# DOWNSTREAM AUGMENTATION GENERATION FOR CONTRASTIVE LEARNING

*Tomohiro Hayase*[*], *Suguru Yasutomi*[*]
Fujitsu Ltd.

*Nakamasa Inoue*
Tokyo Institute of Technology

## ABSTRACT

Contrastive learning has become one of the most promising approaches for learning image representations. However, it heavily relies on heuristic data augmentation techniques, such as Gaussian blurring and color jittering, for making image pairs to be contrastively compared. These augmentations are not always appropriate for downstream tasks that each have their own camera and illumination settings. In this paper, we aim at improving the augmentation process and propose an *augmentation generator*, a network that learns to augment images for contrastive learning. Under the assumption that each downstream task has an optimal implicit augmentation function, the augmentation generator enhances the contrastive learning by estimating it. We demonstrate the effectiveness of our learning framework on two combined datasets, EMNIST-Omniglot and ImageNet-DAISO.

***Index Terms –*** Contrastive learning, data augmentation.

## 1. INTRODUCTION

With the recent successes of large-scale contrastive learning such as MoCo [1, 2] and SimCLR [3, 4], self-supervised learning has proven to be one of the most powerful frameworks for pre-training neural networks in an unsupervised manner. Fine-tuning a pre-trained network is effective in many downstream tasks where only limited data are available for training. Example applications include face recognition [5, 6], medical image recognition [7], human motion recognition [8, 9], and 3D scene recognition [10, 11].

To train robust representations that generalize to unseen tasks, contrastive learning heavily relies on data augmentation. In recent studies, heuristic augmentation techniques, including Gaussian blurring, are commonly utilized to make image pairs that can be contrastively compared. However, if these augmentations are not appropriate for the downstream tasks, overall performance may be degraded in practice. There are previous studies related to automatic augmentation [12], but most of them are on pre-training. In contrast, we focus on finding an optimal augmentation for each downstream task after pre-training and before fine-tuning; thus, this study is complementary to the other contrastive learning frameworks.

In this paper, we present a framework for learning to augment images for contrastive learning. In particular, we propose an algorithm to train an augmentation generator, a network for image augmentation. Under the assumption that each downstream task has its own optimal implicit augmentation function, the augmentation generator estimates it and enhances the contrastive learning. To evaluate our framework, we performed two experiments. The first one evaluated how well the augmentation generator works. We intentionally introduced an implicit augmentation to Omniglot digit images [13] and gave a demonstration of estimating it. The second experiment showed the effectiveness of the framework in a real-world task. We pre-trained a network on the ImageNet dataset [14] and fine-tuned it on the DAISO-100 [15] dataset, which is an image dataset for product-quality management. Our contributions are as follows:

- We propose an augmentation generator, a network that learns to augment images for contrastive learning, by focusing on an embedding space.
- We propose a training algorithm in which an augmentation generator, an eliminator, and a latent noise predictor collaborate with each other in an adversarial manner.
- We demonstrate a real-world application, product quality-management, and show that our framework enhances MoCov2, one of the state-of-the-art contrastive learning frameworks.

## 2. THEORY AND BACKGROUND

**Contrastive Learning.** The goal of contrastive learning is to learn an embedding function

$$F : X \to Y, \tag{1}$$

where $X$ is an input space and $Y$ is an embedding space. For image classification, $Y$ is typically $\mathbb{R}^d$, and the representations $F(X)$ are expected to be well-separated.

The loss $\mathcal{L}_c$ in contrastive learning is often defined over augmented inputs with the following three steps. First, given a mini-batch $B = \{x_i\}_{i=1}^n$, two augmentation functions $t$ and $t'$ are randomly chosen for each $x_i$. Second, embeddings $h_i = H \circ F \circ t(x_i)$ and $h'_i = H \circ F \circ t'(x_i)$ are computed for all $i$, where $H$ is a header network to avoid over-fitting. Finally, the loss $\mathcal{L}_c$ is computed among the set of embeddings $K = \{h_i\}_{i=1}^n \cup \{h'_i\}_{i=1}^n$.

