Supervised Pretraining or Self-supervised Pretraining? A Tale of Two Transfer Learning Paradigms

Anonymous CVPR 2021 submission

Paper ID 4847

Abstract

Transfer learning (TL) has become a standard technique in computer vision and machine learning, which usually helps to improve performance substantially. Previously, the most dominant TL method is supervised pretraining (SP), which uses labeled data to learn a good representation network. Recently, a new pretraining approach – self-supervised pretraining (SSP) – has demonstrated promising results on a wide range of applications. SSP does not require annotated labels. It is purely conducted on input data by solving auxiliary tasks defined on the input data examples. The current reported results show that in certain applications, SSP outperforms SP and the other way around in other applications. There has not been a clear understanding on what properties of data render one approach outperforms the other. Without an informed guideline, ML researchers have to try both methods to find out which one is better empirically. It is usually time-consuming to do so. In this work, we aim to address this problem. We perform a comprehensive comparative study between SP and SSP regarding which one works better under different properties of data, including domain similarity between source and target datasets, the amount of pretraining data, and usage of target data for additional pretraining, etc. Our comparative studies distill a set of novel insights, such as SSP is generally more robust to differences between the source and target domain than SP, and is more robust to the amount of source data than SP. These insights can help ML researchers decide which method to use based on the properties of their applications.

1. Introduction

Pretraining is a commonly used technique in deep learning to learn more effective representations for alleviating overfitting. Given a target task where the amount of training data is limited, training deep neural networks on this small-sized dataset has high risk of overfitting. To address this problem, one can pretrain the feature extraction layers in the network on large-sized external data from some source domain, then finetune these layers on the target data. The abundance of source data enables the network to learn powerful representations that are robust to overfitting. And such representation power can be leveraged to assist in the learning of the target task with more resilience to overfitting.

Arguably, the most popular pretraining approach is supervised pretraining (SP) [28], which learns the weight parameters of a representation network by solving a supervised source task, i.e., correctly mapping input data examples to their labels (e.g., classes, segmentation masks, etc.). While successful, one concern of SP is that it uses labels in the source dataset to learn network weights, which may be biased to the source labels and generalize less well on the target dataset where the classes in the target labels are different from those in the source dataset.

This problem can be potentially alleviated by unsupervised pretraining, which trains the network weights purely based on input data examples without using any labels in the source dataset. Recently, self-supervised learning [27, 39, 1, 12, 16, 35, 43, 3, 23], as an unsupervised pretraining approach, has achieved promising success and outperforms supervised pretraining in a wide range of applications. Similar to SP, self-supervised pretraining (SSP) also solves predictive tasks. But the output labels in SSP are constructed from the input data, rather than annotated by human as in SP. The auxiliary predictive tasks in SSP could be predicting whether two augmented data examples originate from the same original data example [16, 12, 3, 23], inpainting masked regions in images [29], etc. Since SSP does not leverage labels provided by human, it does not have the risk of being biased to labels in a source dataset. On the other hand, the potential pitfall of not using human-annotated labels is that the learned representations by SSP may not be as discriminative as those in SP. In sum, conceptually, SSP and SP both have advantages and disadvantages. It is difficult to judge whether one is better than the other. The existing empirical results show that in certain tasks, SSP outperforms SP [12, 16, 23]; in other tasks, SP performs better than SSP [6]. But these studies did not provide a clear

guidance on what properties of data render one approach works better than the other. Consequently, ML researchers are not informed about how to select the right pretraining method and they have to try both empirically to find out, which is time-consuming and resource-intensive.

To address this issue, we perform comprehensive studies to compare SSP and SP, and investigate which method works better under different properties of data, including domain similarity between source and target datasets, the amount of pretraining data, and the usage of target data for additional pretraining. The studies are performed on 3 source datasets and 3 target datasets from various domains including daily-life objects, general scenes, natural plants, and medical imaging. We summarize the insights distilled from the comparative studies to help ML researchers decide which pretraining method to use based on the specific properties of their applications. The major insights obtained from the studies include:

- When the domain similarity between the source dataset and the target dataset is small, SSP outperforms SP. When domain similarity is large, SP outperforms SSP.
- In the same source domain, when the amount of pretraining data is small, SSP outperforms SP. When the amount is large, SP outperforms SSP.
- SSP is less sensitive to domain difference than SP: on the same target dataset, the performance of SSP pretrained on source datasets that have varying domain difference with the target dataset is relatively stable while that of SP varies a lot.
- When domain similarity is large, SSP is less sensitive to the amount of pretraining data than SP: given a target dataset and a source dataset that have large domain similarity, the performance of SP on the target dataset changes a lot under varying amount of pretraining source data while SSP is relatively stable.
- Combining the target data with source data for pretraining is beneficial for SSP, but not for SP: using target data for additional pretraining in SSP yields better performance, which is not the case in SP.

