Benchmarking Self-Supervised Learning on STL-10: SimCLR Vs BYOL

***Siddharth Maurya, Vijay Kumar**

Dept of Software Engineering, Delhi technological university

***Corresponding Author**

DOI: https://doi.org/10.51584/IJRIAS.2025.10120062

Received: 25 December 2025; Accepted: 30 December 2025; Published: 16 January 2026

ABSTRACT

Self-supervised learning (SSL) has emerged as an effective paradigm for learning visual representations without reliance on labeled data. This study presents a controlled benchmark of two widely adopted SSL methods, SimCLR and BYOL, evaluated on the STL-10 dataset. Both methods are implemented using an identical ResNet-18 backbone and trained under matched computational and optimization settings. Representation quality is assessed through linear probing and k-NN classification. Under these constraints, SimCLR demonstrates stronger performance than BYOL, achieving a linear probe accuracy of 71.21% compared to 69.90% for BYOL. These results emphasize practical considerations in SSL benchmarking and highlight performance trade-offs that arise under resource-limited training regimes.

Key Words: BYOL, SimCLR, SSL

INTRODUCTION

Deep learning has achieved substantial success in computer vision, largely driven by supervised learning on large labeled datasets. However, the cost and scalability of manual annotation limit broader applicability. Self-supervised learning (SSL) addresses this challenge by learning representations from unlabeled data. While early SSL methods based on handcrafted pretext tasks showed limited scalability, recent contrastive and self-distillation approaches have achieved performance competitive with supervised models [1][2].

Among modern SSL methods, contrastive approaches such as SimCLR learn representations through instance discrimination using strong data augmentations, whereas non-contrastive methods like BYOL remove explicit negatives and rely on asymmetric architectures with exponential moving average updates [1][2]. Prior work indicates that these paradigms exhibit distinct optimization and scalability characteristics [3][4].

Despite extensive large-scale benchmarks, evaluating SSL under constrained computational settings remains challenging due to reliance on large batch sizes and extensive hyperparameter tuning. This work addresses this gap through a controlled comparison of SimCLR and BYOL on STL-10 using a shared ResNet-18 backbone and matched training settings, with representation quality evaluated via linear probing and k-NN classification.

Notably, much of the existing literature reports 70–80% linear probe accuracy using high-capacity backbones such as ResNet-50 and long training schedules [1][2]. In contrast, this study demonstrates that comparable performance (~70%) can be achieved with a significantly lighter ResNet-18 model trained for only 50 epochs, highlighting the efficiency and practical viability of SSL under limited computational budgets. Overall, this work provides a reproducible, multi-seed benchmark that aligns small-scale observations with prior large-scale findings while explicitly documenting experimental constraints [5].

Related Work

Self-supervised learning (SSL) has emerged as a prominent paradigm for learning visual representations without explicit human supervision. Early SSL methods relied on handcrafted pretext tasks such as image rotation

prediction, jigsaw puzzles, or colorization. While effective to a degree, these approaches often struggled to scale and generalize to complex downstream tasks. Comprehensive overviews of self-supervised learning for image classification categorize modern approaches into contrastive, clustering-based, and self-distillation frameworks, each with distinct trade-offs in compute, stability, and downstream performance [6].

Contrastive Learning Approaches

Contrastive learning has become one of the most influential directions in self-supervised learning (SSL) for visual representation learning, encompassing a wide range of architectural designs and training strategies [1] [7]. Among these methods, SimCLR introduced a simple yet effective framework that demonstrated competitive performance with supervised learning when trained at scale [1]. SimCLR relies on strong data augmentations and a contrastive loss that maximizes agreement between different augmented views of the same image while minimizing agreement between representations of different images. A key architectural component of SimCLR is the use of a projection head, which maps representations into a latent space where the contrastive objective is applied; empirical results show that this design choice significantly improves representation quality, particularly under linear evaluation protocols [1][8].

