Knowledge-Based Systems 224 (2021) 107090

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Review on self-supervised image recognition using deep neural networks

Kriti Ohri *, Mukesh Kumar

Department of CSE, National Institute of Technology Patna, Bihar 800005, India

ARTICLE INFO

Article history:
Received 19 October 2020
Received in revised form 14 April 2021
Accepted 26 April 2021
Available online 29 April 2021

Keywords:
Self-supervised learning
Unsupervised learning
Semi-supervised learning
Transfer learning
Deep learning
Pretext tasks
Convolutional neural network
Contrastive learning
Online clustering

ABSTRACT

Deep learning has brought significant developments in image understanding tasks such as object detection, image classification, and image segmentation. But the success of image recognition largely relies on supervised learning that requires huge number of human-annotated labels. To avoid costly collection of labeled data and the domains where very few standard pre-trained models exist, self-supervised learning comes to our rescue. Self-supervised learning is a form of unsupervised learning that allows the network to learn rich visual features that help in performing downstream computer vision tasks such as image classification, object detection, and image segmentation. This paper provides a thorough review of self-supervised learning which has the potential to revolutionize the computer vision field using unlabeled data. First, the motivation of self-supervised learning is discussed, and other annotation efficient learning schemes. Then, the general pipeline for supervised learning and self-supervised learning is illustrated. Next, various handcrafted pretext tasks are explained that enable learning of visual features using unlabeled image dataset. The paper also highlights the recent breakthroughs in self-supervised learning using contrastive learning and clustering methods that are outperforming supervised learning. Finally, we have performance comparisons of self-supervised techniques on evaluation tasks such as image classification and detection. In the end, the paper is concluded with practical considerations and open challenges of image recognition tasks in self-supervised learning regime. From the onset of the review paper, the core focus is on visual feature learning from images using the self-supervised approaches.

1. Introduction

The essence of learning is not always direct supervision but the ability to predict in limited external guidance. Self-supervised learning is a form of unsupervised learning where the data itself provides a strong supervisory signal that enables Convolutional Neural Network (ConvNet) to capture intricate dependencies in data without the need for external labels. Essentially, a self-supervised learning task is formulated from a large unlabeled corpus of images and the ConvNet is trained to learn some task (pretext task) designed by the user. For most pretext tasks, the ConvNet predicts a masked area of the image [1] or predicts the correct angle by which the image is rotated [2] etc. The representations learned on the pretext task from the encoder part of the ConvNet are subsequently used for downstream tasks where limited annotated data is available. Self-supervised learning approaches are successful in developing standard pre-trained models for natural language processing like BERT [3], ULMFiT [4], Word2Vec [5], GloVe [6], fastText [7], RoBERTa [8], XLM-R [9], T5 [10] but less in computer vision tasks because of the high-dimensional continuous space that images occupy. Motivated by the BERTs success in NLP as a self-supervised learning model, ActBERT a method to learn video-text pairs in a self-supervised way is proposed [11]. It enables learning of joint video-text representations from large unlabeled video dataset. The earlier models used linguistic features for video-text joint modeling whereas ActBERT leverages three sources of information for cross-model pre-training such as action-related features, region features, and language embeddings. The model is pre-trained with HowTo100M which is a large-scale dataset of 136 million video clips sourced from 1.22M narrated instructional web videos. The pre-trained model is evaluated on five downstream tasks namely text-video clip retrieval, video captioning, video question answering, action segmentation, and action step localization. Results show that actBERT outperforms its counter supervised models in learning video-text representations.

Recently various self-supervised approaches using pretext tasks have surged in the field of computer vision. These pretext tasks designed by the user helps in learning rich representations from the input images. Learning representations of the input

* Corresponding author.
E-mail addresses: kriti.ohri@gmail.com (K. Ohri), mukesh.kumar@nitp.ac.in (M. Kumar).

https://doi.org/10.1016/j.knosys.2021.107090
0950-7051/

signal makes it easier to extract useful characteristics of the data that help in building classifiers or other predictors on top of ConvNets [12]. The cognitive motivation behind self-supervised learning is how infants learn; they are not always presented with the correct answers. Within 3 to 4 months of birth, infants have meaningful expectations about the world around them [13]. Through observation, common sense, and minimal interaction makes infants capable of self-learning. The environment around the infants becomes a source of supervision that helps in developing a general understanding of how things work without constant supervision. A similar concept is mimicked through self-supervised learning in machines, where the data itself contains inherent features that provide supervision for training the model rather than the annotated labels instructing the network of what is right and what is wrong. Hence, the goal of self-supervised learning is to learn representations of the input without the labels that transfer well to the downstream tasks where little labels are available.

There are many popular ConvNet architectures like AlexNet [14], VGG [15], GoogLeNet [16], ResNet [17], DenseNet [18], inceptionV3 [19] etc. which have achieved state of the art performance on various image recognition tasks that are trained on a large labeled dataset. These models are widely used as pre-trained models and are fine-tuned for target tasks which the user expects the network to perform. These pre-trained models have yielded state-of-the-art performance on standard labeled datasets such as ImageNet (1.4 million images with 1000 categories) [20] and OpenImage (59.9 million images with 19,957 categories) [21] etc. The success of these sophisticated architectures relies on large labeled datasets on which they are trained for several GPU hours per day. But in reality, building huge labeled datasets is a challenging task and requires intense manual effort. ImageNet dataset contains around 14*M* labeled images and it took around 22 human years to develop. Though we have publicly available image labeling tools like Amazon SageMaker that guides a human labeler step-by-step to label images, audio, text, etc. but it comes with an additional expense [22]. The situation is, even more, complex and cumbersome for video datasets that are more expensive to label due to the addition of spatial and temporal information. The Kinetics dataset [23], which is used to train ConvNets for human motion or action recognition, contains 500,000 videos belonging to 600 categories, and each video lasts for 10 s. In addition to labeling challenges, learning good representation of the input is an additional reason to look for an alternative solution to supervised learning [24]. Moreover, in a domain like medical, it is hard to obtain annotations because of the privacy concerns and also not knowing what exactly to annotate. Hence, a pre-trained model trained on a large unlabeled dataset can be used in such a domain. However, the benefit of pre-training and fine-tuning on downstream tasks is reduced when the downstream task images belong to a completely different domain than the images that were used for pre-training the network [25].

This paper provides an extensive compilation of different ideas bought forth by the researchers that help in building a pre-trained model using intrinsic properties of the data rather than labeled dataset. The paper highlights the breakthrough in self-supervised learning methods that will give a head start to budding researchers to explore self-supervised learning that will serve as a better alternative in long run. Self-supervised learning is a step forward for building background knowledge and instilling common sense in machines that remain an open challenge in AI since its inception. The paper attempts to summarize and evaluate various self-supervised methods that target learning better representations from the input without labels. The learned representation aid the downstream task (actual task) where small labeled dataset is available. The ongoing research in self-supervised learning indicates that we can bring self-supervised learning paradigm shift to computer vision without much relying on labeled data.

1.1. Different learning schemes

The researchers are always in a quest to formulate different learning schemes that rely on minimal labeled data. Following are the various learning methods that require minimal fine-grained labeling:

Semi-supervised learning: Semi-supervised learning methods use a combination of supervised and unsupervised learning. The model is first trained on a fraction of the dataset that is labeled manually [26]. Once the model is trained it is used to predict the remaining portion of the unlabeled dataset. At last, the network is trained on the full dataset comprising of manually labeled data and pseudo labeled data. Another setup of semi-supervised learning is to train a large capacity model called the "teacher" model with large labeled dataset. This model is then used to predict labels of an unlabeled dataset. The predicted examples are ranked against each concept class and the top-scoring examples are used for pretraining the target model called the "student" model. The final step is to fine-tune the student model with all the available labeled data. It is found that such models have higher accuracy compared with the target model trained only on labeled data [27].

Weakly-supervised learning: Weakly supervised learning refers to learning from coarse-grained labels or noisy labels. In the effort to minimize manual labeling, the researchers exploited Instagram images that are posted by users with hashtags. These hashtags have associated images and form a good source of abundant data (3.5 billion images and 17,000 hashtags). The ConvNet for image recognition is pre-trained with a billion-image version of this large hashtag dataset and fine-tuned on labeled ImageNet dataset. To refine the noisy hashtag labels different label space tests are used such as hashtag mapped to ImageNet synsets, hashtag mapped to WordNet synsets, etc. [28]. The cost of obtaining weak supervision labels is much cheaper than the human labeling process.

Semi-weak supervised learning: Semi-weak supervised learning combines semi-supervised learning and weakly-supervised learning [27]. It uses a framework called the "student–teacher" network that combines both weak and semi-supervised learning. The "teacher" model is first pre-trained with hashtag images that have weak and noisy labels also called the weakly supervised dataset. Further, the model is fine-tuned with ImageNet labeled dataset and then it is used to predict the softmax distribution over the weekly supervised dataset (hashtag images) that was used to pre-train the "teacher" model. Next, the target "student" model is pre-trained with the weakly supervised data with the refined labels from the teacher model. Finally, the labeled data is used to fine-tune the "student" model.

The goal of machine learning has always been to bridge the gap between human and machine level learning. To make machines even more intelligent, the researchers proposed different learning strategies that mimic how humans learn visual concepts of rare instances or objects by just seeing them once. Recently, there are increasing interests in learning methods that learn novel concepts from limited data and also in the learning methods that continually learn without forgetting the prior knowledge. Following are the learning methods that are a step forward towards annotation efficient learning.

Incremental Learning: The goal of incremental learning is to continuously learn and solve new tasks without forgetting the tasks learned in the past by leveraging the data that gets added with time. Incremental learning comes with different variants such as: task incremental learning, class incremental learning, domain incremental learning, etc. In task incremental learning, the model can perform varied tasks right from image classification to object detection to image segmentation and then to

instance segmentation as data gets continuously updated with time. In class incremental learning the model fixes a learning task for e.g., image classification and then it goes from classifying samples of class A to class B and to class C and so on with all sets of classes. In domain incremental learning transition takes from one task happening in one domain (eg. Natural images) to another domain lets say in medical domain. Some methods that achieve incremental learning are zero-shot learning [29], continuous updating of the training set, or using a fixed data representation.

Few shot learning: The goal of few short learning is to effectively learn visual concepts from a training dataset that contains very few instances related to the novel classes. Given N number of novel classes and K samples in each class (K can be small as one)the learner tries to learn visual concepts for instances that are exotic or rare. In this scenario where we have limited training samples for each class, training a deep learning network from scratch results in poor model performance due to overfitting. A standard two-stage approach is adopted for few short learning [30]. In the first stage, a model is pre-trained on a dataset containing a large number of instances in every class (also referred to as a base or train class) to solve a classification task. In the second stage, the model is adapted by transferring the learned parameters from the pre-trained model by removing the last output layer and having a classifier that can classify an instance to one of the novel classes. Finally, the model is finetuned on the limited dataset containing novel classes to perform the actual downstream task of classifying the query to one of the novel classes. Though transfer learning achieves significant improvement in performance than a model made from scratch but the scarcity of data at the second stage, leads to an overfitted model. Hence few short learning requires a transfer learning approach called meta-learning (learn-to-learn approach) [31]. The idea of meta learning is not to supervise to get the right result rather how the answer should behave. Few shot learning with meta learning allows for the training of the learning algorithm itself (Stochastic Gradient Descent) instead of the classifier. To train the learning algorithm, a meta learner module f_θ is implemented which gets optimized for solving few-shot classification task by backpropagating through the classifier and the meta learner f_θ to find gradients with respect to the parameters of the meta learner. The meta learner f_θ is trained using training episodes (S, Q) from base class data (large labeled dataset) by sampling few base classes N and sampling S support examples per class and K query/test samples. The meta learner f_θ then generates the classifier model that predicts the classification scores for test samples. The classification loss (cross-entropy loss) is then minimized by optimizing the parameters of the meta learner by backpropagation. The meta learner f_θ is then evaluated or tested by keeping it fixed which generates a model for novel classes by using the train or support examples of the novel classes and predicting the novel class for the test/query samples. Recent progress in few shot classification using meta learning lead to the use of unlabeled examples along with the labeled data within each training episode [32]. The authors adopted two approaches, one where all unlabeled examples are assumed to belong to the same set of classes as the labeled examples of the episode, as well as the more challenging situation where examples from other distractor classes are also provided. The experimental results show that this scheme learns to improve the prediction of novel classes due to the incorporation of unlabeled examples. In another recent work, the authors leveraged the vast amount of freely available unlabeled video data to perform the task of few-shot video classification. In this semi-supervised few-shot video classification task, millions of unlabeled data are available for each episode during training [33].

Though many new learning techniques have reduced the requirement of fine-grained labeling, still a follow-up question is asked if these labels are really required because scaling the manual labeling process to all the internet images is completely infeasible. Hence, a potential solution proposed by researchers is to learn visual image representations from a large unlabeled dataset by proposing various pretext tasks that are given to the network to solve [34]. The learned weights or the learned representations from the pre-trained model are then used as initialization for downstream computer vision tasks where only some annotations are available.

Unsupervised and self-supervised learning: Self-supervised models use supervisory signals of the partial input that is available to learn a better representation of the input. They leverage the underlying structure in the data to predict the unobserved or hidden part of the input. Whereas, in unsupervised learning, we have samples with no external signals or labels that guide the learning process. Hence calling self-supervised learning as "unsupervised" is not true as it uses far more feedback signals. Details about the structural semantics of the images can be well learned through self-supervised learning than any other form of learning method. The unlabeled data is humungous, and the amount of feedback provided by each sample is huge that aids in learning better representations of the input. According to Misha Denil, unsupervised learning is thinking hard about the model and using whatever data fits the model. Whereas, in self-supervised learning, you think hard about the data and use whatever model fits.

On the other hand, supervised learning not only depends on annotated data but also suffers from issues such as generalization error, adversarial attacks, model brittleness, shortcuts, and spurious correlations. Moreover, the networks trained using supervised learning may not strive hard to learn generalized feature representations and can get away by memorizing the mapping between input and output as the ground truth is always available. Sometimes the ConvNet starts classifying the objects by mere texture without learning rich representation of the object [35,36]. For example, if the texture of the object is scattered randomly around the image, the convNet still predicts the right object without extracting the object from its background. Hence, the supervisory signal can bias the network and lead it to work in an unexpected way indicating that the supervised models are invariant to useful features required for structural understanding [37]. In real sense then our model becomes a texture detector rather than an object detector. Also, the ConvNets trained using supervised learning are brittle when it comes to dealing with dynamic data. Once the supervised model is trained it becomes difficult for the model to adapt to the new data without forgetting the previous knowledge. Hence the supervised model needs to be re-trained with all the data again. With the infeasibility of using supervised learning in all deep learning tasks, the researchers proclaim that the next AI revolution will not be supervised but self-supervised. Fig. 1 shows the supervised learning pipeline, the model learns visual features through the process of training the ConvNet by large amount of labeled data (e.g. ImageNet dataset). After the model is trained, the learned parameters serve as a pre-trained model and are transferred to the target task like object detection or image segmentation by fine-tuning the model. With pre-training, we can use fewer data and can take advantage of models that are already trained with millions of images. During the transfer learning, only general features from the first few layers are transferred to target tasks. However, getting labels for a dataset of the size of ImageNet is quite expensive and and curating datasets of such size for different domains is a laborious task.

