Exploring Tradeoffs of Annotation Cost and Model Accuracy with Contrastive Learning for Yoga Pose Classification

Zolboo Damiran*, Tomokazu Matsui*‡, Yuki Matsuda†‡, Hirohiko Suwa*‡,
Keiichi Yasumoto*‡
*Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan
†Okayama University, Okayama, Japan
‡RIKEN AIP, Tokyo, Japan
Emails: {damiran.zolboo.cw5, m.tomokazu, yukimat, h-suwa, yasumoto}@is.naist.jp

***Abstract*—Yoga pose classification is critical for intelligent environments, health, and fitness applications. This study investigates resource-efficient implementations of classification models using contrastive learning frameworks, including SimCLR, MoCo, and BYOL. We evaluate their performance across varying levels of labeled data, focusing on accuracy, computational efficiency, and robustness. MoCo offers a balanced tradeoff with 87.59% accuracy at 50% labeled data, while BYOL achieves strong results with faster inference. SimCLR, suitable for real-time applications due to faster training, consumes more memory and has slower inference. We apply data augmentation and normalization in the preprocessing pipeline to enhance generalization and address challenges like limited data and class imbalance. These techniques improve the model's resilience and learning efficiency. Our findings guide scalable, energy-efficient, user-centered yoga pose classification models for intelligent environments.**

***Index Terms*—self-supervised learning, contrastive learning, yoga pose classification**

I. Introduction

Yoga pose classification is rapidly evolving, with extremely accurate implications for personal healthcare, fitness tracking, telerehabilitation, and interactive learning systems [1], [2]. Real-time positive feedback, enhanced user engagement, and better exercise safety are all possible through accurately recognizing yoga poses. The capability is appreciable in environments such as smart fitness devices, augmented reality systems, and virtual training platforms. However, classifying yoga poses presents unique challenges because poses are complex, dynamic events that include subtle variations, frequent occlusions, and various body configurations. These challenges make traditional supervised learning methods costly (in terms of time and resources) and less feasible because they require large amounts of labeled data [3].

The need for robust yoga pose classification is growing with the increasing adoption of intelligent systems for fitness and wellness. Systems that detect and analyze body positions can significantly improve user experiences and help prevent injuries by encouraging proper form and posture. However, there is a big problem: insufficient labeled data is available. Labeling large datasets for yoga poses is time-consuming and expensive. Additionally, traditional methods, like keypoint detection models and convolutional neural networks (CNNs), often have trouble dealing with the variations and blocked views that are common in yoga poses. These challenges show the need for other methods to work well with unlabeled data. Self-supervised learning (SSL) has become a promising solution, allowing models to learn functional patterns without needing labeled data. Among SSL techniques, contrastive learning (CL) has been very successful in learning patterns for different computer vision tasks, offering a possible way to address the challenges related to yoga pose analysis.

CL has revolutionized SSL by enabling models to identify and differentiate similar and dissimilar data points without the need for labeled examples [4], [5]. SimCLR and MoCo lead the CL frameworks with their contributions to representation learning. While mass data augmentation paired with large batch sizes facilitates significantly robust representation learning from high-dimensional geometries (mainly to capture subtle changes in angle) [6], it comes at a computational cost that limits its scalability. In contrast to SimCLR, MoCo incorporates a momentum encoder and a dynamic memory queue, which enable training with smaller batch sizes and reduced computational overhead [7]. There is clear evidence that such a combined approach offers notable advantages for applications such as yoga pose classification. A strong alternative is BYOL, which eliminates the need for negative samples and is particularly useful for resource-constrained applications [6].

This study evaluates the performance of these contrastive learning frameworks—SimCLR, MoCo, and BYOL—for yoga pose classification, focusing on their ability to operate efficiently with limited labeled data. In doing so, we aim to address the unique challenges of data scarcity, computational efficiency, and scalability. We consider the impact of preprocessing strategies and look at the tradeoffs between annotation cost and model performance. Meanwhile, we address practical concerns in implementing these models in today's intelligent environments to provide useful suggestions for the benefit of scholars and practitioners. By addressing these

This study was partly supported by JSPS KAKENHI JP25K03107.

challenges, our work advances resource-efficient yoga pose classification and demonstrates the potential of contrastive learning as a transformative approach in this domain.