Subsequent contrastive methods such as MoCo, MoCo-v2, and SwAV further refined contrastive learning by introducing momentum encoders, memory queues, and clustering-based objectives to improve scalability and training efficiency [1][3][9]. These methods can be broadly categorized based on encoder symmetry, the use of momentum updates, and mechanisms for negative sample generation, reflecting trade-offs between scalability, stability, and computational cost [7]. However, a common limitation across contrastive approaches is their reliance on large batch sizes or memory mechanisms to provide sufficient negative samples, which often increases computational and memory requirements and limits practicality in resource-constrained environments [8][10]. Recent work has shown that these limitations can be mitigated through efficient design choices, enabling contrastive methods to achieve strong representation quality even under reduced training budgets and smaller batch sizes [11].

Non-Contrastive and Self-Distillation Methods

Bootstrap Your Own Latent (BYOL) [2] marked a significant departure from contrastive learning by demonstrating that high-quality representations can be learned without explicit negative samples. BYOL employs an online network and a target network, where the target network parameters are updated as an exponential moving average of the online network parameters. The model is trained to predict the target network's representations of one augmented view from the online network's representations of another augmented view of the same image.

Despite initial skepticism regarding representational collapse, BYOL empirically showed strong performance across multiple benchmarks, often outperforming contrastive counterparts such as SimCLR. Subsequent works, including SimSiam and DINO, further explored non-contrastive and self-distillation paradigms, reinforcing the observation that carefully designed architectural asymmetry and optimization dynamics can prevent collapse without negatives.

Benchmarking and Evaluation Protocols

Benchmarking SSL methods typically relies on standardized evaluation protocols such as linear probing, k-nearest neighbor (k-NN) classification, and transfer learning to downstream tasks. Large-scale benchmarks reported in the literature commonly use ImageNet and extensive compute resources, including multi-node GPU clusters or TPUs. While these benchmarks provide valuable insights into scalability and upper-bound performance, they often obscure practical considerations faced by smaller research groups.

Several studies and surveys have emphasized the need for controlled and transparent benchmarking in self-supervised learning, particularly under realistic computational budgets. These works highlight that reported performance is highly sensitive to design choices such as hyperparameter selection, data augmentation strategies, and optimization settings, especially when training is conducted on smaller datasets or limited hardware. Recent

benchmarking analyses further demonstrate that such sensitivities extend to random initialization and evaluation protocol, showing that small performance differences can be unstable across runs and therefore require multi-seed evaluation and careful statistical reporting to ensure reproducibility and meaningful comparison [12]. Positioning of the Present Work

In contrast to large-scale benchmarking efforts, the present study focuses on evaluating SimCLR and BYOL under constrained computational settings using the STL-10 dataset. Training is conducted on commonly accessible hardware, including Kaggle T4 GPUs and consumer-grade RTX 3050 GPUs. By maintaining identical backbone architectures and evaluation protocols, this work aims to provide a fair and reproducible comparison between contrastive and non-contrastive SSL methods in a realistic experimental regime.

Rather than introducing methodological innovations, this study contributes a practical benchmark that complements existing large-scale evaluations. It highlights how performance trends observed in high-resource settings translate to smaller datasets and limited hardware, thereby offering useful insights for practitioners and researchers operating outside large industrial or academic compute infrastructures.

METHODOLOGY

This section describes the dataset, self-supervised learning methods, training setup, evaluation protocols, and experimental design choices used in this study. Special attention is given to reproducibility, statistical robustness, and practical constraints.

Self-Supervised Learning Methods

This study benchmarks two widely adopted self-supervised learning methods: SimCLR and BYOL.

SimCLR employs contrastive learning using a normalized temperature-scaled cross-entropy loss. The model is trained to maximize agreement between different augmented views of the same image while minimizing similarity between representations of different images.

BYOL (Bootstrap Your Own Latent) removes the need for negative samples by using an online network and a target network. The target network parameters are updated using an exponential moving average of the online network parameters, and the online network is trained to predict the target network's representations.

