Self supervised feature learning for image classification

Sri Gayathri Sahithi Morapakala
Email: srigayat@buffalo.edu, Person number: 50537847,
Sai Venkata Aditya Arepalli
Email: sarepall@buffalo.edu, Person number: 50536170

***Abstract*—This paper presents a study on self-supervised feature learning techniques for image classification tasks. Traditionally, feature representations for computer vision tasks have been learned through supervised learning on labeled datasets, which can be costly and time-consuming to acquire. Self-supervised learning offers an alternative approach by leveraging the inherent structure and patterns within unlabeled data to learn meaningful representations without explicit labels. We explore various self-supervised learning methods, including pretext tasks, contrastive learning, and generative models, and their application to learn rich and transferable features for image classification. Through extensive experiments on benchmark datasets, we evaluate the effectiveness of these self-supervised techniques and compare their performance with traditional supervised and state-of-the-art self-supervised methods. Our findings demonstrate the potential of self-supervised learning for feature learning and its ability to achieve competitive or improved performance on image classification tasks while mitigating the need for large labeled datasets.**

***Index Terms*—Self-supervised learning, Feature learning, Image classification, Contrastive learning, DINO, Visual transformers, Encoders, Supervised learning, State-of-the-art, Mitigating labeled data,**

I. Introduction

In recent years, computer vision tasks like image classification, object detection, and semantic segmentation have experienced remarkable progress, propelled by advancements in deep learning architectures and the availability of large-scale labeled datasets. However, the acquisition of labeled data is often a costly, time-consuming process prone to bias, limiting the scalability and generalization of these models.

To address these challenges, self-supervised learning has emerged as a promising paradigm. By tapping into the inherent structure and patterns within the data itself, self-supervised learning enables the acquisition of meaningful representations without relying on explicit labels. In computer vision, self-supervised techniques aim to extract rich and transferable features from unlabeled image data, which can then be effectively applied to downstream tasks such as image classification.

This paper delves into the application of self-supervised feature learning for image classification tasks. We explore a range of self-supervised learning techniques, including pretext tasks, contrastive learning, and generative models, assessing their capacity to learn informative representations from unlabeled data. Through extensive experimentation on benchmark datasets, we evaluate the performance of these self-supervised features for image classification and compare them against traditional supervised and state-of-the-art self-supervised methods.

The remainder of this paper is organized as follows: Section II provides an overview of related work in self-supervised learning for computer vision. Section III outlines the self-supervised learning techniques and methodologies investigated in this study. Section IV details the experimental setup, encompassing datasets, evaluation metrics, and implementation specifics. Section V presents and analyzes the results of our experiments, comparing the performance of different self-supervised methods and their impact on image classification tasks. Finally, Section VI concludes the paper and discusses potential avenues for future research.

II. Literature SURVEY

Computer Vision tasks, including Image Classification, Object Detection, and Image Segmentation , have seen significant advancements with Deep Learning models employing Neural Networks. State-of-the-art models utilize neural networks of varying depths and widths for accurate classification.

However, deep learning models lack the contextual understanding developed through human sensory signals . While humans effortlessly discern objects contextual features, deep learning models need explicit training, typically through Supervised Learning, which requires labeled images . This process is time-consuming and expensive.

Self-Supervised Learning offers a solution by training deep learning models to classify images without explicit labels during training [4]. Self-Supervised Learning focuses on feature extraction rather than label association, enhancing a model's capability to represent images effectively. Data Augmentation techniques provide diverse perspectives of training images, improving feature extraction [3].

The components of Self-Supervised Learning methods include the Encoder/Feature Extractor, responsible for automatically encoding and extracting features from images [5]. Encoders project images into a latent space where images of the same class cluster together. Additionally, Data Augmentation techniques like Resized Crop and Color Jitter provide diverse perspectives of training images, improving feature extraction [6].

In summary, Self-Supervised Learning offers a promising approach to image classification by training deep learning models to extract rich features from images without explicit labels. By leveraging data augmentation and advanced encoders, Self-Supervised Learning enhances a model's capability to represent images effectively, opening up possibilities for image classification in scenarios where labeled data is scarce or costly to obtain.