2. Related Works

2.1. Self-supervised Learning

Self-supervised learning has been widely applied to other application domains, such as computer vision, where various strategies have been proposed to construct auxiliary tasks, based on temporal correspondence [22, 37], cross-modal consistency [36], rotation prediction [8, 34], image inpainting [30], automatic colorization [42], context prediction [24], etc. Some recent works studied self-supervised representation learning based on instance discrimination [39] with contrastive learning. Oord et al. [27] proposed contrastive predictive coding (CPC), which predicts the future in latent space by using powerful autoregressive models, to extract useful representations from high-dimensional data. Bachman et al. [1] proposed a self-supervised representation learning approach based on maximizing mutual information between features extracted from multiple views of a shared context. MoCo [13] and SimCLR [3] learn image encoders by predicting whether two augmented images are created from the same original image. Srinivas [33] proposed to learn contrastive unsupervised representations for reinforcement learning. Khosla et al. [18] investigated supervised contrastive learning, where clusters of points belonging to the same class were pulled together in the embedding space, while clusters of samples from different classes were pushed apart. Klein et al. [20] proposed a contrastive self-supervised learning approach for commonsense reasoning. He et al. [15] proposed an Self-Trans approach which applies contrastive self-supervised learning on top of networks pretrained by transfer learning.

2.2. Comparison of Self-supervised Pretraining and Supervised Pretraining

In [39, 12, 16, 35, 43, 23], supervised pretraining (SP) is compared with self-supervised pretraining (SSP) approaches. It is shown that in certain tasks SSP outperforms SP, but the other way around in other tasks. It is not studied what factors incur such inconsistent results, which renders ML researchers have to try both approaches to find out which one works better empirically without an informed guidance. We aim to bridge this gap in our work, by systematically investigating under what situations SSP performs better than SP and vice versa. Zhai et al. [41] compared a number of representation learning methods including SP and SSP on a single factor: image style. In our work, we make the comparison on several factors, including domain similarity, data amount, usage of target data, etc. Some works [9, 21] performed a comparison of self-supervised learning methods. But they did not compare SSP with SP. Newell and Deng [25] studied what factors affect the performance of SSP. Resnick et al. [32] presented several findings in the study of SSP, such as linear probes are insufficient to evaluate the representations learned by SSP, the bias towards success stemming from the architecture is high, etc. Our work differs from these works in that our goal is to study what factors affect the comparative advantages between SP and SSP.

3. Supervised Pretraining and Self-supervised Pretraining

Figure 1 illustrates supervised pretraining (SP) and self-supervised pretraining (SSP). Both SP and SSP consist of two phrases: pretraining on a source task and finetuning on the target task. The purpose of pretraining is to train

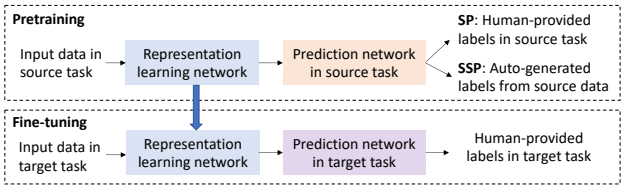

Figure 1. Workflow of supervised pretraining (SP) and self-supervised pretraining (SSP).

the weight parameters of a representation network into a good state. The pretrained weights are used to initialize the representation network in the target task (which has the same architecture as that in the source task). The initialized representation network in the target task is further trained on the target data. The difference between SP and SSP is that: SP performs pretraining using labeled source data in a supervised way while SSP performs pretraining using unlabeled source data in an unsupervised way.

Arguably, supervised pretraining [28] is the most widely used method for pretraining. Given data examples $\mathcal{D}_T$ and their class labels $\mathcal{L}_T$ in a target task T, a neural network $\mathcal{N}_T$ is designed to fulfill this task. $\mathcal{N}_T$ is composed of two parts: a feature extraction sub-network $\mathcal{F}_T$ that learns latent feature representations of $\mathcal{D}_T$ and a prediction sub-network $\mathcal{P}_T$ which is tailored to the predictive task in T. Typically, the majority of weight parameters of $\mathcal{N}_T$ lie in the feature extraction part $\mathcal{F}_T$. Meanwhile, one has access to data examples $\mathcal{D}_S$ and their class labels $\mathcal{L}_S$ in a source task S, where $\mathcal{D}_S$ is in general much larger than $\mathcal{D}_T$. How supervised pretraining (SP) works is: it creates a network $\mathcal{N}_S$ to solve the predictive task in S where the feature extraction sub-network $\mathcal{F}_S$ has the same architecture as $\mathcal{F}_T$, but the prediction sub-network $\mathcal{P}_S$ is tailored to task S. SP trains $\mathcal{N}_S$ on $\mathcal{D}_S$ and $\mathcal{L}_S$ until convergence. Then it uses the weights of $\mathcal{F}_S$ to initialize $\mathcal{F}_T$ and trains $\mathcal{N}_T$ on $\mathcal{D}_T$ and $\mathcal{L}_T$ until convergence. While SP has demonstrated pervasive effectiveness, there is a major concern about this approach. The weights in $\mathcal{F}_S$ are trained by fitting the labels $\mathcal{L}_S$. Therefore they are naturally biased to the classes in $\mathcal{L}_S$. When applied to predicting a new set of classes in $\mathcal{L}_T$, this bias may render the prediction less accurate.

