Additional Positive Enables Better Representation Learning for Medical Images

Dewen Zeng[1(✉)], Yawen Wu[2], Xinrong Hu[1], Xiaowei Xu[3], Jingtong Hu[2], and Yiyu Shi[1(✉)]

[1] University of Notre Dame, Notre Dame, IN, USA
{dzeng2,yshi4}@nd.edu
[2] University of Pittsburgh, Pittsburgh, PA, USA
[3] Guangdong Provincial People's Hospital, Guangzhou, China

Abstract. This paper presents a new way to identify additional positive pairs for BYOL, a state-of-the-art (SOTA) self-supervised learning framework, to improve its representation learning ability. Unlike conventional BYOL which relies on only one positive pair generated by two augmented views of the same image, we argue that information from different images with the same label can bring more diversity and variations to the target features, thus benefiting representation learning. To identify such pairs without any label, we investigate TracIn, an instance-based and computationally efficient influence function, for BYOL training. Specifically, TracIn is a gradient-based method that reveals the impact of a training sample on a test sample in supervised learning. We extend it to the self-supervised learning setting and propose an efficient batch-wise per-sample gradient computation method to estimate the pairwise TracIn for representing the similarity of samples in the mini-batch during training. For each image, we select the most similar sample from other images as the additional positive and pull their features together with BYOL loss. Experimental results on two public medical datasets (i.e., ISIC 2019 and ChestX-ray) demonstrate that the proposed method can improve the classification performance compared to other competitive baselines in both semi-supervised and transfer learning settings.

Keywords: self-supervised learning · representation learning · medical image classification

1 Introduction

Self-supervised learning (SSL) has been extremely successful in learning good image representations without human annotations for medical image applications like classification [1,23,29] and segmentation [2,4,16]. Usually, an encoder is pre-trained on a large-scale unlabeled dataset. Then, the pre-trained encoder is

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43907-0_12.

H. Greenspan et al. (Eds.): MICCAI 2023, LNCS 14220, pp. 119–129, 2023.
https://doi.org/10.1007/978-3-031-43907-0_12

used for efficient training on downstream tasks with limited annotation [19,24]. Recently, contrastive learning has become the state-of-the-art (SOTA) SSL method due to its powerful learning ability. A recent contrastive learning method learns by pulling the representations of different augmented views of the same image (a.k.a positive pair) together and pushing the representation of different images (a.k.a negative pair) apart [6]. The main disadvantage of this method is its heavy reliance on negative pairs, making it necessary to use a large batch size [6] or memory banks [15] to ensure effective training. To overcome this challenge, BYOL [12] proposes two siamese neural networks - the online and target networks. The online network is trained to predict the target network representation of the same image under a different augmented view, requiring only one positive pair per sample. This approach makes BYOL more resilient to batch size and the choice of data augmentations.

As the positive pair in BYOL is generated from the same image, the diversity of features within the positive pair could be quite limited. For example, one skin disease may manifest differently in different patients or locations, but such information is often overlooked in the current BYOL framework. In this paper, we argue that such feature diversity can be increased by adding additional positive pairs from other samples with the same label (a.k.a. True Positives). Identifying such pairs without human annotation is challenging because of the unrelated information in medical images, such as the background normal skin areas in dermoscopic images. One straightforward way to detect positive pairs is using feature similarity: two images are considered positive if their representations are close to each other in the feature space. However, samples with different labels might also be close in the feature space because the learned encoder is not perfect. Considering them as positive might further pull them together after learning, leading to degraded performance.

