Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 32(8): 4913–4925.
DOI: 10.3934/era.2024226
Received: 06 June 2024
Revised: 25 July 2024
Accepted: 05 August 2024
Published: 16 August 2024

Research article

Group-based siamese self-supervised learning

Zhongnian Li, Jiayu Wang, Qingcong Geng and Xinzheng Xu*

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

* **Correspondence:** Email: xxzheng@cumt.edu.cn.

Abstract: In this paper, we introduced a novel group self-supervised learning approach designed to improve visual representation learning. This new method aimed to rectify the limitations observed in conventional self-supervised learning. Traditional methods tended to focus on embedding distortion-invariant in single-view features. However, our belief was that a better representation can be achieved by creating a group of features derived from multiple views. To expand the siamese self-supervised architecture, we increased the number of image instances in each crop, enabling us to obtain an average feature from a group of views to use as a distortion, invariant embedding. The training efficiency has greatly increased with rapid convergence. When combined with a robust linear protocol, this group self-supervised learning model achieved competitive results in CIFAR-10, CIFAR-100, Tiny ImageNet, and ImageNet-100 classification tasks. Most importantly, our model demonstrated significant convergence gains within just 30 epochs as opposed to the typical 1000 epochs required by most other self-supervised techniques.

Keywords: self-supervised learning; average feature; multiple views; classification tasks; siamese network

1. Introduction

In recent years, the field of machine learning has undergone remarkable transformation, among which one of the most influential changes is the evolution of self-supervised learning (SSL) [1–5]. This paradigm offers unprecedented advantages by allowing algorithms to automatically infer meaningful representations from unlabeled datasets. This represents a game-changing initiative as it circumvents the need for expensive labeled data, which has historically been a bottleneck for the scalability of machine learning systems. Among numerous SSL techniques, methods such as Momentum Contrast (MoCo) [6], a simple framework for contrastive learning of visual representations (SimCLR) [7], and Barlow Twins [8] have drawn extensive attention in the field of

visual representation learning. Their performance metrics have been comprehensively evaluated on benchmark datasets such as CIFAR-10 and ImageNet. These classical models are primarily designed to handle single-view features and the extracted embeddings still maintain their characteristics even when subjected to various forms of distortions, such as translation, rotation, and scaling [7].

Although the current self-supervised models have several advantages, they also have their own limitations. For example, MoCo requires a significantly large memory bank [6] to store features, resulting in a sharp increase in computational costs. While SimCLR [7] is efficient, it is well-known for its heavy reliance on data augmentation [7, 9–11] techniques and large-batch training. Although inspired by the interesting principle of redundancy reduction in neuroscience [8], Barlow Twins still cannot avoid the challenges associated with complex hyperparameter tuning. Meanwhile, newer models like Swapping Assignments between multiple Views of the same image (SWAV) [12], Bootstrap Your Own Latent (BYOL) [13], and SimSiam [14], despite their innovative design strategies, still require long training durations, often taking hundreds of epochs to achieve satisfactory convergence. One commonality among these models is that they are all optimized for single-view feature representations, and the immense potential of multi-view features remains unexplored.

Our main inspiration for this research comes from the limitations of single-view image augmentation in the existing field of SSL. In this paper, we proposed an approach called Group-based Siamese Self-Supervised Learning (GSSSL) to improve the inefficiency and limitations of current single-view SSL models. Specifically, the proposed approach explores diverse augmentations of feature groups extracted from multi-view image data for siamese SSL. Besides, we employ the mean aggregation of feature groups to generate distortion-invariant embeddings, which improves the quality of visual representations and significantly reduces training time. Extensive empirical studies are conducted on a range of datasets, including CIFAR-10, CIFAR-100, Tiny ImageNet, and ImageNet-100, and the results demonstrate that our model not only matches existing models but also often outperforms them. It is particularly noteworthy that our framework achieves unprecedented convergence speed, reaching satisfactory performance levels in just 30 epochs, while industry standards typically require around 1000 epochs. By rigorously addressing the inherent limitations and inefficiencies of existing SSL models, our research aims to significantly advance the field. The introduction of our GSSSL framework provides a promising avenue for developing more powerful, efficient, and universal SSL models, thereby expanding their application scope in various domains and industries.

This paper presents the GSSSL framework, which improves upon traditional SSL methods. By leveraging multi-view data and aggregating features from multiple image crops, GSSSL enhances the robustness and efficiency of learned representations. Key contributions include:

1) **Innovative Framework**: GSSSL expands traditional siamese network architectures, creating more robust feature representations by grouping features from multiple views.