Recently, in parallel to supervised pretraining, another pretraining paradigm – self-supervised pretraining (SSP) [27, 39, 1, 12, 16, 35, 43, 3, 23] – has arisen much research interest. Different from SP which is typically conducted in a supervised manner (by labels in source tasks), SSP is undertaken mostly in an unsupervised way without using any human-provided labels. The basic idea of SSP is to construct some auxiliary tasks solely based on the input data itself without using any human-offered annotations and encourage the network to learn meaningful representations by performing the auxiliary tasks well. Given the source data $\mathcal{D}_S$, SSP designs an auxiliary task A and a network $\mathcal{N}_A$ to perform this task. $\mathcal{N}_A$ has a feature extraction sub-network $\mathcal{F}_A$ that has the same architecture as $\mathcal{F}_T$ and a prediction sub-network that is tailored to the task A. SSP trains $\mathcal{N}_A$ on $\mathcal{D}_S$ until convergence. Then it uses the weights of $\mathcal{F}_A$ to initialize $\mathcal{F}_T$, then trains $\mathcal{N}_T$ on $\mathcal{D}_T$ and $\mathcal{L}_T$ until convergence. Since SSP does not leverage labels in the source task, the learned feature extraction sub-network is not biased to these labels and is presumably more generalizable on $\mathcal{D}_T$. However, a potential downside is: $\mathcal{F}_A$ is learned without human supervision, which renders the network less discriminative.

We choose three representative SSP approaches for our studies: MoCo [12], SwAV [2], and Rotation [8]. MoCo and SwAV are based on contrastive learning [11] and Rotation is not. The basic idea of contrastive SSP is: generate augmented examples of original data examples, create a predictive task where the goal is to predict whether two augmented examples are from the same original data example, and learn the representation network by solving this task. MoCo [12] performs contrastive learning by using a queue proposed in [39] that contains a dynamic set of augmented data examples (called keys). The keys are encoded using a momentum encoder. Given a query augmentation, a contrastive loss is defined on the query and keys based on whether they originate from the same data example. SwAV [2] performs contrastive SSP without requiring to compute pairwise comparisons. In SwAV, clustering is performed on the augmentations of data examples. The cluster assignments for different augmentations of the same image are encouraged to be consistent. Specifically, the code of one augmentation is predicted based on the representation of another augmentation. This method is more memory efficient since it does not require a large memory bank. Rotation [8] is a non-contrastive SSP approach which learns image representations by recognizing the 2D rotation that is applied to the image.

4. Design of Studies

In this section, we study several factors that may affect the comparative advantages between SSP and SP, including domain similarity between source dataset and target dataset, amount of pretraining data, and usage of target data for additional pretraining.

Study of domain similarity Given a source dataset (from domain X) where SSP or SP pretraining is performed and a target dataset (from domain Y) where finetuning is performed, we are interested in how the domain similarity between X and Y affects the comparative advantage of SSP and SP. We use three source domains and three target domains, from the following areas: (1) objects in daily life, (2) scenes in daily life, (3) nature, and (4) medical imaging. We use three source datasets: ImageNet, SUN, and ChestX-ray8,

	Caltech-256	Flowers-102	Pneumonia
Visual distance			
ImageNet	1.35	1.73	1.89
SUN	1.09	1.67	1.84
ChestX-ray8	1.99	1.83	1.54
Class similarity			
ImageNet	0.07	0.06	0.01
SUN	0.07	0.04	0.02
ChestX-ray8	0.01	0.01	0.25
ImageNet	+/+	–/+	–/–
SUN	+/+	–/–	–/–
ChestX-ray8	–/–	–/–	+/+

Table 1. Source-target domain similarity. If the visual distance is less than 1.6, a source and a target are considered as in the same domain (denoted by "+"); otherwise, in different domains (denoted by "–"). For class similarity, "+" if the similarity is greater than 0.06, and "–" if otherwise.

from objects, scenes, and medical domains respectively. The details of these three datasets are:

- The ImageNet [5] dataset contains 1,281,167 training and 50,000 validation images from 1,000 classes. These classes cover common objects in daily life, such as monitor, school bus, sleeping bag, teapot, broccoli, etc.
- The SUN-397 [40] dataset contains 130,519 images from 397 scene categories, such as abbey, balcony, cafeteria, etc. Each category has more than 100 images.
- ChestX-ray8 [38] is a widely used medical dataset which contains 112,120 frontal-view X-ray images of 30,805 patients from 15 disease classes, such as emphysema, pneumonia, fibrosis, infiltration, cardiomegaly, etc.

We use three target datasets: Caltech-256, Flowers-102, and Pneumonia from objects, nature, and medical domains respectively. The detailed information of these datasets is as follows.

- The Caltech-256 [10] dataset contains 256 categories of objects in daily life, such as instruments, furniture, animals, food, vehicles, etc. The number of training, validation, and testing images is 7710, 6425, and 6425 respectively.
- The Flowers-102 [26] dataset contains 102 types of flowers. The number of training, validation, and testing images is 6149, 1020, and 1020 respectively.
- Pneumonia [17] is a widely used medical dataset which contains 5,863 X-ray images which are either positive for pneumonia or negative. The number of training, validation, and testing images are 5216, 16, and 624 respectively.

We quantitatively measure the domain similarity between source images and target images based on their visual contents and class labels. To measure visual distance between domains, we use the method in [7]. We sample 1000 images from a source dataset and label them as 0, and sample 1000 images from a target dataset and label them as 1. Then we split the 2000 images into a training set and a test set. We train a classifier on the training set to distinguish whether an image is from the source or target, then measure the classification error ϵ on the test set. The visual difference is defined as $d = 2(1 - 2\epsilon)$. Intuitively, if source images and target images are easy to be told apart (i.e., ϵ is small), then their visual difference is large. In addition to visual difference, we also measure domain similarity in the label space. Given $\{(c_i, f_i)\}_{i=1}^m$ in the source dataset, where c_i is a class name, f_i is the frequency of this class in this dataset, and m is the number of classes in the source dataset, and similarly $\{(c_j, f_j)\}_{j=1}^n$ in the target dataset, we calculate the class similarity of these two domains using this equation: $(\sum_{i=1}^m \sum_{j=1}^n f_i f_j \sigma(c_i, c_j))/(\sum_{i=1}^m f_i)/(\sum_{j=1}^n f_j)$, where $\sigma(c_i, c_j)$ is the cosine similarity between the GloVe [31] embeddings of c_i and c_j. This equation basically measures the average similarity between source labels and target labels.

