its similarities to each image in the query set. In this work, two aggregation functions are considered: average and maximum:

$$S_{set} = \psi(S(i, Q)), \tag{2}$$

where ψ is the aggregation function. We then re-rank the images in the candidate pool based on S_{set}. Given the re-ranked of candidate images, we can determine the pseudo positives either using a threshold T_m or the top-k rule. The mined positives can be added to the query set Q and we can repeat the above two steps for several times to gather more positives if desired. Apart from the mined positives, the rest of the candidate pool is then used as negatives.

Efficient online graph traversal. The proposed method can be considered as an online graph traversal in the image manifold of the query set and the candidate pool. It has two advantages comparing to the standard offline graph traversal methods like (Iscen et al. 2017, 2018; Chang et al. 2019). **First**, it can be applied during training without introducing much computational overhead. **Second**, it is an online approach and the image manifolds can become more reliable along the training, in contrast to (Iscen et al. 2018) in which the image labels are generated in the fixed image manifolds before training.

Loss

We adopt a simple contrastive loss (Hadsell, Chopra, and LeCun 2006). Concretely, the loss function L for a training tuple is:

$$L = \frac{1}{N_p} \sum_{i=1}^{N_p} \left[\sum_{y_i \neq y_j}^{N_{iter}} S_{ij} - \sum_{y_i = y_j}^{N_{iter}} S_{ij} \right], \tag{3}$$

where N_p is the size of query set after memory mining. For a training iteration, N_{iter} is a collection of all images in the mini-batch and the candidate pool for the anchor image. $y_i \neq y_j$ indicates a negative pair while $y_i = y_j$ denotes a positive pair. S_{ij} is the cosine similarity between features.

Experiments and Results

Implementation Details

Network architecture. Following (Weyand et al. 2020), ResNet-101 (He et al. 2016) is used as the backbone network, and only the first four convolutional blocks are kept. The channel size of the backbone output feature map is 2048. For the spatial pooling layer, we adopt the Generalized Mean pooling (Radenović, Tolias, and Chum 2018) (GeM) with the parameter p fixed to be 3. Similar to (Gordo et al. 2017; Cao, Araujo, and Sim 2020), we use a fully-connected layer as the embedding module with an output dimension of 2048, without a careful tuning.

Training details. The training data is a subset of GLDv2 (Ozaki and Yokoo 2019). The dataset contains 1.2M images from 27k landmarks. Unless specified, the size of the offline-computed candidate pool P is set to be 500 for every image, and N_b is set to be 3 for all networks. More details can be found in the supplementary material.

Method	Batch / Mem.	Medium	Hard
		$\mathcal{R}$Oxf / $\mathcal{R}$Par	$\mathcal{R}$Oxf / $\mathcal{R}$Par
ImageNet		23.0 / 52.0	6.5 / 25.9
ImageNet + PCA		40.9 / 65.6	30.5 / 41.7
ImageNet + PCA*		45.0 / 70.7	17.7 / 48.7
Supervised-Arc†		76.2 / 86.8	55.1 / 72.5
A	*nn* / -	57.6 / 68.1	30.3 / 43.7
B	*nn* / neg.	61.0 / 73.4	35.4 / 52.6
C	ours / neg.	65.2 / 75.1	39.9 / 55.7
D	ours / anc.	67.4 / 75.6	42.2 / 56.5
E	ours / ours	73.1 / 77.6	48.3 / 59.5

Table 1: Ablation on pseudo positive mining. All methods use ResNet-101 with GeM pooling. *nn* denotes that taking all the images in the training tuple as positives without selection. *neg.* means that all of the 10^5 features randomly sampled from the memory bank are considered as negatives. * is from (Radenović et al. 2018), and † is from (Weyand et al. 2020) trained with ArcFace loss (Deng et al. 2019).

Ablation Study

We conduct ablation study on two public benchmarks: Oxford and Paris with revisited annotations (Radenović et al. 2018), denoted by $\mathcal{R}$Oxf and $\mathcal{R}$Par, respectively.

Candidate pool size. We empirically find that the performance increases along with P until around 250, after which the performance improves marginally. This is probably because only the hard negatives with similarity higher than 0.4 contribute to the loss, and increasing the size of the pool over 200 brings minimal hard negatives.