2) **Improved Training Efficiency**: GSSSL achieves significant convergence in just 30 epochs, a notable improvement over the typical 1000 epochs required by traditional models.

3) **Empirical Validation**: Extensive evaluations on CIFAR-10, CIFAR-100, Tiny ImageNet, and ImageNet-100 datasets show that GSSSL often outperforms existing SSL models in terms of accuracy and training efficiency.

Overall, GSSSL offers a more efficient and effective approach to SSL, advancing the field toward scalable and accessible machine learning solutions.

2. Related work

This study is closely related to various investigations. Here, we present a succinct summary of these aspects.

Our study is primarily focused on the siamese [15] network architecture for SSL, in which two networks are trained to produce comparable embeddings for identical images. Siamese network architecture was first proposed by Bromley et al. [16], as a novel artificial neural network used for signature verification. Zagoruyko and Komodakis [17] improved the classic siamese networks algorithm, making it more effective in handling image similarity problems. Chen and He [18] introduced that meaningful visual representations can be effectively learned through a simple siamese network structure and stop-gradient operation, which provided a new perspective for unsupervised representation learning. Currently, mainstream SSL methods utilize the siamese network architecture, albeit with some variations in detail.

Notably, MoCo [19] revolutionizes contrastive learning methods by conceptualizing them as dictionary lookup problems, which introduces two queues and momentum encoders to build a comprehensive and coherent dictionary, thereby improving contrastive learning. A key advantage of this approach is that it eliminates the need for a large batch size in training, resulting in significant memory savings.

In contrast, SimCLR [7] employs a siamese network architecture with a single encoder, eliminating the requirement for memory banks, queues, and momentum encoders. It trains the network by focusing on the relationships between positive and negative pairs, where positive pairs consist of two variations of the same image, while all other pairs are considered negatives. This design choice simplifies the approach while maintaining high effectiveness.

In contrast, SwAV [12] integrates contrastive learning with prior clustering techniques, leveraging prior information to compare it with clustering centers instead of relying on numerous negative samples, thereby optimizing resource utilization. The multi-crop operation within SwAV [12] plays a vital role in the field of SSL.

To prevent training collapse, our approach also integrates cross-correlation matrices. These self-supervised methodologies, relying on the siamese network architecture, lead to notable performance improvements despite encountering training inefficiencies.

Our study draws inspiration from SwAV [12]'s multi-crop strategy. However, what distinguishes our approach is our endeavor to utilize a broader array of cropping strategies. This allows us to acquire a collection of view augmentations and utilize these features to generate invariant embeddings without incurring excessive resource usage.

Introducing more than two views to enhance performance imposes a significant memory burden. To address this, SwAV [12]'s proposed multi-crop strategy that incorporates two augmented images at normal resolutions along with multiple (V in number) supplementary low-resolution images. These low-resolution images generally focus on a smaller region of the complete image, thereby minimizing the computational load. Due to the possible introduction of biases through direct resizing, SwAV [12] adopts a mixed approach that involves smaller scales to mitigate bias.

While methods like MoCo [19] primarily focus on improving negative pairs, an alternative approach to enhancing performance is by increasing the number of positive pairs per image. The multicrop technique, initially introduced in SwAV [12], tackles this issue by incorporating smaller crops (96×96)

in addition to the usual two larger ones (224×224). Instead of only comparing the two larger crops or all possible crop pairs, each large crop is matched against all other crops (both large and small). As a result, with 2 large crops and N small crops, the invariance loss is computed 2(N–1) times, amplifying the signal associated with positive pairs.

While the number of additional crops may vary (e.g., 10 in Mugs [20] compared to 6 in SwAV [12]), their direct utilization unavoidably prolongs training time and increases memory consumption. To address this issue, SwAV [12] mitigates memory usage by employing 160×160 large crops and 4 96×96 smaller crops. This approach incurs only a 25% increase in training time compared to the standard configuration with two 224×224 crops, while yielding a 4-point performance improvement. Therefore, the multi-crop strategy is highly successful in improving performance with a minimal increase in computational demands. This approach has become prevalent in recent studies [21–23]. Noteworthy, some studies have observed slight performance enhancements [24], resulting in just a 0.3 point improvement.