Table 4 shows the visual difference and class similarity between source and target datasets. If the visual distance is less than 1.6, a source and a target are considered as in the same domain (denoted by "+"); otherwise, in different domains (denoted by "–"). For class similarity, "+" if the similarity is greater than 0.06, and "–" if otherwise. The third panel in Table 4 shows the results, where the red and blue symbol in each cell correspond to visual similarity and class similarity respectively. A source and a target with "+/+" have high domain similarity, and those with "–/–" have low domain similarity. The results are in accordance with intuitive judgement. For example, ImageNet and Caltech-256 are in the same domain about daily objects; ChestX-ray8 and Pneumonia are in the same domain about medical imaging.

Study of the amount of pretraining data To study how SSP and SP are affected by the amount of pretraining data, we perform the following controlled study: for each source dataset, we use 1%, 10%, and 100% of the dataset for pretraining.

Study of using target data for additional pretraining In existing approaches, pretraining is mostly conducted on source data. We are interested in investigating whether it is helpful to add the training examples in the target task as additional data for pretraining and how this will affect the comparative advantages of SSP and SP. We compare the following pretraining settings: (1) SSP on source data only, on target data only, and on the combined data of source and target; (2) SP on source data only, on target data only, and on the combined data of source and target. For SSP on target data only, we perform self-supervised learning on the training images in the target task. For SSP on the combined data

of source and target, we perform self-supervised learning on source images and target training images. For SP on target data only, it is the same as finetuning on the target dataset with random initialization of the network. For SP on the combined data, it is a multi-task learning problem which trains classification models for the source task and the target task simultaneously. Given the models pretrained using the above-described strategies, we finetune these models on the target dataset.

5. Experiments

5.1. Experimental settings

All source tasks and target tasks are about image classification. We used ResNet-50 [14] and ResNet-18 for classification. The architecture of ResNet-50 and ResNet-18 is the same for different tasks, except the number of output units (which is the same as the number of categories in a task). The SP and the MoCo model using ResNet-50 pretrained on the full ImageNet dataset were borrowed from [14] and [12]. The rest of the models were pretrained from scratch.

The hyperparameter settings and choices of data augmentation methods mostly follow the best practice proposed in previous works. For SP on source tasks, the applied data augmentation includes random resize with a ratio sampled from $[0.08, 1]$, random crop of 224×224, and the ImageNet AutoAugment policy. The weights were optimized using SGD with an initial learning rate of 0.1 and a momentum of 0.9. For the ResNet-50 backbone, SP was performed for 200 epochs on every source dataset with a minibatch size of 128. For the ResNet-18 backbone, SP was performed for 100 epochs. The learning rate was reduced by 0.1 every 50 epochs.

For SSP, we experimented MoCo [13], SwAV [2], and Rotation [8]. For MoCo on source datasets, we used the data augmentation methods in [4] and followed their hyperparameter settings, where SGD was used as the optimizer with an initial learning rate of 0.015 and a momentum of 0.9. For the ResNet-50 backbone, MoCo was pretrained for 200 epochs with a batch size of 128. For the ResNet-18 backbone, MoCo was pretrained for 120 epochs. The learning rate was adjusted using cosine learning rate scheduling. For SwAV on source datasets, we used the data augmentation methods in [2] and used SGD as the optimizer with an initial learning rate of 0.6 and a momentum of 0.9. We pretrained SwAV for 100 epochs with a batch size of 32. The learning rate was adjusted using cosine learning rate scheduling. For Rotation on source datasets, the data augmentation methods were the same as those in SP. The networks were pretrained based on the rotation prediction task, where each image was rotated by 4 different angles including $0°$, $90°$, $180°$, and $270°$. SGD was applied as the optimizer with an initial learning rate of 0.1, a mini-batch size of 32, an epoch number of 100, and a momentum of 0.9.

For finetuning on target datasets, we utilized the same data augmentation methods as in supervised pretraining. Adam [19] was used as the optimizer with an initial learning rate of 0.001. The finetuning was performed for 200 epochs on Caltech-256 and Flower-101, and 30 epochs on Pneumonia. Cosine scheduling with a period of 6 was used to adjust the learning rate. The finetuning of each model was repeated three times with different random initializations and the results were averaged on the three runs. Implementation was based on PyTorch and the experiments were conducted on 8 Tesla V100 GPUs.

6. Results

6.1. Results on domain similarity

We perform controlled studies to investigate how domain similarity between source and target datasets affects the performance of SSP and SP. To rule out the influence of the amount of pretraining data, in the study of this section we make the number of pretraining examples in all source datasets the same. This number is chosen to be 90,000. For a source dataset whose size is larger than 90,000, we randomly sample 90,000 data examples from this dataset. Studies on the full source datasets (without sub-sampling) are performed in the next two sections.