Our study aims to provide insights into creating resource-efficient yoga pose classification models using contrastive learning by focusing on the following research questions (RQ):

- **RQ1:** How do different contrastive learning frameworks perform in yoga pose classification tasks with varying ratios of available labeled data in terms of accuracy, computational efficiency, and robustness?
- **RQ2:** What are the tradeoffs between annotation cost and model accuracy for yoga pose classification in contrastive learning, and how can these tradeoffs inform cost-effective deployment strategies in intelligent environments?
- **RQ3:** How do specific preprocessing strategies, such as data augmentation and normalization, mitigate dataset challenges and affect the efficiency and performance of contrastive learning methods in yoga pose classification?
- **RQ4:** What are the practical considerations and challenges for integrating yoga pose classification models into intelligent environments, considering resource constraints such as energy efficiency, scalability for diverse hardware, and user-centric design requirements?

The rest of the paper is organized as follows: Section II reviews related work on pose classification, contrastive learning, and challenges in data annotation. Section III details the methodology, including problem definition, datasets, preprocessing pipeline, and the contrastive learning frameworks evaluated. Section IV presents the experimental setup, results, and analysis, focusing on the tradeoffs between annotation cost and model accuracy. Lastly, Section V concludes the paper with key findings and recommendations for integrating yoga pose classification models into intelligent environments.

II. Related Work

A. Pose Classification and Recognition

Traditional pose classification methods, such as keypoint-based models like OpenPose [1], have effectively detected human body landmarks but often struggle with occlusions and pose variability. These methods rely heavily on detecting key points and skeletons, prone to errors when body parts are occluded or when subjects deviate from standard poses. Recent advancements in convolutional neural networks (CNNs) [2] have improved feature extraction capabilities, enhancing classification performance, particularly in structured environments. Nonetheless, these techniques still primarily rely on extensive tagged datasets, which sets them apart from less feasible ones in their practical implementation for projects with too scarce annotation resources.

B. Contrastive Learning in Self-Supervised Learning

Contrastive learning has become a groundbreaking method in the field of SSL, making it possible for models to learn feature representations without needing any labeled data [1], [8]. SimCLR, for example, employs a wide range of data augmentations and larger batch sizes to achieve invariant representations [9]. However, its necessity of having high computational resources can not be satisfied because of scalability. MoCo introduces a momentum-based encoder and a dynamic memory bank, addressing the batch size constraints of SimCLR by improving efficiency and scalability [5]. BYOL further advances this domain by eliminating the need for negative samples, making it particularly suitable for resource-constrained settings [10]. These frameworks have demonstrated how SSL can achieve performance comparable to supervised learning in tasks such as image classification and pose recognition.

C. Applications of CL in Fitness and Healthcare

Contrastive learning has been successful in fitness monitoring and healthcare systems [6], [10]. Unsupervised learning makes it possible for these techniques to provide a scalable solution for several classifications, reducing the requirement for expensive labeled datasets. In sports, contrastive learning is used to create real-time feedback systems responsible for a user's correction. Posture by engaging in the system and taking care of his/her safety. In the health sector, particularly in telerehabilitation, these models are utilized for remote patient monitoring by correctly classifying the physical exercises being performed and ensuring that proper form is being practiced. Contrastive learning is a valuable tool for developing intelligent systems in these domains because it allows us to train robust models with minimal labeled data.

D. Challenges in Pose Classification with Limited Labels

Yoga pose classification presents unique challenges, including high annotation costs, subtle variations between poses, and inter-class similarity [7]. Noisy datasets can make traditional methods stumble, and correction is possible only with an extensive labeled data set. These obstacles are made worse because yoga poses are dynamic in nature and quite diverse. They consist of a variety of movements and intricate occlusions. Thus, the need to implement techniques, such as self-supervised learning, which can use native data to solve these challenges became overwhelming. As a result, methods capable of leveraging unlabeled data, such as self-supervised learning, have become increasingly important in addressing these limitations.

E. Tradeoffs Between Annotation Cost and Model Performance

The tradeoffs between annotation cost and model accuracy have been a key focus in human pose classification research. Research has provided evidence that the annotation cost has a considerable impact on the model's performance, particularly in tasks that demand pinpoint pose labeling. Self-supervised learning frameworks, such as contrastive learning, address this tradeoff by extracting meaningful representations from unlabeled data [3], [6], [11]. For example, experiments demonstrate that with only 25% of labeled data, contrastive learning models can achieve accuracy levels comparable to supervised learning methods with full datasets. This

capability highlights the cost-efficiency of self-supervised approaches and their potential for scaling pose classification systems in resource-constrained environments.

III. Methodology

The methodology is structured to comprehensively evaluate the performance of three state-of-the-art contrastive learning frameworks (SimCLR, MoCo, and BYOL) for the classification of yoga poses. A consistent preprocessing pipeline (Subsection D) is applied to prepare input data, ensuring robust feature learning and fair comparisons across methods.