To ensure a fair comparison, both methods share common architectural and training components:

- Backbone architecture: ResNet-18
- Projection head: two-layer multilayer perceptron (MLP)
- Optimization strategy: AdamW optimizer with cosine learning rate scheduling

Hyperparameters such as batch size, learning rate, number of epochs, and augmentation strategies are selected following commonly reported settings in prior SSL literature, subject to hardware limitations.

Training Protocol and Statistical Robustness

Recent SSL benchmarking studies emphasize that single-run evaluations are insufficient due to the sensitivity of self-supervised models to random initialization, data ordering, and optimization noise. Reporting performance across multiple random seeds, along with variance estimates, is increasingly recommended to ensure statistical robustness and reproducibility.

Following these recommendations, each model in this study is trained using three different random seeds, and results are reported using mean accuracy, standard deviation, and variance. This design choice mitigates the risk of drawing conclusions from favorable single-run artifacts and aligns with best practices in contemporary SSL evaluation.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue XII December 2025

Performance is reported using:

- Mean accuracy
- Standard deviation
- Variance

Evaluation Protocol

Representation quality is evaluated using two standard SSL evaluation strategies:

Linear Probing:

A linear classifier is trained on top of frozen representations learned by the SSL model. This protocol measures the linear separability of learned features and is widely used as a primary evaluation metric in SSL literature.

k-Nearest Neighbor (k-NN) Classification:

A non-parametric evaluation is performed using a k-NN classifier with k = 20. This approach assesses representation quality without additional training and provides insight into the intrinsic structure of the learned feature space.

Both evaluation methods are applied consistently across all runs and models.

Dataset Scope and Experimental Constraints

The experimental evaluation in this study is limited to the STL-10 dataset. While this dataset provides a meaningful balance between complexity and computational feasibility, reliance on a single dataset restricts the generalizability of conclusions.

This limitation arises from constrained computational resources, including limited access to high-end GPUs and restricted training time. Expanding evaluation to additional datasets such as CIFAR-10, CIFAR-100, or ImageNet subsets would require significantly greater computational capacity.

Despite this limitation, the controlled experimental design ensures that observed performance differences between SimCLR and BYOL are attributable to methodological differences rather than confounding variables.

Experiments

Experimental Setup

All experiments were conducted under limited computational resources to reflect realistic constraints faced by smaller research labs and student researchers. Training was performed using NVIDIA Tesla T4 GPUs (Kaggle). No TPU clusters or multi-node GPU setups were used.

Both SimCLR and BYOL were implemented from scratch using PyTorch, ensuring full control over architectural and training choices. Care was taken to keep the experimental conditions as consistent as possible across methods to allow a fair comparison.

Dataset

The STL-10 dataset consists of 96×96 color images across 10 object classes, with 5,000 labeled training images, 8,000 test images, and 100,000 unlabeled images. Its higher resolution compared to CIFAR datasets makes it well suited for evaluating representation learning under moderate data availability while remaining computationally tractable.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue XII December 2025

Self-supervised pretraining was performed on the unlabeled split using standard train/test partitions. No dataset-specific preprocessing was applied beyond the augmentations required for self-supervised learning, ensuring consistency across methods. Downstream performance was evaluated using a linear classifier trained on frozen representations from the labeled training split, following standard linear evaluation protocols in self-supervised learning.

Data Augmentation Strategy

Both methods relied on **strong data augmentation**, which is critical for effective self-supervised representation learning.

The following augmentations were applied to generate two correlated views of each input image:

- Random resized cropping
- Random horizontal flipping
- Color jitter (brightness, contrast, saturation, hue)
- Random grayscale conversion
- Gaussian blur

The same augmentation pipeline was used for both SimCLR and BYOL to eliminate augmentation-induced bias.

Model Architecture

A **ResNet-based backbone** was used as the encoder for both methods. On top of the backbone:

- **SimCLR** employed a **non-linear projection head** consisting of a multi-layer perceptron (MLP), mapping representations to a contrastive embedding space.
- **BYOL** used:
  - An **online network** with an encoder and projection head
  - A **target network** with identical architecture, updated using an **exponential moving average (EMA)** of the online network parameters
  - A predictor network applied only to the online branch

Architectural dimensions and depths were kept identical wherever applicable.