Figure 2 and Figure 3 show the top-1 image classification accuracy (%) in the study of domain similarity, using ResNet-50 and ResNet-18 as backbone respectively. Different sub-figures are for different target datasets. On the x-axis are source datasets which have decreasing domain similarity with the target dataset from left to right. "+" and "-" denote the domain similarity (red for visual similarity and blue for class similarity) between source and target datasets. During finetuning, each model is trained three times with different random initialization. The result is averaged on the three runs. From these figures, we make the following observations. **First**, on most source-target dataset pairs that have strong domain similarity ("+/+"), including (ImageNet, Caltech-256), (SUN, Caltech-256), (ChestX-ray8, Pneumonia), SP achieves better accuracy than SSP methods (including MoCo, SwAV, and Rotation). This shows that when the domain similarity is large, SP is more effective than SSP. The possible reason is: when the domain similarity is large, the classes in source labels have a lot of overlap with those in target labels. Due to this overlap, it is beneficial to transfer the semantic information in source labels to the target task. SP utilizes sources labels while SSP does not. Hence SP works better than SSP. **Second**, on source-target dataset pairs where the domain similarity is small ("–/–"), including (ChestX-ray8, Caltech-256), (SUN, Flowers-102), (ChestX-ray8, Flowers-102), (ImageNet, Pneumonia), and (SUN, Pneumonia), in most cases SSP methods achieve

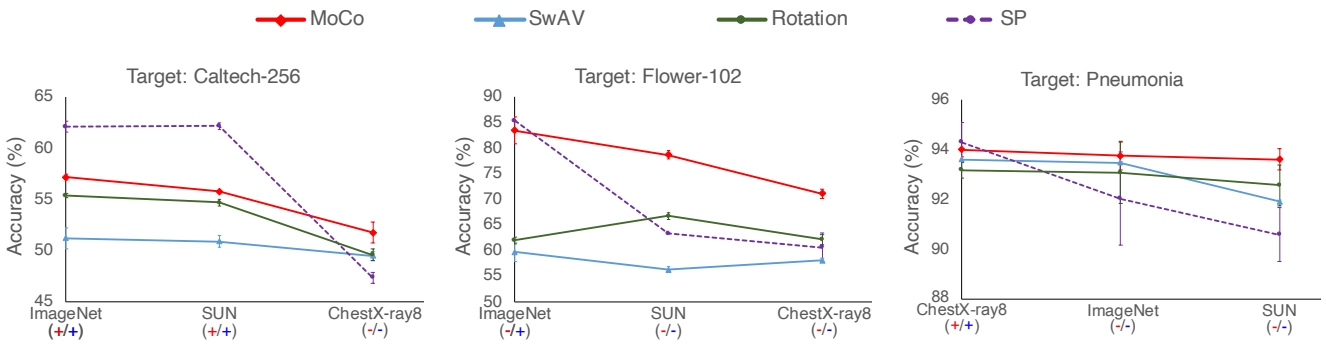

Figure 2. Top-1 image classification accuracy (%) in the study of domain similarity, using ResNet-50 as backbone. Different sub-figures are for different target datasets. On the x-axis are source datasets, which have decreasing domain similarity with the target dataset from left to right. "+" and "-" denote the domain similarity (red for visual similarity and blue for class similarity) between source and target datasets. SSP methods include MoCo, SwAV, and Rotation.

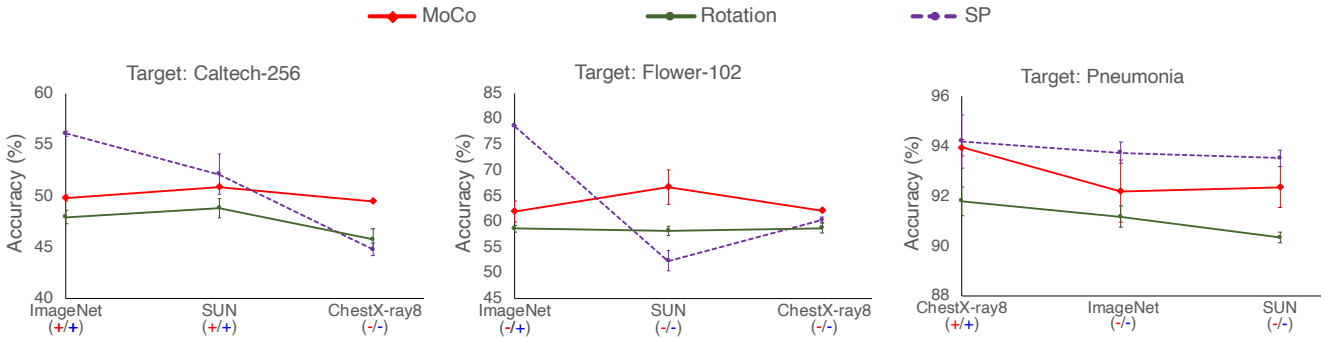

Figure 3. Top-1 image classification accuracy (%) in the study of domain similarity, using ResNet-18 as backbone.