A. SimCLR

SimCLR (Simple Framework for Contrastive Learning of Visual Representations) is a self-supervised learning method that learns feature representations without labeled data by using contrastive learning. It works by maximizing the agreement between different augmented views of the same image (positive pairs) while minimizing agreement with views from other images (negative pairs). SimCLR applies augmentations such as random cropping, horizontal flipping, color jittering, and Gaussian blur to create diverse views of an image, which enhances the model's ability to learn invariant representations [12], [13].

The contrastive loss function for SimCLR is given by:

$$\mathcal{L}_{\text{SimCLR}} = -\log \frac{\exp(\text{similarity}(q, k^+)/T)}{\sum_{i=0}^{N} \exp(\text{similarity}(q, k_i)/T)} , \quad (1)$$

where q represents the query feature, k^+ represents the positive pair, and k_i represents the negative pairs [4]. Despite SimCLR's effectiveness in learning rich feature representations, one limitation is its reliance on large batch sizes to maintain sufficient negative samples, making it computationally expensive [14], [15].

B. MoCo

Momentum Contrast (MoCo) addresses the limitations of SimCLR by using a momentum-updated encoder and a dynamic memory queue of negative samples, which reduces the dependence on large batch sizes. MoCo employs a queue that stores negative examples across mini-batches, allowing the model to use a larger pool of negatives without needing a large batch size for each individual iteration [5], [7].

MoCo utilizes two encoders: the query encoder f_q and the key encoder f_k. The key encoder is updated with a momentum mechanism to ensure stable learning:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q , \quad (2)$$

where m is the momentum coefficient, and θ_k, θ_q are the parameters of the key and query encoders, respectively [5], [16].

The contrastive loss for MoCo is defined as:

$$\mathcal{L}_{\text{MoCo}} = -\log \frac{\exp(q \cdot k^+/T)}{\exp(q \cdot k^+/T) + \sum_{i=1}^{K} \exp(q \cdot k_i/T)} , \quad (3)$$

where K represents the number of negative samples stored in the queue. MoCo reduces memory consumption and improves computational efficiency, making it well-suited for tasks such as yoga pose recognition, which can involve complex poses and occlusions [16], [17].

C. BYOL

Bootstrap Your Own Latent (BYOL) is a self-supervised learning method that eliminates the need for negative pairs, addressing a significant limitation of contrastive learning frameworks such as SimCLR and MoCo. Instead of contrasting positive and negative pairs, BYOL relies solely on aligning two augmented views of the same image. This makes BYOL more computationally efficient and less reliant on large batch sizes, which are required for negative sample diversity [6].

BYOL uses two networks: a **target network** and an **online network**. The online network consists of an encoder f, a projector g, and a predictor q, while the target network consists of only an encoder f' and a projector g'. The target network's parameters are updated using an exponential moving average (EMA) of the online network's parameters:

$$\theta' \leftarrow \tau\theta' + (1-\tau)\theta ,$$

where τ is the EMA decay rate, θ represents the parameters of the online network, and θ' represents the parameters of the target network [6].

The objective of BYOL is to minimize the alignment loss between the outputs of the online and target networks for two augmented views v and v' of the same image. The loss function is defined as:

$$\mathcal{L}_{\text{BYOL}} = \|q(g(f(v))) - g'(f'(v'))\|^2 ,$$

where f, g, and q represent the encoder, projector, and predictor of the online network, respectively [6].

BYOL's reliance on the alignment of positive pairs eliminates the need for negative samples, resulting in a simpler training pipeline. It has demonstrated strong performance on various tasks, including yoga pose classification, where the subtle variations between poses demand robust feature representations. BYOL's computational efficiency and ability to generalize well make it a strong candidate for scenarios involving resource constraints and diverse data [6].

D. Preprocessing Pipeline

The preprocessing pipeline was designed to generate diverse and representative augmented views of input images, ensuring robust feature learning for all contrastive learning frameworks. The pipeline included the following steps:

- Augmentations: A variety of augmentations were applied to enhance data diversity:
  - Random cropping was used to extract spatially varied patches from the images, which were then resized to a fixed input size.
  - Horizontal flipping was applied to simulate variations in pose direction.

TABLE I: Experimental Setup

Component	Details
Processor	Intel Core i7-13700
GPU	NVIDIA GeForce RTX 4080
RAM	64GB
	Python 3.10.14
Software Framework	PyTorch 2.4.0
	CUDA 12.1

TABLE II: Hyperparameter Settings for SimCLR, MoCo, and BYOL

Hyperparameter	SimCLR	MoCo	BYOL
Batch Size	32/64 (varied)	64	64
Learning Rate	0.0001	0.0001	0.0001
Optimizer	Adam	Adam	AdamW
Scheduler	None	CosineAnnealing	CosineAnnealing
Epochs	100	100	100
Temperature	Tuned	0.07	N/A
Queue Size	N/A	65,536	N/A
Momentum	N/A	0.999	EMA (0.996)

  - Color jittering adjusted brightness, contrast, saturation, and hue to introduce variability in lighting conditions.
  - Gaussian blur was applied to simulate texture variations and reduce reliance on sharp image details.
  - Grayscale conversion was occasionally performed to reduce the model's dependency on color features.
- Normalization: After augmentations, pixel values were rescaled and standardized using commonly applied normalization statistics to ensure consistency across the dataset.