Training Protocol

Both models were trained for **50 epochs** using the same optimizer and learning rate schedule.

Key training details:

- **Batch size:** 512 (SimCLR and BYOL)
- **Optimizer:** Stochastic Gradient Descent (SGD) with momentum
- **Weight decay:** Applied uniformly across methods
- **Learning rate scheduling:** Cosine decay

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue XII December 2025

- **Temperature parameter:** Used only for SimCLR's contrastive loss
- **EMA decay:** Applied to BYOL target network

No method was given additional tuning advantages.

Evaluation Protocol

To evaluate the quality of learned representations, a **linear probing** protocol was employed:

1. The pretrained encoder was frozen.
2. A single linear classifier was trained on top of the frozen representations using labeled STL-10 data.
3. Classification accuracy on the test set was reported as the primary metric.

This protocol measures the **linear separability** of learned features and is a standard evaluation method in self-supervised learning literature.

Multiple Random Seeds and Reproducibility

To account for stochasticity in training, each experiment was conducted using **multiple random seeds**. Specifically, training was repeated with random seeds **40, 41, and 42**.

For each method, results are reported across seeds, and **mean performance and variance** are analyzed in subsequent sections. This approach ensures that observed trends are not artifacts of a single favorable initialization.

Resource Constraints and Practical Considerations

Unlike many prior benchmarks that rely on large-scale GPU or TPU clusters, this study intentionally operates under **restricted compute budgets**. This design choice emphasizes:

- Practical reproducibility
- Accessibility for academic and small-lab settings
- Relevance to edge and resource-limited environments

Despite these constraints, both methods demonstrate competitive performance, validating their applicability beyond large-scale industrial setups.

RESULTS

Linear Probe Performance

The quality of learned representations was evaluated using a linear probing protocol on the STL-10 dataset. Table 1 reports the classification accuracy obtained by training a linear classifier on frozen features learned via self-supervised pretraining.

Table 1- Linear Probe accuracy across multiple seeds.

Method	Epochs	Seed	Linear Probe Accuracy (%)
Sim CLR	50	40	70.79

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue XII December 2025

Sim CLR	50	41	71.50
Sim CLR	50	42	71.34
BYOL	50	40	69.58
BYOL	50	41	70.00
BYOL	50	42	70.14

Across the evaluated random seeds, both methods achieved competitive performance, with accuracies exceeding 69% after 50 epochs of pretraining.

k-NN Evaluation Performance

In addition to linear probing, representation quality was further evaluated using a **k-nearest neighbor (k-NN) classifier**, which provides a non-parametric assessment of the learned feature space. For this evaluation, feature embeddings extracted from the frozen encoders were used to classify test samples based on their nearest neighbors in the training set, with $k = 20$.

Table 2- k-NN accuracy across multiple seeds.

Method	**Epochs**	**Seed**	**k-NN Accuracy (%)**
SimCLR	50	40	66.90
SimCLR	50	41	67.29
SimCLR	50	42	67.36
BYOL	50	40	67.58
BYOL	50	41	67.47
BYOL	50	42	68.66

Across all evaluated seeds, both methods achieved stable k-NN performance, with accuracies exceeding 66%.

Statistical Summary Across Seeds

To reduce the influence of randomness in initialization and data ordering, results were aggregated across multiple seeds. The mean and variability of linear probe accuracy are summarized below.

Table 3- Summary of mean and variability of linear probe accuracy.

Method	**Mean Accuracy (%)**	**Standard Deviation**	**Variance**
SimCLR	71.21	**0.30**	0.09
BYOL	69.90	**0.23**	0.05

Both methods exhibit low variance across runs, indicating stable training behavior under the selected hyperparameters.

