Self-Supervised Pretraining for Enhancing Supervised Learning in Low-Data Regimes

Anirudh Kumar Singh
School of Computer Science & Engineering
Lovely Professional University
Jalandhar, Punjab, India
anirudhsingh3019@gmail.com

Anand Kumar
School of Computer Science & Engineering
Lovely Professional University
Jalandhar, Punjab, India
anandk.cs20@gmail.com

***Abstract*—**In the real-world situation, availability of labelled data is very limited and they are costly too which increases the difficulty for deep learning models to get high accuracy. Self-Supervised Learning (SSL) came as a powerful technique to grip unlabelled data for pretraining, thereby improving the accuracy like ability of models in low-data regimes. In this study, we investigate the role of SSL pretraining on convolutional neural network (CNN) architectures, which focuses on the STL-10 dataset. We compare the performance of MobileNet, ResNet, and EfficientNet models under SSL pretraining, followed by fine-tuning on the labelled subset of STL-10. The results indicate that EfficientNet secures the highest accuracy of 88%, leaving the other models behind. Our research highlights the effectiveness of self-supervised pretraining in improving classification performance with limited labelled data.

***Keywords*—**Self-supervised learning, low-data regime, CNN, STL-10 dataset, convolutional neural networks, EfficientNet**.**

I. INTRODUCTION

Deep learning has significantly transformed the field of artificial intelligence, enabling breakthroughs in image recognition, natural language processing, and autonomous systems. The success of deep learning models, particularly convolutional neural networks (CNNs), is largely attributed to their ability to learn hierarchical features from large-scale labelled datasets. However, in practical scenarios, obtaining sufficient labelled data remains a significant challenge. Many real-world applications, such as medical diagnostics, autonomous driving, and remote sensing, require expert annotations, which are costly, time-consuming, and sometimes infeasible to acquire in large quantities. In such cases, the dependency on supervised learning methods becomes a limitation, preventing the effective application of deep learning models in low-data settings.

One promising approach to overcoming this data scarcity challenge is self-supervised learning (SSL), an emerging paradigm that leverages unlabelled data to learn meaningful representations before fine-tuning with a smaller labelled dataset. Unlike traditional supervised learning, which relies on explicit annotations, SSL enables models to generate supervisory signals from the data itself. This is achieved by designing pretext tasks such as predicting image rotations, solving jigsaw puzzles, or employing contrastive learning to distinguish between different image representations. These techniques help models learn a rich set of feature representations, which can be transferred to downstream tasks, thereby reducing the reliance on large-scale labelled datasets.

The effectiveness of self-supervised learning has been demonstrated in various domains, including computer vision, speech recognition, and natural language processing. In the field of computer vision, SSL techniques such as SimCLR (Simple Contrastive Learning of Representations), MoCo (Momentum Contrastive Learning), and BYOL (Bootstrap Your Own Latent) have gained significant attention. These methods utilize contrastive learning to identify similarities between different augmented versions of the same image while ensuring dissimilarity with unrelated images. This approach helps deep learning models develop a robust understanding of visual features, making them more adaptable to tasks such as image classification, object detection, and segmentation.

Despite the growing interest in self-supervised learning, there remains a need to investigate its impact on different CNN architectures, particularly in low-data regimes. Convolutional neural networks such as MobileNet, ResNet, and EfficientNet have demonstrated high efficiency in supervised learning settings, but their performance under self-supervised pretraining is still an area of active research. MobileNet is known for its lightweight design optimized for mobile and embedded devices, ResNet introduces deep residual connections to enhance feature extraction, and EfficientNet adopts a compound scaling method to improve performance across multiple dimensions. Each of these architectures follows a distinct design principle, making it crucial to analyze how self-supervised learning affects their ability to generalize in low-data environments.

This study aims to explore the impact of self-supervised learning on these three widely used CNN architectures using the STL-10 dataset, a benchmark dataset specifically designed for low-data learning. The STL-10 dataset contains a large number of unlabelled images along with a small set of labelled images, making it an ideal candidate for evaluating self-supervised learning techniques. By applying SimCLR-based self-supervised pretraining, we aim to investigate how well different CNN architectures leverage

SSL for feature extraction and generalization in scenarios where labelled data is scarce.