Other approaches have surfaced to alleviate the computational load associated with introducing additional crops to the encoder. These methods utilize nearest-neighbors in the embedding space. In Nearest-Neighbor Contrastive Learning of visual Representations (NNCLR) [25], the corresponding positive crop is replaced with its nearest neighbor in the latent space. Conversely, Mean Shift (MSF) [26] establishes a k-nearest neighbor (k-NN) graph in the embedding space, achieving a comparable outcome to multi-crop by enhancing signals associated with positive pairs. Unified Vison Contrastive Learning (UniVCL) [27] further utilizes this strategy, incorporating augmentation techniques such as edge or node masking with a k-NN graph in the latent space. All these approaches greatly improve performance with lower computational requirements compared to multi-crop. In MSF, the adoption of a k-NN graph results in a mere 6% increase in training time.

Contrary to our more generalized and simplified approach, we push this concept to its utmost limit. We segment images into a predefined number of small patches with overlaps and apply data augmentation. The aim is to acquire feature groups enriched with more diverse characteristics, enhancing the performance of SSL. Empirical findings highlight significant advancements from this methodology, allowing our GSSSL technique to achieve profound convergence within a limited number of training epochs, a feat not realized by other dominant SSL methods.

3. Method

In this research, setting itself apart from traditional SSL techniques, the GSSSL method adopts a more enriched feature-centric approach. It harnesses the feature sets derived from both branches of the siamese network to compute the cross-correlation matrix loss. Additionally, an augmented triplet loss [28] is utilized to gauge the distance between the features of the positive and negative feature groups [29]. Algorithm 1 shows the procedure for implementing this methodology.

To elaborate, for a given image x, we generate two distorted views using data augmentation techniques similar to those used in other SSL methods. Subsequently, these distorted views, denoted as a and b, are further divided into n fixed-sized image patches through overlapping random cropping. Each cropped image patch is subsequently augmented using the z_n image patches denoted as $a_1, \ldots, a_n$ and $b_1, ..., b_n$. Here, a_i and b_i respectively represent the $i - th$ augmented image patch from a and b.

Algorithm 1 GSSSL PyTorch Pseudocode

Require: x: Batch of input data.
Require: F: encoder network.
Require: A: Augmented images from x.
Require: B: Another set of augmented images from x.
Require: $A_1, \ldots, A_n$: n image patches from A, extracted using overlapping random crops.
Require: $B_1, \ldots, B_n$: n image patches from B, extracted using overlapping random crops.

for $x = 1$ to *loaders* **do**
  A, B = augment(x)
  // augment n fixed-size image patches
  $A_1, \ldots, A_n$ = extract patches & augment(A)
  $B_1, \ldots, B_n$ = extract patches & augment(B)
  // calculate projection
  $z_1, \ldots, z_n = F(A_1), \ldots, F(A_n)$
  $w_1, \ldots, w_n = F(B_1), \ldots, F(B_n)$
  // calculate loss
  L_{BT1} = total($L_{BT}(z_i, \bar{W})$ for i in range(n))
  L_{BT2} = total($L_{BT}(w_i, \bar{Z})$ for i in range(n))
  $Loss_{L_{BT}} = L_{BT1} + L_{BT2}$
  $Loss = Loss_{L_{BT}} + Loss_{rank-k}(\bar{Z}, \bar{W}, C)$
  // optimization step
  $Loss$.backward()
  $Optimizer$.step()
end for

For the augmented image patches a_i and b_i, we obtain their embeddings h_{a_i}, h_{b_i}, and projections z_{a_i}, z_{b_i}. Here, $h_{a_i} = f(a_i; \theta)$, $h_{b_i} = f(b_i; \theta)$, $z_{a_i} = g(h_{a_i})$, and $z_{b_i} = g(h_{b_i})$. Finally, we normalize the learned projections z_{a_i}, z_{b_i}. The function $f(\cdot; \theta)$ represents a deep neural network (e.g., ResNet-18) [30] with trainable parameters θ, g is a simpler neural network with only two fully connected layers. We define F as $F = g(f(\cdot; \theta))$. The entire process is illustrated in Figure 1.