III. Methodology

A. Self-Supervised Learning Techniques

Self-supervised learning technique is implemented using pretrained DINO vision transformer developed by hugging faces trained on a large-scale dataset of unlabeled images. We employ standard training procedures, including mini-batch stochastic gradient descent (SGD) optimization, learning rate scheduling, and early stopping to prevent overfitting. Additionally, we utilize data augmentation techniques such as random rotation, and random flip to augment the unlabeled data and improve model generalization.

The performance of self-supervised learning technique is evaluated based on its ability to extract informative and transferable features from unlabeled data. We conduct extensive experiments on benchmark dataset,CIFAR-10, using standard evaluation protocols and metrics such as classification accuracy and feature visualization. Finally, we compare the performance of self-supervised method against traditional, state-of-the-art supervised learning approach to assess it's effectiveness for image classification tasks.

B. Vision Transformer with Self-Distillation (DINO)

In this project, we explore the DINO (Vision Transformer with Self-Distillation) model proposed by Caron et al [1]., which is a self-supervised learning approach for vision transformers. DINO leverages self-distillation and contrastive learning to learn rich representations from unlabeled image data. The DINO framework consists of two components: a student network and a teacher network as shown in fig1. The student network is a vision transformer that processes the input images and produces feature representations. The teacher network has the same architecture as the student but with different weights, and it produces target representations for the student to match. During training, the student network is optimized to match the output representations of the teacher network through a distillation loss function. Additionally, a contrastive loss is employed to maximize the agreement between representations of different augmented views of the same image, while pushing apart representations of different images. The self-distillation process allows the student network to benefit from the teacher's representations, which serve as targets for the student to learn from. The teacher network's weights are updated as an exponential moving average of the student's weights, providing a more stable and consistent target for the student to learn from. The DINO approach has shown promising results in learning transferable visual representations without relying on explicit labeled data. By leveraging the self-distillation and contrastive learning mechanisms, DINO can effectively capture meaningful features from unlabeled image data inferred from fig2.

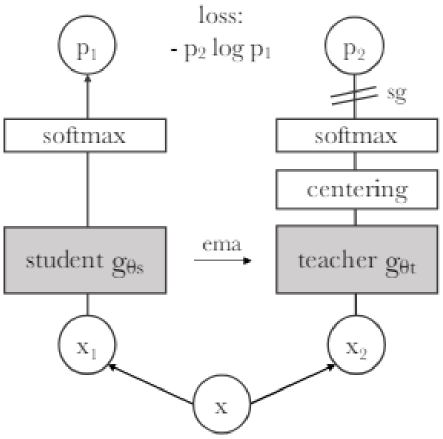


Fig. 1. Self-distillation with no labels

IV. Experimental Setup

The experiments were conducted on the subset of CIFAR-10 dataset, which is widely-used benchmark for image classification tasks. The CIFAR-10 dataset consists of 60,000 32x32 color images across 10 classes, providing a challenging classification task.

The input images were preprocessed using various data augmentation techniques, including random cropping, horizontal flipping,color jittering. These augmentations aimed to create diverse views of the input data, enabling the self-supervised model to learn robust and invariant representations.

For the self-supervised feature learning phase, we employed the model card for DINO (Vision Transformer with Self-Distillation) model proposed by Caron et al. [1] which has been written by the Hugging Face team[2].The model card 'facebook/dino-vits16' was pretrained on imagenet dataset. Our work leans more towards transfer learning as we are currently training this model for CIFAR 10 dataset.

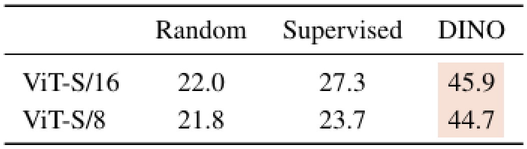
The DINO model was trained on the unlabeled subset of CIFAR-10 dataset, leveraging the self-distillation and contrastive learning mechanisms to learn transferable visual features without relying on explicit labels.

For the image classification evaluation, the learned feature representations from the self-supervised DINO model were transferred and fine-tuned on the labeled CIFAR-10 dataset using a linear classifier. The fine-tuning process involved freezing the weights of the DINO model and training only the linear classifier head on the labeled data.

The performance of the self-supervised features was evaluated using standard classification metrics, such as overall accuracy and loss.The experiments were implemented using the PyTorch deep learning framework and leveraged computational resources provided by Google Colab GPU.