The mean accuracy, standard deviation, and variance for the k-NN evaluation are summarized in Table 4.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue XII December 2025

Table 4- Summary of mean accuracy, standard deviation, and variance for the k-NN evaluation

Method	Mean Accuracy (%)	Standard Deviation	Variance
SimCLR	67.18	0.20	0.04
BYOL	67.90	**0.53**	0.28

Training Dynamics: Epochs vs Loss

To analyze optimization behavior, training loss curves for SimCLR and BYOL over 50 epochs are shown in Figures 1 and 2. Both methods converge stably without divergence or collapse across all seeds. SimCLR exhibits a faster early loss reduction due to its contrastive objective, while BYOL shows a smoother, gradual decay driven by its EMA-based bootstrapping. This confirms the robustness of both training pipelines under limited computational resources.

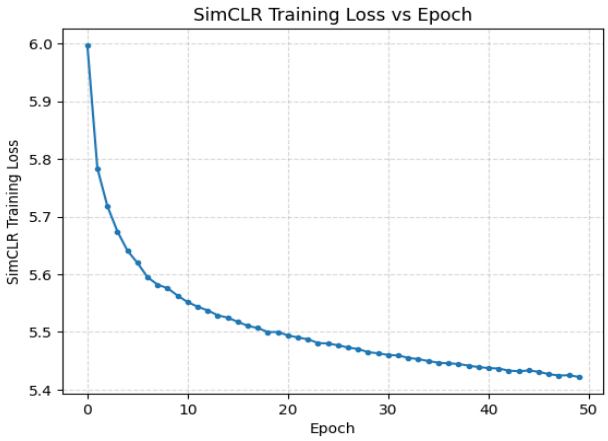

Fig. 1- SimCLR Training Loss Vs Epoch

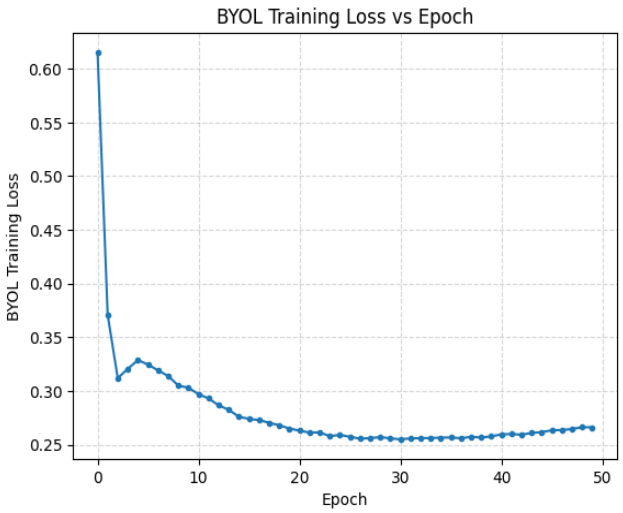

Fig. 2- BYOL Loss Vs Epoch

Feature Space Visualization Using t-SNE

To qualitatively evaluate representation structure, t-SNE was applied to frozen feature embeddings and colored by class labels. As shown in Figure 3, both SimCLR and BYOL learn semantically meaningful spaces with clear class-wise clustering. BYOL exhibits lightly tighter intra-class clusters, while SimCLR shows clearer inter-class separation for some classes, consistent with trends reported in prior SSL literature.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue XII December 2025