During training, for a batch of d images we denote as $A = [a^1, ..., a^d]$ and $B = [b^1, ..., b^d]$, where a^j and b^j represent the $j - th$ image in the batch, and we first augment the images as described above to obtain $A_1, ..., A_n$ and $B_1, ..., B_n$, where $A_i = [a_i^1, ..., a_i^d]$, $B_i = [b_i^1, ..., b_i^d]$. Next, we pass the augmented image patches into the encoder to get the features $z_i = F(A_i)$ and $w_i = F(B_i)$, and concatenate them into $Z = [z_1, ..., z_n]$ and $W = [w_1, ..., w_n]$. Of course, we can also obtain $\overline{Z}$ and $\overline{W}$ from Z and W . We believe that $\overline{Z}$ and $\overline{W}$ encompass more comprehensive feature information.

In this work, we utilize cross-correlation matrices to prevent model collapse:

$$L_{BT}(W, Z) = \sum_i (1 - C_{ii}(W, Z))^2 + \lambda \sum_i \sum_{j \neq i} C_{ij}(W, Z)^2 \tag{3.1}$$

Here, λ is a positive constant that balances the importance of the first and second terms in the loss function, and the matrix C, which differs from the one used in Barlow Twins [8], is computed along the batch dimension between the projections z_i and the averaged projections $\overline{W}$, as well as between the

projections w_i and the averaged projections $\overline{Z}$. Additionally, C is the cross-correlation matrix computed between the outputs of the two identical networks along the batch dimension:

$$C_{\mathrm{ij}}(W,Z) = \frac{\sum_d Z_{d,i}\overline{W}_{d,j}}{\sqrt{\sum_d (Z_{d,i})^2 \sum_d \left(\overline{W}_{d,j}\right)^2}} \tag{3.2}$$

Therefore, we obtain the first loss function:

$$Loss_{L_{BT}} = L_{BT1}(Z_i, \overline{W}) + L_{BT2}(W_i, \overline{Z}) \tag{3.3}$$

Let the feature dimension of the image be k, and the dimensions of each matrix are as follows: $C \in R^{k\times k}$, $z_i \in R^{d\times k}$, $w_i \in R^{d\times k}$, $Z = [z_1, ..., z_n] \in R^{n\times d\times k}$, $W = [w_1, ..., w_n] \in R^{n\times d\times k}$, where d denotes the number of images in a batch.

Furthermore, we introduce a rank-k triplet loss [24] to mitigate misclassification of negative feature pair groups. A cosine similarity matrix is constructed between the averaged embeddings $\overline{Z}$ and $\overline{W}$. Positive feature pair groups are selected from the corresponding positions in $\overline{Z}$ and $\overline{W}$, while negative feature pair groups are assembled from alternate positions. We propose that the diagonal elements of this matrix should surpass the $k-th$ largest value within their respective rows, a departure from the principles of the hardest triplet loss.