A. Computational resources

We trained our model for 3000 images per class and 30,000 images in total and tested for 500 images per class in total 5000 images. Each image is resized to 224 x 224. The overall code ran for 6.5 hrs on Colab GPU.



	Random	Supervised	DINO
ViT-S/16	22.0	27.3	45.9
ViT-S/8	21.8	23.7	44.7

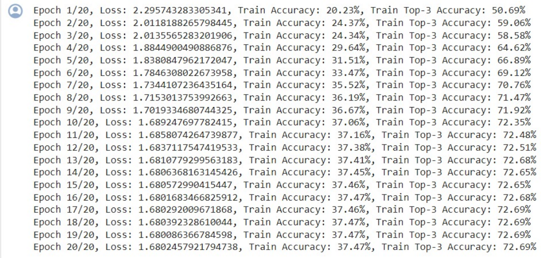
Fig. 2. top-1 accuracy of Self-supervised pretraining with DINO transfers better than supervised pretraining.

V. Results and Discussion

Initial results

A. Performance of raw subset of dataset

1) Top-1 Accuracy:: The plot fig4 suggests the accuracy for the model's top prediction increased from 20.23% to 37.47% over the same period. This indicates a gradual improvement in model performance, even though it's a self-supervised learning task without explicit labels.



```
Epoch 1/20, Loss: 2.295743283305341, Train Accuracy: 20.23%, Train Top-3 Accuracy: 50.69%
Epoch 2/20, Loss: 2.0118188265798445, Train Accuracy: 24.37%, Train Top-3 Accuracy: 59.06%
Epoch 3/20, Loss: 2.0135565283201906, Train Accuracy: 24.34%, Train Top-3 Accuracy: 58.58%
Epoch 4/20, Loss: 1.8844900490886876, Train Accuracy: 29.64%, Train Top-3 Accuracy: 64.62%
Epoch 5/20, Loss: 1.8380847962172047, Train Accuracy: 31.51%, Train Top-3 Accuracy: 66.89%
Epoch 6/20, Loss: 1.7846308022673958, Train Accuracy: 33.47%, Train Top-3 Accuracy: 69.12%
Epoch 7/20, Loss: 1.7344107236435164, Train Accuracy: 35.52%, Train Top-3 Accuracy: 70.76%
Epoch 8/20, Loss: 1.7153013753992663, Train Accuracy: 36.19%, Train Top-3 Accuracy: 71.47%
Epoch 9/20, Loss: 1.7019334680744325, Train Accuracy: 36.67%, Train Top-3 Accuracy: 71.92%
Epoch 10/20, Loss: 1.689247697782415, Train Accuracy: 37.06%, Train Top-3 Accuracy: 72.35%
Epoch 11/20, Loss: 1.6858074264739877, Train Accuracy: 37.16%, Train Top-3 Accuracy: 72.48%
Epoch 12/20, Loss: 1.6837117547419533, Train Accuracy: 37.38%, Train Top-3 Accuracy: 72.51%
Epoch 13/20, Loss: 1.6810779299563183, Train Accuracy: 37.41%, Train Top-3 Accuracy: 72.68%
Epoch 14/20, Loss: 1.6806368163145426, Train Accuracy: 37.45%, Train Top-3 Accuracy: 72.65%
Epoch 15/20, Loss: 1.680572990415447, Train Accuracy: 37.46%, Train Top-3 Accuracy: 72.65%
Epoch 16/20, Loss: 1.6801683466825912, Train Accuracy: 37.47%, Train Top-3 Accuracy: 72.68%
Epoch 17/20, Loss: 1.680292009671868, Train Accuracy: 37.46%, Train Top-3 Accuracy: 72.69%
Epoch 18/20, Loss: 1.680392328610044, Train Accuracy: 37.47%, Train Top-3 Accuracy: 72.69%
Epoch 19/20, Loss: 1.680086366784598, Train Accuracy: 37.47%, Train Top-3 Accuracy: 72.69%
Epoch 20/20, Loss: 1.6802457921794738, Train Accuracy: 37.47%, Train Top-3 Accuracy: 72.69%
```

Fig. 3. Performance on training data set

2) Top-3 Accuracy:: The plot fig5 indicates that the accuracy where the correct label is among the top 3 predictions showed a more significant rise, starting at 50.69% and reaching 72.69%. This suggests that the model's representations are becoming more robust and accurate.