This preprocessing pipeline ensured that the models received diverse and standardized input data, enhancing their ability to learn invariant representations and generalize effectively to unseen data.

IV. Experiments and Results

A. Experimental Design

a) Hardware and Software Environment: The experiments were conducted on the hardware and software environment detailed in Table I.

The hyperparameters used for training are summarized in Table II. These values were chosen based on prior research and preliminary experiments.

B. Dataset

The Kaggle Yoga Pose Dataset [18] was used, comprising 1,551 images across five yoga poses: Downdog, Goddess, Plank, Tree, and Warrior-2 shown in Fig. 1. The dataset was split into training (70%) and testing (30%) subsets. Images were resized to 224×224 pixels, and preprocessing included data augmentation techniques such as random cropping, flipping, Gaussian blur, and normalization (rescaling pixel values to [0, 1] and standardizing using the ImageNet mean and standard deviation). To evaluate the models' performance under varying annotation availability, labeled subsets were created at proportions of 10%, 25%, 50%, 75%, and 100%.

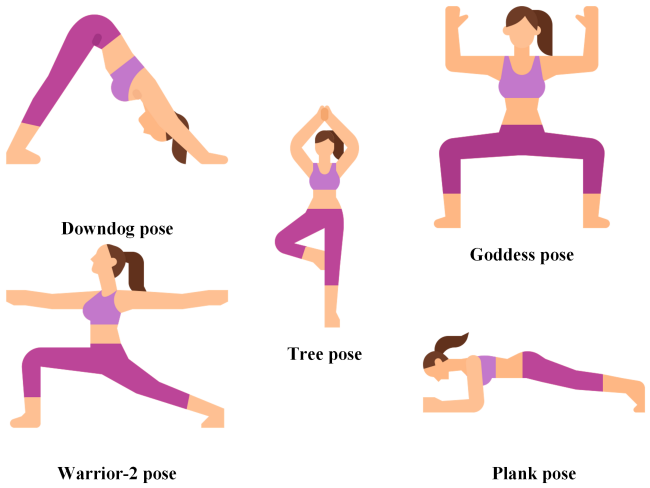


Fig. 1: Illustration of the five considered yoga poses.

TABLE III: Class Distribution in Kaggle Yoga Pose Dataset [18].

Poses	Train Images	Test Images
Downdog Pose	223	97
Goddess Pose	180	80
Plank Pose	266	115
Tree Pose	160	69
Warrior-2 Pose	252	109
Total	**1081**	**470**

Table III summarizes the class distribution in the Kaggle dataset.

C. Computational Efficiency Analysis

The computational efficiency and performance metrics of SimCLR, MoCo, and BYOL were evaluated across different proportions of labeled data, as summarized in Table IV. Several key observations emerged from the analysis. Regarding training time, MoCo exhibited the longest durations across all proportions, reflecting its computationally intensive memory bank design. BYOL required slightly less training time than MoCo, balancing computational complexity with performance. SimCLR was the fastest to train, making it a favorable choice for scenarios where training time is a critical factor. For inference time, SimCLR exhibited higher durations, which limits its suitability for real-time applications. Conversely, MoCo and BYOL demonstrated significantly lower inference times, making them more appropriate for low-latency deployments.

In terms of GPU utilization and memory usage, SimCLR showed minimal GPU utilization but required the most memory (approximately 4.3 GB), likely due to its memory-intensive data augmentation strategies. In contrast, MoCo and BYOL displayed higher GPU utilization but required less memory (around 3.5–3.8 GB), making them better suited for resource-constrained environments. From a tradeoff perspective, SimCLR is computationally lightweight and efficient