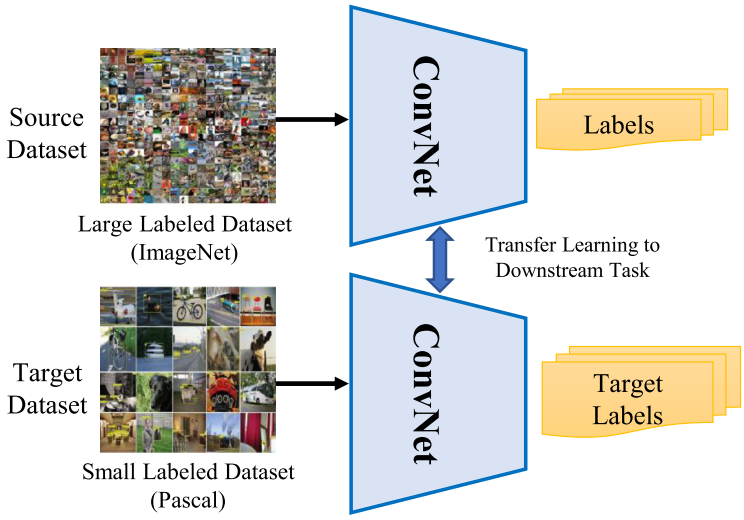


Fig. 1. Supervised learning pipeline, the learned features from the pre-trained model trained on large labeled dataset (e.g. ImageNet dataset) are transferred to the target task where limited labeled dataset (e.g. Pascal dataset) is available.

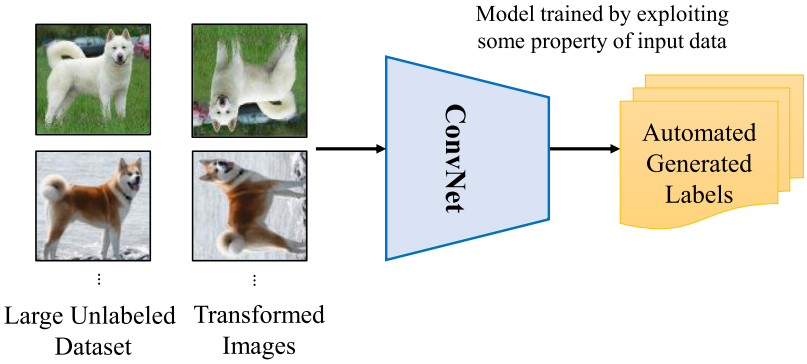


Fig. 2. ConvNet trained to generate automated labels using self-supervised learning without human annotating the input images.

1.2. Self-supervised learning

The key to unleashing unlabeled data is self-supervised learning, which is gaining importance in recent years. Though the use of self-supervised learning is not new and it can be traced back to 1989 by Jürgen Schmidhuber, in his paper "Making the World Differentiable" which explains how two self-supervised recurrent networks can interact to attack fundamental credit assignment problem [38]. However, self-supervised learning gained momentum in recent years due to the explosion of huge amount of unlabeled data. The primary objectives of self-supervised learning are (1) To deploy state-of-the-art deep learning models with performance matching the supervised counterpart without relying on huge labeled dataset. (2) To learn generalized and semantically meaningful representations from unlabeled data that help during downstream tasks like image classification, image segmentation, object detection, etc. (3) To harness huge amount of data that is available for free by replacing the supervised pre-training with self-supervised pre-training. (4) To have a more practical approach to learning as possessed by humans. Fig. 2 shows self-supervised learning where the ConvNet generates pseudo labels (e.g. degree to which the input image is rotated) from the rotated input image by exposing the relationship between parts of the input data.

Concepts in self-supervised learning: We will now discuss various concepts related to self-supervised learning and its vocabulary.

Pretext or auxiliary task: To learn visual features from unlabeled data certain tasks are pre-designed for the ConvNet to solve. The term "pretext" means that the task is done before the actual target task is undertaken whose mere purpose is to learn generalizable representations of the input both at a low level and high level as shown in Fig. 3 [39]. For most of the pretext tasks, a part of the data is withheld or some transformation is applied for which the network predicts the missing part or the correct transformation applied. The label generated by the network is the property of the data itself.

Pseudo labels: Pseudo labels are generated automatically based on the type of pretext task the network solves [39]. For example, if an input image is taken and a transform is applied to it let us say rotation and passed to the ConvNet, the ConvNet predicts the property of the transform i.e., angle by which the image is rotated.

Pre-trained model: It is a ConvNet that contains the learned representations/useful behavior of data that is trained on a large unlabeled dataset using a pretext task. Mostly the model is pre-trained on ImageNet without labels on a pretext task and subsequently fine-tuned on a smaller labeled dataset [37].

Transfer learning: Transfer learning is the operation of transferring the pre-training features from the pre-trained model to solve the downstream tasks like image classification, object detection, and image segmentation, etc. Through transfer learning the model achieves higher accuracy with much less labeled data and requires less computation time than models that do not use transfer learning. Whereas building the model from scratch initialized with random weights results in an inefficient solution

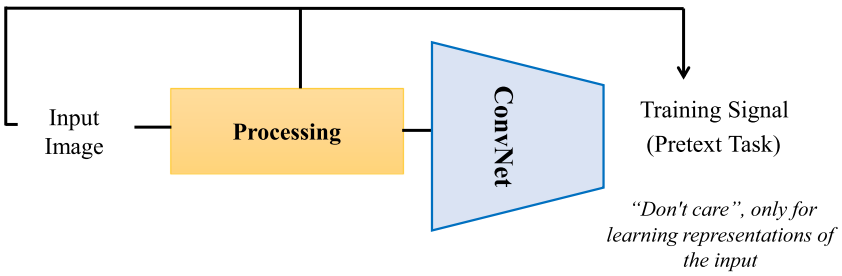


Fig. 3. Self-supervised learning using pretext task to learn good representation of the input.

as the network starts from a point where it does not know anything [40]. Two major transfer learning scenarios that evaluate the learned representation are (1) Linear classifier (ConvNet as fixed feature extractor) (2) Fine-tuning the ConvNet for downstream tasks.

Linear classification: To evaluate the learned representation a linear classifier is trained on top of the ConvNet trained on the large unlabeled data set (ImageNet). The last fully connected layer of the pre-trained model is removed, and the rest of the ConvNet is frozen on which a classifier is trained. Evaluation is often performed on the same dataset that was used to train the network on the pretext task. Typical datasets for linear classification include ImageNet, Places205, Pascal VOC07, COCO14, etc.

Fine-tuning the model: The second scheme to evaluate the learned representation is not only to replace and retrain the classifier on top of the ConvNet but also fine-tune the weights of the pre-trained network through backpropagation. It is possible to fine-tune all the layers of the ConvNet, or we can keep some of the earlier layers fixed and only fine-tune some higher-level layers of the network.

Type of architecture: The quality of the visual representations learned using pretext task depends significantly on the type of ConvNet architecture used and the ability of the network to scale with increased data. The impact of the architecture is found on the quality of the representation learned and on the accuracy of the downstream tasks. The popular convolutional neural architectures used for pre-training are- ResNet50, ResNet50 v1, ResNet50 v2 and also their scaled-up versions [34]. It has also been observed that low-capacity model like Alexnet [14] has not shown much improvement with more data as compared to ResNet.

Downstream tasks: These tasks are specific to the problem that defines what the model actually does (primary task). Whereas, pretext task is a secondary task undertaken to achieve the primary tasks. Many computer vision downstream tasks exist such as image classification, object detection, image segmentation, etc. Table 1 shows the image datasets used for downstream tasks.

1.3. Self-supervised learning pipeline

The general pipeline of self-supervised learning is shown in Fig. 4. In the first stage, as shown in Fig. 4(a), the ConvNet is trained on a pretext task (e.g. image rotation) using a large corpus of unlabeled data. The network learns useful representations and predicts the degree by which the image is rotated. In the second stage as shown in Fig. 4(b), the rotation prediction head from the pre-trained model is removed by keeping the remaining network fixed and a linear classifier is trained on top of it with a new dataset where fewer labels are available. Complex downstream tasks can also be performed such as image segmentation and object detection by fine-tuning the entire network or partial network (deeper layers) using backpropagation as shown in Fig. 4(c).

2. Image representation or feature learning techniques using self-supervised learning

This section summarizes the early works on traditional unsupervised learning, and also the recent handcrafted pretext tasks and contrastive instance learning schemes designed to learn rich representation of the input. Broadly the representations learned using self-supervised learning are categorized into six categories: reconstruction from a corrupted or partial image, reconstruction of the image from an altered view of the image, image generation, spatial context prediction, transformation prediction, instance discrimination or contrastive learning and clustering based schemes.

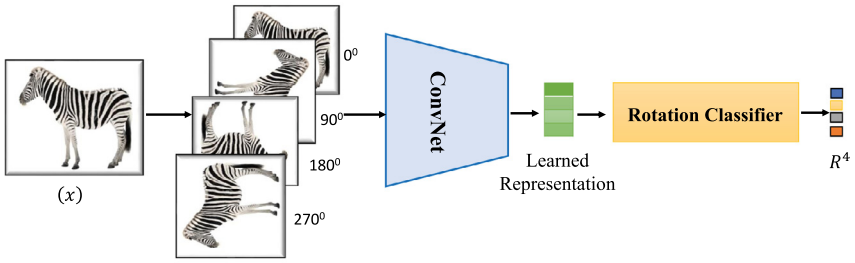
2.1. Reconstruction from a corrupted or partial image

One of the early works in unsupervised learning is the classical autoencoder that learns representations by compressing the input image at a low dimensional bottleneck layer. The encoder–decoder assumes that there exists a high degree of correlation/structure in the data. The encoder compresses the data into an intermediate representation and the decoder takes the intermediate representation and reconstructs the input image. Traditional autoencoders happen to underperform as it only compresses the input without learning rich representations of the input [41]. The denoising autoencoder is an enhancement to the traditional autoencoder that prevents the network to learn an identity function by introducing pixel level noise in the image and the network is forced to reconstruct the original image as shown in Fig. 5. Another version of denoising autoencoder is stacked autoencoder [42] in which the layers of the autoencoders are stacked to initialize a deep architecture that encodes and decodes the data across various layers. As the layers are stacked, the model learns a better representation of the input. The downside of the autoencoders is that the noise added to the images is random and unsystematic which does not contribute to semantic learning of the input. Some improvements were bought by denoising autoencoders by corrupting the input but the corruption was localized and random which did not account for greater learning at the semantic level. The scheme also results in a train and evaluation gap as training is done on noisy images and evaluation is done on clean images.

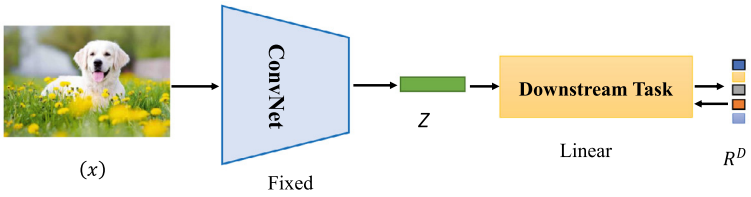
Instead of formulating the problem as a compression task, many researchers formulated the problem as a context-based prediction task. An ad-hoc approach is implemented wherein the user designs a task for the ConvNet to solve that aims in learning rich representation of the input. Image inpainting is one such pretext task that forces the network to predict the masked area of pixels from other parts of the input [1]. The model is trained on large number of unlabeled images with corresponding masked out regions. Fig. 6 shows the masked input image passed to the encoder that captures the semantic context of the image to a latent feature representation and the decoder reconstructs the actual image by filling realistic image content in the masked region.

Table 1
Summary of commonly used image datasets used in downstream image recognition tasks.

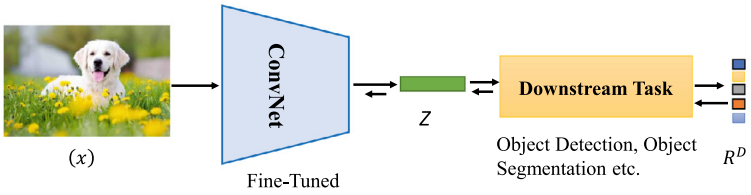
Category	Dataset	Train size	Classes	References
Natural	Caltech 101	3060	102	(L. Fei-Fei et al. 2004)
Natural	CIFAR-10	50000	10	(Krizhevsky, 2009)
Natural	CIFAR-100	50,000	100	(Krizhevsky & Hinton, 2009)
Natural	DTD	3760	47	(Cimpoi et al., 2014)
Natural	Flowers 102	2040	102	(Nilsback & Zisserman, 2008)
Natural	Oxford-IIIT Pets	3680	37	(Parkhi et al., 2012)
Natural	Sun 397	87,003	397	(Xiao et al., 2010)
Natural	Caltech-UCSD	6033	200	(Lin et al. 2015)
Natural	Stanford cars	8144	196	(Jonathan Krause et al. 2013)
Natural	Food 101	120,216	251	(Bossard et al. 2014)
Natural	FGVC Aircraft	3334	100	(Maji et al. 2013)
Natural	Stanford Dogs	12, 000	120	(Aditya Khosla et al. 2011)
Natural	SVHN	73, 257	10	(Netzer et al. 2011)



(a) Stage 1: Train the ConvNet on a pretext task (e.g. image rotation) using a large unlabled dataset e.g. ImageNet.



(b) Stage 2: Train linear classifier on learned features with fewer labels.



(c) Stage 2: Fine-tune network for complex downstream tasks with fewer labels.

Fig. 4. The general two stage pipeline of self-supervised learning. First, the model is pre-trained on a pretext task (e.g., rotation) without the labels, and then the linear classifier is trained on top of the ConvNet removing the prediction head and finally fine-tuned for complex downstream tasks with a small labeled dataset.

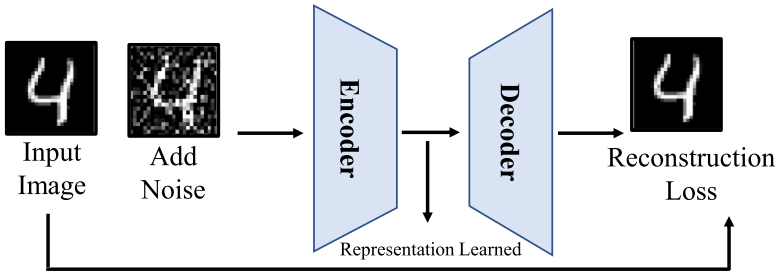


Fig. 5. Denoising autoencoder learns the compressed representation of the input at the encoder by exploiting the redundant information of the input image. The decoder reconstructs the input image which is close to the original image by minimizing reconstruction loss.