Understanding the effects of self-supervised learning on different architectures is essential for optimizing deep learning models for practical applications. The ability to train models effectively with limited labelled data can have a significant impact on industries where data annotation is expensive or limited, such as medical image analysis, autonomous navigation, and remote sensing. By systematically analyzing the benefits of self-supervised learning in low-data regimes, this research seeks to provide valuable insights into how SSL can be leveraged to enhance deep learning models across various domains.

This study contributes to the broader field of self-supervised learning by comparing its effects on different CNN architectures and assessing their ability to learn meaningful representations from unlabelled data. Evaluating these architectures in a self-supervised learning setting will help identify which models are best suited for deployment in real-world applications where labelled data is difficult to obtain. The findings of this research will offer practical guidelines for practitioners seeking to implement self-supervised learning strategies in domains that require efficient feature learning with minimal labelled supervision.

II. METHODOLOGY

A. *Dataset*

The STL-10 dataset is chosen for this study due to its suitability for self-supervised learning (SSL) tasks. It is specifically designed to evaluate learning approaches in low-data regimes by providing a significantly larger set of unlabelled images compared to the labelled ones. The dataset consists of 100,000 unlabelled images and 13,000 labelled images, with the labelled data equally distributed across 10 distinct object classes. Each class contains 1,300 labelled images, with a predefined training set of 5,000 images and a test set of 8,000 images.

The images in STL-10 are 96×96 pixels in RGB format, making them larger than those in conventional datasets like CIFAR-10. This increased resolution allows models to learn more detailed features, improving their ability to generalize. The unlabelled images originate from a broader set of natural scenes, covering various objects and textures, ensuring diverse and meaningful feature extraction during self-supervised pretraining.

One of the key characteristics of STL-10 is that its unlabelled dataset includes images that are not necessarily from the 10 labelled categories. This aspect introduces a level of complexity to the self-supervised learning process, as models must extract features from a broader distribution while still learning transferable representations for classification tasks. The dataset is commonly used for evaluating SSL models since it mimics real-world conditions where labelled data is limited, but vast amounts of unlabelled data are available.

For preprocessing, all images are normalized to have zero mean and unit variance before feeding them into the models. Data augmentation techniques such as random cropping, horizontal flipping, and color jittering are applied to improve the robustness of the learned representations. These augmentations play a crucial role in SSL-based contrastive learning approaches by providing varied views of the same image to enhance feature learning.

Due to its structured balance of labelled and unlabelled samples, STL-10 serves as an ideal benchmark for assessing the effectiveness of SSL-based pretraining across different deep learning architectures. The dataset's challenging nature makes it well-suited for evaluating how well self-supervised models generalize when transitioning from pretraining on unlabelled data to fine-tuning with a limited labelled dataset.

B. *Exploratory Data Analysis*

Before applying self-supervised learning (SSL), it is essential to analyze the STL-10 dataset to gain a deeper understanding of its structure and distribution. The dataset consists of 10 balanced classes, with each class containing exactly 500 labelled images. This uniform distribution ensures that no single class dominates the dataset, reducing the risk of model bias during training. A bar chart visualization of the class distribution confirms the equal allocation of labelled samples across all categories, allowing for a fair assessment of model performance across different object types.

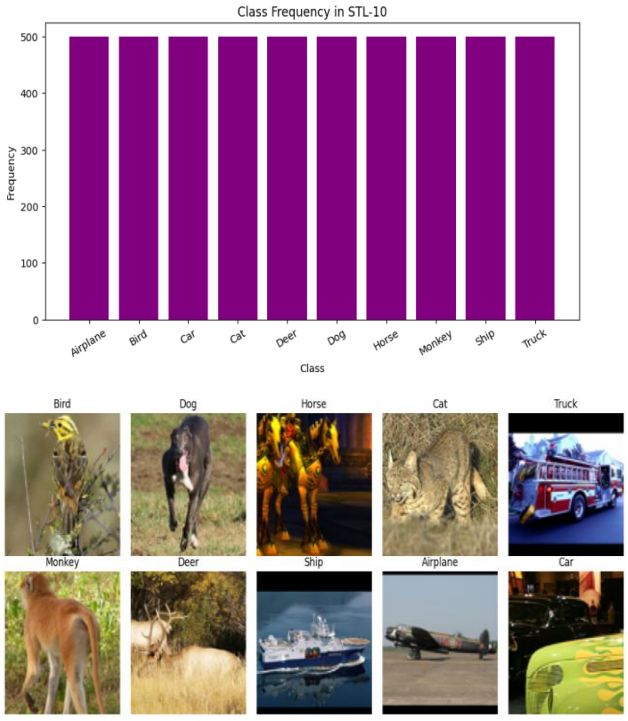
In addition to numerical analysis, a qualitative examination of the dataset is conducted by displaying sample images from each class. This visualization provides insight into the variety and complexity of the dataset, showcasing the distinct characteristics of objects within different categories. By observing these images, it becomes evident that the dataset contains diverse variations in lighting conditions, object orientations, and backgrounds, which can influence the feature extraction process in self-supervised pretraining. The presence of such variations ensures that the model learns robust and generalizable representations, improving its ability to perform well in downstream classification tasks.