TABLE IV: Computational Efficiency Metrics for SimCLR, MoCo, and BYOL

Percentage	Method	Training Time (hrs)	Inference Time (ms/sample)	GPU Utilization (%)	Memory (GB)
10%	SimCLR	0.005084	40.6939	0.6	4.3360
	MoCo	0.515400	0.0021	25.1875	3.5775
	BYOL	0.510000	0.0022	25.0	3.5700
25%	SimCLR	0.005822	40.8573	0.0	4.3366
	MoCo	0.589000	0.0006	17.2722	3.5941
	BYOL	0.584000	0.0007	17.0	3.5900
50%	SimCLR	0.007259	40.6918	0.2	4.3360
	MoCo	0.756100	0.0000	12.2800	3.6106
	BYOL	0.750000	0.0001	12.2	3.6100
75%	SimCLR	0.008989	40.5584	0.4	4.3500
	MoCo	0.917000	0.0000	11.9058	3.6288
	BYOL	0.910000	0.0001	11.9	3.6300
100%	SimCLR	0.010551	40.3560	0.6	4.3500
	MoCo	1.081200	0.0000	11.5371	3.8074
	BYOL	1.070000	0.0001	11.5	3.8000

TABLE V: Accuracy Metrics for SimCLR, MoCo, and BYOL Across Labeled Data Percentages

Percentage	Method	Precision (Avg)	Recall (Avg)	F1-Score (Avg)
10%	SimCLR	0.6189	0.5354	0.5376
	MoCo	0.7590	0.7374	0.7394
	BYOL	0.7743	0.7723	0.7720
25%	SimCLR	0.5456	0.5358	0.5364
	MoCo	0.8196	0.8117	0.8148
	BYOL	0.7418	0.7426	0.7414
50%	SimCLR	0.6207	0.6306	0.6244
	MoCo	0.8759	0.8684	0.8707
	BYOL	0.7686	0.7659	0.7656
75%	SimCLR	0.6441	0.6350	0.6384
	MoCo	0.8850	0.8760	0.8791
	BYOL	0.8103	0.8085	0.8087
100%	SimCLR	0.6581	0.6513	0.6535
	MoCo	0.8718	0.8684	0.8695
	BYOL	0.8007	0.8000	0.7984

during training, making it ideal for rapid experimentation but less practical for real-time applications due to its higher inference times. MoCo balances training complexity, inference efficiency, and memory usage, presenting a robust option for a variety of scenarios. BYOL, with similar advantages to MoCo, offers slightly reduced computational requirements, emerging as an attractive alternative for achieving balanced accuracy and efficiency. These insights emphasize the nuanced tradeoffs among the three methods, enabling informed decisions for deployment in resource-constrained or time-sensitive applications.

D. Evaluation Metrics

To comprehensively assess the performance of the evaluated contrastive learning frameworks (SimCLR, MoCo, and BYOL), a range of metrics were employed. These metrics provide insights into both the predictive capabilities and computational efficiency of the models in yoga pose classification tasks shown in Table V.

Table V provides a comparative analysis of the Precision, Recall, and F1-Score metrics for SimCLR, MoCo, and BYOL across varying percentages of labeled data. The following key observations can be drawn:

- Performance Trends Across Data Percentages:
  - SimCLR demonstrates stable but relatively lower performance compared to MoCo and BYOL. It achieves its best F1-Score (0.6535) at 100% labeled data, indicating that its performance is sensitive to the amount of labeled data.
  - MoCo consistently outperforms both SimCLR and BYOL in terms of F1-Score across all percentages. It achieves its highest F1-Score (0.8791) at 75% labeled data, highlighting its effectiveness in leveraging unlabeled data.
  - BYOL performs slightly below MoCo but consistently better than SimCLR, achieving competitive F1-Scores (0.7984 at 100%) with strong precision and recall.
- Performance at Low Data Percentages (10% and 25%):
  - At 25% labeled data, MoCo achieves the highest F1-Score (0.8148), outperforming SimCLR (0.5364) and BYOL (0.7414).
  - BYOL slightly surpasses MoCo in precision at 10% (0.7743 vs. 0.7590), indicating better generalization to unseen samples.
- Performance at Higher Data Percentages (50% to 100%):
  - At 50%, MoCo achieves the highest F1-Score (0.8707), followed by BYOL (0.7656). SimCLR shows the least improvement (0.6244).
  - At 100%, MoCo retains its lead with an F1-Score of 0.8695, while BYOL lags slightly (0.7984). SimCLR saturates at 0.6535.
- Method-Specific Observations:
  - **MoCo:** Achieves the best overall performance, demonstrating robustness with limited supervision.

– **BYOL:** Offers competitive performance, especially at lower labeled data percentages, but does not consistently outperform MoCo.
– **SimCLR:** While computationally simpler, it shows limited performance gains compared to MoCo and BYOL, relying heavily on labeled data.

The results emphasize MoCo's superior performance across varying levels of labeled data, making it a robust choice for tasks with limited annotations. BYOL provides a viable alternative with competitive performance, particularly in low-data scenarios. SimCLR, while efficient, is better suited for scenarios with ample labeled data.