[*] These authros have contributed equally.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

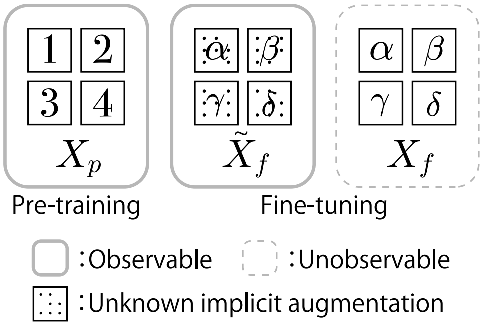

**Fig. 1**: Problem setting we consider. We assume that images for fine-tuning are implicitly augmented. Corresponding pure images are not observable.

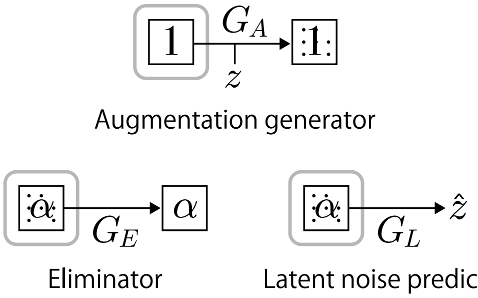

**Fig. 2**: Proposed three modules $G_A, G_E$, and $G_L$ with a noise vector $z$ for learning augmentation.

Examples of definitions of $\mathcal{L}_c$ include the losses in SimCLR and MoCo. Both are based on InfoNCE [16]:

$$\ell(q, k_+) = -\log \frac{e^{q\cdot k_+/\tau}}{e^{q\cdot k_+/\tau} + \sum_{k_-} e^{q\cdot k_-/\tau}}, \tag{2}$$

where $q$ is a query, $k_+$ is a positive key, and $k_- \in K_-$ is a negative key. SimCLR makes $K_-$ from the current batch $K_- = K \setminus \{q, k_+\}$, and computes $\mathcal{L}_c = (2n)^{-1}\sum_i(\ell(h_i, h'_i) + \ell(h'_i, h_i))$. MoCo maintains a bank of momentum queues.

For downstream tasks, the head is replaced by a new one. Specifically, if the down-stream task is a $c$-class classification, a classification head $C : Y \to \mathbb{R}^c$ is utilized. In practice, a linear layer is often a reasonable choice for $C$.

**Data augmentation.** Set a data augmentation function as

$$g_a : X \times Z_a \to X, \tag{3}$$

where $X$ is an image space, $Z_a$ is a parameter space for randomization, and $a$ is the name of the augmentation. For example, Gaussian blurring augmentation $g_{\text{blur}}$ takes two inputs, an image $x \in X$ and a parameter $z = (p, \lambda) \in [0, 1]^2 = Z_{\text{blur}}$. It applies to the image a blurring filter with a Gaussian distribution with standard deviation $\lambda\sigma_{\text{max}}$ if and only if $p$ is smaller than $p_{\text{th}}$, where $\sigma_{\text{max}} \in \mathbb{R}^+$ is the maximum value of the standard deviation and $p_{\text{th}} \in [0, 1]$ is a threshold to determine whether or not to apply this augmentation. Note that $\sigma_{\text{max}}$ and $p_{\text{th}}$ are hyper-parameters. In practice, compositions of different augmentation functions are used in contrastive learning.

## 3. METHOD

This section presents the proposed learning framework of augmentation generation for contrastive learning. First, we describe the assumptions; then, we present the two training stages: (i) the training augmentation generator and (ii) the re-training of the embeddings.

**Assumptions.** Let $X_p, \tilde{X}_f \subset X$ be finite sets of images for pre-training and fine-tuning, respectively. As shown in Fig. 1, we assume that the images in $\tilde{X}_f$ are implicitly augmented, and the corresponding original images $X_f$ are not observable. This simulates a real-world setting where there may be implicit changes in the camera and illumination settings. Additionally, we assume that the pre-training dataset $X_p$ is large enough and is similar to $X_f$ in the sense that there is a constant $c > 0$ with

$$c\mu_f \leq \mu_p, \tag{4}$$

where $\mu_p$ and $\mu_f$ are respectively the probability distributions of the pre-training and fine-tuning images.