To solve this problem, we propose BYOL-TracIn, which improves vanilla BYOL using the TracIn influence function. Instead of quantifying the similarity of two samples based on feature similarity, we propose using TracIn to estimate their similarity by calculating the impact of training one sample on the other. TracIn [22] is a gradient-based influence function that measures the loss reduction of one sample by the training process of another sample. Directly applying TracIn in BYOL is non-trivial as it requires the gradient of each sample and careful selection of model checkpoints and data augmentations to accurately estimate sample impacts without labels. To avoid per-sample gradient computation, we introduce an efficient method that computes the pairwise TracIn in a mini-batch with only one forward pass. For each image in the mini-batch, the sample from other images with the highest TracIn values is selected as the additional positive pair. Their representation distance is then minimized using BYOL loss. To enhance positive selection accuracy, we propose to use a pre-trained model for pairwise TracIn computation as it focuses more on task-related features compared to an on-the-fly model. Light augmentations are used on the samples for TracIn computation to ensure stable positive identification. To the best of our knowledge, we are the first to incorporate additional positive pairs from different images in BYOL. Our extensive empirical results show that our proposed

method outperforms other competing approaches in both semi-supervised and transfer learning settings for medical image classification tasks.

2 Related Work

Self-supervised Learning. Most SSL methods can be categorized as either generative [10,28] or discriminative [11,21], in which pseudo labels are automatically generated from the inputs. Recently, contrastive learning [6,15,27] as a new discriminative SSL method has dominated this field because of its excellent performance. SimCLR [6] and MoCo [15] are two typical contrastive learning methods that try to attract positive pairs and repulse negative pairs. However, these methods rely on a large number of negative samples to work well. BYOL [12] improves contrastive learning by directly predicting the representation output from another view and achieves SOTA performance. As such, only positive pairs are needed for training. SimSiam [7] further proves that stop-gradient plays an essential role in the learning stability of siamese neural networks. Since the positive pairs in BYOL come from the same image, the feature diversity from different images of the same label is ignored. Our method introduces a novel way to accurately identify such positive pairs and attract them in the feature space.

Influence Function. The influence function (IF) was first introduced to machine learning models in [20] to study the following question: which training points are most responsible for a given prediction? Intuitively, if we remove an important sample from the training set, we will get a large increase in the test loss. IF can be considered as an interpretability score that measures the importance of all training samples on the test sample. Aside from IF, other types of scores and variants have also been proposed in this field [3,5,14]. Since IF is extremely computationally expensive, TracIn [22] was proposed as an efficient alternative to estimate training sample influence using first-order approximation. Our method extends the normal TracIn to the SSL setting (i.e., BYOL) with a sophisticated positive pair selection schema and an efficient batch-wise per-sample gradient computation method, demonstrating that aside from model interpretation, TracIn can also be used to guide SSL pre-training.

3 Method

3.1 Framework Overview

Our BYOL-TracIn framework is built upon classical BYOL method [12]. Figure 1 shows the overview of our framework. Here, we use x_1 as the anchor sample for an explanation, and the same logic can be applied to all samples in the mini-batch. Unlike classical BYOL where only one positive pair (x_1 and x'_1) generated from the same image is utilized, we use the influence function, TracIn, to find another sample (x'_3) from the batch that has the largest impact on the anchor sample. During training, the representations distance of x_1 and x'_3 will also be minimized. We think this additional positive pair can increase the variance and diversity of

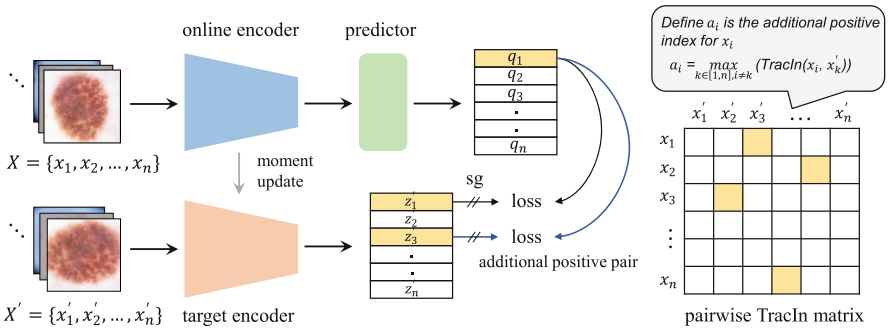

Fig. 1. Overview of the proposed BYOL-TracIn framework. X and X' represent two augmentations of the mini-batch inputs. BYOL-TracIn minimizes the similarity loss of two views of the same image (e.g., q_1 and z_1') as well as the similarity loss of the additional positive (e.g., z_3') identified by our TracIn algorithm. sg means stop-gradient.