To solve this task the network must acquire a general understanding of the structure of different objects, their colors, and gain a deeper semantic understanding of the entire scene. This task will only be easy for the ConvNet to solve if it can recognize the object in the image. The loss function used in the scheme is the joint loss of reconstruction objective ($L2$ loss) and adversarial loss. The

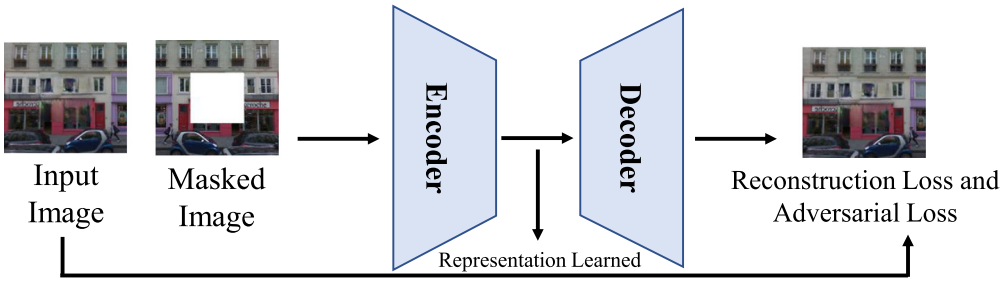
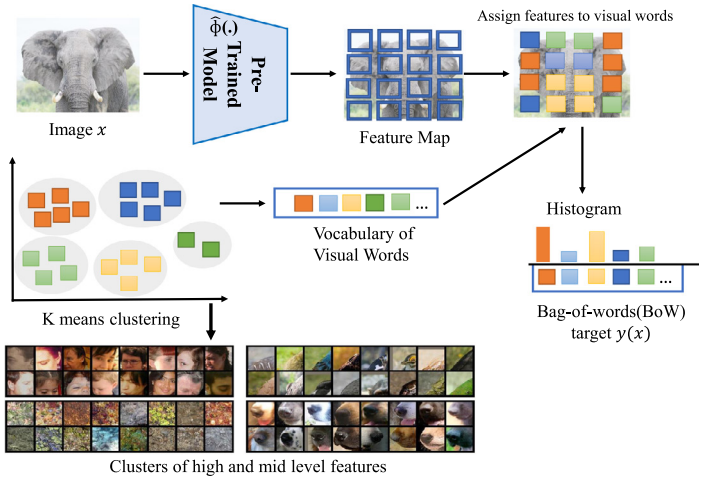
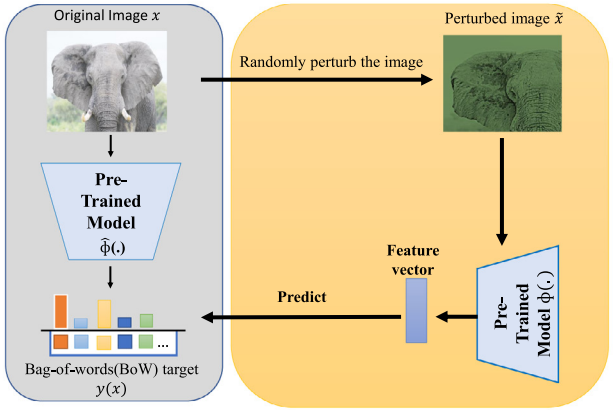


Fig. 6. Image inpainting pretext task of reconstructing the masked region of the input image. The ConvNet is trained on pairs of masked-original images by reducing the reconstruction and adversarial loss.



(a) Learning representations through prediction of Bags of Visual Words. A pre-trained feature extractor $\hat{\Phi}(.)$ extracts the features across all the images in the dataset to form clusters of high and mid-level representations that represents a vocabulary of visual words. Then the image is represented as a bag of visual words by assigning features to visual words [43].



(b) In the second stage the image x is perturbed to give $\tilde{x}$ and is send as an input to a second network $\Phi(.)$. The network then predicts BoW and compares with the actual BoW of the original image x. The feature extractor $\Phi(.)$ is then used for downstream computer vision tasks [43].

Fig. 7. A two stage process of predicting bag of visual words.

reconstruction loss captures the overall structure of the missing region and the adversarial loss is incorporated to get a sharp prediction of the masked region by picking up a particular mode from the distribution. In another way, the network can be thought of as an adversarial generative network, where the generator tries to generate the images by filling the masked region and the discriminator tries to find the discrepancies between the predicted and actual image patch. Once the pre-training is complete, the decoder is removed and the learned representation is used for the downstream tasks. Though the scheme reserves the fine details of the image, reconstruction is tough and ambiguous as there are multiple ways to fill the missing region in the image. Moreover, there exists a train-evaluation gap as training is done on masked images and evaluation is done on non-masked images.

Another self-supervised learning approach proposed is predicting bag-of-visual words (BoW) as shown in Fig. 7 that is inspired by the natural language processing method of predicting bag of words. Instead of words, the scheme uses visual words that encode discrete visual concepts useful for downstream tasks like image classification and object detection [43]. Fig. 7(a) shows the unlabeled images fed to a pre-trained self-supervised feature extractor $\hat{\Phi}(.)$ that extracts features across the entire dataset and clusters the similar features to form a vocabulary of visual words. Further, a histogram for each image x is created representing the count of individual high level and mid-level features present. Then as a self-supervised task, the pre-trained ConvNet $\hat{\Phi}(.)$ predicts the bag of words representation for original image x. Correspondingly another network $\Phi(.)$ is trained to predict bag of visual words of a perturbed version $\widetilde{x}$ of the image x as shown in Fig. 7(b). Further cross entropy loss is calculated between the predicted bag of words and the original bag of words $y(x)$ which is backpropagated to refine the model. To solve this task the ConvNet must learn to detect visual clues that are invariant to perturbations and also contextual features. The advantage of the scheme is that the representations learned are invariant to transformations and it learns contextual reasoning skills by inferring missing visual words of missing image regions. On the other hand, it loses low-level details and spatial information of the input image.

2.2. Reconstruction of the image from an altered view of image

Another set of pretext tasks the ConvNet solves is to predict the correct view of the image from its altered view. Image colorization is one such pretext task that forces the ConvNet to predict probable color (ab channel) of an image from a grayscale (L channel) image [44]. By solving this pretext task, the ConvNet learns an image representation by predicting the color values of an input 'grayscale' image. The network is trained on millions of pairs of colored and grayscale images with negligible cost as they are freely available. To solve the colorizing task, the network has to recognize different objects present in the image and group related parts together to tint them with the same color. Therefore, a visual representation of the input image is learned in the process of performing the pretext task that is useful in performing downstream tasks. An illustration of a grayscale image with its predicted color image using an encoder–decoder architecture based on ConvNet is shown in Fig. 8. Once the network is trained on the pretext task, the decoder part of the network is removed for performing downstream tasks. The downside of the scheme is that the reconstruction of the image is hard and ambiguous as several possible solutions exist for coloring the image. As color mapping is not deterministic, the network colors the object with the combination of all colors which leads to a greyish colored image. Also, the network is forced to evaluate on grayscale images resulting in loss of information. Though the scheme is useful when we want to color the old grayscale films.

Further to the above work, an enhancement was bought by a scheme called the split-brain autoencoder trained on pairs of grayscale and colored images. Fig. 9 shows a given color image X that is split it into grayscale channels X_1 and color channels X_2. The ConvNet is split into two distinct subnetworks $\mathcal{F}1$ and $\mathcal{F}2$. $\mathcal{F}1$ predicts the color channel from the grayscale channel and $\mathcal{F}2$ predicts the grayscale channel from the colored channel. The two complementary channels $\hat{X}_1$ and $\hat{X}_2$ are then aggregated to predict the reconstructed image of the original image on which the cross-entropy loss is calculated [45]. The goal of performing such a pretext task is to induce representations that transfer well to the downstream tasks. The same setup can also be applied for images with depth and color in which the pretext task forces the network to predict one from another. This method ensures backward consistency by doing two-way prediction from grayscale to color image and from color to grayscale image, and together the reconstructed image should be close to the original image. It is a challenging task as the reconstruction is hard and ambiguous because there are multiple possible ways to color an image. However, recently authors have devised new schemes using variational autoencoders and latent variables for incorporating diverse colorization in the reconstructed images [47].

2.3. Image generation

Generation based self-supervised methods are used for learning image representations that involve the process of generating images or high-resolution images using Generative Adversarial Networks. Most of the methods for image generation do not need any human-annotated labels. GANs learn to create realistic images that are similar to the input images but not present in the input dataset. The intuition behind this is that if we can get a model to generate a realistic image of a person's face for example, then the model must have also learned a lot about human faces in general. In GANs two networks compete with each other. A generator samples z vector from a latent space to generate a reconstructed or a realistic fake image. On the other hand, the discriminator network competes with the generator and tries to distinguish between the fake samples coming from the generator and the real samples. Following the game-theoretic approach, the discriminator forces the generator to generate realistic images, while the generator forces the discriminator to improve its differentiation ability. During the training, the two networks are competing against each other and make each other stronger [48]. GAN has served as a foundational work that has helped in the creation of various successful architectures such as DC GAN [49], WGAN [50], WGAN-GP [51], Progressive GAN [52], SN-GAN [53], SAGAN [54], BIGAN [55], BigGAN [56], StyleGAN [57], LOGAN [58] etc. After initial success in using GANs for unsupervised learning, GANs have been surpassed by self-supervised based approaches. One such unsupervised method is Large Scale Adversarial Representation Learning (BigBiGAN) for image generation and representation learning [46]. This method allows for the extraction of features in an unsupervised way from a Generative Adversarial Network and scales up the previously existing algorithms such as BIGAN [55] and BigGAN leading to an improved GAN model [56]. The traditional GANs take the latent variable and convert it into an actual image but from the representational learning perspective, we have to go from the image to its latest space. BigBiGAN, builds upon the state-of-the-art BigGAN model, extending it to representation learning by adding an encoder and modifying the discriminator. Fig. 10 shows the generator part of the network $\mathcal{G}$ that samples z from a latent space to generate reconstructed images $\hat{\mathrm{x}}$ that discriminator tells if they are fake or real. The encoder part of the generator $\mathcal{E}$ maps the images back to a latent space $\hat{\mathrm{z}}$. The latent space $\hat{\mathrm{z}} \sim \mathcal{E}(\mathrm{x})$ and $\mathrm{z} \sim P_z$ correspond to different images form a feature space that can be used for downstream recognition tasks by training logistic regression classifier using a dataset where fewer labels are available. On the other hand, the discriminator part of the network not only takes x and z pairs as input but also the joint distribution of x and z. The loss includes the aggregation of unary data term s_x, s_z, as well as the joint term s_{xz} which ties the data and latent distributions together. The discriminator loss ℓ trains the network to distinguish between the two joint data-latent distributions from the encoder and the generator, pushing it to predict positive values for encoder input pairs $(\mathrm{x}, \mathcal{E}(\mathrm{x}))$ and

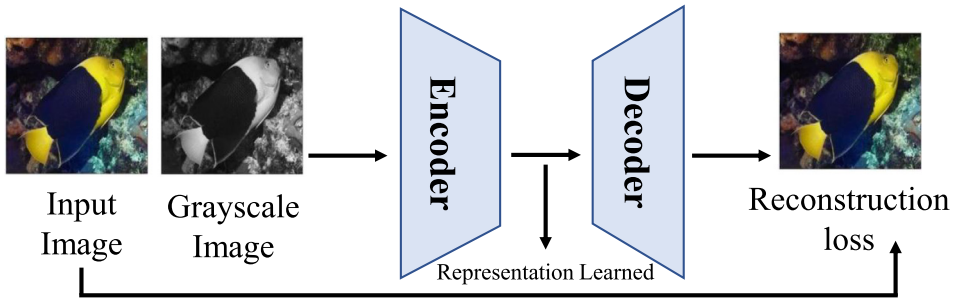


Fig. 8. Colorization as a pretext task, the ConvNet predicts the color image from a grayscale image.

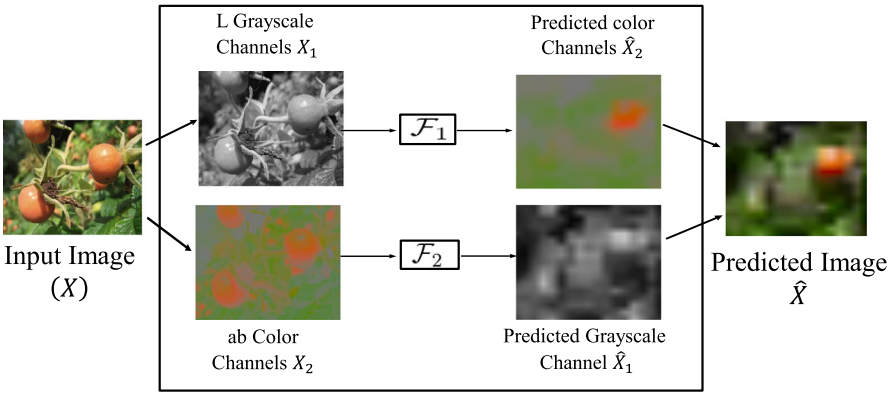


Fig. 9. Split-Brain Autoencoder composed of two disjoint sub-networks $\mathcal{F}1$ and $\mathcal{F}2$, each trained to predict one channel from another. Network $\mathcal{F}1$ performs automatic colorization, whereas network $\mathcal{F}2$ performs grayscale predictions. Combining the two channels give the predicted reconstructed image [45].

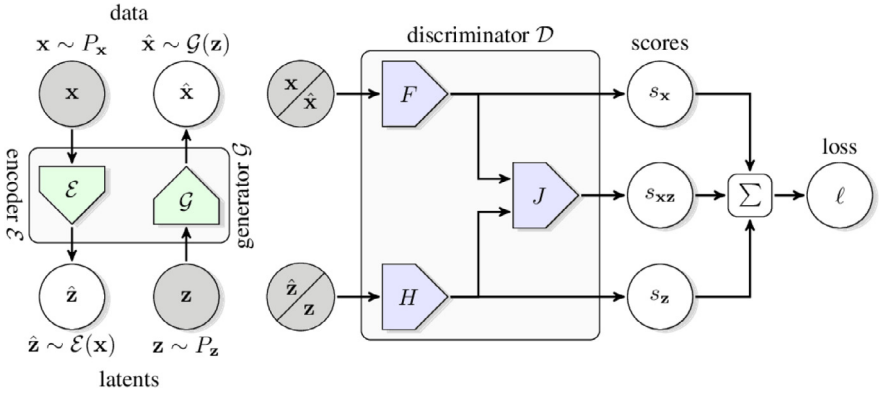


Fig. 10. Representation learning using BigBiGAN framework. The joint discriminator $\mathcal{D}$ calculates the loss ℓ. The inputs to the discriminator $\mathcal{D}$ are data-latent pairs, either ($\mathrm{x} \sim P_x$, $\hat{\mathrm{z}} \sim \mathcal{E}(\mathrm{x})$), sampled from the data distribution P_x and encoder $\mathcal{E}$ outputs, or ($\hat{\mathrm{x}} \sim \mathcal{G}(\mathrm{z})$, $\mathrm{z} \sim P_z$), sampled from the generator $\mathcal{G}$ outputs and the latent distribution P_z. The loss ℓ consists of the unary data term s_x and the unary latent term s_z, as well as the joint term s_{xz} which ties the data and latent distributions [46].

negative values for generator input pairs ($\mathcal{G}$(z), z). To generate data in a particular domain necessitates that the model should understand the semantics of the said domain. Despite the success of GANs, they are faced with few challenges such as (a) harder to train because the parameters oscillate and rarely converge and (b) the learning process is inhibited because the discriminator overpowers the generator and it fails to create real-like fakes. Another work proposed is to use pretext task for e.g., rotation for better GAN discriminators. The rotation pretext task encourages the discriminator to learn meaningful representations that are not forgotten during training [59].