### 3.1. Training Augmentation Generator

The augmentation generator $G_A$ is a network to output augmented images, to which a contrastive loss is applied. As shown in Fig. 2, its role is to estimate the implicit augmentation in $\tilde{X}_f$. More specifically, $G_A$ takes an image with a latent noise vector as input:

$$G_A : X \times Z \to X, \tag{5}$$

where $X$ is an image space and $Z$ is a set of random vectors. This definition is inspired by (3).

Further, we introduce two auxiliary networks, an eliminator $G_E : X \to X$ and a latent noise predictor $G_L : X \to Z$. As shown in Fig. 2, the former learns to eliminate augmentations, and the latter predicts the latent noise vector $z \in Z$. They collaborate with each other and help in the training of the augmentation generator.

The proposed training algorithm is summarized in Algorithm 1. It consists of two steps: discriminator training and generator training.

**Discriminator Training.** Motivated by adversarial training techniques [17], we introduce two discriminators,

$$D_f, D_p : Y \to [0, 1]. \tag{6}$$

The discriminator $D_f$ (resp. $D_p$) determines if the representation comes from $X_f$ (resp. $X_p$). This step minimizes the following loss and updates the parameters of the discriminators:

$$\mathcal{L}(D_f, D_p) = \mathbb{E}_{x_p, \tilde{x}_f, z}[\ell_{D_f}(x_p, \tilde{x}_f, z) + \ell_{D_p}(x_p, \tilde{x}_f)], \tag{7}$$

$$\begin{aligned} \ell_{D_f}(x_p, \tilde{x}_f, z) = &- \log\left(D_f \circ F\left(\tilde{x}_f\right)\right) \\ &- \log\left(1 - D_f \circ F \circ G_A\left(x_p, z\right)\right), \end{aligned} \tag{8}$$

$$\begin{aligned} \ell_{D_p}(x_p, \tilde{x}_f) = &- \log\left(D_p \circ F\left(x_p\right)\right) \\ &- \log\left(1 - D_p \circ F \circ G_E\left(\tilde{x}_f\right)\right). \end{aligned} \tag{9}$$

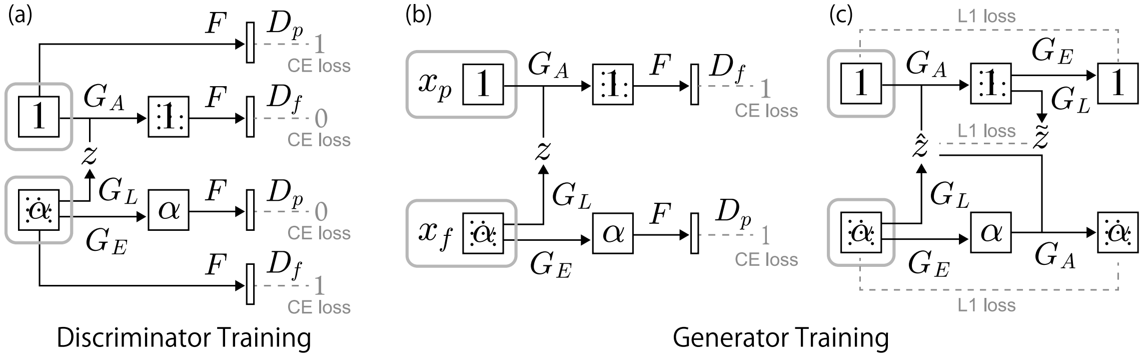

**Fig. 3**: Discriminator and generator training. The forward paths are illustrated. (a) discriminator loss, (b) generator loss, and (c) consistency loss.