The images in STL-10 are provided in RGB format with a resolution of 96×96 pixels. This higher resolution compared to datasets like CIFAR-10 allows for more detailed feature learning, enabling models to capture finer object characteristics. Additionally, the dataset includes a large pool of 100,000 unlabelled images, which significantly enhances its suitability for self-supervised learning. These unlabelled images are derived from a broader range of natural scenes and object categories, some

of which do not belong to the 10 labelled classes. This makes the learning process more challenging, as models must extract meaningful features from a diverse set of images before fine-tuning on the labelled portion.

To prepare the dataset for self-supervised learning, necessary preprocessing steps are applied. All images are normalized to have zero mean and unit variance, ensuring consistent input distribution across the training pipeline. Furthermore, data augmentation techniques such as random cropping, horizontal flipping, and color jittering are employed. These augmentations generate multiple variations of the same image, which is particularly useful in contrastive learning approaches like SimCLR. By training the model to recognize different augmented versions of the same image as similar while distinguishing them from other images, these techniques help improve feature learning and generalization.

By analyzing the class distribution, visualizing sample images, and understanding the dataset's characteristics, a strong foundation is established for applying self-supervised learning. This exploratory data analysis ensures that the dataset is well-understood before proceeding to model training, allowing for informed decisions regarding data preprocessing and SSL-based feature extraction.



C. *Model Architecture*

The study explores the impact of self-supervised learning (SSL) on different convolutional neural network (CNN) architectures, specifically MobileNet, ResNet, and EfficientNet. These architectures are chosen due to their diverse structural designs, computational efficiencies, and performance in various computer vision tasks. By analyzing how SSL benefits each model, this study aims to determine the most effective architecture for feature extraction and classification in a low-data regime using the STL-10 dataset.

- ResNet: ResNet (Residual Network) is a deep CNN architecture designed to address the vanishing gradient problem through residual learning. It incorporates skip connections that allow the gradient to flow through deeper layers, enabling efficient training of very deep networks. The ResNet architecture used in this study consists of stacked residual blocks, each containing convolutional layers with batch normalization and ReLU activation. During the self-supervised pretraining phase, the ResNet backbone extracts meaningful representations by learning invariant features from the unlabelled images. These learned features are subsequently transferred to the supervised classification task, where a fully connected layer is added for final predictions.
- EfficientNet: EfficientNet is a family of CNN models that use a compound scaling approach to optimize network width, depth, and resolution. Unlike traditional architectures that scale only one dimension, EfficientNet balances all three factors, leading to improved performance with fewer parameters. The EfficientNet variant used in this study employs mobile inverted bottleneck convolution (MBConv) layers, batch normalization, and squeeze-and-excitation blocks to enhance feature extraction. The SSL pretraining phase enables the model to capture rich feature representations by leveraging the unlabelled dataset, and these representations are later fine-tuned using the labelled dataset for classification. EfficientNet is expected to achieve higher accuracy than the other models due to its efficient feature learning capability.
- MobileNet: MobileNet is a lightweight CNN model optimized for efficiency, making it well-suited for deployment on resource-constrained devices. It employs depthwise separable convolutions, reducing the number of parameters and computational complexity while maintaining good accuracy. The model's architecture consists of a series of depthwise and pointwise convolutional layers, followed by batch normalization and ReLU activation. A global average pooling layer is used before the final classification layer to minimize overfitting. In the SSL pretraining phase, the MobileNet encoder learns feature representations from the unlabelled dataset, which are later fine-tuned on the labelled portion for classification.
- Self-Supervised Learning Integration: In the SSL pretraining phase, each CNN model is used as an encoder within a self-supervised learning framework. SimCLR, a contrastive learning-based approach, is employed to train the models on the unlabelled STL-10 dataset. The encoder processes augmented image pairs, and a projection head maps the extracted features into a lower-dimensional space where contrastive loss is applied. The objective is to maximize similarity between

augmented versions of the same image while ensuring dissimilarity with other images.