Positive selection in mini-batches. As Figure 4 shows, the proposed pseudo positive selection strategies improve the performance over the nearest-neighbour baseline *nn* in general. In particular, applying an absolute threshold on the unaugmented similarity performs better than others, only beaten by its multi-scale variant on $\mathcal{R}$Par. Furthermore, the similarity based on the augmented images is unreliable and harms the retrieval performance. This proves the importance of feeding a version without data augmentation for each image during training. Based on the unaugmented similarity, we experiment on several values of N_b (i.e. 1, 3, 5, 7), and find that $N_b = 3$ is the optimal choice. In the rest of the experiments, a threshold of $T_b = 0.65$ with the unaugmented similarity is adopted, with $N_b = 3$.

Performance gain of each component. We compare our method with several baselines and show the performance gain brought by each proposed component in Table 1. Firstly, the ImageNet-pretrained network can be seen as a lower bound. As reported in (Radenović et al. 2018), applying a PCA/whitening on the features can significantly increase the performance. Notably, the rest of the results in

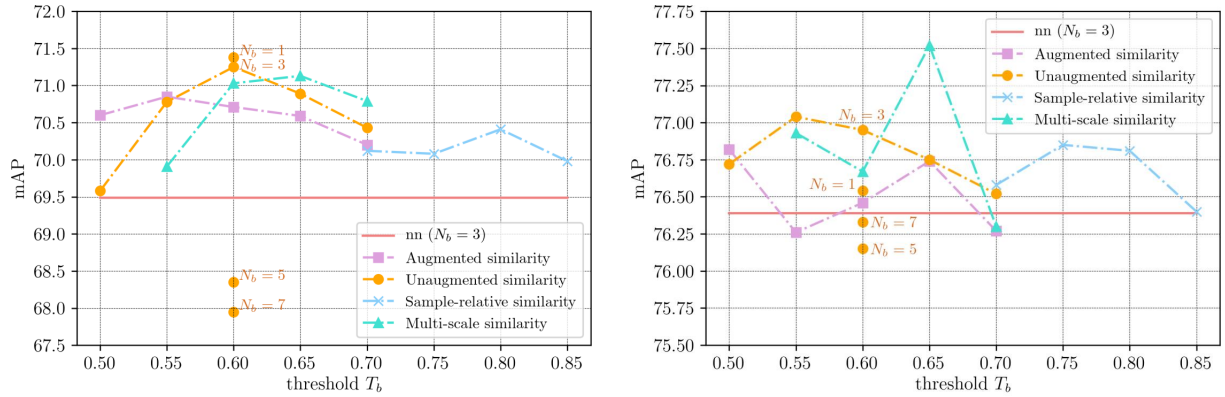

Figure 4: Ablation on the positive selection in mini-batches. The results are obtained on $\mathcal{R}$Oxf (left) and $\mathcal{R}$Par (right) with medium protocol. *nn* considers all the N_b images in each training tuple to be positives, without selection.

Table 1 do not involve PCA post-processing. Training with offline-computed nn positives in the mini-batch (denoted by method *A*) raises the performance to a relatively high level, i.e. above 60 with medium setup. Method *B* is method *A* with a memory bank in which features are only regarded as negatives. This gives notable gains in both $\mathcal{R}$Oxf and $\mathcal{R}$Par. Method *C* replaces the nn with the proposed pseudo positive selection, which improves the mAP by around 4 and 3 points in $\mathcal{R}$Oxf and $\mathcal{R}$Par, respectively. If the proposed memory mining is adopted (denoted by method *E*), the performance is further boosted prominently. However, if we perform the mining in the memory only using the anchor image itself (i.e. without the selected positives in the training tuple), the performance drops significantly (method *D*). This validates the importance of mining with a query set.