**Algorithm 1** : Training Augmentation

```
Input: Learning rate: $\eta$, pretraining dataset $X_p$, downstream
    dataset for fine-tuning $X_f$, pretrained base network $F$
    while loss does not converge do
        Pick minibatches
        $g_{\theta_N} \leftarrow \partial_{\theta_N}\mathcal{L}(D_f, D_p)$, $N = D_f, D_p$.
        $g_{\theta_N} \leftarrow \partial_{\theta_N}\mathcal{L}(G_A, G_E, G_L)$, $N = G_A, G_E, G_L$.
        Optimize$(\theta_N; g_{\theta_N})$, $N = D_f, D_p, G_A, G_E, G_L$.
    end while
```

Fig. 3a illustrates the forward paths to compute these losses. Note that $F$ is the embedding function in (1).

**Generator Training.** In this step, the augmentation generator and the eliminator try to fool the two discriminators by minimizing the following loss:

$$\begin{aligned}\mathcal{L}(G_A, G_E, G_L) = \mathbb{E}_{x_p,\tilde{x}_f,z}[&\ell_{G_A}(x_p, z) + \ell_{G_E}(\tilde{x}_f, z)\\ &+ \lambda\ell_r(x_p, \tilde{x}_f, z)]. \qquad (10)\end{aligned}$$

This loss consists of two generator losses (Fig. 3b) and a consistency loss (Fig. 3c). The former are given by

$$\ell_{G_A}(x_p, z) = -\log\left(D_f \circ G_A \circ F\,(x_p, z)\right), \qquad (11)$$

$$\ell_{G_E}(\tilde{x}_f) = -\log\left(D_p \circ F \circ G_E\,(\tilde{x}_f)\right). \qquad (12)$$

The latter is a cycle-consistency's [18] extension that incorporate with the latent noise,

$$\begin{aligned}\ell_r(\tilde{x}_f, x_p, z) =& \|G_A(G_E(\tilde{x}_f), G_L(\tilde{x}_f)) - \tilde{x}_f\|_1\\ &+ \|G_E \circ G_A(x_p, z) - x_p\|_1\\ &+ \|G_L \circ G_A(x_p, z) - z\|_1. \qquad (13)\end{aligned}$$

### 3.2. Re-training of embedding

After training the augmentation generator, we apply it to contrastive learning. By (4), we have

$$\mathbb{E}_{\mu_p}[\ell(F, G_A(F, z))] \geq c\mathbb{E}_{\mu_f}[\ell(F, G_A(F, z))]. \qquad (14)$$

Thus, to minimize the right-hand side of (14) without observations from $\mu_f$, we re-train $F$ by minimizing the following while freezing $G_A$:

$$\mathcal{L}(F) = \frac{1}{|X_p|}\sum_{x \in X_p} \ell(F(x), G_A(F(x), z)), \qquad (15)$$

where the latent noise vectors are uniformly sampled from the set $\tilde{Z} = \{\tilde{z} : \tilde{z} = G_L(x), x \in \tilde{X}_f\}$.

## 4. EXPERIMENTS

To examine the effectiveness of the proposed framework, we performed experiments with two pairs of datasets, EMNIST-Omniglot and ImageNet-DAISO.

### 4.1. Experimental Setup

**EMNIST-Omniglot.** This small-scale experiment was to show how the augmentation generator works. EMNIST ByClass [21], which consists of 814,255 handwritten digit and Latin alphabet images, was used as a pre-training dataset $X_p$. Omniglot [13], which consists of 32,460 handwritten character images collected from various languages, was divided into two subsets: a clean fine-tuning set $X_f^C$ (24,345 images) and a clean test set $X_t^C$ (8,115 images). To demonstrate the condition of Fig. 1, we intentionally made augmented datasets $X_f^D$ and $X_t^D$ by drawing random dots on the images, as shown in Fig. 4. We only used $\tilde{X}_f = X_f^D$ for fine-tuning; the reported results are for both $X_t^C$ and $X_t^D$. ResNet-18 [22] was used as the backbone embedding network. The generators $G_A, G_E$, and $G_L$ were perceptrons with four layers, each with 912 hidden units. Except for the last layer, $G_E$ shared its layers with $G_L$.