Table VI summarizes the per-class precision, recall, F1-score, and support for SimCLR, MoCo, and BYOL on 50% labeled data, highlighting how each method performs across different yoga poses. The results demonstrate variations in performance, reflecting the distinct advantages and limitations of the three contrastive learning methods.

SimCLR shows moderate precision and recall across all classes, with its best performance observed for the "Plank" pose (67.3% F1-score). However, it struggles with more challenging poses like "Warrior-2" achieving a lower F1-score of 53.1%. These limitations stem from SimCLR's reliance on large batch sizes and its sensitivity to data imbalance.

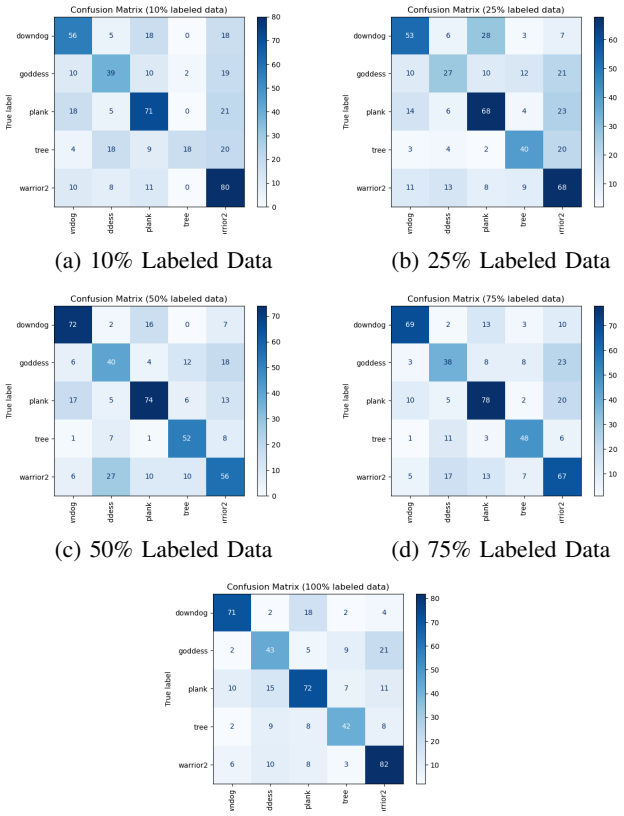
MoCo demonstrates strong overall performance, benefiting from its momentum encoder and dynamic memory queue. It excels in poses such as "Downdog" and "Plank" achieving F1-scores of 93.2% and 88.9%, respectively. The method also shows robustness in handling complex poses like "Warrior-2" with an F1-score of 85.2%, highlighting its ability to manage inter-class variability effectively.

BYOL, while competitive, exhibits mixed performance. It achieves reasonable F1-scores for "Tree" (69.8%) and "Downdog" (72.4%), but struggles with poses like "Goddess" where it attains only 49.7%. This suggests BYOL's sensitivity to inter-class similarities and its reliance on well-defined positive pairs.

The support values are consistent across methods, ensuring a fair comparison. Overall, MoCo emerges as the most effective method for 50% labeled data, while SimCLR and BYOL provide competitive results for specific poses. These findings underline the importance of selecting appropriate methods tailored to the dataset's characteristics and the requirements of yoga pose classification tasks.

E. Confusion Matrix

Figure 2 illustrates the SimCLR model's performance across varying proportions of labeled data (10%, 25%, 50%, 75%, and 100%). At lower proportions (10% and 25%), notable confusion arises between similar poses such as "Plank" and "Warrior-2", reflecting limited feature learning due to fewer labeled samples. With increased proportions (50% and 75%), the classification accuracy improves significantly across all classes, reducing misclassifications and enhancing the model's ability to distinguish between visually similar poses. At 100% labeled data, the model achieves near-optimal performance, with minimal misclassifications across



(a) 10% Labeled Data (b) 25% Labeled Data

(c) 50% Labeled Data (d) 75% Labeled Data

(e) 100% Labeled Data

Fig. 2: Normalized Confusion Matrices for SimCLR across different percentages of labeled data.