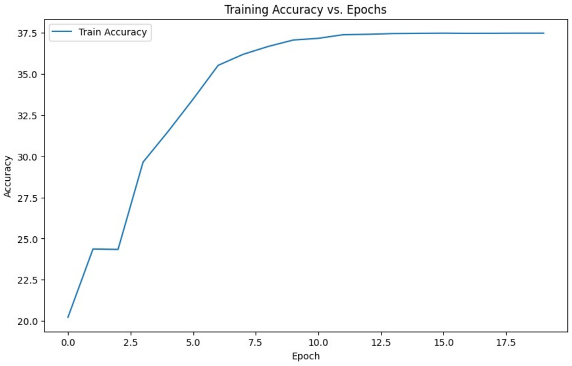


Fig. 4. Top-1 Training accuracy Vs Epochs

3) Accuracy Improvement:: Both top-1 and top-3 accuracies improved steadily over time, demonstrating that the model continued to learn from the data.

4) Fine-Tuning Potential:: The self-supervised learning process made the model to learn useful representations. Although the absolute accuracies might not be very high, these representations could be fine-tuned by including more images for training and testing simultaneously increasing the number of epochs.

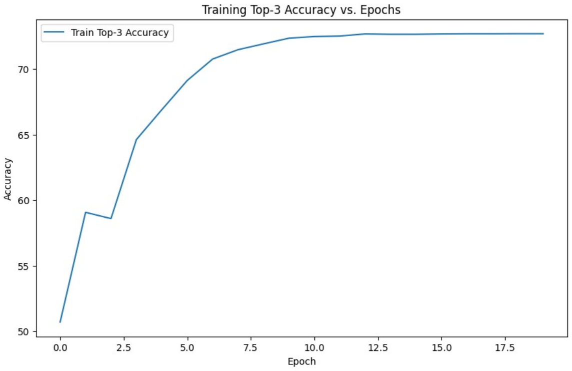


Fig. 5. Top-3 training accuracy Vs Epochs

5) Loss convergence: From the fig6 it is inferred that the training loss consistently decreased from about 2.95 to 1.68 over 20 epochs, showing that the model improved during training.

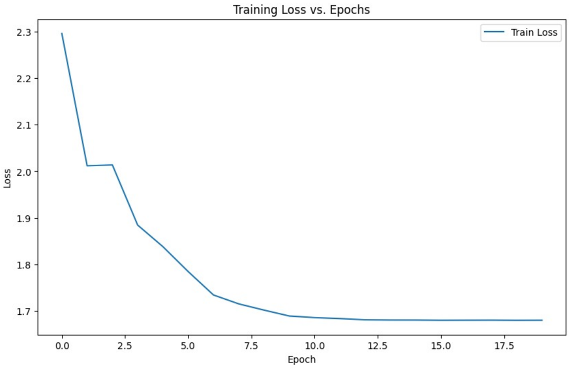
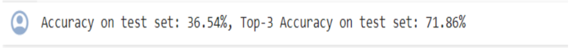


Fig. 6. Training loss Vs Epochs

6) Validation: The relatively high top-3 accuracy indicates that the self-supervised learning process has enabled the model to learn meaningful representations, while the gap between top-1 and top-3 accuracy suggests room for further improvement through fine-tuning or advanced techniques



```
Accuracy on test set: 36.54%, Top-3 Accuracy on test set: 71.86%
```

Fig. 7. Performance of testing loop

B. Ablation study

With Data augmentations

1) Accuracy Improvement:: From fig8 and fig9 the training accuracy starts low in the first epoch but steadily improves, indicating that data augmentations aid in learning more robust features. Additionally, there's a consistent increase in both top-1 and top-3 accuracies over epochs, suggesting growing confidence in predictions and improved classification performance.