Details in memory mining. In Table 2, we compare the different options in the design of memory mining. For the similarity aggregation methods, *mean* and *max* perform better in $\mathcal{R}$Oxf and $\mathcal{R}$Par, respectively. As the selection strategy, *topk* performs better than *threshold*. This is in contrast to the case in mini-batches, where *threshold* is better. It is because the potential positives from mini-batches are the *top-ranked* images from the candidate pool, which are very likely to be easy positives. Hence, the taking thresholds on the similarity is relatively reliable. Whereas in the memory bank, mining is more difficult and similarities become less reliable. In terms of sparsity, the performance seems not sensitive. This is probably because the size of the query set is relatively small. The last conclusion to draw from Table 2 is that the mAP increases with the mining iterations, while bringing negligible computational overhead. In the rest of the experiments, *avg* and *topk* are adopted with 4 iterations in the mining.

Mining accuracy at training. We analyze the precision of positive mining along the training, which is defined as the ratio between the true positives mined and the total number of mined pseudo positives. We find that the precision of mining in both mini-batches and the memory bank increases gradually along the training. The plots of the precision are displayed in the supplementary material.

Aggre.	Selection	Medium	Hard
		$\mathcal{R}$Oxf / $\mathcal{R}$Par	$\mathcal{R}$Oxf / $\mathcal{R}$Par
avg	topk	71.3 / 77.0	45.5 / 58.3
avg	topk w/ spa.	70.4 / 77.1	44.8 / 58.8
avg	threshold	68.4 / 76.3	42.0 / 57.4
max	topk	70.6 / 77.2	44.5 / 58.7
max	topk w/ spa.	70.4 / 77.2	44.0 / 58.5
max	threshold	61.1 / 71.5	34.9 / 47.6
avg	topk 1×	69.5 / 76.2	42.8 / 57.2
avg	topk 4×	73.1 / 77.6	48.3 / 59.5

Table 2: Ablation on mining positives in the memory bank. *topk* denotes taking the top $k = 5$ images in the ranked candidate pool. T_m = 0.6 for *threshold*. *spa.* refers to retaining only the similarity scores above 0.6 to enforce the sparsity. *1*× or *4*× means that the mining is performed for one or four iterations (otherwise two iterations).

Comparison with Other Methods

We compare InsCLR with the state-of-the-art methods in Table 3, including the large-scale retrieval results by adding $\mathcal{R}$1M. Following the convention in image retrieval and for a fair comparison, all the self-supervised methods (including SimCLR, MoCov2, SwAV etc.) start with ImageNet-pretrained networks and are used to fine-tune on GLDv2.

Supervised methods. Table 3 shows that InsCLR achieves outstanding mAP on $\mathcal{R}$Oxf and $\mathcal{R}$Par for both medium and hard setup. In particular, the mAP of InsCLR on $\mathcal{R}$Oxf medium is on par with the state-of-the-art supervised methods. Namely, only *R101-GeM (GLDv2-clean)* performs better than InsCLR on both $\mathcal{R}$Oxf and $\mathcal{R}$Par, when spatial verification (SP) is not considered. In particular, with the same architecture, InsCLR enhances the mAP on $\mathcal{R}$Oxf medium/hard from 45.0/17.7 (ImageNet pretrained) to 73.1/48.3, comparing to 76.2/55.1 attained with full supervision. Even with SP, *DELF-R-ASMK+SP (GLD)* only performs better than InsCLR when $\mathcal{R}$1M is added. However, SP is much slower at run-time and memory-consuming.