better accuracy than SP, though there are a few exceptions. These results show that in general when domain similarity is small, SSP methods work better than SP. The possible reason is: when the domain similarity is small, the classes in source labels are largely different from those in target labels. When the representations are learned by fitting source labels, they are biased to source classes and generalize less well on the target dataset which has a different set of classes. SSP avoids using source labels, hence is not prone to such a bias. **Third**, SSP methods are less sensitive to domain difference than SP. From left to right on the x-axis of each sub-figure, the source datasets have increasing domain difference with the target dataset. When the domain difference is increasing, in most cases the accuracy of SP changes drastically while the accuracy of SSP methods remains relatively stable. For example, on Caltech-256, the difference between the highest and lowest accuracy is about 6% (absolute) for MoCo and 2% for SwAV. In contrast, for SP, the difference between the highest and lowest accuracy is about 15% (absolute). The possible reason is that SP uses source labels for pretraining; when the domain difference varies significantly, the labels change substantially, which makes the learned representations vary a lot. In contrast, since SSP methods do not leverage source labels for pretraining, they are less sensitive to the change of source labels. **Fourth**, in each sub-figure, when the domain similarity between source datasets and the target dataset is decreasing (from left to right on the x-axis), in most cases the accuracy of SSP methods and SP decreases. For example, in terms of domain similarity with Caltech-256, we have the following order: ImageNet, SUN $>$ ChestX-ray8. As can be seen, the accuracy corresponding to these source datasets decreases. This is because pretraining on a closer source domain can make the learned representations more suitable for the target domain.

6.2. Results on the amount of pretraining data

In this section, we study how SSP methods and SP perform when the amount of pretraining data varies. Given a source dataset, we create two subsets by randomly sampling 1% and 10% of data examples. The experimental results are shown in Figure 4 and Figure 5, using ResNet-50 and ResNet-18 as backbone respectively. Each sub-figure corresponds to a target dataset. On the x-axis are increasing percentages of a source dataset.

From these two figures, we make the following observations. **First**, when the amount of pretraining data is small

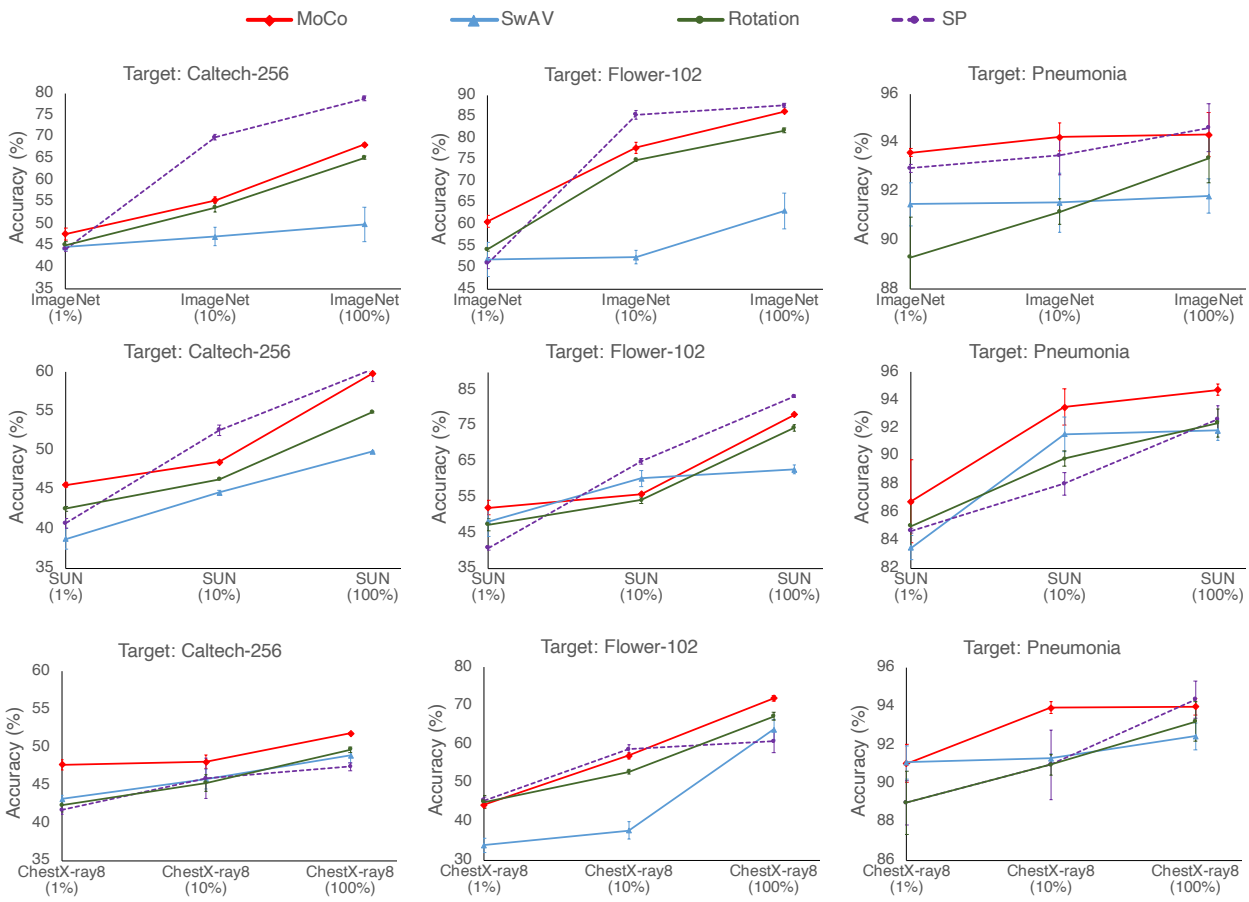

Figure 4. Top-1 image classification accuracy (%) in the study of the amount of pretraining data, using ResNet-50 as backbone. On the x-axis are source datasets with increasing percentages of examples used for pretraining. Different sub-figures are for different target datasets. SSP methods include MoCo, SwAV, and Rotation.