**ImageNet-DAISO.** In this large-scale challenging experiment, the augmentation generator learned real-world lighting changes. ImageNet-1k [14], which consists of 1.2 million images, is used as the pre-training dataset $X_p$. DAISO-100 [15], which consists of 160,000 images of 100 miscellaneous products provided with metadata of lighting conditions, was equally divided into four subsets $X_f^R$, $X_t^R$, $X_f^S$, and $X_t^S$. Here, $R$ and $S$ denote images taken under room light and spotlight conditions, respectively, Either $X_f^R$ or $X_f^S$ was used for fine-tuning to evaluate whether the augmentation generator helped to learn a representation that was robust against changes in lighting conditions. Both $X_t^R$ and $X_t^S$ were used for testing. ResNet-50 was used as the backbone embedding

**Table 1**: Classification accuracy on Omniglot and DAISO test sets. $A \rightarrow B$ denotes $X_f^A$ $(= \tilde{X}_f)$ is used for fine-tuning and $X_t^B$ is used for testing. $D$: images intentionally augmented by Drawing random dots. $C$: Clean images. $S$: Images taken under Spotlight. $R$: Images taken under Room light.

| | Method | Re-training set | Additional augmentaion | EMNIST-Omniglot | | ImageNet-DAISO | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $D \rightarrow C$ | $D \rightarrow D$ | $S \rightarrow R$ | $S \rightarrow S$ | $R \rightarrow S$ | $R \rightarrow R$ |
| | MoCov2 (baseline) | $X_p$ | - | 3.25 | 22.97 | 67.45 | 94.74 | 71.10 | 93.99 |
| | MoCov2 + Ours | $X_p$ | $G_A$ | **55.22** | 64.60 | **71.91** | 94.84 | **74.49** | 94.16 |
| Reference | MoCov2 | $X_p \cup \tilde{X}_f$ | - | 37.97 | **76.03** | 59.20 | 93.06 | 69.28 | 92.08 |
| | SimCLR [3] | $X_p \cup \tilde{X}_f$ | - | - | - | 54.74 | 94.61 | 65.20 | 94.53 |
| | SimSiam [19] | $X_p \cup \tilde{X}_f$ | - | - | - | 55.50 | 94.85 | 66.33 | **94.62** |
| | SwAV [20] | $X_p \cup \tilde{X}_f$ | - | - | - | 55.17 | **94.86** | 64.98 | 94.38 |

network. We used U-Net [23] for $G_A, G_E$, and $G_L$. An input channel that corresponded to $z \in Z$ was added to $G_A$. Except for the last layer, $G_E$ shared its layers with $G_L$.

In both experiments, the embedding network was first pre-trained with the contrastive loss of MoCov2 [1]. The hyper-parameters were the defaults of its official implementation. To train the augmentation generator, we used momentum SGD with a learning rate of $\lambda = 3 \times 10^{-4}$ with a batch size of 256. For EMNIST-Omniglot, the number of training epochs was 100. For ImangeNet-DAISO, it was 25 for $S$ and was 28 for $R$. After that, only a new linear head was trained with the ground-truth labels for the evaluation. We report the classification accuracy on each validation dataset.

### 4.2. Evaluation Results

Table 1 summarizes the experimental results. We can see that our method outperformed the MoCov2 baseline in all conditions. In particular, we obtained significant performance improvements in cases where the fine-tuning and test conditions were different (i.e., $D \rightarrow C$, $S \rightarrow R$ and $R \rightarrow S$). This confirms that our augmentation generator worked as expected. Even when the baseline used all images $X_p \cup \tilde{X}_f$ for training the embeddings, our method outperformed it in most cases. We can see that the reference model was over-fitted to the dataset for fine-tuning, and as a result, it performed well only on $D \rightarrow D$, where the fine-tuning and testing images share exactly the same intentional augmentation. The overfitting caused performance degradation on ImageNet-DAISO. In contrast, our method successfully trained representations that were robust against changes in lighting conditions by avoiding over-fitting with the augmentation generator.