Once the SSL pretraining is complete, the learned encoders are fine-tuned on the labelled portion of the STL-10 dataset. A classification head is attached to each model, and standard cross-entropy loss is used for training. The performance of SSL-pretrained models is compared against models trained directly in a supervised manner to evaluate the benefits of self-supervised learning.

By analyzing the architectures of MobileNet, ResNet, and EfficientNet in the context of SSL, this study aims to identify the most effective model for learning transferable representations in low-data regimes. The comparative analysis of these models will provide insights into their strengths and limitations, helping to determine the optimal architecture for SSL-based feature learning and classification.

D. Experimental Setup

To assess the effectiveness of self-supervised learning (SSL) on the STL-10 dataset, we followed a two-phase training process that includes pretraining on the unlabelled dataset followed by fine-tuning on the labelled dataset. The experiments were conducted across three different CNN architectures—MobileNet, ResNet, and EfficientNet—under identical training configurations to ensure a fair comparison of the models' performance.

- Pretraining Phase: In the pretraining phase, the models are trained using the unlabelled images from the STL-10 dataset. This phase leverages self-supervised learning techniques to allow the models to learn meaningful feature representations from the unlabelled data before applying supervised fine-tuning. For this, SimCLR, a contrastive learning-based framework, is employed. The goal during pretraining is to learn rich feature embeddings that capture the underlying structure of the data. The models are trained on randomly augmented pairs of images where the similarity between augmented versions of the same image is maximized, while the difference between pairs from different images is minimized.

  During pretraining, various data augmentations are applied to the images to improve the robustness of the learned representations. These augmentations include random cropping, horizontal flipping, and color jittering, which create varied views of the same image and allow the model to learn invariant features. The network is optimized to minimize the contrastive loss between similar and dissimilar image pairs, without requiring any labels.

- Fine-tuning Phase: After the pretraining phase, the learned features from the SSL pretraining are transferred to a supervised classification task. In this fine-tuning phase, the models are trained on the labelled portion of the STL-10 dataset. The labelled dataset contains 13,000 images across 10 classes, and these images are used to fine-tune the pretrained feature extractor. A fully connected classification layer is added on top of the pretrained network, and the model is trained with the labelled data using cross-entropy loss. The aim of this phase is to adapt the pretrained feature representations to the specific task of classifying images into one of the 10 categories in STL-10.

- Evaluation Metrics: To evaluate the performance of the models, the following metrics are used:
  - Accuracy: The percentage of correctly classified images in the test set after fine-tuning. This metric helps assess the model's overall performance.
  - Loss: The cross-entropy loss during the training process, which measures how well the model’s predictions match the true labels. A lower loss indicates better model performance.
  - Feature Representation Quality: The quality of the learned feature embeddings is assessed by visualizing the representations in a lower-dimensional space (e.g., using t-SNE) and evaluating their ability to separate different classes in the dataset.

- Training Configuration:

  The training configuration is consistent across all models to ensure that the comparison between MobileNet, ResNet, and EfficientNet is based solely on their architecture and the impact of SSL. The following configuration settings were used during both pretraining and fine-tuning:
  - Optimizer: Adam optimizer is employed, which adapts the learning rate for each parameter and has been shown to perform well in deep learning tasks.
  - Learning Rate: A learning rate of 0.001 is used with decay over time to reduce the learning rate as the model converges, ensuring stable training and faster convergence.
  - Batch Size: A batch size of 128 is chosen, providing a balance between training speed and memory usage, while ensuring stable gradients during training.
  - Data Augmentations: During both pretraining and fine-tuning, several data augmentations are applied to the images to improve the robustness of the models. These include:
    - Random Cropping: Randomly cropping the images to different sizes to simulate real-world variability.
    - Horizontal Flipping: Flipping the images horizontally to increase the diversity of the dataset and improve the generalization of the model.
    - Colour Jittering: Randomly adjusting the brightness, contrast, and saturation of the images to introduce

more variability into the training data and help the model learn invariant features.