(e.g., 1% of source dataset), SSP methods outperforms SP in general. For example, pretrained on 1% of each source dataset, MoCo outperforms SP on most target datasets. One possible reason is: when the pretraining data is small, SP has a risk of overfitting to the labels of the small source dataset and generalizes less well on target datasets. In contrast, SSP methods perform pretraining in an unsupervised manner, which does not have the risk of overfitting to labels of small-sized source data. **Second**, when the amount of pretraining data is large (e.g., 100% of source dataset), SP outperforms SSP methods. For example, pretrained on 100% ImageNet data or 100% SUN data, SP achieves better performance than MoCo, SwAV, and Rotation on most target datasets. The possible reason is: when the dataset is large, SP is less likely to suffer overfitting. On the contrary, the diverse labels contained in the large dataset enables SP to learn more discriminative representations than SSP. **Third**, when domain similarity is large, SSP methods are less sensitive to data amount than SP. For example, with ResNet-50 as backbone, when the pretraining ImageNet data increases from 1% to 100%, the relative improvement of MoCo, SwAV, and Rotation on Caltech-256 are less than 40% whereas the relative improvement of SP are over 60%. **Fourth**, for both SSP and SP, increasing the amount of training data leads to better performance. This is not surprising since deep learning methods are data hungry.

6.3. Results on using target data for pretraining

In this study, we compare the following pretraining settings: on source data only, on target data only, and on the combined data of source and target. The source dataset is SUN. Target data refers to data examples in the training set of a target dataset. The backbone is ResNet-50. We study one SSP approach: MoCo. Figure 6 shows the results. From this Figure, we see that combining target data with source data for pretraining is beneficial for SSP (MoCo), but not beneficial for SP. As can be seen, pretraining MoCo on the combined data performs better than on source-only and than on target-only. However, SP on the combined data performs worse than on source-only. SSP is benefited because SSP is

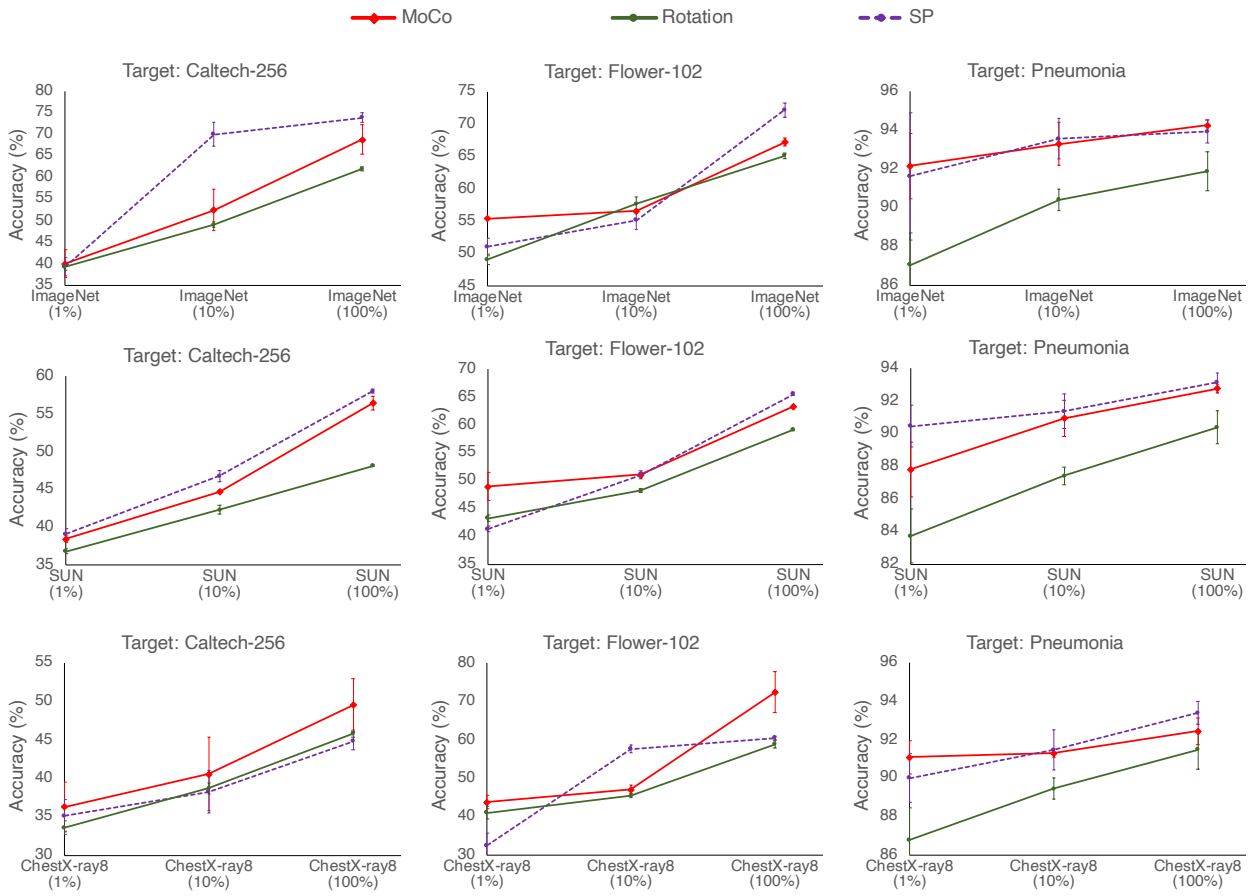

Figure 5. Top-1 image classification accuracy (%) in the study of the amount of pretraining data, using ResNet-18 as backbone.