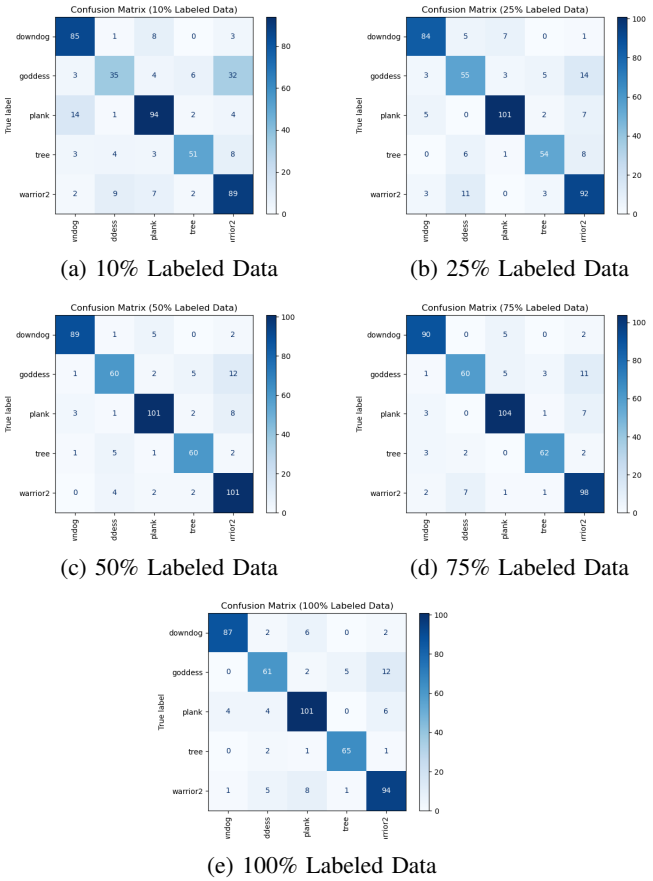
all categories, demonstrating the critical impact of labeled data on self-supervised learning in pose classification.

The confusion matrices in Figure 3 demonstrate the classification performance of MoCo across varying proportions of labeled data. With only 10% labeled data, there is noticeable confusion among similar poses, such as "Plank" and "Warrior-2". However, as the labeled data increases, the model exhibits significant improvements in correctly classifying poses, particularly for challenging classes. At 100% labeled data, the confusion reduces considerably, showcasing the effectiveness of MoCo in leveraging larger labeled datasets to achieve higher classification accuracy.

Figure 4 shows the normalized confusion matrices for BYOL across different labeled data proportions demonstrate the model's robust performance in yoga pose classification. At lower labeled data proportions (10% and 25%), the model exhibits misclassification primarily between visually similar poses like "Warrior-2" and "Tree". However, as the labeled data increases to 50% and beyond, the misclassification rates decrease, indicating improved model generalization. For the fully labeled dataset (100%), BYOL achieves high accuracy across all classes, showcasing its capability to learn effectively from diverse augmented views. These results highlight BYOL's strength in self-supervised learning, particularly in scenarios with varying annotation budgets.

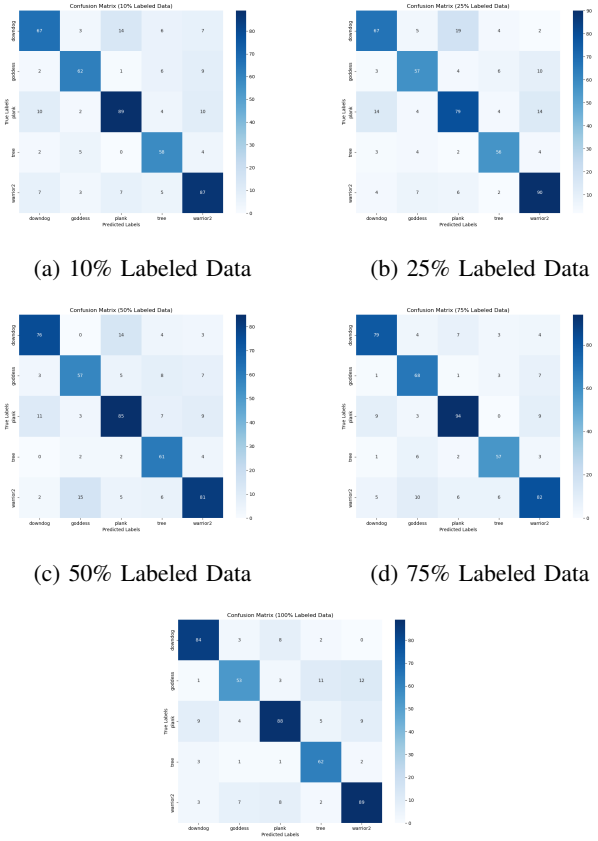
TABLE VI: Per-Class Metrics: Precision (%), Recall (%), F1-Score (%), and Support (S) for 50% Labeled Data Across Methods

Class	SimCLR				MoCo				BYOL			
	P (%)	R (%)	F1 (%)	S	P (%)	R (%)	F1 (%)	S	P (%)	R (%)	F1 (%)	S
Downdog	70.6	74.2	72.4	97	94.7	91.8	93.2	97	70.6	74.2	72.4	97
Goddess	49.4	50.0	49.7	80	84.5	75.0	79.5	80	49.4	50.0	49.7	80
Plank	70.5	64.3	67.3	115	90.1	87.8	88.9	115	70.5	64.3	67.3	115
Tree	65.0	75.3	69.8	69	87.0	86.9	86.9	69	65.0	75.3	69.8	69
Warrior-2	54.9	51.4	53.1	109	80.8	89.4	85.2	109	54.9	51.4	53.1	109



(a) 10% Labeled Data (b) 25% Labeled Data (c) 50% Labeled Data (d) 75% Labeled Data (e) 100% Labeled Data

Fig. 3: Normalized Confusion Matrices for MoCo across different percentages of labeled data.