2.4. Spatial context predictions

Spatial context predictions exploit the spatial relationships among image parts. Context prediction is one such pretext task that forces the network to predict the correct spatial orientation of two randomly chosen patches of an image [60]. The network is trained on pairs of (image-patch and neighboring-patch) that are chosen randomly to form a large, unlabeled corpus of data. The goal of training is to assign similar representations to semantically similar patches, for example, the representation of ears of a cat coming from different images of cat should be semantically similar to each other. The network learns to associate semantically similar patches using the nearest-neighbor matching principle. The network is forced to predict the spatial arrangement of patches for which the input image is divided into 3×3 grid of non-overlapping patches. Fig. 11 shows the ConvNet that predicts the spatial arrangement of the two input patches. The inputs to the shared ConvNet are the two random image patches, one is the anchor patch and the other is the query patch. Given the two patches, the network predicts the relative position

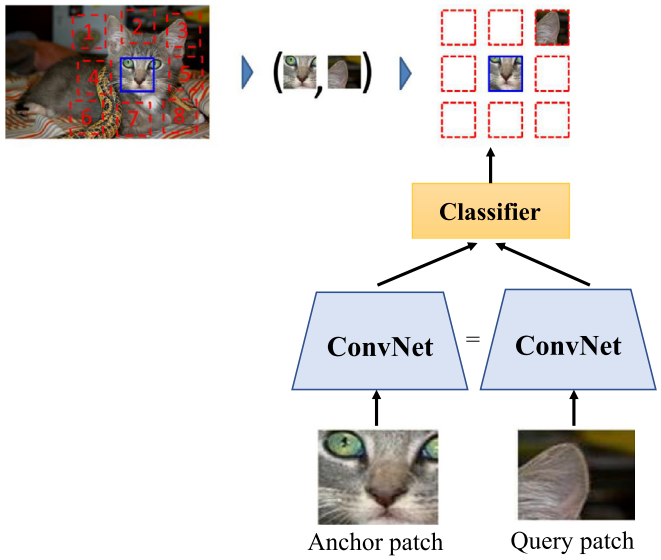


Fig. 11. Visualization of the contextual patch prediction task, the shared ConvNet receives the anchor image patch and the query image patch and it predicts the location of the query patch corresponding to the anchor patch by choosing one position among the eight positions.

of the query patch with respect to the anchor patch by choosing one location among 8 possibilities using cross-entropy loss. To solve this task, the model has to learn some semantics in terms of recognizing object parts and their spatial relationship with each other. To prevent the network from cheating, the authors jittered the patches by dropping some color channels to avoid chromatic aberration and also created the grid of non-overlapping patches to prevent the network from getting clues from boundary pixels. The downside of the scheme is that the model is trained on patches and evaluation is done on complete images.

Follow-up work to context prediction is jigsaw puzzle that is harder to solve than context prediction [61]. The pretext task forces the ConvNet to learn input representation by solving a jigsaw puzzle created from an input image. The network is trained on pairs of shuffled and ordered puzzle patches of the images which are available for free and forms a large corpus of unlabeled images. In this method, 3×3 grid of patches is shuffled through a random permutation and passed through the ConvNet that predicts the correct order of the shuffled patches as shown in Fig. 12. To accomplish this pretext task, ConvNet needs to identify objects, their shape, and their associations with their sub-parts. For a 3×3 shuffled puzzle, we have 9! (362880 ways) possible permutations for arranging the patches of the image which is quite a large sample space. To prevent searching across a large sample space, only a subset of possible permutations is used such as 64 permutations with the highest hamming distance. The representations learned encapsulates geometrical understanding of the input which helps perform downstream tasks.

Another beautiful piece of work is Contrastive Predicting Coding (CPC) which is a self-supervised technique that is generic and multimodal in approach. The scheme provides a common framework for handling different modalities like speech, image, text, and reinforcement learning in 3D environments [62]. In the context of computer vision, the model is trained by input images of size 256×256 from an unlabeled ImageNet dataset with the corresponding overlapping grid of image patches. Fig. 13 shows the input image divided into a grid of overlapping patches of size 64×64 with 50% overlap that results in 7×7 grid of patches. Each patch is then encoded by a ResNet encoder resulting in a grid of patch embeddings. The pretext task forces the network to predict the representation of a particular patch from the patch that lies above it. To do so, the network needs to reason about the object and its associated parts. It uses a special type of contrastive loss function for self-supervised learning called the InfoNCE loss [63]. The loss contrasts the representation of the predicted patch, the correct patch, and all the negatives coming from the same image and other images. The intuition is that the predictions should be more similar to the true patch embedding versus the negative patches that come from the rest of the images and the same image. The scheme achieves hard goals and learns representations spread across different modalities. The downside of the approach is that the training is slow as the images have to be divided into several patches. Additionally, the scheme also encounters train and evaluation gap, as training is done on patches, and evaluation is done on images.

2.5. Transformation prediction

Image rotation is a pretext task intended for the network to solve to predict the correct angle by which the image is rotated. The network is pre-trained on pairs of rotated image and rotation-angle by randomly rotating images of an unlabeled dataset by 0°, 90°, 180°, and 270° [2]. Fig. 14 shows the input image rotated by multiple of 90° angle and passed through a ConvNet that predicts the angle by which the input image is rotated resulting in a four-class classification task. To accomplish the task the network has to understand the location, type, and pose of objects in an image. The scheme is simple to implement and at the same time semantically meaningful. Whereas, the downside of rotation prediction assumes that the training images are captured with canonical orientations. Moreover, it results in a train-evaluation gap because the training is done on rotated images and evaluation is done on upright images. On similar lines, researchers have also worked on relative geometric transformation prediction where the network predicts the correct transformation by which the image is transformed [64].

2.6. Instance discrimination or contrastive learning

Based on the way visual features are learned, some schemes rely on instance level discrimination rather than patch level predictions or image generation. Many visual common-sense schemes discussed such as colorization, jigsaw, relative patch prediction, and rotation, etc. are adhoc hand-crafted pretext tasks and many of them rely on patches. Dividing the image into several patches increases the batch size by manifolds and also increases the training time. Moreover, it results in a train-evaluation gap because the training is done on patches, and evaluation is done on complete images. Hence, researchers proposed the concept of contrastive learning that works at the instance level where instances of the same image form a positive pair, and any other image act negative to the pair. In other words, contrastive learning provides a framework that tries to learn a feature space that pools together representations that are related and push apart representations that are not related.

In contrastive learning, the ConvNet is trained on a large corpus of millions of unlabeled images containing examples of similar and dissimilar images. Contrastive learning at the instance level is an approach to learn useful features by solving the pretext task which compares the anchor image, negative and positive (APN) representations from a large unlabeled dataset. Fig. 15 shows a batch of two images where each image forms its own class. To generate a positive pair, we take the image and augment it in two different ways. We refer to one of the augmented view of the image as an anchor image and the other view of the image as positive. Any different image to the anchor image in the

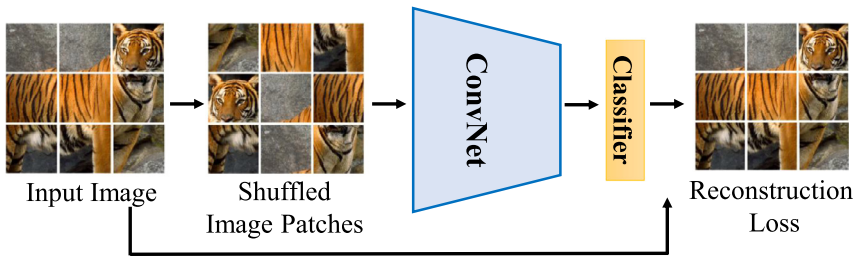


Fig. 12. Jigsaw image puzzle task. An image containing non-overlapping patches in the original image on the left, the randomly permuted patches in the middle, and the ConvNet predicts correct spatial arrangement of patches on the right.

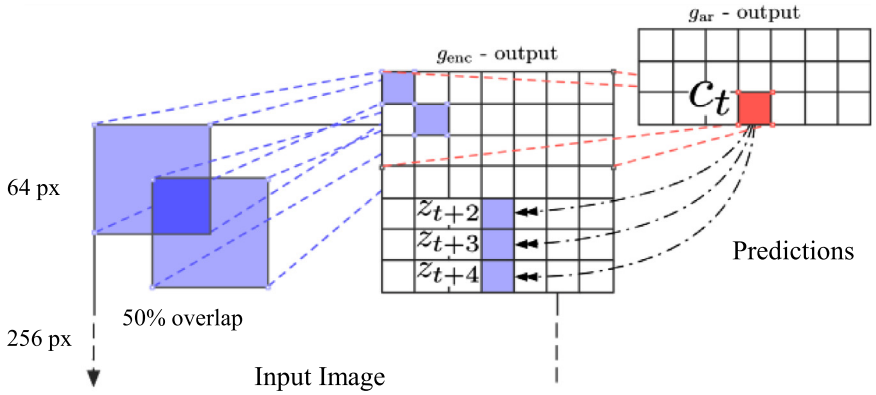


Fig. 13. Visualization of Contrastive Predictive Coding for images. A PixelCNN autoregressive model is used to make the predictions of bottom patches from top image patches [62].

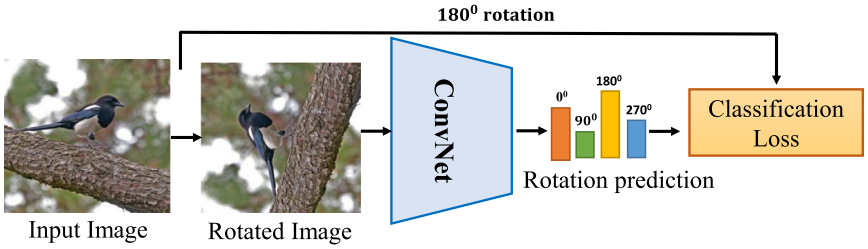


Fig. 14. Illustration of self-supervised task of predicting the angle by which the image is rotated.

dataset will be negative to the anchor-positive pair. The objective is to maximize the similarity between two views of the same image and repel them from the views coming from different images [65]. Hence, contrastive learning can reason about multiple images at once whereas jigsaw or rotation pretext tasks always reason about a single image independently. Moreover, the patch prediction tasks are not fine-grained due to the non-availability of negatives from other images. Recently many contrastive learning methods at the instance level have been proposed which have shown promising results in learning good feature representation that helps perform the downstream tasks. Despite being unsupervised these schemes have outperformed supervised pre-training in learning image representations.

Concepts in Contrastive Learning: We will now discuss various concepts related to contrastive learning that aids in learning rich representations of the input at the instance level.

Objective of Contrastive Learning: To learn a representation or a feature space that pulls representation that come from the same images and repel representations that come from different images.

Data Augmentation: Handcrafted pretext tasks for representation learning such as dividing the images into patches, rotation, colorization or masking the images, etc. have been taken over by a bunch of automated augmentations in contrastive learning. The purpose of argumentation in contrastive learning is very much different than in supervised learning as the task is very different. We would not like the ConvNet to find an easy way of doing a contrastive task just by learning one feature. The network should be able to learn many different kinds of features before it does instance discrimination. A good argumentation strategy is the most important ingredient for contrastive learning that will force the network to learn rather than cheat. Augmentation of the images allows ConvNet to learn rich and generalizable features in a self-supervised learning environment. The goal of data augmentation is to generate anchor, positive and negative (APN) images that are used in contrastive learning. Augmenting the images makes the task harder for the ConvNet as it cannot get away without learning rich representation of the input. Also, contrastive learning needs more data augmentation than supervised learning due to the non-availability of the image labels. A well-tuned composition of augmentations stands out and leads to substantial improvement gains in the performance of the downstream tasks [66]. Fig. 16 shows the typical data augmentation methods used for visual feature learning such as color transformation, scaling, random cropping, flipping (horizontally, vertically), etc.

Encoder: The encoder part of the network extracts the feature representations of the images. It takes two different augmented

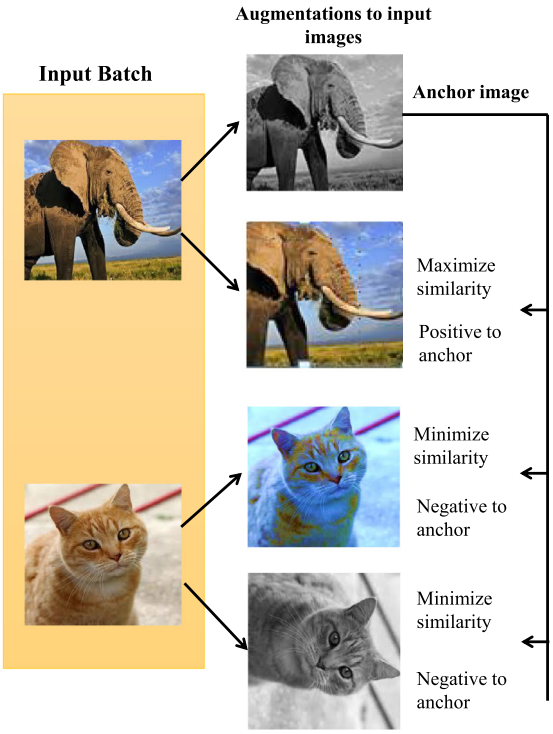


Fig. 15. Contrastive learning helps distinguish between similar and dissimilar objects using self-supervised learning. For each image in the batch, a random transform is applied to get a pair of two images that represent different instances of the same image. Contrastive self-supervised pretext task makes the representation of anchor and positive image close together and negative representations away.

images x_i and x_j of the same image x and extracts representation h_i and h_j in the form of vectors as shown in Fig. 17. The encoder can be generic and replaceable with many architectures. In many works, the authors have used ResNet-50 and its variants as the ConvNet encoder.

Nonlinear Projection Head: The representations h_i and h_j are passed through a nonlinear projection head to produce embeddings z_i and z_j on which the contrastive loss is computed as shown in Fig. 17. While some methods calculate the loss on representation h from the encoder part of the network. It is found beneficial to define the contrastive loss on z rather than h [66].