Fig. 5 shows some qualitative results. As can be seen, the generator learned the augmentation function that changes the density of dots, even though our method does not aim to reproduce the augmentation at the image level. The final classification performance was improved because this augmentation enhanced the contrastive learning. In the future, it would be interesting to estimate other types of implicit real-world augmentation such as 3D viewpoint changes and occlusion.

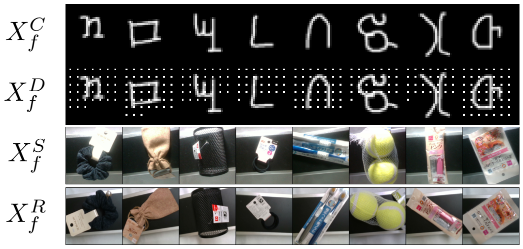

**Fig. 4**: Example images from fine-tuning datasets.

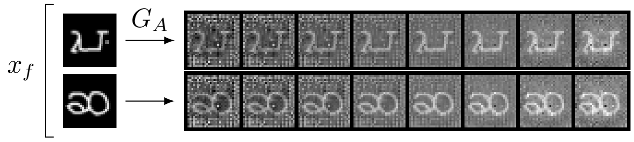

**Fig. 5**: Examples of generated augmentation for several latent variables in the EMNIST-Omniglot setting.

## 5. CONCLUSION

We presented an augmentation generator, a network for image augmentation in contrastive learning. We proposed a learning algorithm, in which an eliminator and a latent noise predictor collaboratively help the augmentation generator to learn how to augment images for downstream tasks. Our experiments showed the effectiveness of the proposed algorithm in a real-world task. In the future, applying our method to video data and audio data would be interesting; so would improving the augmentation generation algorithms.

## 6. REFERENCES

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] X. Chen, H. Fan, R. Girshick, and K. He, "Im-

proved baselines with momentum contrastive learning," *arXiv2003.04297*, 2020.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020.

[4] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Nakamasa Inoue, "Teacher-assisted mini-batch sampling for blind distillation using metric learning," in *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2021.

[6] Chun-Hsien Lin and Bing-Fei Wu, "Domain adapting ability of self-supervised learning for face recognition," in *International Conference on Image Processing (ICIP)*, 2021.

[7] S. Cornelissen, J.A. van der Putten, T. G. W. Boers, J.B. Jukema, K.N. Fockens, J.J.G.H.M. Bergman, F. van der Sommen, and P.H.N. de With, "Evaluating self-supervised learning methods for downstream classification of neoplasia in barrett's esophagus," in *International Conference on Image Processing (ICIP)*, 2021.

[8] Yiqun Liu, Yi Zeng, Jian Pu, Hongming Shan, Peiyang He, and Junping Zhang, "Selfgait: A spatiotemporal representation learning method for self-supervised gait recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[9] Ci-Siang Lin and Yu-Chiang Frank Wang, "Self-supervised bodymap-to-appearance co-attention for partial person re-identification," in *International Conference on Image Processing (ICIP)*, 2021.

[10] Jianrong Wang, Ge Zhang, Zhenyu Wu, Xuewei Li, and Li Liu, "Self-supervised depth estimation via implicit cues from videos," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[11] Yawen Lu, Yuhao Zhu, and Guoyu Lu, "3D SceneFlowNet: Self-supervised 3D scene flow estimation based on graph cnn," in *International Conference on Image Processing (ICIP)*, 2021.

[12] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le, "AutoAugment: Learning augmentation strategies from data," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[13] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] Takashi Katoh, Kanata Suzuki, Shioe Kuramochi, Tomotake Sasaki, and Hiromichi Kobashi, "Dataset of annotated images of sundry objects — benchmark for performance degradation caused by domain shifts," in *ICLR 2021 Workshop "Generalization beyond the training distribution in brains and machines"*, 2021.

[16] Oriol Vinyals Aaron van den Oord, Yazhe Li, "Representation learning with contrastive predictive coding," *arXiv1807.03748*, 2018.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Xinlei Chen and Kaiming He, "Exploring simple siamese representation learning," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[21] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.