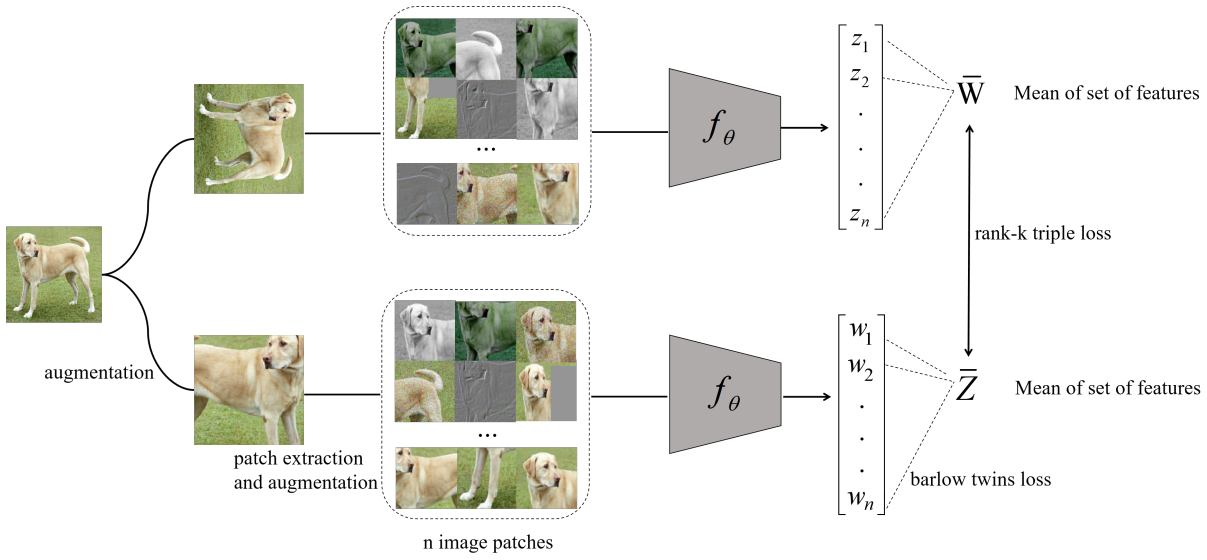

Figure 1. During the training process, two distinct data augmentation operations are randomly sampled from the same family of augmentations and applied to each data example to acquire two correlated views. Then, the image is randomly cropped into n fixed-size image patches with overlapping. We subsequently apply data augmentation to these n fixed-size image patches, including color jitter, greyscale, horizontal flip, Gaussian blur, and solarization. Similar to other SSL methods, these image patches are then fed into the encoder f to obtain representations z, w.

The "hardest triplet loss" is a loss function employed in training deep learning models, primarily focused on enhancing feature representations within embedding spaces. Built upon the foundation of

triplet loss, it addresses the challenge of selecting "easy" and "hard" sample pairs to improve model performance. In triplet loss, a training sample is represented as a triplet comprising an anchor sample, a positive sample, and a negative sample. The goal is to ensure that the distance between the anchor and positive samples is smaller than that between the anchor and negative samples. However, in practice, many sample pairs can easily satisfy this condition, diminishing the effectiveness of the loss function.

To counter this issue, the hardest triplet loss introduces the concept of hard example mining. Specifically, it chooses sample pairs that are more challenging to distinguish-those in which the negative sample is closer to the anchor sample. This approach aims to emphasize samples that are harder to differentiate, enhancing the discriminative ability of the embedding space.

In implementation, the hardest triplet loss first computes distances between all possible positive and negative sample pairs, subsequently selecting the most challenging pairs to calculate the loss. This ensures that the model focuses on pairs that are harder to classify when updating parameters, thereby enhancing the separability of the embedding space.

However, the application of the hardest triplet loss may be limited in unsupervised learning [31], as it often lacks class label information to determine relationships between positive and negative samples. Consequently, we introduce the rank-k triplet loss [24] to circumvent the misclassification of two samples from the same class as negative feature pair groups. Therefore, we obtained a second loss function, which computes the cosine similarity matrix between $\overline{Z}$ and $\overline{W}$ via dot product. Specifically, we extract the k-th largest value from the similarity matrix to represent the k-th nearest negative sample similarity. The average loss across all samples is then calculated using the triplet loss.

$$Loss_{rank-k}(Z, W, C) = \frac{1}{d}\sum_{i=1}^{d} max(0, C + sim(\overline{Z}_i, \overline{W}_{i,rank-k}) - sim(\overline{Z}_i, \overline{W}_i)) \tag{3.4}$$

Here, C is a margin deciding whether or not to drop a triplet. d is the number of samples in the batch size.

In this work, we distinguish ourselves from traditional SSL by offering a broader range of views and utilizing the feature sets from the outputs of the two branches of the siamese network to compute the cross-correlation matrix, while simultaneously minimizing redundancy among these components. We also employ the rank-k triplet loss to prevent misjudgment of negative feature set pairs.

4. Experiments

In this section, we conduct experiments to validate the effectiveness of the GSSSL method. To begin, in Section 4.1, we introduce the experimental setup. Next, in Section 4.2, we present the results of linear evaluations performed at different epochs. Finally, in Section 4.3, we perform ablation experiments on the number of image patches to emphasize the significance of the number of image patches in the convergence of the GSSSL method.

4.1. Experimental setup

Datasets: We validate the effectiveness of GSSSL using four benchmark datasets, namely, CIFAR-10, CIFAR-100, Tiny ImageNet, and ImageNet-100. For each of these datasets, we employ unlabeled training images for SSL. These datasets encompass 10, 100, 200, and 100 classes,

respectively. CIFAR-10 and CIFAR-100 consist of 50,000 training images and 10,000 test images, with dimensions $32 \times 32 \times 3$. Tiny ImageNet contains 200 classes, 100,000 training images, 10,000 test images, with image dimensions of $64 \times 64 \times 3$. ImageNet-100, a subset of ImageNet, comprises 100 classes, around 126,600 training images, and 5,000 test images, with dimensions 224×224. To validate the effectiveness of GSSSL, we compare it against state of the art SSL methods: BYOL [13], SimCLR [7], Variance-Invariance-Covariance Regularization (VICReg) [23], SwAV [12], Barlow Twins [8], and Mixed Barlow Twins [32].