the features of the same label, leading to better clustering in the feature space and improved learning performance. The pairwise TracIn matrix is computed using first-order gradient approximation which will be discussed in the next section. For simplicity, this paper only selects the top-1 additional sample, but our method can be easily extended to include top-k ($k > 1$) additional samples.

3.2 Additional Positive Selection Using TracIn

Idealized TracIn and Its First-order Approximation. Suppose we have a training dataset $D = \{x_1, x_2, ..., x_n\}$ with n samples. $f_w(\cdot)$ is a model with parameter $w \in \mathbb{R}$, and $\ell(w, x_i)$ is the loss function when model parameter is w and training example is x_i. The training process in iteration t can be viewed as minimizing the training loss $\ell(w_t, x_t)$ and updating parameter w_t to w_{t+1} using gradient descent (suppose only $x_t \in D$ is used for training in each iteration). Then the idealized TracIn of one sample x_i on another sample x_k can be defined as the total loss reduction by training x_i in the whole training process.

$$\text{TracInIdeal}(x_i, x_k) = \sum_{t:x_t=x_i}^{T} (\ell(w_t, x_k) - \ell(w_{t+1}, x_k)). \tag{1}$$

where T is the total number of iterations. If stochastic gradient descent is utilized as the optimization method, we can approximately express the loss reduction after iteration t as $\ell(w_{t+1}, x_k) - \ell(w_t, x_k) = \nabla \ell(w_t, x_k) \cdot (w_{t+1} - w_t) + O(||\Delta w_t||^2)$. The parameter change in iteration t is $\Delta w_t = w_{t+1} - w_t = -\eta_t \nabla \ell(w_t, x_t)$, in which η_t is the learning rate in iteration t, and x_t is the training example. Since η_t is usually small during training, we can ignore the high order term $O(||\Delta w_t||^2)$, and the first-order TracIn can be formulated as:

$$\text{TracIn}(x_i, x_k) = \sum_{t:x_t=x_i}^{T} \eta_t \nabla \ell(w_t, x_k) \cdot \nabla \ell(w_t, x_i). \tag{2}$$

The above equation reveals that we can estimate the influence of x_i on x_k by summing up their gradient dot products across all training iterations. In practical BYOL training, the optimization is usually done on mini-batches, and it is impossible to save the gradients of a sample for all iterations. However, we can use the TracIn of **the current iteration** to represent the similarity of two samples in the mini-batch because we care about the pairwise relative influences instead of the exact total values across training. Intuitively, if the TracIn of two samples is large in the current iteration, this means that the training of one sample can benefit the other sample a lot because they share some common features. Therefore, they are similar to each other.

Efficient Batch-wise TracIn Computation. Equation 2 requires the gradient of each sample in the mini-batch for pairwise TracIn computation. However, it is prohibitively expensive to compute the gradient of samples one by one. Moreover, calculating the dot product of gradients on the entire model is computationally and memory-intensive, especially for large deep-learning models where there could be millions or trillions of parameters. Therefore, we work with the gradients of the last linear layer in the online predictor.