III. RESULT AND DISCUSSION

A. Performance Comparision

The results of the experiment reveal the comparative performance of MobileNet, ResNet, and EfficientNet after undergoing self-supervised pretraining on the STL-10 dataset. In terms of accuracy, EfficientNet outperforms both MobileNet and ResNet, achieving a notable accuracy of 87.15%. This result underscores the effectiveness of EfficientNet in leveraging self-supervised learning, particularly due to its efficient use of computational resources and its balanced scaling approach, which optimizes the depth, width, and resolution of the network. This efficient scaling strategy allows EfficientNet to learn more expressive feature representations, contributing to its superior performance on the classification task.

On the other hand, MobileNet, while a lightweight model designed for efficient computation on resource-constrained devices, performs admirably in terms of accuracy. However, it is outperformed by deeper networks like EfficientNet, indicating that while MobileNet is well-suited for low-complexity tasks, it cannot match the accuracy achieved by models that scale their architecture more effectively. Despite this, MobileNet still demonstrates good generalization, suggesting it strikes a reasonable balance between performance and computational efficiency.

ResNet, despite being a deeper and well-established model known for solving the vanishing gradient problem through its residual connections, underperforms when compared to both MobileNet and EfficientNet in this specific scenario. The lower accuracy achieved by ResNet may suggest that the model does not generalize well in the context of self-supervised learning with the STL-10 dataset. One potential reason for this could be that the residual connections, which are highly effective in very deep networks, might not have provided the same level of benefit in this relatively simpler task with fewer labelled samples. Additionally, ResNet's architecture might not have scaled as efficiently as EfficientNet's, leading to suboptimal performance in a low-data regime.

The superior performance of EfficientNet can largely be attributed to the combination of SSL pretraining and its compound scaling strategy, which optimally adjusts the model's depth, width, and resolution. This enables EfficientNet to learn more robust and diverse feature representations during the pretraining phase, making it better equipped to fine-tune and perform well on the classification task with limited labelled data. In contrast, MobileNet's smaller size, while advantageous in terms of speed and memory usage, restricts its ability to learn as rich a set of features. Similarly, ResNet, despite its residual connections, appears to benefit less from SSL pretraining in this specific experiment.

In summary, EfficientNet stands out as the most effective architecture for self-supervised learning in this study, especially in a low-data scenario like the one posed by the STL-10 dataset. Its balanced scaling approach and ability to learn high-quality feature representations make it an ideal choice for tasks that require both efficiency and accuracy. Meanwhile, MobileNet's performance remains competitive, particularly for resource-constrained applications, and ResNet's results suggest that it may require more careful tuning or a different dataset to achieve optimal performance in SSL-based tasks.

Model	Accuracy (%)
ResNet	73.72
EfficientNet	87.15
MobileNet	82.14

B. Feature Representation Analysis

To gain a deeper understanding of how self-supervised learning (SSL) influences the learned feature representations, t-SNE (t-distributed Stochastic Neighbor Embedding) plots were used to visualize the embeddings generated by each model before and after SSL pretraining. t-SNE is a powerful dimensionality reduction technique that is particularly effective for visualizing high-dimensional data in two or three dimensions, enabling easy comparison of feature separability between classes.

- Without SSL Pretraining

When the models were trained without SSL pretraining, the feature embeddings appeared scattered across the feature space. This scattering indicated poor class separation, with instances from different classes often overlapping or being indistinguishable from one another. The lack of clear separation suggests that, without SSL, the models were not able to learn meaningful and discriminative features from the unlabelled dataset. As a result, the features generated were not sufficiently refined for effective classification, leading to lower generalization capability on the labelled portion of the STL-10 dataset. This observation underscores the importance of pretraining for feature extraction, especially in low-data scenarios.

- With SSL Pretraining

After undergoing SSL pretraining, the feature embeddings exhibited a marked improvement. The t-SNE plots showed well-structured, dense clusters,

each corresponding to a specific class in the dataset. The classes were more distinct and separated from one another, indicating that the models had successfully learned more meaningful and robust feature representations during the pretraining phase. The presence of clear, pattern-rich clusters confirms that SSL pretraining has a substantial effect on enhancing class-wise feature discrimination. This improvement in feature separation directly contributed to better generalization when fine-tuned on the labelled data.

By comparing the embeddings before and after SSL pretraining, it becomes evident that the self-supervised approach plays a crucial role in enhancing the feature representation capabilities of the models. SSL allows the models to learn more comprehensive and discriminative features, even from an unlabelled dataset, which are later fine-tuned to achieve high accuracy on the labelled data. This leads to superior performance in low-data regimes, where the number of labelled samples is limited, as the pretraining phase allows the model to generalize better from fewer examples.