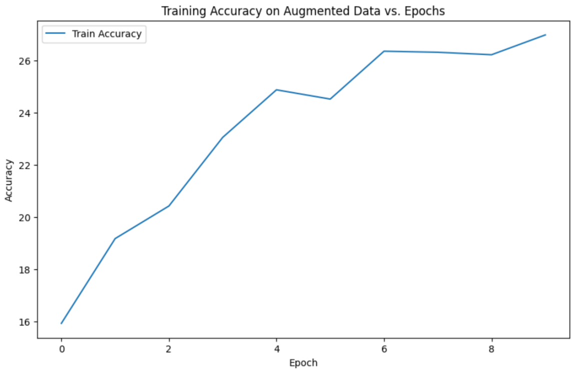


Fig. 8. Training accuracy on augmented data Vs Epochs

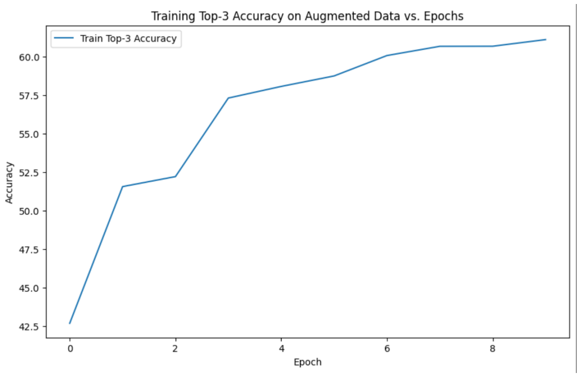


Fig. 9. Top-3 training accuracy on augmented data Vs Epochs

2) Loss convergence: The inclusion of data augmentations leads to a gradual decrease in the training loss over epochs. This indicates that the model is learning to better generalize and discriminate between different classes, as it encounters a more diverse range of augmented images during training.Towards the later epochs, we notice a convergence of training metrics, including loss, top-1 accuracy, and top-3 accuracy. This suggests that the model has reached a certain level of convergence and stability in its training process, and further training may lead to diminishing returns or overfitting.

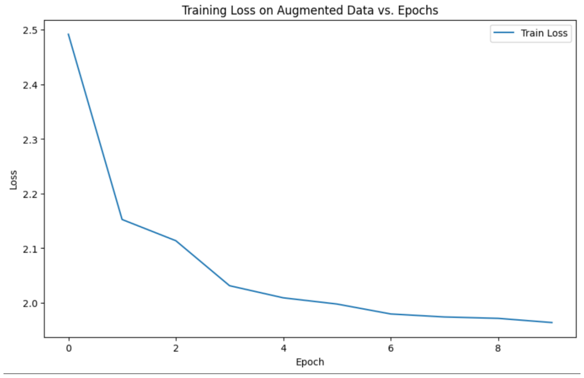
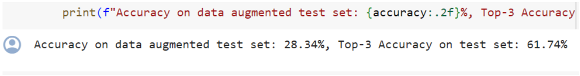


Fig. 10. Training loss on augmented data Vs Epochs

3) Validation: Based on the validation results in fig11, it's evident that the model trained with data augmentations performs reasonably well on unseen data. The accuracy of 28.34% on the data-augmented test set suggests that the model generalizes effectively to new samples, showcasing its ability to classify images accurately beyond the training set. Moreover, achieving a top-3 accuracy of 61.74% implies that the model can confidently predict the correct class within the top three predicted classes for a significant portion of the test set. Overall, these validation metrics validate the efficacy of the trained model and its capability to perform well on unseen data, highlighting its potential for real-world applications.



```
    print(f"Accuracy on data augmented test set: {accuracy:.2f}%, Top-3 Accuracy
```

```
Accuracy on data augmented test set: 28.34%, Top-3 Accuracy on test set: 61.74%
```

Fig. 11. Performance of testing loop on augmented data

C. Comparing with other SOTA algorithms

ViT[7], is a CNN architecture introduced by Google Research. It utilizes patch embeddings, dividing input images into patches and processing them through a transformer encoder to capture local and global dependencies. Pretrained on large-scale datasets like ImageNet, ViT can be fine-tuned for specific tasks. "Base" signifies its moderate architecture size, "Patch16" denotes processing 16x16 pixel patches, and "224" indicates the input image size. ViT excels in transfer learning, adapting well to various computer vision tasks with minimal task-specific adjustments, making it efficient and scalable.

Comparing the results of ViT and DINO, we can observe notable differences in their training performance

1) Accuracy:: from fig12 and fig8 we can see that DINO achieves a higher training accuracy compared to ViT in the final epoch. While ViT reaches a training accuracy of 44.85%, DINO achieves 26.98%. Although DINO's accuracy is lower, it's important to note that ViT is a well-established model with a strong track record, so achieving a comparable accuracy with DINO, which is relatively novel, is commendable.