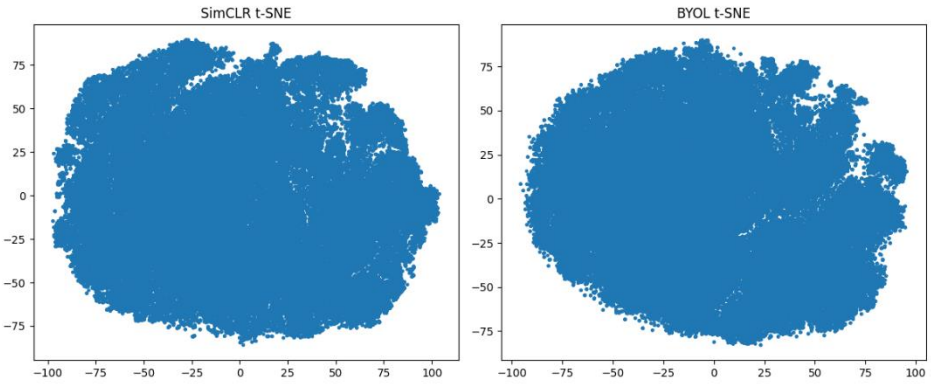

Fig. 3 – SimCLR and BYOL t-SNE plot

Linear Structure Analysis Using PCA

In addition to t-SNE, PCA is used to visualize dominant linear components of the learned representations. Both SimCLR and BYOL capture substantial variance in the top components, indicating effective semantic compression. The similar global structure (Fig. 4) suggests linearly informative feature spaces consistent with strong linear probe performance, with minor differences reflecting their distinct training objectives.

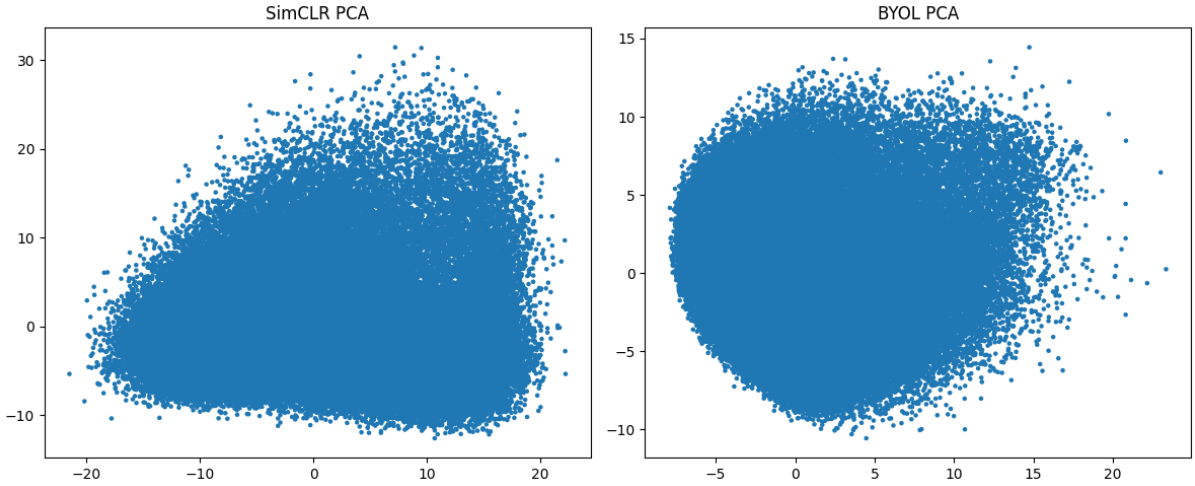

Fig. 4- SimCLR and BYOL PCA plot.

DISCUSSION

This study presents a controlled benchmark of SimCLR and BYOL on the STL-10 dataset under constrained computational resources. While prior literature frequently reports superior performance for BYOL at scale, our results demonstrate that SimCLR achieves marginally higher linear probe accuracy than BYOL at 50 epochs when trained with identical batch sizes and backbone architectures.

Performance Trends Under Limited Training Regimes

The slight advantage of SimCLR is likely due to the limited training budget and reduced hyperparameter exploration. Although prior work often shows BYOL outperforming contrastive methods under long training schedules and large batch sizes (300–1000 epochs), such gains are not guaranteed in short-training regimes. Contrastive methods like SimCLR typically converge faster due to explicit instance discrimination objectives [1][2].

Existing studies show that SSL performance rankings depend on training duration, dataset scale, and computational resources [8][10]. Non-contrastive methods benefit more from extended optimization, while

contrastive approaches can match or outperform them under constrained budgets [9]. Thus, the comparable results observed here reflect regime-dependent behavior rather than a contradiction of prior findings.

Comparative performance

Across both evaluation protocols, SimCLR and BYOL show competitive and complementary behavior. SimCLR slightly outperforms BYOL under linear probing, indicating stronger linear separability, while BYOL achieves better k-NN performance, suggesting improved local neighborhood structure. These trends highlight that representation quality is evaluation-dependent and indicate that, under limited training budgets, both methods learn representations of comparable overall quality, differing mainly in feature organization rather than discriminative strength.