Similarity Measure: Some mechanism is required to compute the similarity between representations or embeddings of anchor-positive, anchor-negative pairs. To compute the similarity between representations z_i and z_j we have various similarity measures such as dot product, cosine similarity, or bi-linear transformations, etc. One of the most common similarity metrics used is cosine similarity that computes the amount of similarity between two representations by outputting a scalar score indicating the degree of similarity. E.g., consider an image on which two random transformations are applied to get a pair of two augmented images x_i and x_j. Each image in that pair is passed through an encoder to get representations. Then a non-linear fully connected layer is applied to get representations z. The task is to maximize the similarity between these two representations z_i and z_j. The cosine similarity is calculated on projected representations z_i and z_j which is defined in Eq. (1).

$$\text{sim}(z_i, z_j) = \frac{z_i^T . z_j}{(\tau \|z_i\| \|z_j\|)} \tag{1}$$

$\|z_i\|$ or $\|z_j\|$ represents normalized feature vector or normalized embeddings on which the loss function is computed. τ is the adjustable temperature parameter that restricts the range of similarity scores from softmax by increasing entropy that stabilizes the loss.

Loss Function : Cosine similarly and temperature parameter acts as a basis for contrastive loss function. A loss function is defined on top of the self-supervised training that penalizes the network for getting different representations for different versions of the same image. An image rotated by two different angles should have the same consistent representations even though they are rotated with different angles. The augmentations like color distortion or gaussian blur interferes and corrupts the input data and forces the network to learn from a diverse set of features. In each case, the original image and the transformed image should give the same predictions, and create the same features in intermediate representations. Hence, the loss function is minimized if the similarity between the query image and the positive embedding is more and maximized if the dissimilarity between the two is more. Based on the loss, the encoder and projection head representations improve over time and the representations obtained place similar instances of images closer in the space and negatives far apart. Widely used loss functions in the self-supervised setup is the negative contrastive estimation (NCE) loss [63], triplet loss [68], N-pair loss [69], and InfoNCE [62]. Generally, the contrastive schemes focus on comparing the embeddings with contrastive loss called "NT-Xent loss" (Normalized Temperature-Scaled Cross-Entropy Loss) that is defined in Eq. (2).

$$\ell_{ij} = -log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} exp(sim(z_i, z_k)/\tau)} \tag{2}$$

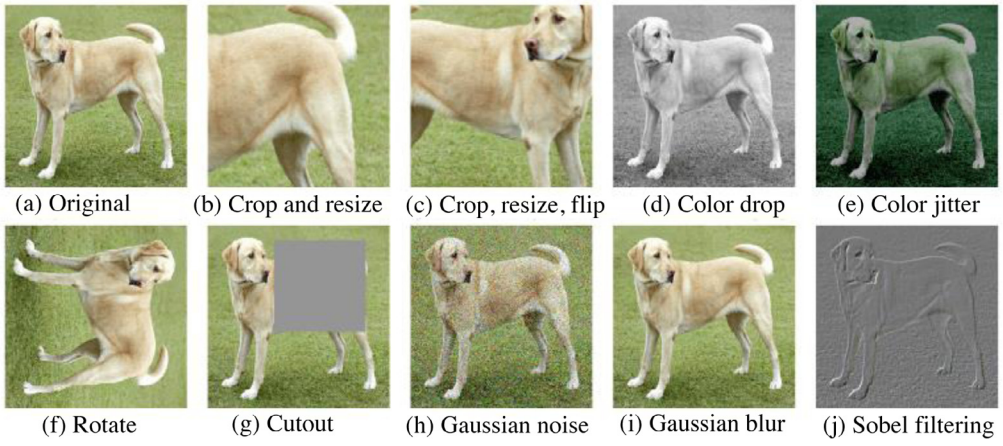


Fig. 16. Typical data augmentation methods used for visual feature learning in contrastive learning using self-supervised setup [66].

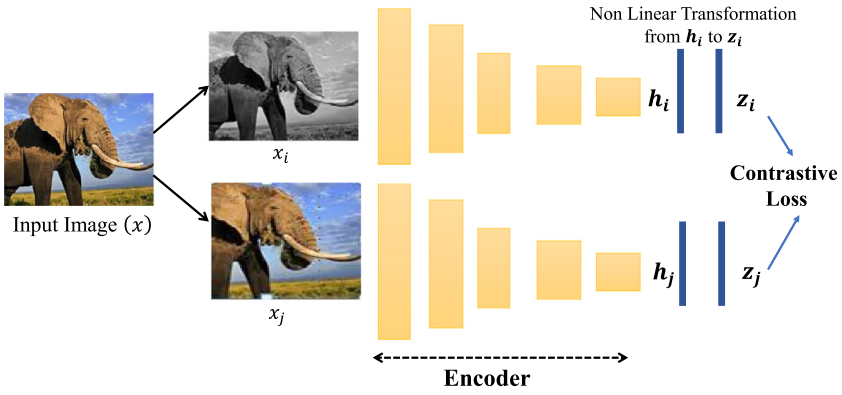


Fig. 17. Nonlinear transformation from h to z and contrastive loss computed on the representation z [66].

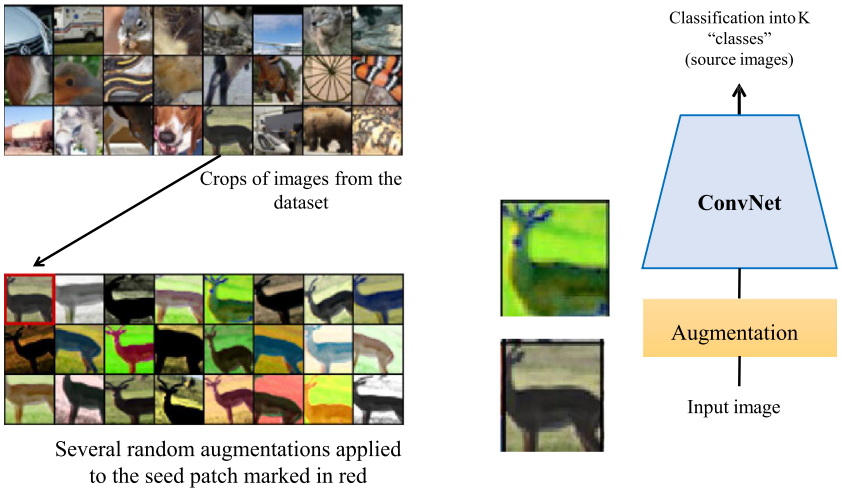


Fig. 18. Discriminating unsupervised feature learning with Exemplar Convolutional Neural Networks. Exemplary patches sampled from the unlabeled dataset (left up). Several random augmentations applied to the exemplary patch (left down). The original ("seed") patch is marked in red. Given a distorted crop from an exemplary patch, ConvNet classifies the distorted crop to one of the K surrogate classes [67].

Eq. (2) depicts the loss function for a positive pair of examples x_i and x_j and the goal is to identify positive pair of each z_i and repel others. Where N is the number of samples, $2N$ are the transformed pairs, $2(N-1)$ negative pairs, $sim(z_i, z_j)$ represents the cosine similarity as discussed in Eq. (1). The term in the numerator is the positive pairs and the terms in the denominator is the negative pairs. τ is a hyperparameter called the adjustable temperature parameter. 1 represents as "indicator function", the term $1_{[k!=i]}$ will become 1 if the condition matches else it is 0.

2.6.1. Contrastive learning prediction schemes

Exemplar ConvNets [67] is one of the earliest methods to work at the instance discrimination level. Fig. 18 shows patches of size 32×32 sampled from the unlabeled dataset of images at varying positions and are scaled to form the initial training set. One of the input patches is selected and several random transformations are applied to give rise to many patches that vary in the degree of perturbations but not in terms of content. Similarly, this process is applied to all the cropped images of the dataset. All the distorted crops from a randomly sampled 'seed' image patch form a surrogate class. Now as a self-supervised prediction task, given a distorted crop exacted from an image the ConvNet classifies it to one of the K surrogate classes as shown in Fig. 18. For the network to predict the right surrogate class the network needs to be invariant to any transformations related to geometry and color. One of the downsides of this scheme is that number of surrogate classes is equal to the number of samples in the dataset that results in a large number of classes and inhibits scalability. A revised scheme is to treat the task as a metric learning task were given the cropped image; the representation learned should be similar to the crop coming from the same image source and different from others.

Recent work on instance discrimination has raised the performance of self-supervised learning at power to supervised learning. The objective of contrastive learning is to make the embeddings of the query (anchor image) similar to the positive key and dissimilar to the negative key embeddings. Each input image is split into a query and a key formed by performing two different sets of augmentations on the image. The authors have suggested various methods for handling negatives such as: end-to-end mechanism, memory bank, and momentum encoder. In end-to-end mechanism, as shown in Fig. 19, the query (original samples) is passed through query encoder and the keys (augmented versions of positive and negative samples) are passed through the key encoder (same shared encoder for both query and keys) that produces the embeddings for both the query and the keys [70,71]. The loss is calculated over the different pairs (query-keys) and the shared encoder is updated by back-propagating through all the samples during training maintaining consistency between the queries and keys. One of the downside of the approach is that the number of negatives is limited to the size of the GPU memory as the batch size cannot be larger than the GPU memory.

The extension to the above work is to use a memory bank which is much more memory efficient than using large batch

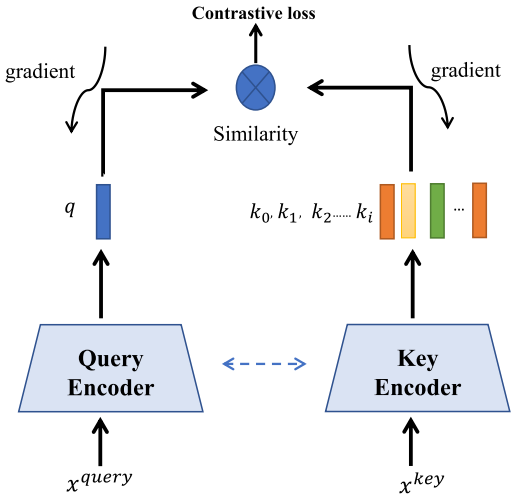


Fig. 19. End-to-end approach for contrastive learning.

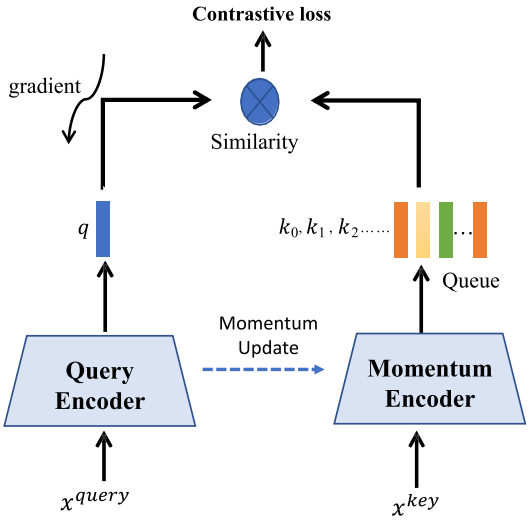


Fig. 21. Momentum Encoder for contrastive learning.

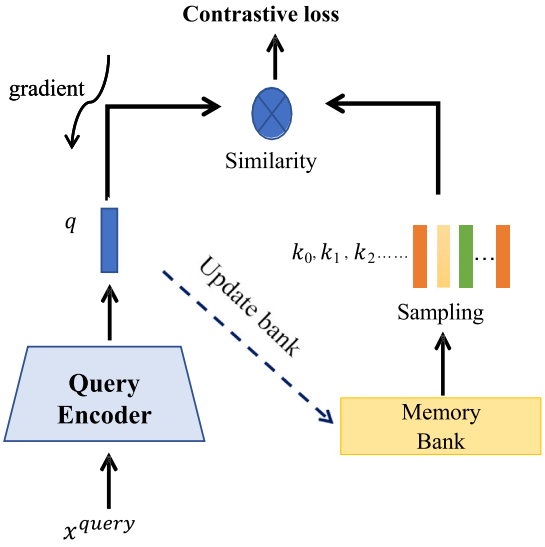


Fig. 20. Memory bank approach for contrastive learning.

sizes. The memory bank or the dictionary contains the embeddings of all the negative and positive samples as shown in Fig. 20. The query is passed through the encoder to get the embedding which is compared to the subset of keys that are randomly sampled from the memory bank. Contrastive loss is then calculated and backpropagated through the query encoder and not through the memory bank. The memory bank is updated once in a while via exponential moving average to make sure the memory bank slowly updates itself and is in sink with the query representation. The advantage of the scheme is that we can have many negative but, on the downside, the keys get obsolete with corresponding queries as the memory bank is not updated frequently [71,72].

To address the issues faced by memory bank, another scheme is proposed called Momentum Contrast (MoCO v1) that relies on a memory bank but uses a different approach for updating it [71]. Each input image is split into a query and a key formed by performing two different sets of augmentations. Fig. 21 shows the momentum encoder, the query is passed through the encoder and all the keys are passed through the momentum encoder to produce the embeddings. A similarity measure takes these embeddings and measures the similarity between the pairs (query-key). Contrastive loss is then calculated and backpropagated through the query encoder and the parameters of the momentum encoder are updated using momentum update with the new weights of the query encoder at every iteration. The momentum encoder relies on the memory bank but with a different update scheme that is slowly pursuing via exponential moving average (momentum update). This prevents the outdated keys and queries from being collected and the scheme becomes memory efficient. Another important feature of MoCo is the queue that follows the first-in-first-out scheme, that is to say, the older key representations are discarded as compared to the new key embeddings as the training proceeds. MoCo outperforms the end-to-end and memory bank approach by having more consistent and updated key representations and also decouples the batch size from the negatives. Further, MoCo v1 was enhanced to MoCo v2 [73] by adding MLP head, adding more argumentations similar to SimCLR [71] and cosine learning rate. The results also show that MoCo v2 largely closes the gap between unsupervised and supervised representation learning in many image recognition tasks and can serve as an alternative to ImageNet supervised pre-training in several applications.