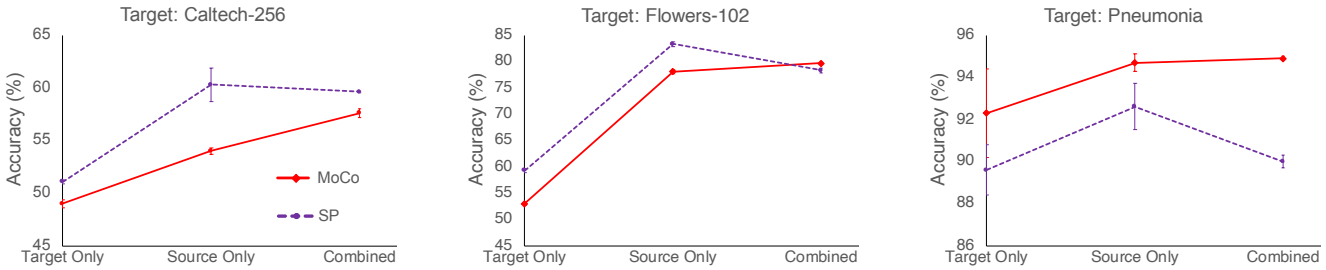

Figure 6. Top-1 image classification accuracy (%) in the study of using target data for additional pretraining, with ResNet-50 as the backbone.

unsupervised and adding unlabeled target data yields more pretraining examples. SP is not benefited probably because the model will be finetuned on labeled target data anyway and therefore it is not very necessary to use the labeled dataset for pretraining.

7. Conclusions

In this work, we make a comprehensive comparative study about self-supervised pretraining (SSP) and supervised pretraining (SP), regarding which method works better under different properties of data. Specifically, we study how the domain similarity between source and target dataset, amount of pretraining data, and usage of target data for additional pretraining affect the comparative advantages of SSP and SP. On 3 source datasets and 3 target datasets from various domains, on varying amount of pretraining data, we conduct experiments and distill a set of novel insights, such as SSP is generally more robust to differences between the source and target domain than SP, and is more robust to the amount of source data than SP. These insights can potentially help ML researchers to decide which pretraining method to use based on the properties of their applications and foster the development of new SSP and SP methods.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In Advances in Neural Information Processing Systems, pages 15509–15519, 2019. 1, 2, 3

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems, 33, 2020. 3, 5

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709, 2020. 1, 2, 3

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297, 2020. 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255. IEEE, 2009. 4

[6] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 2051–2060, 2017. 1

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. The Journal of Machine Learning Research, 17(1):2096–2030, 2016. 4

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728, 2018. 2, 3, 5

[9] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In Proceedings of the IEEE International Conference on Computer Vision, pages 6391–6400, 2019. 2

[10] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 4

[11] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1735–1742. IEEE, 2006. 3

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722, 2019. 1, 2, 3, 5

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722, 2019. 2, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 5

[15] Xuehai He, Xingyi Yang, Shanghang Zhang, Jinyu Zhao, Yichen Zhang, Eric Xing, and Pengtao Xie. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. medRxiv, 2020. 2

[16] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. arXiv preprint arXiv:1905.09272, 2019. 1, 2, 3

[17] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 172(5):1122–1131, 2018. 4

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. arXiv preprint arXiv:2004.11362, 2020. 2

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations, 12 2014. 5

[20] Tassilo Klein and Moin Nabi. Contrastive self-supervised learning for commonsense reasoning. arXiv preprint arXiv:2005.00669, 2020. 2

[21] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1920–1929, 2019. 2

[22] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In Advances in Neural Information Processing Systems, pages 317–327, 2019. 2

[23] 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. 1, 2, 3

[24] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9339–9348, 2018. 2

[25] Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? arXiv preprint arXiv:2003.14323, 2020. 2

[26] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pages 722–729. IEEE, 2008. 4

[27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018. 1, 2, 3

[28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009. 1, 3

[29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2536–2544, 2016. 1

[30] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2536–2544, 2016. 2

[31] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014. 4

[32] Cinjon Resnick, Zeping Zhan, and Joan Bruna. Probing the state of the art: A critical look at visual representation evaluation. arXiv preprint arXiv:1912.00215, 2019. 2

[33] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. arXiv preprint arXiv:2004.04136, 2020. 2

[34] Yu Sun, Xiaolong Wang, Liu Zhuang, John Miller, Moritz Hardt, and Alexei A. Efros. Test-time training with self-supervision for generalization under distribution shifts. In ICML, 2020. 2

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. arXiv preprint arXiv:1906.05849, 2019. 1, 2, 3

[36] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6629–6638, 2019. 2

[37] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2566–2576, 2019. 2

[38] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2097–2106, 2017. 4

[39] 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. 1, 2, 3

[40] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3485–3492. IEEE, 2010. 4

[41] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark. arXiv preprint arXiv:1910.04867, 2019. 2

[42] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In European conference on computer vision, pages 649–666. Springer, 2016. 2

[43] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In Proceedings of the IEEE International Conference on Computer Vision, pages 6002–6012, 2019. 1, 2, 3