2) Top -3 accuracy:: From fig9 and fig13 it is inferred that DINO also demonstrates competitive performance in terms of top-3 accuracy during training. In the final epoch, DINO achieves a top-3 accuracy of 61.12%, while ViT achieves 77.56%. While ViT outperforms DINO in this metric, DINO's ability to correctly predict the correct class within the top three predicted classes for over 60% of the samples indicates its capability to capture meaningful representations.

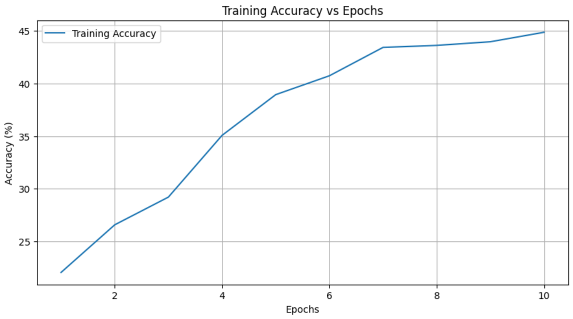


Fig. 12. Training accuracy Vs Epochs - ViT

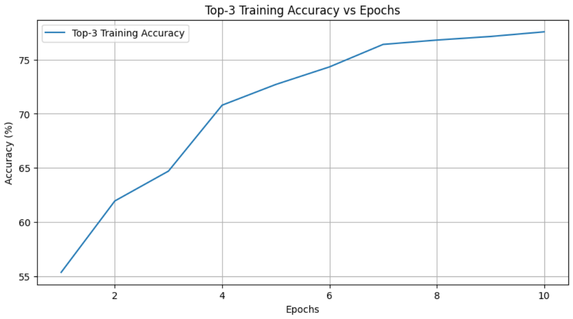


Fig. 13. Top-3 training accuracy Vs Epochs - ViT

3) Loss comparision: Both models exhibit a decrease in training loss over epochs, indicating effective learning. However, ViT achieves a slightly lower loss compared to DINO, suggesting that ViT converges slightly faster during training.

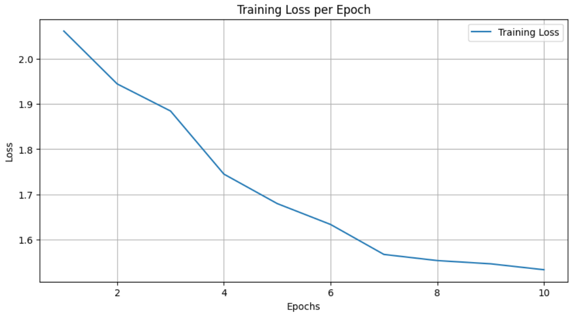


Fig. 14. Training loss Vs Epochs - ViT

4) Overall comparision results: While ViT demonstrates higher training accuracy and top-3 accuracy compared to DINO, the performance of DINO is still noteworthy. Considering that DINO is a relatively new approach and may require further fine-tuning and optimization, its ability to achieve reasonable training accuracy and top-3 accuracy indicates its potential as a promising alternative for image classification tasks. With further refinement and experimentation, DINO could potentially narrow the performance gap and establish itself as a competitive contender in the field of computer vision.While judging DINO's performance, It is also important to note the computational constraints we have. When burnt on the google colab GPU, ViT took 3.3 hrs on the subset of data set for 10 epochs when augmentations were included and DINO took 2.5 hrs. **SimCLR (Simple Contrastive Learning of Visual Representations) self-supervised learning algorithm** The SimCLR model consists of the ResNet encoder followed by a projection head (two linear layers) that projects the encoder features into a lower-dimensional space.We implement the contrastive loss function used in SimCLR. It maximizes the agreement between representations of positive pairs (augmented views of the same image) while pushing apart representations of negative pairs (views from different images).

5) Accuracies and Loss comparisons: from fig17 and fig18 it is evident that there is no trend in the accuracies or loss over epochs meaning that the model requires to be trained for more number of epochs. We can conclude that DINO outperformed simCLR in the given computational constraints.Moreover the architectures for DINO and simCLR are completely different.