Image Augmentations: Each input image undergoes two transformations to produce the distorted views shown in Figure 1. The augmentation pipeline includes random cropping, resizing to 224×224, horizontal flipping, color jittering, conversion to grayscale, Gaussian blurring, and solarization. While the first two transformations (cropping and resizing) are always applied, the last five are randomly applied with certain probabilities. These probabilities differ for the last two transformations (blurring and solarization). We employ the same augmentation parameters as BYOL [13]. After obtaining the two distorted views, image patches are extracted and subjected to the same data augmentation used in VICReg.

Architecture: The encoder consists of a ResNet-18 network (excluding the final classification layer, yielding 2048 output units) followed by a projector network. The projector network comprises two linear layers, each containing 4096 hidden units and 512 output units. Batch normalization layers and rectified linear units follow the first two layers of the projector.

Optimization: We follow the optimization protocol outlined in BYOL [13]. Employing the Layer-wise Adaptive Rate Scaling (LARS) optimizer [33], we train for 30 epochs with a batch size of 100. The batch sizes of compared methods are as follows: SimCLR, BYOL, and SwAV use a batch size of 4096. VICReg and Barlow Twins use a batch size of 2048. Additionally, Mixed Barlow Twins uses a batch size of 256. The learning rate for weights is set to 0.3, while biases and batch normalization parameters have a learning rate of 0.0048. The learning rate is scaled by the batch size and divided by 256. We incorporate a 10-epoch learning rate warm-up, followed by a learning rate reduction by a factor of 1000 using a cosine decay schedule. The optimal trade-off parameter (λ) for the loss function is determined to be 5×10^{-3}. A weight decay parameter of 1.5×10^{-6} is employed. Biases and batch normalization parameters are exempt from LARS adaptation and weight decay.

4.2. Experiment analysis

In this subsection, we assess the performance of GSSSL through linear evaluations conducted on CIFAR-10, CIFAR-100, Tiny ImageNet, and ImageNet-100 at various epochs, using a patches setting of 10. Following the linear evaluation protocol, we train linear classifiers on top of the ResNet18 backbone pretrained with GSSSL and other SSL methods for different epochs.

In this study, considering the potential increase in computational overhead with an increase in the number of image patches, experiments were conducted to compare the time consumed at convergence. The time unit is measured in minutes. Table 1 displays the linear evaluation results at different epochs compared to other methods. Table 2 displays the computation time for different methods on various datasets. The results illustrate that GSSSL demonstrates rapid convergence within a few epochs, outperforming the other comparative methods that require more training epochs. The linear evaluation outcomes across the datasets emphasize GSSSL's enhanced convergence, indicating its ability to learn more effective representations.

Table 1. Accuracy of different methods on various datasets. The underline indicates the highest accuracy at 1000/400 epochs. The performance of our method on various datasets is shown in bold.

Methods	CIFAR-10		CIFAR-100		Tiny ImageNet		ImageNet-100	
	Accuracy	Epochs	Accuracy	Epochs	Accuracy	Epochs	Accuracy	Epochs
SimCLR	0.910	1000	0.662	1000	0.488	1000	0.776	400
BYOL	<u>0.926</u>	1000	<u>0.708</u>	1000	0.510	1000	<u>0.802</u>	400
VICReg	0.921	1000	0.685	1000	–	–	0.792	400
SwAV	0.923	1000	0.658	1000	–	–	0.740	400
Barlow Twins	0.922	1000	0.702	1000	0.503	1000	0.791	400
Mixed Barlow Twins	0.919	1000	0.685	1000	<u>0.514</u>	1000	-	-
GSSSL	**0.925**	**30**	**0.709**	**30**	**0.514**	**30**	**0.798**	**30**

Table 2. Time required to achieve the accuracy of Table 1 for different methods on various datasets. The performance of our method on various datasets is shown in bold.