As current deep learning frameworks (e.g., Pytorch and TensorFlow) do not support per-sample gradient when the batch size is larger than 1, we use the following method to efficiently compute the per-sample gradient of the last layer. Suppose the weight matrix of the last linear layer is $W \in \mathbb{R}^{m \times n}$, where m and n are the numbers of input and output units. $f(q) = 2 - 2 \cdot \langle q, z \rangle / (\|q\|_2 \cdot \|z\|_2)$ is the standard BYOL loss function, where q is the online predictor output (a.k.a., logits) and z is the target encoder output that can be viewed as a constant during training. We have $q = Wa$, where a is the input to the last linear layer. According to the chain rule, the gradient of the last linear layer can be computed as $\nabla_W f(q) = \nabla_q f(q) a^T$, in which the gradient of the logits can be computed by:

$$\nabla_q f(q) = 2 \cdot \left(\frac{\langle q, z \rangle \cdot q}{\|q\|_2^3 \cdot \|z\|_2} - \frac{z}{\|q\|_2 \cdot \|z\|_2} \right). \tag{3}$$

Therefore, the TracIn of sample x_i and x_k at iteration t can be computed as:

$$\begin{aligned} \text{TracIn}(x_i, x_k) &\approx \eta_t \nabla_W f(q_i) \cdot \nabla_W f(q_k) \\ &= \eta_t (\nabla_q f(q_i) \cdot \nabla_q f(q_k))(a_i \cdot a_k). \end{aligned} \tag{4}$$

Equation 3 and 4 tell us that the per-sample gradient of the last linear layer can be computed by using the inputs of this layer and the gradient of the output logits for each sample, which can be achieved with only one forward pass on the mini-batch. This technique greatly speeds up the TracIn computation and makes it possible to be used in BYOL.

Using Pre-trained Model to Increase True Positives. During the pre-training stage of BYOL, especially in the early stages, the model can be unstable and may focus on unrelated features in the background instead of the target features. This can result in the selection of wrong positive pairs while using TracIn. For example, the model may identify all images with skin diseases on

the face as positive pairs, even if they are from different diagnostics, as it focuses on the face feature instead of the diseases. To address this issue, we suggest using a pre-trained model to select additional positives with TracIn to guide BYOL training. This is because a pre-trained model is more stable and well-trained to focus on the target features, thus increasing the selected true positive ratio.

4 Experiments and Results

4.1 Experimental Setups

Datasets. We evaluate the performance of the proposed BYOL-TracIn on four publicly available medical image datasets. **(1) ISIC 2019 dataset** is a dermatology dataset that contains 25,331 dermoscopic images among nine different diagnostic categories [8,9,25]. **(2) ISIC 2016 dataset** was hosted in ISBI 2016 [13]. It contains 900 dermoscopic lesion images with two classes benign and malignant. **(3) ChestX-ray dataset** is a chest X-ray database that comprises 108,948 frontal view X-ray images of 32,717 unique patients with 14 disease labels [26]. Each image may have multiple labels. **(4) Shenzhen dataset** is a small chest X-ray dataset with 662 frontal chest X-rays, of which 326 are normal cases and 336 are cases with manifestations of Tuberculosis [18].

Training Details. We use Resnet18 as the backbone. The online projector and predictor follow the classical BYOL [12], and the embedding dimension is set to 256. On both ISIC 2019 and ChestX-ray datasets, we resize all the images to 140×140 and then crop them to 128×128. Data augmentation used in pre-training includes horizontal flipping, vertical flipping, rotation, color jitter, and cropping. For TracIn computation, we use one view with no augmentation and the other view with horizontal flipping and center cropping because this setting has the best empirical results in our experiments. We pre-train the model for 300 epochs using SGD optimizer with momentum 0.9 and weight decay $1 \times e^{-5}$. The learning rate is set to 0.1 for the first 10 epochs and then decays following a concise learning rate schedule. The batch size is set to 256. The moving average decay of the momentum encoder is set to 0.99 at the beginning and then gradually updates to 1 following a concise schedule. All experiments are performed on one NVIDIA GeForce GTX 1080 GPU.