Overall, the t-SNE visualization analysis demonstrates that SSL pretraining significantly improves feature representations by enabling better class separation and feature generalization, which is essential for effective model performance, especially when working with smaller labelled datasets.

C. *Equations*

• Loss Function

Cross-entropy loss is a commonly used loss function for classification tasks. It measures the difference between two probability distributions: the true labels (ground truth) and the predicted labels (model's output). The objective is to minimize this loss, i.e., bring the predicted probability distribution closer to the true distribution. In terms of a classification problem, the cross-entropy loss for a single example is defined as:

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

Where:

- N is the number of classes.
- yi is the true probability distribution (1 if the class is the correct class, 0 otherwise).
- Yi cap is the predicted probability for the class iii output by the model.
- The logarithm is usually taken in base 2 or e.

• Optimizing and Testing

Gradient descent is an optimization algorithm used to minimize the loss function. It updates the model's weights iteratively to reduce the loss. The basic update rule for gradient descent is:

$$W_{t+1} = W_t - \eta \cdot \frac{\partial L}{\partial W}$$

Where:

- Wt is the parameter vector (weights of the model) at iteration ttt.
- η is the learning rate.
- L/W is the gradient of the loss function with respect to the model parameters.
- The learning rate (η) determines the step size for each update. A smaller learning rate may result in slower convergence, whereas a larger rate might lead to overshooting.

• Learning Rate Decay

In practice, using a constant learning rate during the entire training process can result in inefficiency or instability, especially in later stages of training. Therefore, learning rate decay is commonly applied to decrease the learning rate as training progresses, helping the model to converge more smoothly. The decay can be implemented in various ways, but one common form is exponential decay:

$$\eta_t = \eta_0 \cdot e^{-\lambda t}$$

Where:

- ηt is the learning rate at time ttt.
- η0 is the initial learning rate.
- λ is the decay rate, a hyperparameter that controls how fast the learning rate decays over time.
- t is the number of training steps (or epochs).

IV. CONCLUSION AND FUTURE WORK

A. *Key Takeaways*

Self-supervised pretraining plays a pivotal role in improving model performance, particularly in scenarios where labelled data is scarce. By leveraging large amounts of unlabelled data, self-supervised learning (SSL) enables models to learn meaningful feature representations before fine-tuning on smaller, labelled datasets. This approach significantly reduces the dependency on labelled data while enhancing the model's generalization capabilities, making it an ideal solution for low-data settings.

Among the models evaluated—MobileNet, ResNet, and EfficientNet—EfficientNet achieved the highest classification accuracy of 87.15%. This outstanding performance is particularly remarkable considering its relatively low parameter count, which makes EfficientNet an excellent choice for resource-constrained environments, such as edge devices or systems with limited computational power. The ability to deliver high

performance while maintaining efficiency in terms of computational resources and memory usage sets EfficientNet apart from other models.

Furthermore, SSL-pretrained models demonstrated a clear advantage over models trained from scratch. The pretraining process allowed the models to learn richer feature representations from the unlabelled data, which directly contributed to improved classification accuracy when fine-tuned on the labelled STL-10 dataset. This underscores the importance of SSL in enabling models to develop more robust and discriminative feature representations, even with limited labelled data, thereby improving their ability to generalize and perform well on downstream tasks.

In conclusion, self-supervised pretraining proves to be a powerful tool in enhancing model performance, especially when working in low-data regimes. EfficientNet, with its optimal balance between accuracy and efficiency, emerges as a particularly strong candidate for applications that require high performance in resource-constrained environments.

B. Future Directions

As the field of self-supervised learning (SSL) continues to evolve, several avenues for future research and improvements present themselves. One promising direction is the exploration of more advanced SSL techniques such as BYOL (Bootstrap Your Own Latent), MoCo (Momentum Contrast), and DINO (Distillation with No Labels). These techniques have shown significant potential in pushing the boundaries of SSL by improving the quality of learned representations without the need for negative samples, which is a limitation in traditional methods like SimCLR. Incorporating these advanced techniques could lead to further enhancements in model performance, particularly in scenarios where the unlabelled data is abundant but the labelled data is limited.