Methods	CIFAR-10	CIFAR-100	Tiny ImageNet	ImageNet-100
	Time (minutes)	Time (minutes)	Time (minutes)	Time (minutes)
SimCLR	457	499	914	461
BYOL	458	534	918	465
VICReg	543	669	–	528
SwAV	1080	1095	–	1088
Barlow Twins	527	675	1054	530
Mixed Barlow Twins	353	390	781	–
GSSSL	**135**	**164**	**284**	**343**

Specifically, compared to SimCLR, BYOL, VICReg, SwAV, and Barlow Twins, our method exhibits comparable accuracy on CIFAR-10 and CIFAR-100 datasets while significantly reducing training time. Although our method slightly lags behind some competitors in terms of accuracy on Tiny ImageNet and ImageNet-100 datasets, it excels in training efficiency by markedly shortening the training time. The experimental data indicates that the computational time of our method over 30 training epochs is roughly one-third of the minimum computational time of other methods, further confirming the significant advantage of our method in terms of time efficiency.

These results indicate that our proposed GSSSL method has certain advantages in enhancing image representation learning and achieves significant improvements in training efficiency.

4.3. Ablation study for different patch sizes

In this subsection, we evaluate the performance of GSSSL on CIFAR-10 and CIFAR-100 using varying patch sizes. We set the number of training epochs for SSL to 30 and employ a batch size of 100 for the ResNet18 backbone. The comparison results for different patch sizes are presented in Table 3. Notably, GSSSL exhibits improved performance as the patch size increases.

We conducted ablation experiments on the selection of loss functions in the key modules. Table 4 presents the results of training for 30 epochs on the CIFAR-10 and CIFAR-100 datasets, respectively. From the results, it can be observed that the combination of the proposed loss functions has certain advantages.

Table 3. Accuracy with different number of image patches on different datasets.

patches	CIFAR-10	CIFAR-100
	30 Epochs	30 Epochs
5	0.907	0.689
10	0.925	0.709
15	0.928	0.717
20	0.929	0.724
25	0.929	0.730
30	0.930	0.733
35	0.931	0.733

Table 4. Comparison of results on CIFAR-10 and CIFAR-100

	CIFAR-10	**CIFAR-100**
Loss_{BT}	0.911	0.690
Loss_{BT} + Loss_{rank-k}	0.925	0.709

We generalize the applicability of the proposed method by applying it to SimCLR and MoCo and conducting experiments on the CIFAR-10 dataset. As shown in Table 5, similar significant improvements in performance and efficiency are observed from the experimental results.

Table 5. Performance comparison of SimCLR and MoCo with multi-image patches on CIFAR-10 dataset.

Method	**Accuracy**	**Time (minutes)**
SimCLR (1000 Epoches)	0.910	457
MoCo (800 Epoches)	0.897	367
SimCLR + Multi-Image Patches (30 Epochs)	0.924	156
MoCo + Multi-Image Patches (30 Epochs)	0.912	84

5. Discussion

This paper introduces a novel SSL approach termed GSSSL. The primary objective of this study is to address the limitations of single-view features in SSL. The proposed method incorporates a siamese network architecture, utilizing feature groups enriched with diverse characteristics for distortion-invariant embedding. It synergistically combines the cross-correlation matrix loss from the cutting-edge Barlow Twins method and rank-k triplet loss to enhance representation learning performance, particularly within a limited number of training epochs. We conduct linear evaluation

comparisons under various SSL configurations and ablation studies with different numbers of image patches to validate the efficacy of GSSSL. Experimental outcomes indicate that our approach achieves remarkable convergence within a few epochs, outperforming state of the art SSL techniques across multiple datasets.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, et al., Graph self-supervised learning: A survey, *IEEE Trans. Knowl. Data Eng.*, **35** (2022), 5879–5900. https://doi.org/10.1109/TKDE.2022.3172903
2. S. Liu, A. Mallol-Ragolta, E. Parada-Cabeleiro, K. Qian, X. Jing, A. Kathan, et al., Audio self-supervised learning: A survey, *Patterns*, **3** (2022), 100616. https://doi.org/10.1016/j.patter.2022.100616
3. S. Albelwi, Survey on self-supervised learning: auxiliary pretext tasks and contrastive learning methods in imaging, *Entropy*, **24** (2022), 551. https://doi.org/10.3390/e24040551
4. L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (2021), 4037–4058. https://doi.org/10.1109/TPAMI.2020.2992393
5. P. Fang, X. Li, Y. Yan, S. Zhang, Q. Kang, X. Li, et al., Connecting the dots in self-supervised learning: A brief survey for beginners, *J. Comput. Sci. Technol.*, **37** (2022), 507–526. https://doi.org/10.1007/s11390-022-2158-x
6. K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum Contrast for unsupervised visual representation learning, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 9726–9735. https://doi.org/10.1109/CVPR42600.2020.00975
7. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in *Proceedings of the 37th International Conference on Machine Learning*, (2020), 1597–1607.
8. J. Zbontar, L. Jing, I. Misra, Y. LeCun, S. Deny, Barlow Twins: Self-supervised learning via redundancy reduction, in *Proceedings of the 38th International Conference on Machine Learning*, (2021), 12310–12320.
9. S. Huang, X. Jin, Q. Jiang, L. Liu, Deep learning for image colorization: Current and future prospects, *Eng. Appl. Artif. Intell.*, **114** (2022), 105006. https://doi.org/10.1016/j.engappai.2022.105006