Method	Medium		Hard	
	$\mathcal{R}$Oxf / $\mathcal{R}$1M	$\mathcal{R}$Par / $\mathcal{R}$1M	$\mathcal{R}$Oxf / $\mathcal{R}$1M	$\mathcal{R}$Par / $\mathcal{R}$1M
Supervised training				
R101 - GeM (ImageNet) (Radenović et al. 2018)	45.0 / 25.6	70.7 / 46.2	17.7 / 4.7	48.7 / 20.3
R101 - R-MAC (Gordo et al. 2017)	60.9 / 39.3	78.9 / 54.8	32.4 / 12.5	59.4 / 28.0
R101 - GeM - AP (Revaud et al. 2019)	67.5 / 47.5	80.1 / 52.5	42.8 / 23.2	60.5 / 25.1
R101 - GeM - AP (GLD) (Revaud et al. 2019)	66.3 / –	80.2 / –	42.5 / –	60.8 / –
R101 - DELG (Cao, Araujo, and Sim 2020)	73.2 / 54.8	82.4 / 61.8	51.2 / 30.3	64.7 / 35.5
R101 - GeM (GLDv2-clean) (Weyand et al. 2020)	76.2 / -	86.8 / -	55.1 / -	72.5 / -
DELF - ASMK + SP (Radenović et al. 2018)	67.8 / 53.8	76.9 / 57.3	43.1 / 31.2	55.4 / 26.4
DELF - R-ASMK + SP (GLD) (Teichmann et al. 2019)	76.0 / 64.0	80.2 / 59.7	52.4 / 38.1	58.6 / 29.4
Self-supervised training based on automatic annotation				
VGG16 - MAC (Radenović, Tolias, and Chum 2016)	58.4 / 39.1	66.8 / 42.4	30.5 / 17.9	42.0 / 17.7
R101 - GeM (GLD) (Radenović, Tolias, and Chum 2018)	64.7 / 45.2	77.2 / 52.3	38.5 / 19.9	56.3 / 24.7
R101 - GeM (Siméoni, Avrithis, and Chum 2019)	65.3 / 46.1	77.3 / 52.6	39.6 / 22.2	56.6 / 24.8
R101 - GeM + DSM (Siméoni, Avrithis, and Chum 2019)	65.3 / 47.6	77.4 / 52.8	39.2 / 23.2	56.2 / 25.0
Self-supervised training				
DeepCluster (Caron et al. 2018)	29.8 / 9.8	49.1 / 13.3	9.0 / 0.9	26.0 / 3.2
SimCLR (Chen et al. 2020a)	22.2 / 9.4	50.5 / 14.4	6.3 / 2.2	19.6 / 1.1
MoCov2 (Chen et al. 2020b)	27.3 / 11.0	65.1 / 17.4	6.1 / 0.8	38.4 / 3.2
BYOL (Grill et al. 2020)	11.0 / 1.9	28.4 / 3.7	2.3 / 0.1	8.8 / 0.2
PCL (Li et al. 2020)	29.2 / 10.3	59.3 / 17.6	7.9 / 0.5	28.9 / 2.6
SwAV (Caron et al. 2020)	19.9 / 7.1	38.5 / 10.4	3.7 / 0.1	10.5 / 0.4
InsCLR (ours)	73.1 / 56.2	77.6 / 56.7	48.3 / 29.6	59.5 / 29.2

Table 3: Comparison to state-of-the-art methods on large-scale retrieval. *Automatic annotation* means additional computer vision systems are used to annotate the images before training. *SP* refers to the spatial verification using local features. Results of all the unsupervised methods are obtained using their official code with careful hyper-parameter tuning, with the same network architecture as InsCLR. Note that all the methods in this table are built on ImageNet-pretrained networks.

Method	Labels	Val set	Test set
(Weyand et al. 2020)	Yes	23.30	25.57
ImageNet pretrained	No	0.89	0.52
InsCLR	No	13.39	13.71

Table 4: Retrieval task on GLDv2 (% mAP@100).

Method	Labels	INSTRE
ImageNet (w/o PCA)	-	32.7
(Iscen et al. 2018)	No	57.7
InsCLR w/o P.M. (*nn*=1)	No	55.6
InsCLR	No	76.2

Table 5: Evaluation on INSTRE. † denotes the result of method (Gordo et al. 2017) implemented by (Iscen et al. 2017). *P.M.* denotes the positive mining in InsCLR.

Self-supervised methods. In the self-supervised regime, InsCLR surpasses all the self-supervised methods. We attempted to train SCAN on our task but it failed to converge in the clustering step. Lastly, although clustering-based methods like DeepCluster, PCL and SwAV implicitly take into account the intra-class variation into the representation learning, they hardly bring improvements. It shows that learning intra-class invariance explicitly from positive and negative pairs is superior for the task at hand.

Evaluation on More Benchmarks

GLDv2 retrieval task. We directly evaluate the trained InsCLR model on the GLDv2 retrieval task. As shown in Table 4, InsCLR can achieve performance 13.39% and 13.71% on validation and test set, respectively. This is a surprisingly large improvement comparing to the ImageNet-pretrained baseline, given that no labels are used.