A very popular scheme based on end-to-end approach is a simple framework for contrastive learning of visual representations (SimCLR) [66] that has served as a base for many recent contrastive learning schemes. SimCLR adopts contrastive learning that attempts to attract different augmented views of the same image and repel augmented views coming from other images. Fig. 22 shows an input image x on which two separate sets of augmentations are applied resulting in two correlated views of the same image $\tilde{x}_i$ and $\tilde{x}_j$ (positive pair). The transformation applied can be a combination of random cropping, random color distortion, and gaussian blur. All other pairs in the batch are sampled as dissimilar images (negatives) to the positive pair. Each correlated view of the same image is then passed through individual ResNet-50 encoder $f(.)$ to get representations h_i and h_j. The two representations are then passed through an MLP based nonlinear projection head $g(.)$ resulting in a lower dimension representations z_i and z_j. Next, the similarity between two correlated versions of an image is calculated using cosine similarity on representations z_i and z_j. Ideally, the similarities between augmented images of the same object will be high while the similarity between different objects will be lower. Finally, the loss function for the contrastive learning objective is calculated, which helps to identify the invariant features of each input image and maximize the ability of the network to identify different transformations

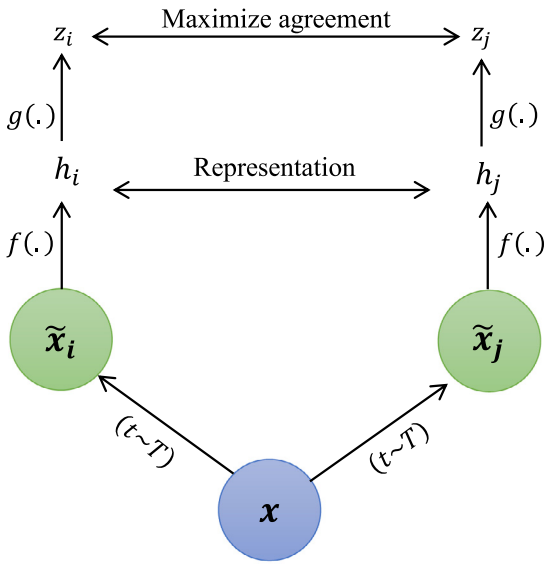


Fig. 22. A simple framework for contrastive learning of visual representations. Two separate data augmentation are applied to the input image x to obtain two correlated views $\tilde{x}_i$ and $\tilde{x}_j$. A base encoder network $f(.)$ and a projection head $g(.)$ are trained to maximize agreement using a contrastive loss. After training is completed, the projection head $g(.)$ is thrown away and encoder $f(.)$ and representation h are used for downstream tasks [66].

of the same image. SimCLR uses a contrastive loss called "NT-Xent loss" (Normalized Temperature-Scaled Cross-Entropy Loss) for the augmented pairs in the batch that are taken one by one and the softmax function is applied to get the probability of the two images being similar. SimCLR improves upon the previous state-of-the-art self-supervised learning methods but also beats the supervised learning method on ImageNet classification by incorporating nonlinear projection head over representations h_i and h_j and implementing strong data argumentation techniques. The representation before the non linear projection head is used for downstream tasks. The model is trained on varying batch sizes from 256 to 8192, a batch size of 8192 is trained for a max of 100 epochs. The optimizer used is LARS with a learning rate of 4.8 (=0.3 $\times$ BatchSize/256) and weight decay of 10^{-6}. The linear warmup is used for the first 10 epochs, and then the learning rate is decayed with the cosine decay schedule. One of the limitations of SimCLR is that the numbers of negatives are limited by the batch size. However, the performance becomes better with large batch sizes and a higher number of training epochs. The authors of simCLR extended the work to simCLR v2, where the dataset has large amount of unlabeled data and very little labeled data. The unlabeled data is used to pre-train a large model (teacher model) in an unsupervised way. Next, the model is fine-tuned on a small subset of data that is labeled in a supervised fashion. Lastly, distillation or self-learning is done with unlabeled examples for refining and transferring the task-specific knowledge to a smaller network also called a student network [74]. Using distillation, the large network can be distilled back to a smaller ResNet50 network by retaining almost the same accuracy as that of the larger model. The distillation is not just on a labeled dataset but also uses the labels produced by the teacher model over the entire unlabeled dataset to train the smaller network or the teacher network.

Another recent work on contrastive learning is the Pretext-Invariant Representations learning (PIRL) that focuses on learning representations that are invariant to the pretext tasks using a memory bank of negatives [75]. In conventional pretext tasks given an image I on which a transformation t is applied (eg. rotation or jigsaw puzzle), the ConvNet predicts the property of the transform applied to the image. Under this setup, it is observed that the last layers of the ConvNet include low-level information of the transform, so if the ConvNet received a rotated image, the last layer features will change dramatically according to the pretext task applied. As a result, the network lands up learning less semantic features at the higher layers which do not transfer well to downstream tasks. Whereas what is expected is that the representations learned are invariant to these transformations that transfer well to the downstream tasks. Instinctively this makes sense because even if an image is divided into patches and shuffled, it does not change the visual semantics of the image. Motivated by the idea of generating representations that are invariant to transformations, PIRL is proposed as shown in Fig. 23. The input to the ConvNet (ResNet-50) is a pair of original image I and transformed image I^t. Each pair is passed through the shared ConvNet to produce feature embedding and are finally sent to the linear projection to produce the representation of the original image and its corresponding transformed image. The image I and any pretext transformed version of this image I^t are related samples and any other sample is unrelated sample. Hence by training the network like this, representations contain very little information about the transform t. Finally, a loss function (Noise Contrastive Estimator) is added that penalizes for getting different representations for the positive pair. The loss function puts the embedding of related images close and pushes away the embedding of unrelated or random images. Once the model is trained, the linear projection heads are removed and the encoder is used for downstream tasks. The network is trained using mini-batch SGD using an initial learning rate of 1.2×10^{-1} and a final learning rate of 1.2×10^{-4} with a cosine learning rate decay [76]. The network is trained for 800 epochs using a batch size of 1024 images. The important thing that has made contrastive learning work so well is the availability of large number of negatives. PIRL uses a memory bank containing a moving average of learned representations of all the original images that enable a large number of negatives used during training. PIRL is one of the recent works that learns good visual representations of images irrespective of the pretext task used.

Contrastive schemes are computationally expensive as they push away the representations that come from different images while pull the representations that come from different views of the same image. Computing all the pairwise comparisons on a large dataset is intractable and requires a lot of computation. Hence, most contrastive schemes rely on approximation and take only a subset of examples for comparison. Moreover, in instance-based learning, every sample is treated as its own class. This makes it unreliable in conditions where it compares an input sample against other samples coming from the same class.

2.7. Clustering based schemes

Clustering is another scheme for learning representation using self-supervised learning. The clustering schemes extract representations from the input images using a feature extractor and club semantically similar image features together. The model is then trained on these cluster assignments which serve as pseudo labels. Some of the clustering based approaches are DeepCluster [78], and SeLA [79]. Most of the clustering approaches are offline, which means they require at least one forward pass of the entire dataset to calculate the cluster assignment. Hence this becomes computationally expensive for large datasets. On the other hand, contrastive schemes based on noise contrastive estimation generally operate by comparing different pairs of images and then calculating a contrastive loss which again becomes computationally expensive. Addressing the challenges, recently a new clustering-based self-supervised based approach for image representation learning has attracted attention that combines online

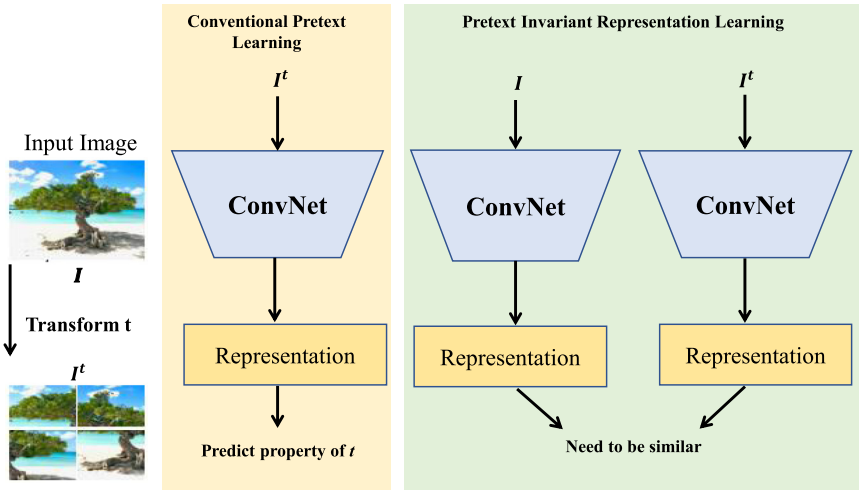


Fig. 23. Pretext-Invariant Representation Learning (PIRL). Given an input image I, a pretext task t of rotation and jigsaw is applied to give the transformed image I^t. I and I^t are sent to the shared ConvNet resulting in the feature embeddings. The network learns representations that are invariant to the transformation t and retains semantic information by keeping the representations of the image I and its transformed counterpart I^t close together and distancing from others [75].

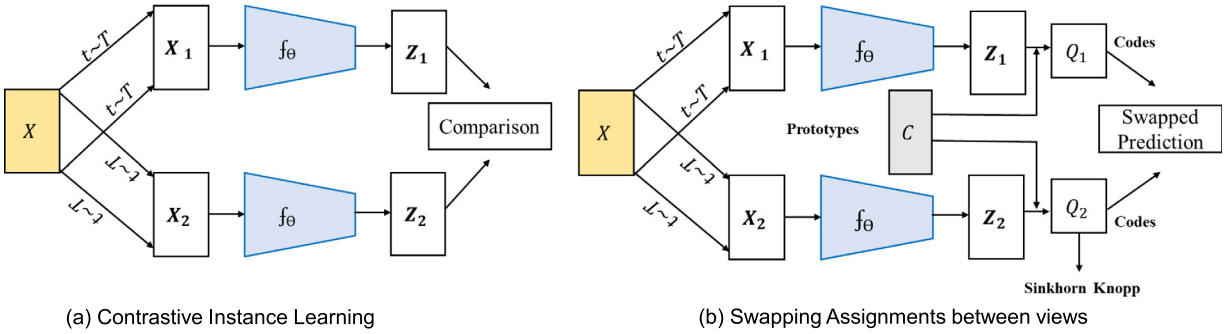


Fig. 24. Contrastive instance learning (left) vs. SwAV (right). Swapping Assignments between Views (SwAV), first the codes are obtained by assigning features to prototype vectors. Then to solve a "swapped" prediction problem, the codes from one data augmented view is predicted using the other view. Thus, SwAV does not directly compare image features unlike contrastive instance learning. Prototype vectors are learned along with the ConvNet parameters by backpropagation [77].

clustering mechanism with contrastive learning. The researchers have proposed a self-supervised approach to learn features by Swapping Assignments between multiple Views of the same image (SwAV) [77,80]. It uses an online clustering mechanism to learn better representations by grouping similar features together by comparing representations with cluster centroids. The objective is not only to make the positive pairs of samples close to each other but also, to make sure that all other features that are similar to each other club together. As a result, negative comparison with all the images lying in the large mini-batch is prevented thereby leading to reduced computational overhead. For example, in a feature space, the features of sheep should be closer to the features of goats (as both are animals) but should be far from the features of cars. Fig. 24 shows SwAV framework, the ResNet-50 network receives different augmented views of the same image (views can be more than two) generating the embedding. This embedding vector then goes to a shallow non-linear network f_θ that produces a projection vector denoted by Z. The representation or features generated are not directly compared to each other, like in contrastive learning. Rather mapping of features is done to their nearest neighbor in a set of K trainable prototype vectors ($C = [c1, \ldots, c_K]$). This method then maps feature encoding of the augmented views of images into discrete codebook C containing the set of prototype vectors or clusters $[c1, \ldots, c_K]$ and use it to look up to the codes that are most similar to the features. Then as a swapped prediction problem, the code Q of the view of an image is predicted from the representation Z of another view of the same image. The online clustering problem is treated as an optimal transport problem using Sinkhorn–Knopp algorithm that enforces an equipartition constraint of assigning samples to clusters [81].

SwAV uses a different augmentation strategy as opposed to SimCLR and MoCo. It uses multi-scale cropping and creates multiple views of the single image. In Multi-crop augmentation, full high-resolution images (eg: 224 × 224) from the dataset is taken to generate standard or high-resolution cropped images (representing global views of the image) and then additional low-resolution images (eg: 96 × 96) is sampled along with the global views. Fig. 25 shows the setup off swapped prediction problem wherein z_s and z_t are the representations of the two views of the image and q_s and q_t are the respective codes generated. Now as a swapped prediction problem, the code of the image is predicted from another view of the same image. The goal is to minimize the cross-entropy loss between the two views of the same image. If two different views of the same image contain similar information, then it should be possible to predict its code from one or the other feature. Recently, a self-supervised method based on SwAV is proposed called SEER that works with high dimensional complex data. The model is pre-trained on billions of random, unlabeled and uncurated public Instagram images, and is fine-tuned on ImageNet in a supervised fashion. SEER outperforms the state-of-the-art self-supervised models, attaining 84.2 percent top-1 accuracy on ImageNet [82].

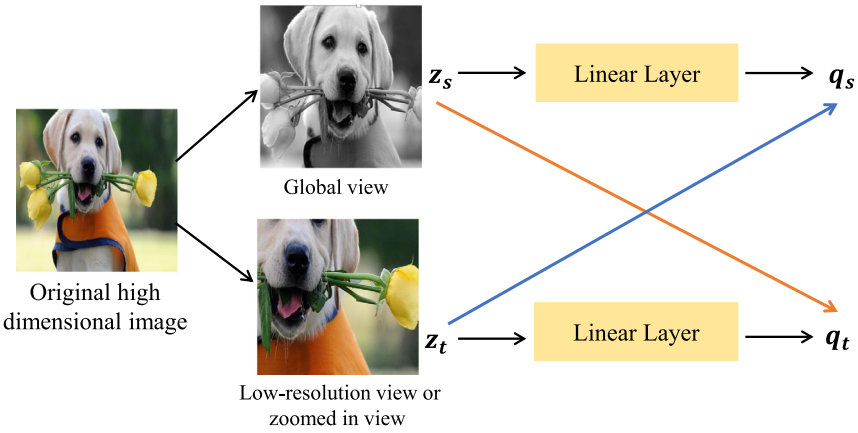


Fig. 25. Setting up the swapped prediction problem between two separate views of the same image [77].

2.7.1. Training in contrastive schemes

Contrastive learning employs a variety of optimization algorithms for effective training of the model [83]. The training of ConvNet involves learning the parameters in a way that minimizes contrastive loss that serves as an unsupervised objective function. Many contrastive schemes such as [72,73,75,84] have been trained using mini-batch Stochastic Gradient Descent (SGD). One of the most important hyperparameters for SGD is the learning rate which in practice should gradually be decreased over time to prevent the overshooting of the objective. But it takes a lot of time to navigate a gentle slope. Hence, we need to add momentum to the gradient descent which is used in most deep learning approaches.

Another popular optimization method known as gradient descent with adaptive learning rate (Adam) [85] has been used in a few methods [62,86,87]. Adam is a combination of RMSProp [85] and momentum which incorporates the first-order momentum of the gradient term. Both first and second moments are corrected to bias to account for their initialization at zero.

For large batch size training as in end-to-end contrastive learning schemes [66,73,77] a standard SGD based optimizer and using large learning rates becomes highly unstable. It results in lower model performance and training may diverge. To stabilize the training, Layer-wise Adaptive Rate Scaling (LARS) [88] optimizer is used along with cosine decay schedule [76]. LARS uses a different learning rate for each layer and not for each weight. And the magnitude of the update is controlled with respect to the weight norm for stabilizing the training speed. The LARS optimizer is initialized with the learning rate and the LARS coefficient η defines how much we trust the layer to change its weights during one update. Once the LARS optimizer is defined, a scheduler is initialized with an initial warm up learning rate for few initial warm up epochs till the learning rate is gradually increased to the target learning rate. After the warm-up period, the learning rate is decayed with the cosine decay schedule without restarts.