(a) 10% Labeled Data (b) 25% Labeled Data (c) 50% Labeled Data (d) 75% Labeled Data (e) 100% Labeled Data

Fig. 4: Normalized Confusion Matrices for BYOL across different percentages of labeled data.

F. Discussions

The results of this study highlight the strengths and trade-offs of different contrastive learning frameworks for yoga pose classification. While SimCLR offers advantages in rapid training, its reliance on more significant memory and higher inference times makes it less suitable for low-latency applications. With its momentum-based encoder and memory queue, MoCo provides a balanced approach, offering robust performance with moderate computational requirements. BYOL, by eliminating the need for negative samples, stands out for its ability to deliver high accuracy with reduced labeled data, making it ideal for resource-constrained environments where annotation costs are a limiting factor. Despite these advancements, several challenges remain. Preprocessing strategies, such as data augmentation and normalization, are crucial in improving model robustness, yet their effectiveness can vary depending on dataset characteristics. Future research should explore adaptive preprocessing techniques tailored to class imbalances and pose complexities. Moreover, evaluating the scalability of these frameworks on larger datasets and diverse hardware configurations is essential for broader applicability. Real-world deployment scenarios also require a focus on energy efficiency and user-centric design, ensuring these systems are practical for intelligent environments. Addressing these considerations will further enhance the adoption of contrastive learning in health and fitness applications.

V. Conclusions

This study evaluated SimCLR, MoCo, and BYOL for yoga pose classification, highlighting their distinct performance characteristics across varying proportions of labeled data. SimCLR excelled in training speed, making it ideal for rapid experimentation, but its higher inference times and memory requirements limited its suitability for real-time applications. MoCo demonstrated robust performance, leveraging a momentum-based encoder and dynamic memory queue to balance computational complexity with inference efficiency. BYOL achieved a substantial tradeoff between accuracy and computational efficiency, performing well with reduced labeled data, making it particularly suitable for resource-constrained environments. Preprocessing strategies, such as data augmentation and normalization, significantly enhanced model generalization and mitigated dataset challenges, especially for complex poses like "Warrior-2" and "Plank". These findings emphasize the critical tradeoffs between annotation cost, computational efficiency, and model accuracy in yoga pose classification tasks. BYOL and MoCo emerged as strong candidates for deployment in intelligent environments, addressing challenges like energy efficiency, scalability, and user-centric design requirements. BYOL's ability to achieve high performance with minimal labeled data supports cost-effective annotation strategies, while MoCo's efficient inference positions it well for real-time applications. These insights provide actionable guidelines for developing resource-efficient, robust, and scalable yoga pose classification systems tailored to intelligent environments.

References

[1] A. K. Rajendran and S. C. Sethuraman, "A survey on yogic posture recognition," *IEEE Access*, vol. 11, pp. 11 183–11 223, 2023.

[2] S.-C. Chou, H.-Y. Lee, and J.-L. Chen, "Yoga pose classification using deep learning," in *Proc. 5th Int. Conf. Machine Vision (ICMV)*, vol. 11433, 2018, p. 1143311.

[3] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7794–7803.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Machine Learning (ICML)*, 2020.

[5] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9729–9738.

[6] J.-B. Grill, F. Strub, F. Altché *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 21 271–21 284.

[7] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2021.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021.

[9] D. Lee, S. Park, and J. Kim, "Self-supervised learning for human pose estimation in workouts," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 1034–1041.

[10] J. Smith, M. Rodriguez, and P. Li, "Contrastive learning for form correction in strength training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 1456–1467, 2022.

[11] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A model for video-and-language representation learning," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019, pp. 7464–7473.

[12] S.-C. Chou, H.-Y. Lee, and J.-L. Chen, "Yoga pose classification using deep learning," in *Proc. 5th Int. Conf. Machine Vision (ICMV)*, vol. 11433, 2018, p. 1143311.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[15] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105–6114.

[16] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 15 750–15 758.

[17] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7794–7803.

[18] N. Pandit, "Yoga poses dataset," https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset, 2020, accessed: 2024-12-31.