INSTRE benchmark. To showcase the generalization of InsCLR, we fine-tune an ImageNet-pretrained ResNet-50 with GeM ($p = 3$) on another instance retrieval benchmark: INSTRE (Wang and Jiang 2015). As shown by Table 5, InsCLR significantly outperforms (Iscen et al. 2018). Moreover, the proposed positive mining within mini-batches and the memory again boosts the performance by a large margin (i.e. 55.6 to 76.2).

Conclusion

We present a new SSL method built on the instance-level constrastive learning for instance retrieval. This sets it apart from existing SSL methods that commonly learn from image-level contrast. InsCLR can learn intra-class invariance by mining informative positives from both mini-batches and memory bank during training. Extensive experiments demonstrate that InsCLR can achieve comparable performance to supervised methods on instance retrieval.

References

Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5297–5307.

Babenko, A.; and Lempitsky, V. 2015. Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*.

Cao, B.; Araujo, A.; and Sim, J. 2020. Unifying Deep Local and Global Features for Image Search. *arXiv*, arXiv–2001.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 132–149.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*.

Chang, C.; Yu, G.; Liu, C.; and Volkovs, M. 2019. Explore-exploit graph traversal for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9423–9431.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4690–4699.

Gordo, A.; Almazán, J.; Revaud, J.; and Larlus, D. 2016. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, 241–257. Springer.

Gordo, A.; Almazan, J.; Revaud, J.; and Larlus, D. 2017. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2): 237–254.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap Your Own Latent-A New Approach to Self-Supervised Learning. *Advances in Neural Information Processing Systems*, 33.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, 1735–1742. IEEE.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2018. Mining on manifolds: Metric learning without labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7642–7651.

Iscen, A.; Tolias, G.; Avrithis, Y.; Furon, T.; and Chum, O. 2017. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2077–2086.

Li, J.; Zhou, P.; Xiong, C.; Socher, R.; and Hoi, S. C. 2020. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*.

Li, S.; Chen, D.; Liu, B.; Yu, N.; and Zhao, R. 2019. Memory-based neighbourhood embedding for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 6102–6111.

Noh, H.; Araujo, A.; Sim, J.; Weyand, T.; and Han, B. 2017. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, 3456–3465.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ozaki, K.; and Yokoo, S. 2019. Large-scale landmark retrieval/recognition under a noisy and diverse dataset. *arXiv preprint arXiv:1906.04087*.

Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; and Zisserman, A. 2007. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Radenović, F.; Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2018. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5706–5715.

Radenović, F.; Tolias, G.; and Chum, O. 2016. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, 3–20. Springer.

Radenović, F.; Tolias, G.; and Chum, O. 2018. Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1655–1668.

Revaud, J.; Almazán, J.; Rezende, R. S.; and Souza, C. R. d. 2019. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE International Conference on Computer Vision*, 5107–5116.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4104–4113.

Siméoni, O.; Avrithis, Y.; and Chum, O. 2019. Local features and visual words emerge in activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11651–11660.

Teichmann, M.; Araujo, A.; Zhu, M.; and Sim, J. 2019. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5109–5118.

Tolias, G.; Avrithis, Y.; and Jégou, H. 2016. Image search with selective match kernels: aggregation across single and multiple images. *International Journal of Computer Vision*, 116(3): 247–261.

Tolias, G.; Sicre, R.; and Jégou, H. 2015. Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879*.

Van Gansbeke, W.; Vandenhende, S.; Georgoulis, S.; Proesmans, M.; and Van Gool, L. 2020. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, 268–285. Springer.

Wang, S.; and Jiang, S. 2015. INSTRE: a new benchmark for instance-level object retrieval and recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(3): 1–21.

Wang, X.; Zhang, H.; Huang, W.; and Scott, M. R. 2020. Cross-Batch Memory for Embedding Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6388–6397.

Weyand, T.; Araujo, A.; Cao, B.; and Sim, J. 2020. Google Landmarks Dataset v2-A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2575–2584.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3733–3742.

Xie, J.; Zhan, X.; Liu, Z.; Ong, Y. S.; and Loy, C. C. 2020. Delving into Inter-Image Invariance for Unsupervised Visual Representations. *arXiv preprint arXiv:2008.11702*.