3. Performance comparisons

Recently, there is a rapid surge in self-supervised learning methods for computer vision tasks that have started to outperform supervised learning methods. In this section, we evaluate self-supervised methods on various standardized datasets and a variety of downstream tasks. Most of the time the model is pre-trained on a large unlabeled dataset such as ImageNet on a pretext task and fine-tuned on a smaller labeled dataset. Table 2 shows the linear classification top-1 accuracy on top of the features learned by the network pre-trained on VOC7 without labels.

Table 2
Linear classification top-1 accuracy on top of features learned using self-supervised approach pre-trained on VOC7 without labels and object detection with fine-tuned features on VOC7+12 using Faster-CNN. The backbone architecture used is AlexNet having 61M parameters and ResNet50 having 25.6M parameters.

Method	Architecture	Parameters	Classification	Detection
Supervised	AlexNet	61M	79.9	56.8
Supervised	ResNet50	25.6M	87.5	81.3
Inpaint [1]	AlexNet	61M	56.5	44.5
Color [44]	AlexNet	61M	65.6	46.9
BiGAN [55]	AlexNet	61M	60.1	46.9
Context [60]	AlexNet	61M	65.3	51.1
DeepCluster [80]	AlexNet	61M	72	55.4
Rotation [2]	ResNet50	25.6M	63.9	72.5
Jigsaw [61]	ResNet50	25.6M	64.5	75.1
LA [89]	ResNet50	25.6M	69.1	–
NPID [72]	ResNet50	25.6M	76.6	79.1
PIRL [75]	ResNet50	25.6M	81.1	80.7
MoCo [71]	ResNet50	25.6M	–	81.4
SwAV [77]	ResNet50	25.6M	88.9	82.6

Similarly, for object detection, the pre-trained model is fine-tuned on VOC7+12 using Faster-RCNN. We see that the performance of contrastive schemes such as MoCo, PIRL, SwAV outperforms in comparison to other self-supervised models.

Fig. 26 depicts ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pre-trained on ImageNet). The performances of self-supervised schemes have shown exceptional results and are heading towards supervised learning. SimCLR achieves the same classification accuracy as ResNet-50 trained on supervised learning at the cost of increased width of ResNet-50 or increased parameters. SimCLR performs well due to the large batch of negative examples, output projection head, stronger data augmentation, and longer training time. But the performance gains are even more for SwAV than SimCLR which are attributed to factors such as multi-scale cropping, generating multiple views of a single image, and adopting an online clustering mechanism to group similar features. Networks having large parameters do give higher linear evaluation accuracy than training on ResNet50 and shrinks the gap with supervised training. However, large models are tough to handle and also computationally expensive, hence researchers are coming up with ways where a large trained model is distilled back to standard ResNet50 after it is pre-trained on a large subset of unlabeled data and finetuned on the small subset of data with labels [74]. On the architectural design front, a recent work called RegNets is proposed which is a new family of ConvNets that is capable of scaling to billions of parameters and can be optimized to fit different runtime and memory constraints [90].

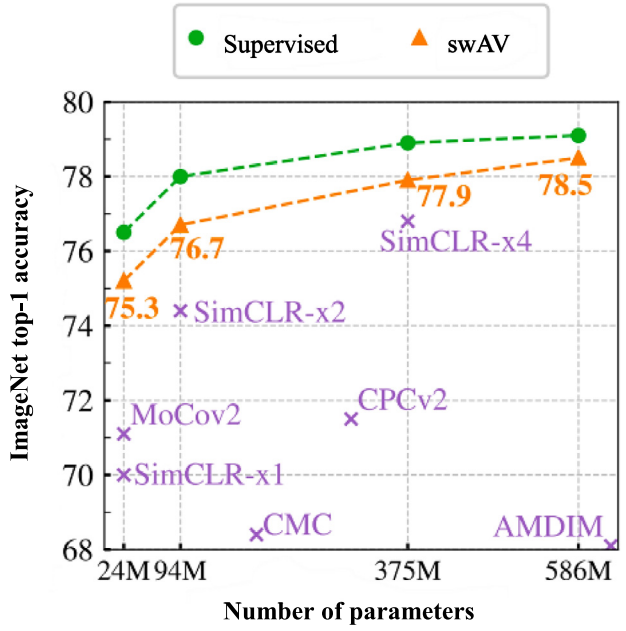


Fig. 26. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised schemes. The pre-trained model is trained on ImageNet dataset without labels. Results show further gains in the accuracy as the number of parameters are increased beyond 24*M* by increasing the width of ResNet with a factor × 2, × 4, and × 5 for both SimCLR and SwAV. However, SwAV outperforms SimCLR with an increased number of parameters [77].

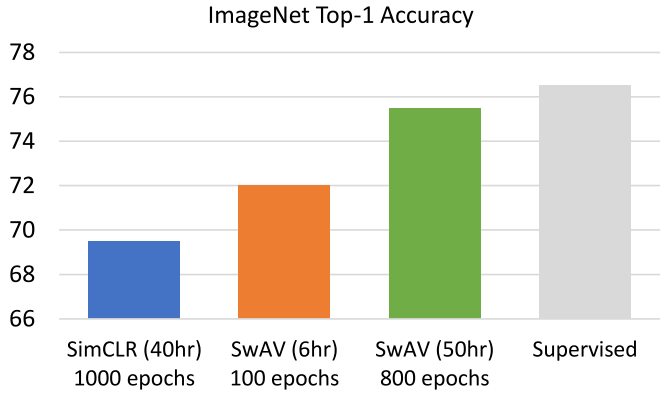


Fig. 27. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with SwAV and SimCLR. The results show SwAV converges fast in a smaller number of epochs in comparison to SimCLR and reaches a performance of 75% when number of epochs are increased to 800.

Fig. 27 depicts ImageNet classification accuracy for self-supervised learning schemes. ResNet50 is pre-trained on ImageNet without labels. The network is frozen and a linear classifier is trained on top of the network. The gray bar depicts the performance of supervised pre-training of ImageNet and the blue bar indicates the performance of SimCLR after 1000 epochs of training, that last for 40 h on 64 GPUs. The orange bar indicates the performance of swAV that converges fast and reaches better performance with 100 epochs which lasts for 6 h. Longer training of SwAV for 800 epochs lasting 50 h results in 76% top one accuracy on ImageNet as shown in the green bar.

Table 3 shows the progression of top 1 accuracy (classification) on ImageNet dataset using contrastive self-supervised learning. The ResNet-50 ConvNet learns a linear classifier on top of frozen representation trained on ImageNet without labels. The performance of SwAV outperforms all other self-supervised approaches and is only 1% away from supervised learning on the ImageNet classification task. The self-supervised methods show further increase in performance as the number of parameters increase.

Table 3
Linear classification on ImageNet. Top-1 accuracy on ImageNet dataset trained on frozen features from different self-supervised methods with a standard ResNet-50 containing 24*M* parameters.

Method	Arch.	Parameters	Top1
Supervised	R50	24	76.5
Colorization [44]	R50	24	39.6
Jigsaw [61]	R50	24	45.7
NPID [72]	R50	24	54
BigBiGAN [46]	R50	24	56.6
LA [89]	R50	24	58.8
NPID++ [75]	R50	24	59
MoCo [73]	R50	24	60.6
SeLa [91]	R50	24	61.5
PIRL [75]	R50	24	63.6
CPC [62]	R50	24	63.8
PCL [62]	R50	24	65.9
SimCLR [66]	R50	24	70
MoCov2 [73]	R50	24	71.1
SwAV [77]	R50	24	75.3

Table 4
Linear classification performance on four datasets (ImageNet, VOC07, Places205, iNaturalist datasets) using the setup [92]. The linear classifiers are trained on image representations generated by the ConvNet trained on ImageNet (without labels) using self-supervised approach. Numbers with † are measured using 10-crop evaluation.

Method	Parameters	Transfer dataset			
		ImageNet	VOC07	Places205	iNat.
ResNet-50 using evaluation setup of [92]					
Supervised	25.6M	75.9	87.5	51.5	45.4
Colorization [92]	25.6M	39.6	55.6	37.5	–
Rotation [2]	25.6M	48.9	63.9	41.4	23
NPID++ [72]	25.6M	59	76.6	46.4	32.4
MoCo [71]	25.6M	60.6	–	–	–
Jigsaw [61]	25.6M	45.7	64.5	41.2	21.3
PIRL [75]	25.6M	63.6	81.1	49.8	34.1
SwAV [77]	25.6M	75.3	88.9	56.7	48.6
Different architecture or evaluation setup					
NPID [72]	25.6M	54	–	45.5	–
BigBiGAN [46]	25.6M	56.6	–	–	–
AET [64]	61M	40.6	–	37.1	–
DeepCluster [80]	61M	39.8	–	37.5	–
Rot. [34]	61M	54	–	45.5	–
LA [89]	25.6M	60.2†	–	50.2†	–
CMC [93]	51M	64.1	–	–	–
CPC [62]	44.5M	48.7	–	–	–
CPC-Huge [94]	305M	61	–	–	–
BigBiGAN-Big [46]	86M	61.3	–	–	–
AMDIM [95]	670M	68.1	–	55.1	–

Table 4 shows the results of transfer learning from learned representation on Image-classification task on four datasets (ImageNet, VOC07, Places205, and iNaturalist datasets) using the setup of [92]. Linear classifiers are trained on the representation learned obtained by self-supervised models pre-trained on ImageNet without labels. SwAV substantially outperforms their covariant counterparts and produces comparable accuracy to the state-of-the-art supervised model on linear classification on various datasets.

Table 5 shows the ablation of SwAV, MoCo v2, and SimCLR, and PIRL which are all trained under the same setup. MoCo v2 included certain details that were a part SimCLR such as the MLP head, color distortion and Gaussian blur augmentation (agu+), and cosine learning decay (cos). It was found that the performance of MoCo v2 increased to 67.5% from 60.6%. Further, an

improvement of 3.5% was seen by increasing the training time from 200 to 800 epochs. However, SwAV achieves the state-of-the-art performance when trained in the small-batch setting, with fewer epochs, and by using multi-crop argumentation. The authors of SwAV suggest that multi-crop augmentation strategy is generic and can be implemented in various contrastive learning schemes, to further enhance the performance on downstream tasks.

4. Practical consideration

To implement self-supervised techniques, we need to consider some practical considerations that will enable the learning of good representations. In self-supervision, a pretext task holds a very important part that is intended for the network to solve though it is not the primary task. Nevertheless, it is done to learn rich image representations that benefit the downstream tasks and enable the network to show high performance on a limited labeled dataset. However, the pretext task should be chosen wisely and in conjunction with the downstream task. Although numerous pretext tasks have been proposed in contrastive learning, still research is going on to identify the right pretext task for a given problem. Another consideration is to prevent the network from cheating, as often networks "cheat" and find an easy way to solve the pretext task hence shortcut prevention is essential. Many schemes like jigsaw and context prediction form a grid of non-overlapping patches that prevent the network from learning via boundary pixels, also the color channels are jittered to prevent easy learning [1,61]. Also, a well-designed and strong data augmentation and type of network architecture used for self-supervised learning makes a big difference in the kind of results in which we are interested. Impact of strong data augmentation on self-supervised schemes has been thoroughly studied recently in SimCLR [66] and SwAV [77]. Fig. 28 shows linear evaluation (ImageNet top-1 accuracy) under single augmentation as well as on composition of data augmentations. Applying single augmentation results in low performance than applying composition of augmentations. E.g., applying crop and then color augmentation leads to 56.3% accuracy than just applying color augmentation. This insight was laid in SimCLR paper, where the authors stated the color distortion is very important for contrastive learning. If you have two crops of the same image, the intensity histogram may be similar for both the crops as a result color distortion becomes the second important transformation that makes the task harder for the ConvNet, and improves the quality of the representation. A well-tuned augmentation strategy can lead to substantial information gains as seen in SimCLR and SwAV.

Scaling of batch size and training steps has also been found to impact the performance of self-supervised models. As the training epochs increases the accuracy of the models tends to increase for self-supervised learning methods as shown in Fig. 29. Self-supervised learning benefits more from scaling up the training and enhancing argumentations than in supervised training.

4.1. Open challenges

Self-supervised learning methods have achieved great success and are obtaining good performance that is close to supervised models on image recognition tasks. However, there are certain challenges faced by self-supervision which are as follows:

Resource intensive and requires careful attention to detailing: Contrastive learning requires long batch training time as well as complex resource setups like many TPUs [96]. For example, SimCLR uses 128 TPUs for large batch training during self-supervised learning. Also, careful attention has to be paid to detailing like data augmentation and pretext tasks. As

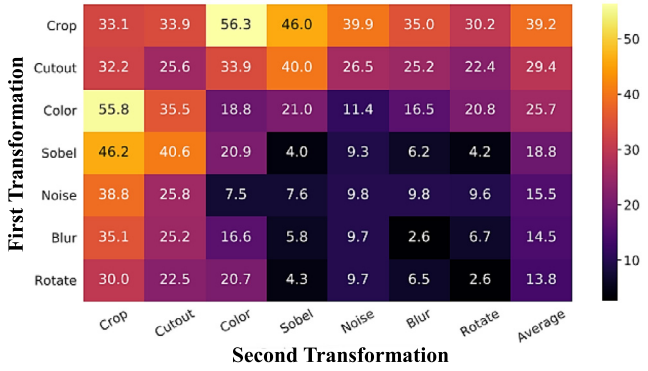


Fig. 28. Linear evaluation (ImageNet top-1 accuracy) on pre-trained network trained on ImageNet without labels under single and composition of data augmentations [66].

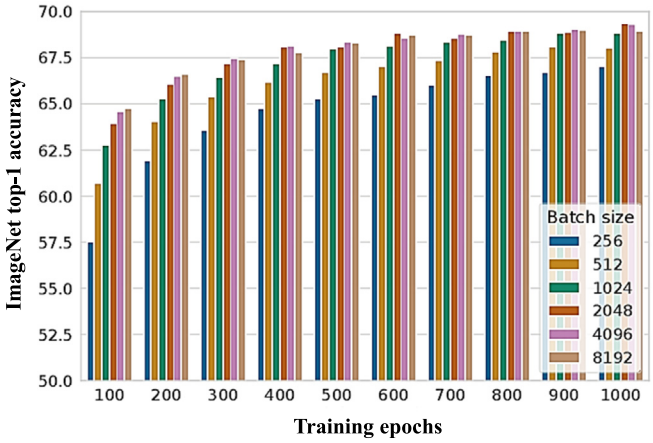


Fig. 29. Top 1 ImageNet accuracy on ResNet-50 trained with different batch size and training steps. Each bar is a single run from scratch [66].

the focus of contrastive learning is instance discrimination a lot many negative and positive pairs have to be generated and hence the selection of appropriate sample selection strategy should be done [97]. Recent work called "Bootstrap Your Own Latent (BYOL)" proposed a new approach to self-supervised learning where the authors considered the reconstruction of features instead of the inputs. It claims to perform better than state-of-the-art contrastive methods by avoiding large number of negative pairs [98].