Baselines. We compare the performance of our method with a random initialization approach without pre-training and the following SOTA baselines that involve pre-training. (1) BYOL [12]: the vanilla BYOL with one positive pair from the same image. (2) FNC [17]: a false negative identification method designed to improve contrastive-based SSL framework. We adapt it to BYOL to select additional positives because false negatives are also equal to true positives for a particular anchor sample. (3) FT [30]: a feature transformation method used in contrastive learning that creates harder positives and negatives to improve the learning ability. We apply it in BYOL to create harder virtual positives. (4) FS: using feature similarity from the current mini-batch to select the top-1 additional positive. (5) FS-pretrained: different from the FS that uses the current

Table 1. Comparison of all methods on ISIC 2019 and ChestX-ray datasets in the semi-supervised setting. We also report the fine-tuning results on 100% datasets. BYOL-Sup is the upper bound of our method. BMA represents the balanced multiclass accuracy.

Method	ISIC 2019			ChestX-ray		
	10%	50%	100%	10%	50%	100%
		BMA ↑			AUC ↑	
Random	0.327(.004)	0.558(.005)	0.650(.004)	0.694(.005)	0.736(.001)	0.749(.001)
BYOL [12]	0.399(.001)	0.580(.006)	0.692(.005)	0.699(.004)	0.738(.003)	0.750(.001)
FNC [17]	0.401(.004)	0.584(.004)	0.694(.005)	0.706(.001)	0.739(.001)	0.752(.002)
FT [30]	0.405(.005)	0.588(.008)	0.695(.005)	0.708(.001)	0.743(.001)	0.751(.002)
FS	0.403(.006)	0.591(.003)	0.694(.004)	0.705(.003)	0.738(.001)	0.752(.002)
FS-pretrained	0.406(.002)	0.596(.004)	0.697(.005)	0.709(.001)	0.744(.002)	0.752(.002)
BYOL-TracIn	0.403(.003)	0.594(.004)	0.694(.004)	0.705(.001)	0.742(.003)	0.753(.002)
BYOL-TracIn-pretrained	**0.408(.007)**	**0.602(.003)**	**0.700(.006)**	**0.712(.001)**	**0.746(.002)**	**0.754(.002)**
BYOL-Sup	0.438(.006)	0.608(.007)	0.705(.005)	0.714(.001)	0.748(.001)	0.756(.003)

model to compute the feature similarity on the fly, we use a pre-trained model to test whether a well-trained encoder is more helpful in identifying the additional positives. (6) BYOL-Sup: the supervised BYOL in which we randomly select one additional positive from the mini-batch using the label information. This baseline is induced as the upper bound of our method because the additional positive is already correct. We evaluate two variants of our method, BYOL-TracIn and BYOL-TracIn-pretrained. The former uses the current training model to compute the TracIn for each iteration while the latter uses a pre-trained model. For a fair comparison, all methods use the same pre-training and finetuning setting unless otherwise specified. For FS-pretrained and BYOL-TracIn-pretrained, the pre-trained model uses the same setting as BYOL. Note that this pre-trained model is only used for positive selection and not involves in training.

4.2 Semi-supervised Learning

In this section, we evaluate the performance of our method by finetuning with the pre-trained encoder on the same dataset as pre-training with limited annotations. We sample 10% or 50% of the labeled data from ISIC 2019 and ChestX-ray training sets and finetune the model for 100 epochs on the sampled datasets. Data augmentation is the same as pre-training. Table 1 shows the comparisons of all methods. For ISIC 2019, we report the balanced multiclass accuracy (BMA, suggested by the ISIC challenge). For ChestX-ray, we report the average AUC across all diagnoses. We conduct each finetuning experiment 5 times with different random seeds and report the mean and std.