Stability Across Random Seeds

Both methods exhibit low variance across random seeds, indicating stable and reproducible training behavior under the selected configurations. Although SimCLR shows slightly higher mean accuracy, the narrow performance gap and overlapping variance suggest that the difference is not statistically conclusive at this scale. Prior benchmarking studies caution that small accuracy differences between self-supervised methods often fall within statistical variability and should be interpreted carefully rather than as definitive evidence of superiority [12].

Consistency with Prior Work

The overall performance trends observed in this study remain consistent with other studies by [3], [11], which report that BYOL typically matches or exceeds contrastive methods given sufficient compute and training duration. Our findings do not contradict these results but rather contextualize them within practical constraints, highlighting that performance rankings may vary depending on training budgets, dataset scale, and evaluation protocol.

Importantly, the strong performance of SimCLR in this setting underscores its continued relevance as a robust and efficient baseline for self-supervised learning when computational resources are limited.

CONCLUSION

This study presents a controlled benchmark of two prominent self-supervised learning methods, SimCLR and BYOL, on the STL-10 dataset under constrained computational resources. By using identical backbone architectures, training conditions, and evaluation protocols, the work ensures a fair comparison between contrastive and non-contrastive approaches. Both methods learned strong and stable representations, achieving high linear probe performance. With a fixed 50-epoch budget, SimCLR achieved slightly higher accuracy than BYOL, though BYOL remained competitive. The results show that performance depends on training duration and resource constraints, and demonstrate that effective self-supervised learning is feasible without large-scale compute. The results highlight the influence of training duration and resource limitations on performance and show that effective self-supervised learning is feasible without large-scale compute, with contrastive methods offering favorable efficiency in low-resource settings.

Conflict of Interest – None

REFERENCES

1. Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. International Conference on Machine Learning (ICML).(2020)
2. Grill, J.-B., Strub, F., Altché, F., et al. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. NeurIPS. (2020)
3. Chen, X., & He, K. Exploring Simple Siamese Representation Learning. CVPR. (SimSiam) (2021) pg (3-7)

4. Caron, M., Misra, I., Mairal, J., et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV. (DINO) (2021) pg (1-6).
5. Wenwen Qiang, Jingyao Wang, Changwen Zheng et al. On the Universality of Self-Supervised Learning. arXiv:2405.01053v5 . May 2025.
6. Dey, D., Edher, H., Rao, L.M., Saini, D.K. (2026). Self-supervised Learning in Image Classification. In: Senjyu, T., So-In, C., Joshi, A. (eds) Smart Trends in Computing and Communications. SmartCom 2025. Lecture Notes in Networks and Systems, vol 1464. Springer, Singapore. https://doi.org/10.1007/978-981-96-7520-3_23
7. A. Khan, S. AlBarri and M. A. Manzoor, "Contrastive Self-Supervised Learning: A Survey on Different Architectures," 2022 2nd International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 2022, pp. 1-6
8. Purushwalkam, S., & Gupta, A. Demystifying Contrastive Self-Supervised Learning. ECCV Workshops. (2020) pg (1-4).
9. He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. CVPR. (2020) pg(1-8).
10. Oord, A. van den, Li, Y., & Vinyals, O. Representation Learning with Contrastive Predictive Coding. arXiv preprint arXiv:1807.03748. (2018)
11. Addepalli, S., Bhogale, K., Dey, P., Babu, R.V. Towards Efficient and Effective Self-supervised Learning of Visual Representations. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13691. Springer, Cham. https://doi.org/10.1007/978-3-031-19821-2_30 (2022)
12. Markus Marks, Manuel Knott, Neehar Kondapaneni, Elijah Cole, Thijs Defraeye, Fernando Perez-Cruz, Pietro Perona, “A Closer Look at Benchmarking Self-Supervised Pre-training with Image Classification” Jul 2024.