Beyond Contrastive learning: In another work, the authors claim that the starting layers of the ConvNet learn useful representation by only a single high-resolution image provided sufficient data augmentation is used. However, at deep layers gap with manual supervision cannot be closed even if millions of unlabeled images are used for training. The proposed scheme uses a high-resolution image and generates 1 million augmented crops on which the model is trained. It is observed that representations learned from the first few layers are of the same quality as in the case of representations learned when a linear classifier is trained with supervised and unsupervised learning on millions of images [99].

Learning beyond a single object in an image: Most of the self-supervised pre-trained models are trained using images that have a single dominant object like in the case of ImageNet dataset. Whereas in applications like self-driving cars the scene contains multiple objects and distinguishing between two similar scenes is quite a challenging task [100].

Table 5
Top-1 linear classifier accuracy on ImageNet on top of frozen features from a ResNet-50 pre-trained using self-supervised learning approach. The performance of the self-supervised models is studied as the function of MLP head, augmentation (Multi-crop), cosine learning decay cos, number of epochs, and batch size. $2 \times 160 + 4 \times 96$ indicates 2 crops of size 160×60 and 4 crops of size 96×96.

Method	Unsupervised pre-training					
	MLP	Multi-crop	cos	epochs	batch	ImageNet top-1 accuracy
MoCo v2	✓	2×224	✓	200	256	67.5
SimCLR	✓	2×224	✓	200	256	61.9
SimCLR	✓	2×224	✓	200	8192	66.6
SwAV	✓	$2 \times 160 + 4 \times 96$	✓	200	256	72.0
SwAV	✓	$2 \times 224 + 6 \times 96$	✓	200	256	72.7
	Results of longer unsupervised pre-training					
MoCo v2	✓	2×224	✓	800	256	71.1
SimCLR	✓	2×224	✓	1000	4096	69.3
PIRL	✓	2×224	✓	800	1024	63.6
SwAV	✓	$2 \times 224 + 6 \times 96$	✓	400	256	74.3

Learning beyond structured images: satellite images and medical images (microscopic images) have no or very less structure to exploit as a result it becomes difficult to find context in them. Hence schemes like relative patch prediction or jigsaw puzzle are inefficient to deal with such images [101].

Dataset biases: In self-supervised learning task, the data itself provides strong supervision to solve the pretext tasks. As a result, the feature representations learned using self-supervised objectives are influenced by the data on which the model is trained. Such biases are hard to minimize with the increase in the size of the datasets.

Augmentation strategy for new domains: Constructing a pre-trained model with datasets containing medical and satellite images, may demand a different augmentation strategy as compared to what we are using with natural image dataset (ImageNet).

5. Conclusions

Self-supervised methods have shown results at power with supervised learning by leveraging the huge amount of unlabeled data that is available for free. The effectiveness of self-supervised learning techniques has been found in complex downstream tasks such as image classification, object detection, image segmentation, etc., where limited labeled data is available. The unlabeled data can be utilized which is available for free and present in abundance for building effective pre-trained models. The greatest benefits of pre-training are currently in low data regimes where limited annotation data is available. The paper has done an extensive review on various handcrafted pretext tasks as well as various self-supervised methods that follow contrastive approach or instance discrimination. The paper also highlights the state-of-the-art self-supervised methods that are showing significant results in comparison to supervised learning. Finally, this work concludes by discussing some practical considerations and open challenges in image recognition tasks using self-supervised learning.

CRediT authorship contribution statement

Kriti Ohri: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing. **Mukesh Kumar:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

All authors approved the version of the manuscript to be published.

References

[1] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, Alexei A Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.
[2] Spyros Gidaris, Praveer Singh, Nikos Komodakis, Unsupervised representation learning by predicting image rotations, 2018, arXiv preprint arXiv:1803.07728.
[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[4] Jeremy Howard, Sebastian Ruder, Universal language model fine-tuning for text classification, 2018, arXiv preprint arXiv:1801.06146.
[5] Yoav Goldberg, Omer Levy, Word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method, 2014, arXiv preprint arXiv:1402.3722.
[6] Jeffrey Pennington, Richard Socher, Christopher D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[7] Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, Bag of tricks for efficient text classification, 2016, arXiv preprint arXiv:1607.01759.
[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, arXiv preprint arXiv:1907.11692.
[9] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov, Unsupervised cross-lingual representation learning at scale, 2019, arXiv preprint arXiv:1911.02116.
[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2019, arXiv preprint arXiv:1910.10683.
[11] Linchao Zhu, Yi Yang, Actbert: Learning global-local video-text representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8746–8755.
[12] Yoshua Bengio, Aaron Courville, Pascal Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[13] A. Emin Orhan, Vaibhav V. Gupta, Brenden M. Lake, Self-supervised learning through the eyes of a child, 2020, arXiv e-prints, arXiv–2007.
[14] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[15] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, Kilian Q Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.
[21] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al., Openimages: A public dataset for large-scale multi-label and multi-class image classification, 2017, Dataset available from https://github.com/openimages, 2(3), 2-3.
[22] Ameet V. Joshi, Amazon's machine learning toolkit: Sagemaker, in: Machine Learning and Artificial Intelligence, Springer, 2020, pp. 233–243.
[23] Joao Carreira, Andrew Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.
[24] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, Michael S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.
[25] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, Samy Bengio, Transfusion: Understanding transfer learning for medical imaging, in: Advances in Neural Information Processing Systems, 2019, pp. 3347–3357.
[26] Olivier Chapelle, Bernhard Scholkopf, Alexander Zien, Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews], IEEE Trans. Neural Netw. 20 (3) (2009) 542–542.
[27] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, Dhruv Mahajan, Billion-scale semi-supervised learning for image classification, 2019, arXiv preprint arXiv:1905.00546.
[28] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, Laurens van der Maaten, Exploring the limits of weakly supervised pretraining, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, 181–196.
[29] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, Zeynep Akata, Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly, IEEE Trans. Pattern Anal. Mach. Intell. 41 (9) (2018) 2251–2265.
[30] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, Jia-Bin Huang, A closer look at few-shot classification, 2019, arXiv preprint arXiv:1904.04232.
[31] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, Bernt Schiele, Meta-transfer learning for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 403–412.
[32] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, Richard S. Zemel, Meta-learning for semi-supervised few-shot classification, 2018, arXiv preprint arXiv:1803.00676.
[33] Linchao Zhu, Yi Yang, Label independent memory for semi-supervised few-shot video classification, IEEE Ann. Hist. Comput. (01) (2020) 1–1.
[34] Alexander Kolesnikov, Xiaohua Zhai, Lucas Beyer, Revisiting self-supervised visual representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, 1920–1929.
[35] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Texture and art with deep neural networks, Curr. Opin. Neurobiol. 46 (2017) 178–186.
[36] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, Wieland Brendel, Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, 2018, arXiv preprint arXiv:1811.12231.
[37] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al., A large-scale study of representation learning with the visual task adaptation benchmark, 2019, arXiv preprint arXiv:1910.04867.
[38] Jiirgen Schmidhuber, Making the world differentiable: On using self-supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environm nts, 1990.
[39] Longlong Jing, Yingli Tian, Self-supervised visual feature learning with deep neural networks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2020).
[40] Dumitru Erhan, Aaron Courville, Yoshua Bengio, Pascal Vincent, Why does unsupervised pre-training help deep learning? in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, 201–208.
[41] Yoshua Bengio, Learning deep architectures for AI, Now Publishers Inc, 2009.
[42] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, Pierre-Antoine Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 1096–1103.
[43] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, Matthieu Cord, Learning representations by predicting bags of visual words, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6928–6938.
[44] Richard Zhang, Phillip Isola, Alexei A. Efros, Colorful image colorization, in: European Conference on Computer Vision, Springer, 2016, pp. 649–666.
[45] Richard Zhang, Phillip Isola, Alexei A. Efros, Split-brain autoencoders: Unsupervised learning by cross-channel prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1058–1067.
[46] Jeff Donahue, Karen Simonyan, Large scale adversarial representation learning, 2019, arXiv preprint arXiv:1907.02544.
[47] Carl Doersch, Tutorial on variational autoencoders, 2016, arXiv preprint arXiv:1606.05908.
[48] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, 2014, arXiv preprint arXiv:1406.2661.
[49] Alec Radford, Luke Metz, Soumith Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015, arXiv preprint arXiv:1511.06434.
[50] Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 214–223.
[51] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, Improved training of wasserstein gans, 2017, arXiv preprint arXiv:1704.00028.
[52] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, Progressive growing of gans for improved quality, stability, and variation, 2017, arXiv preprint arXiv:1710.10196.
[53] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida, Spectral normalization for generative adversarial networks, 2018, arXiv preprint arXiv:1802.05957.
[54] Han Zhang, Ian Goodfellow, Dimitris Metaxas, Augustus Odena, Self-attention generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7354–7363.
[55] Jeff Donahue, Philipp Krähenbühl, Trevor Darrell, Adversarial feature learning, 2016, arXiv preprint arXiv:1605.09782.
[56] Andrew Brock, Jeff Donahue, Karen Simonyan, Large scale GAN training for high fidelity natural image synthesis, 2018, arXiv preprint arXiv:1809.11096.
[57] Tero Karras, Samuli Laine, Timo Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.
[58] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, Timothy Lillicrap, Logan: Latent optimisation for generative adversarial networks, 2019, arXiv preprint arXiv:1912.00953.
[59] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, Neil Houlsby, Self-supervised gans via auxiliary rotation loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12154–12163.
[60] Carl Doersch, Abhinav Gupta, Alexei A. Efros, Unsupervised visual representation learning by context prediction, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1422–1430.
[61] Mehdi Noroozi, Paolo Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, in: European Conference on Computer Vision, Springer, 2016, pp. 69–84.
[62] Aaron van den Oord, Yazhe Li, Oriol Vinyals, Representation learning with contrastive predictive coding, 2018, arXiv preprint arXiv:1807.03748.
[63] Michael Gutmann, Aapo Hyvärinen, Noise-contrastive estimation: A new estimation principle for unnormalized statistical models, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 297–304.
[64] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, Jiebo Luo, Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2547–2555.
[65] William Falcon, Kyunghyun Cho, A framework for contrastive self-supervised learning and designing a new approach, 2020, arXiv preprint arXiv:2009.00104.
[66] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton, A simple framework for contrastive learning of visual representations, 2020, arXiv preprint arXiv:2002.05709.

[67] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, Thomas Brox, Discriminative unsupervised feature learning with exemplar convolutional neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (9) (2015) 1734–1747.
[68] Matthew Schultz, Thorsten Joachims, Learning a distance metric from relative comparisons, Adv. Neural Inform. Process. Syst. 16 (2004) 41–48.
[69] Kihyuk Sohn, Improved deep metric learning with multi-class n-pair loss objective, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 1857–1865.
[70] Raia Hadsell, Sumit Chopra, Yann LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'06, 2, IEEE, 2006, pp. 1735–1742.
[71] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.
[72] Zhirong Wu, Yuanjun Xiong, Stella X Yu, Dahua Lin, Unsupervised feature learning via non-parametric instance discrimination, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3733–3742.
[73] Xinlei Chen, Haoqi Fan, Ross Girshick, Kaiming He, Improved baselines with momentum contrastive learning, 2020, arXiv preprint arXiv:2003.04297.
[74] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, Geoffrey Hinton, Big self-supervised models are strong semi-supervised learners, 2020, arXiv preprint arXiv:2006.10029.
[75] Ishan Misra, Laurens van der Maaten, Self-supervised learning of pretext-invariant representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6707–6717.
[76] Ilya Loshchilov, Frank Hutter, Sgdr: Stochastic gradient descent with warm restarts, 2016, arXiv preprint arXiv:1608.03983.
[77] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, Armand Joulin, Unsupervised learning of visual features by contrasting cluster assignments, 2020, arXiv preprint arXiv:2006.09882.
[78] Mathilde Caron, Piotr Bojanowski, Armand Joulin, Matthijs Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 132–149.
[79] Yuki Markus Asano, Christian Rupprecht, Andrea Vedaldi, Self-labelling via simultaneous clustering and representation learning, 2019, arXiv preprint arXiv:1911.05371.
[80] Mathilde Caron, Piotr Bojanowski, Armand Joulin, Matthijs Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 132–149.
[81] Marco Cuturi, Sinkhorn distances: lightspeed computation of optimal transport, NIPS 2 (3) (2013) 4.
[82] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al., Self-supervised pretraining of visual features in the wild, 2021, arXiv preprint arXiv:2103.01988.
[83] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, Fillia Makedon, A survey on contrastive self-supervised learning, Technologies 9 (1) (2021) 2.
[84] Piotr Bojanowski, Armand Joulin, Unsupervised learning by predicting noise, in: International Conference on Machine Learning, PMLR, 2017, pp. 517–526.
[85] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[86] Aravind Srinivas, Michael Laskin, Pieter Abbeel, Curl: Contrastive unsupervised representations for reinforcement learning, 2020, arXiv preprint arXiv:2004.04136.
[87] Hakim Hafidi, Mounir Ghogho, Philippe Ciblat, Ananthram Swami, Graphcl: Contrastive self-supervised learning of graph representations, 2020, arXiv preprint arXiv:2007.08025.
[88] Yang You, Igor Gitman, Boris Ginsburg, Large batch training of convolutional networks, 2017, arXiv preprint arXiv:1708.03888.
[89] Chengxu Zhuang, Alex Lin Zhai, Daniel Yamins, Local aggregation for unsupervised learning of visual embeddings, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6002–6012.
[90] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, Piotr Dollár, Designing network design spaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10428–10436.
[91] Yuki Markus Asano, Christian Rupprecht, Andrea Vedaldi, Self-labelling via simultaneous clustering and representation learning, 2019, arXiv preprint arXiv:1911.05371.
[92] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, Ishan Misra, Scaling and benchmarking self-supervised visual representation learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6391–6400.
[93] Yonglong Tian, Dilip Krishnan, Phillip Isola, Contrastive multiview coding, 2019, arXiv preprint arXiv:1906.05849.
[94] Olivier Henaff, Data-efficient image recognition with contrastive predictive coding, in: International Conference on Machine Learning, PMLR, 2020, pp. 4182–4192.
[95] Philip Bachman, R. Devon Hjelm, William Buchwalter, Learning representations by maximizing mutual information across views, 2019, arXiv preprint arXiv:1906.00910.
[96] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, Phillip Isola, What makes for good views for contrastive learning, 2020, arXiv preprint arXiv:2005.10243.
[97] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, Philipp Krahenbuhl, Sampling matters in deep embedding learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2840–2848.
[98] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al., Bootstrap your own latent: A new approach to self-supervised learning, 2020, arXiv preprint arXiv:2006.07733.
[99] Yuki M Asano, Christian Rupprecht, Andrea Vedaldi, A critical analysis of self-supervision, or what we can learn from a single image, 2019, arXiv preprint arXiv:1904.13132.
[100] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, Oscar Beijbom, nuscenes: A multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11621–11631.
[101] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al., Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer, JAMA 318 (22) (2017) 2199–2210.