From Table 1, we have the following observations: (1) Compared to Random, all the other methods have better accuracy, which means that pre-training can indeed help downstream tasks. (2) Compared to vanilla BYOL, other pre-training methods show performance improvement on both datasets. This shows that additional positives can increase feature diversity and benefit BYOL learning. (3) Our BYOL-TracIn-pretrained consistently outperforms all other unsu-

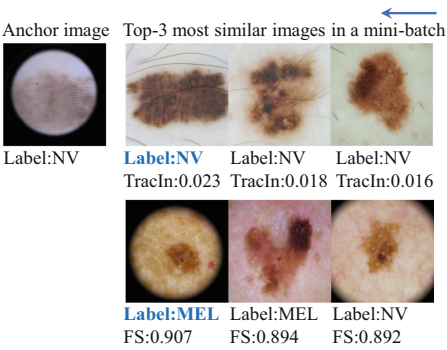

Fig. 2. Comparison of TracIn and Feature Similarity (FS) in selecting the additional positive during training on ISIC 2019.

Table 2. Transfer learning comparison of the proposed method with the baselines on ISIC 2016 and Shenzhen datasets.

Method	ISIC 2016 Precision ↑	Shenzhen AUC ↑
Random	0.400(.005)	0.835(.010)
BYOL [12]	0.541(.008)	0.858(.003)
FNC [17]	0.542(.007)	0.862(.006)
FT [30]	0.559(.011)	0.876(.005)
FS	0.551(.003)	0.877(.004)
FS-pretrained	0.556(.004)	0.877(.006)
BYOL-TracIn	0.555(.012)	0.880(.007)
BYOL-TracIn-pretrained	**0.565(.010)**	**0.883(.001)**
BYOL-Sup	0.592(.008)	0.893(.006)

pervised baselines. Although BYOL-TracIn can improve BYOL, it could be worse than other baselines like FT and FS-pretrained (e.g., 10% on ISIC 2019). This is because some additional positives identified by the on-the-fly model may be false positives, and attracting representations of such samples will degrade the learned features. However, with a pre-trained model in BYOL-TracIn-pretrained, the identification accuracy can be increased, leading to more true positives and better representations. (4) TracIn-pretrained performs better than FS-pretrained in all settings, and the improvement in BMA could be up to 0.006. This suggests that TracIn can be a more reliable metric for assessing the similarity between images when there is no human label information available. (5) Supervised BYOL can greatly increase the BYOL performance on both datasets. Yet our BYOL-TracIn-pretrained only has a marginal accuracy drop from supervised BYOL with a sufficient number of training samples (e.g., 100% on ISIC 2019).

To further demonstrate the superiority of TracIn over Feature Similarity (FS) in selecting additional positive pairs for BYOL, we use an image from ISIC 2019 as an example and visualize the top-3 most similar images selected by both metrics using a BYOL pre-trained model in Fig. 2. We can observe that TracIn accurately identifies the most similar images with the same label as the anchor image, whereas two of the images selected by FS have different labels. This discrepancy may be attributed to the fact that the FS of these two images is dominated by unrelated features (e.g., background tissue), which makes it unreliable. More visualization examples can be found in the supplementary.

4.3 Transfer Learning

To evaluate the transfer learning performance of the learned features, we use the encoder learned from the pre-training to initialize the model on the downstream datasets (ISIC 2019 transfers to ISIC 2016, and ChestX-ray transfers to Shenzhen). We finetune the model for 50 epochs and report the precision and AUC on ISIC 2016 and Shenzhen datasets, respectively. Table 2 shows the comparison

results of all methods. We can see that BYOL-TracIn-pretrained always outperforms other unsupervised pre-training baselines, indicating that the additional positives can help BYOL learn better transferrable features.

5 Conclusion

In this paper, we propose a simple yet effective method, named BYOL-TracIn, to boost the representation learning performance of the vanilla BYOL framework. BYOL-TracIn can effectively identify additional positives from different samples in the mini-batch without using label information, thus introducing more variances to learned features. Experimental results on multiple public medical image datasets show that our method can significantly improve classification performance in both semi-supervised and transfer learning settings. Although this paper only discusses the situation of one additional pair for each image, our method can be easily extended to multiple additional pairs. However, more pairs will introduce more computation costs and increase the false positive rate which may degrade the performance. Another limitation of this paper is that BYOL-TracIn requires a pre-trained model to start with, which means more computation resources are needed to demonstrate its effectiveness.