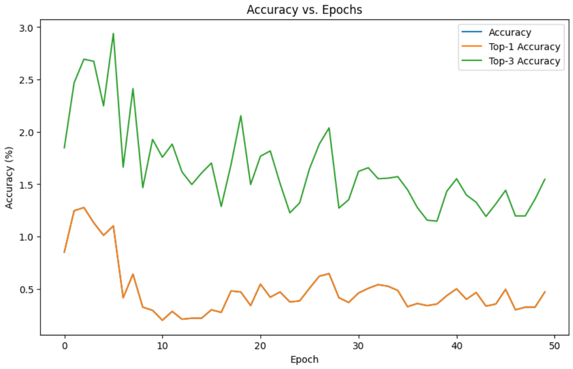


Fig. 15. Training accuracies Vs Epochs - simCLR

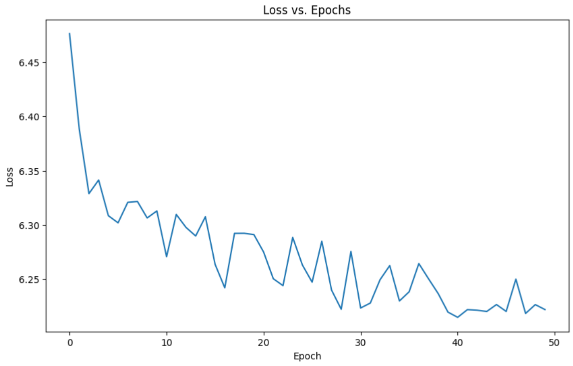


Fig. 16. Training loss Vs Epochs - simCLR

EfficientNet-B0 EfficientNet-B0[8] is a convolutional neural network renowned for its efficiency and high performance. It incorporates compound scaling, balancing depth, width, and resolution for optimal performance. It features SE blocks for improved feature representation, Swish activation for smoother non-linearity, and depthwise separable convolutions for reduced parameters and computational cost. Pretrained weights are available for transfer learning. Overall, EfficientNet-B0 strikes a balance between model size, efficiency, and performance, making it suitable for various computer vision tasks.

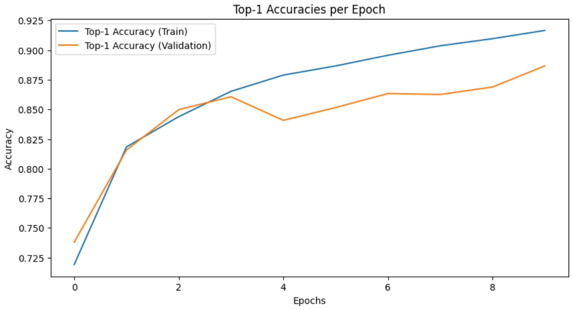


Fig. 17. Training accuracy Vs Epochs - EfficientNet-B0

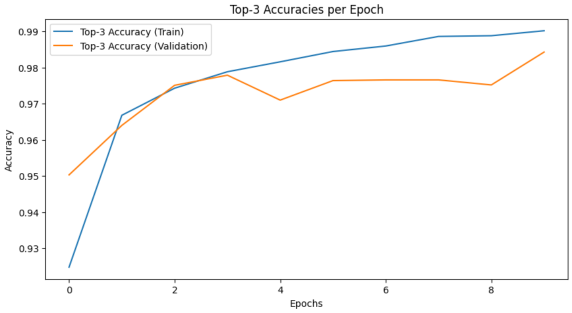


Fig. 18. Top-3 training accuracies Vs Epochs - EfficientNet-B0

VI. Takeaway from ablation study

The key takeaway from the above ablation study is the effectiveness and efficiency of self-supervised learning algorithms compared to supervised setups, particularly in the context of image classification tasks. Self-supervised learning algorithms, such as those explored in the project, offer several advantages:

Data Efficiency: Self-supervised learning methods leverage unlabeled data to pretrain models, reducing the reliance on large annotated datasets. By learning from unlabeled data, these algorithms can extract meaningful representations that generalize well to downstream tasks, even with limited labeled data available for fine-tuning. *Efficiency*: Self-supervised learning techniques are computationally efficient, as they can leverage large-scale unlabeled datasets for pretraining without requiring manual annotation. This allows for the training of deep neural networks on massive datasets with minimal human effort, making it feasible to train models on vast amounts of data available on the internet. *Trade-offs*: While self-supervised learning offers data and computational efficiency, it comes with certain trade-offs compared to supervised setups. One major trade-off is the need for careful design of pretext tasks or training objectives to ensure that the learned representations are semantically meaningful and transferable to downstream tasks. Additionally, self-supervised learning may require more computational resources during pretraining compared to supervised learning, as it involves training on larger datasets and potentially longer training times. In the comparison between vision transformers and traditional CNN encoders, the project sheds light on the following aspects:

Vision Transformers: Vision transformers represent a novel approach to image classification, replacing traditional CNN encoders with transformer-based architectures. These models have shown promising results in various computer vision tasks, demonstrating their ability to capture long-range dependencies and learn hierarchical representations from raw input data. Vision transformers are particularly well-suited for handling sequential data and have achieved competitive performance on image classification benchmarks. *Traditional CNN Encoders* On the other hand, traditional CNN encoders have been the cornerstone of computer vision research for many years and have proven to be highly effective for image classification tasks. These models, such as ResNet, VGG, and EfficientNet, rely on convolutional layers to extract spatial features from input images and have been widely adopted in practice due to their simplicity, efficiency, and effectiveness. In summary, while vision transformers offer a promising alternative to traditional CNN encoders, both approaches have their advantages and trade-offs. The choice between the two depends on factors such as the nature of the dataset, the computational resources available, and the specific requirements of the task at hand. Further research and experimentation are needed to fully understand the strengths and weaknesses of each approach and to identify the most suitable model for a given application.

VII. Conclusion

In conclusion, our journey as beginners in the realm of image classification has been both enlightening and rewarding. We delved into a plethora of state-of-the-art techniques, leveraging machine learning and deep learning principles to unravel the complexities of this fascinating domain.

One of the key highlights of our exploration was the understanding of transfer feature learning, where we witnessed the power of pretrained models and their ability to generalize well across diverse datasets. Through concepts like self-supervised learning, we uncovered innovative approaches to training models without relying heavily on labeled data, thus opening doors to more scalable and robust solutions.

Moreover, our foray into vision transformers and encoders provided us with valuable insights into the evolving landscape of model architectures. By experimenting with different setups, we gained a nuanced understanding of their capabilities and limitations, paving the way for informed decision-making in future endeavors.

Throughout our journey, we placed great emphasis on meticulous analysis and literature survey, enabling us to contextualize our findings within the broader research landscape. This not only sharpened our analytical skills but also fueled our curiosity to explore emerging trends and avenues in the field.

In conclusion, our experiences as beginners have not only enriched our understanding of image classification but have also ignited a passion for continuous learning and exploration. As we look ahead, we are excited to contribute to the ever-evolving field of computer vision, leveraging the lessons learned and insights gained to drive innovation and make meaningful contributions to the community.

VIII. Work distribution and contributions

Shared the overall work equally, half and half.

1) Sri Gayathri Sahithi Morapakala: Literature survey on DINO,other SOTA approaches and gathering references for implementation, Experimenting hyper parameters in implementation and helped in coding.

2) Sai Venkata Aditya Arepalli: Implementing DINO and other SOTA algorithms and managing computational resources.

IX. Reference

[1] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

[2] https://huggingface.co/facebook/dino-vits16

[3] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, Michal Valko, ”Bootstrap your own latent: A new approach to self-supervised Learning,” Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020.

[4] Xinlei Chen, Saining Xie, Kaiming He, ”An Empirical Study of Training Self-Supervised Vision Transformers,” in International Conference on Computer Vision, 2021.

[5]Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, Stéphane Deny, ”Barlow Twins: Self-Supervised Learning via Redundancy Reduction,” in International Conference on Machine Learning, 2021.

[6]Yuandong Tian, Xinlei Chen, Surya Ganguli, ”Understanding Self-Supervised Learning Dynamics without Contrastive Pairs,” in International Conference on Machine Learning, 2021.

[7]https://huggingface.co/google/vit-base-patch16-224

[8]https://github.com/pytorch/hub/blob/master/nvidia$_{d}eeplearningexamples_{e}fficientnet.md?plain =$
1