10. M. Xu, S. Yoon, A. Fuentes, D. S. Park, A comprehensive survey of image augmentation techniques for deep learning, *Pattern Recognit.*, **137** (2023), 109347. https://doi.org/10.1016/j.patcog.2023.109347
11. C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *J. Big Data*, **6** (2019). https://doi.org/10.1186/s40537-019-0197-0
12. M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, (2020), 9912–9924.
13. J. B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, et al., Bootstrap Your Own Latent-a new approach to self-supervised learning, in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, (2020), 21271–21284.
14. X. Chen, K. He, Exploring simple siamese representation learning, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 15745–15753. https://doi.org/10.1109/CVPR46437.2021.01549
15. S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, (2005), 539–546. https://doi.org/10.1109/CVPR.2005.202
16. J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, et al., Signature verification using a "siamese" time delay neural network, *Int. J. Pattern Recognit. Artif. Intell.*, **7** (1993), 669–688. https://doi.org/10.1142/S0218001493000339
17. S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 4353–4361. https://doi.org/10.1109/CVPR.2015.7299064
18. X. Chen, K. He, Exploring simple siamese representation learning, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 15745–15753. https://doi.org/10.1109/CVPR46437.2021.01549
19. R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, et al., A cookbook of self-supervised learning, preprint, arXiv:2304.12210.
20. P. Zhou, Y. Zhou, C. Si, W. Yu, T. K. Ng, S. Yan, Mugs: A multi-granular self-supervised learning framework, preprint, arXiv:2203.14415.
21. M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, et al., Emerging properties in self-supervised vision transformers, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 9630–9640. https://doi.org/10.1109/ICCV48922.2021.00951
22. J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, et al., iBOT: Image bert pre-training with online tokenizer, preprint, arXiv:2111.07832.
23. A. Bardes, J. Ponce, Y. LeCun, VICReg: Variance-Invariance-Covariance Regularization for self-supervised learning, preprint, arXiv:2105.04906.
24. G. Wang, K. Wang, G. Wang, P. H. S. Torr, L. Lin, Solving inefficiency of self-supervised representation learning, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 9485–9495. https://doi.org/10.1109/ICCV48922.2021.00937

25. D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, A. Zisserman, With a little help from my friends: Nearest-Neighbor Contrastive Learning of visual Representations, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 9568–9577. https://doi.org/10.1109/ICCV48922.2021.00945

26. S. A. Koohpayegani, A. Tejankar, H. Pirsiavash, Mean Shift for self-supervised learning, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 10306–10315. https://doi.org/ 10.1109/ICCV48922.2021.01016

27. S. Tang, F. Zhu, L. Bai, R. Zhao, C. Wang, W. Ouyang, Unifying visual contrastive learning for object recognition from a graph perspective, in *Computer Vision-ECCV 2022*, (2022), 649–667. https://doi.org/10.1007/978-3-031-19809-0_37

28. F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 815–823. https://doi.org/10.1109/CVPR.2015.7298682

29. R. Miao, Y. Yang, Y. Ma, X. Juan, H. Xue, J. Tang, et al., Negative samples selecting strategy for graph contrastive learning, *Inf. Sci.*, **613** (2022), 667–681. https://doi.org/10.1016/j.ins.2022.09.024

30. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90

31. H. B. Barlow, Unsupervised learning, *Neural Comput.*, **1** (1989), 295–311. https://doi.org/10.1162/neco.1989.1.3.295

32. W. G. C. Bandara, C. M. De Melo, V. M. Patel, Guarding Barlow Twins against overfitting with mixed samples, preprint, arXiv:2312.02151.

33. Y. You, I. Gitman, B. Ginsburg, Large batch training of convolutional networks, preprint, arXiv:1708.03888.

AIMS Press