References

1. Azizi, S., et al.: Big self-supervised models advance medical image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3478–3488 (2021)
2. Bai, W., et al.: Self-supervised learning for cardiac MR image segmentation by anatomical position prediction. In: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.-T., Khan, A. (eds.) MICCAI 2019. LNCS, vol. 11765, pp. 541–549. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32245-8_60
3. Barshan, E., Brunet, M.E., Dziugaite, G.K.: Relatif: identifying explanatory training samples via relative influence. In: International Conference on Artificial Intelligence and Statistics, pp. 1899–1909. PMLR (2020)
4. Chaitanya, K., Erdil, E., Karani, N., Konukoglu, E.: Contrastive learning of global and local features for medical image segmentation with limited annotations. Adv. Neural. Inf. Process. Syst. **33**, 12546–12558 (2020)
5. Chen, H., et al.: Multi-stage influence function. Adv. Neural. Inf. Process. Syst. **33**, 12732–12742 (2020)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
7. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758 (2021)
8. Codella, N., et al.: Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). arXiv preprint arXiv:1902.03368 (2019)

9. Combalia, M., et al.: Bcn20000: Dermoscopic lesions in the wild. arXiv preprint arXiv:1908.02288 (2019)
10. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430 (2015)
11. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
12. Grill, J.B., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Adv. Neural. Inf. Process. Syst. **33**, 21271–21284 (2020)
13. Gutman, D., et al.: Skin lesion analysis toward melanoma detection: a challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic). arXiv preprint arXiv:1605.01397 (2016)
14. Hara, S., Nitanda, A., Maehara, T.: Data cleansing for models trained with sgd. In: Advances in Neural Information Processing Systems 32 (2019)
15. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
16. Hu, X., Zeng, D., Xu, X., Shi, Y.: Semi-supervised contrastive learning for label-efficient medical image segmentation. In: de Bruijne, M., Cattin, P.C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., Essert, C. (eds.) MICCAI 2021. LNCS, vol. 12902, pp. 481–490. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87196-3_45
17. Huynh, T., Kornblith, S., Walter, M.R., Maire, M., Khademi, M.: Boosting contrastive self-supervised learning with false negative cancellation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2785–2795 (2022)
18. Jaeger, S., Candemir, S., Antani, S., Wáng, Y.X.J., Lu, P.X., Thoma, G.: Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. Quant. Imaging Med. Surg. **4**(6), 475 (2014)
19. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020)
20. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: International Conference on Machine Learning, pp. 1885–1894. PMLR (2017)
21. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
22. Pruthi, G., Liu, F., Kale, S., Sundararajan, M.: Estimating training data influence by tracing gradient descent. Adv. Neural. Inf. Process. Syst. **33**, 19920–19930 (2020)
23. Sowrirajan, H., Yang, J., Ng, A.Y., Rajpurkar, P.: Moco pretraining improves representation and transferability of chest x-ray models. In: Medical Imaging with Deep Learning, pp. 728–744. PMLR (2021)
24. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X.: Embracing imperfect datasets: a review of deep learning solutions for medical image segmentation. Med. Image Anal. **63**, 101693 (2020)
25. Tschandl, P., Rosendahl, C., Kittler, H.: The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data **5**(1), 1–9 (2018)

26. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2097–2106 (2017)
27. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: self-supervised learning via redundancy reduction. In: International Conference on Machine Learning, pp. 12310–12320. PMLR (2021)
28. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
29. Zhang, Y., Jiang, H., Miura, Y., Manning, C.D., Langlotz, C.P.: Contrastive learning of medical visual representations from paired images and text. In: Machine Learning for Healthcare Conference, pp. 2–25. PMLR (2022)
30. Zhu, R., Zhao, B., Liu, J., Sun, Z., Chen, C.W.: Improving contrastive learning by visualizing feature transformation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10306–10315 (2021)