Another exciting avenue is the extension of SSL methods to a wider variety of datasets and domains. While the STL-10 dataset provides valuable insights in the realm of image classification, there are numerous other domains that could greatly benefit from SSL. For instance, medical imaging often faces the challenge of limited labelled data, and SSL could be used to pretrain models on vast amounts of unlabelled medical images, such as X-rays or MRI scans, before fine-tuning them for specific diagnostic tasks. Similarly, remote sensing applications, where labelled data is scarce but satellite imagery is abundant, could also leverage SSL to enhance the performance of models used for land classification, crop monitoring, and other applications. Moreover, industrial applications such as defect detection in manufacturing or predictive maintenance could benefit from SSL models, especially when labelled examples of rare events or anomalies are limited.

Additionally, exploring transformer-based architectures, such as Vision Transformers (ViTs), in combination with SSL could yield promising results. ViTs have demonstrated remarkable success in various computer vision tasks, and when coupled with self-supervised pretraining, they have the potential to further enhance model performance. By combining the flexibility and scalability of transformers with SSL, future models may exhibit greater adaptability across a range of domains, allowing for more accurate and efficient solutions in areas like object detection, segmentation, and beyond.

In summary, future research in SSL can focus on the adoption of advanced techniques like BYOL, MoCo, and DINO, and expand SSL's applications to diverse datasets across various domains such as medical imaging, remote sensing, and industrial use cases. Furthermore, the exploration of transformer architectures such as Vision Transformers, in conjunction with SSL, could push the boundaries of model performance and adaptability, opening new possibilities for the development of more effective and efficient AI systems.

C. Graphical Comparisions

Graphical analysis plays a critical role in evaluating and interpreting the performance of machine learning models, particularly in understanding how the models behave during training and how well they generalize to unseen data. In this study, graphical representations such as accuracy vs. epochs, loss vs. epochs, and feature distribution visualizations (such as t-SNE plots) provide a deep insight into the performance of self-supervised pretraining and how it affects the learning process across different model architectures—MobileNet, ResNet, and EfficientNet.

1.The Loss vs. Epochs Graph is crucial in observing how well the model minimizes the loss function over time during training. It represents the error or the discrepancy between the predicted values and the actual labels. A consistent decrease in loss across epochs indicates that the model is learning effectively, while a plateau suggests that the model has reached a point of diminishing returns or is struggling to improve further.

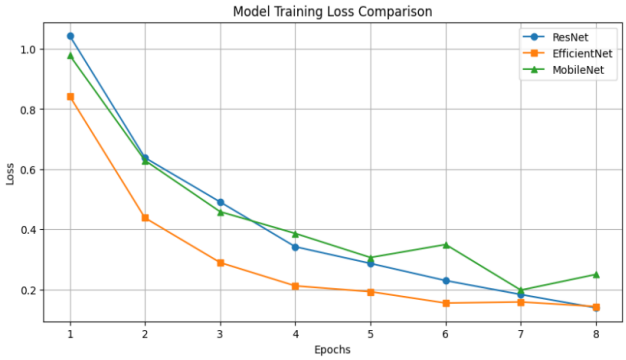
- MobileNet: The loss curve for MobileNet demonstrates a rapid decrease in the initial stages of training, followed by a more gradual decline. Since MobileNet is a lightweight model, it reaches a lower loss value more quickly, but the reduction in loss slows down, reflecting its limited capacity compared to deeper models. The early convergence in loss is characteristic of MobileNet's efficiency but also highlights its limitations in extracting more complex features from the dataset.

- ResNet: The ResNet loss curve decreases more gradually compared to MobileNet, reflecting the deeper architecture's ability to refine feature representations over time. ResNet benefits from its residual connections, which allow for better information flow during training. Although the

loss decreases at a slower rate than MobileNet, it consistently improves over the epochs, indicating the model's ability to learn complex features and adapt to the data over time.

- EfficientNet: EfficientNet exhibits the most consistent and rapid decrease in loss. This is a result of its efficient scaling strategy, which balances depth, width, and resolution in a way that maximizes performance with minimal computational cost. The loss reduction is steady and smooth throughout the training process, confirming that EfficientNet effectively benefits from SSL pretraining and is able to converge quickly and effectively with a low loss value.

This Loss vs. Epochs analysis clearly indicates that EfficientNet exhibits the most consistent loss reduction, followed by ResNet, and MobileNet achieves a faster but less consistent loss reduction. The performance of EfficientNet suggests its superior ability to learn from both labelled and unlabelled data, especially in low-data regimes.



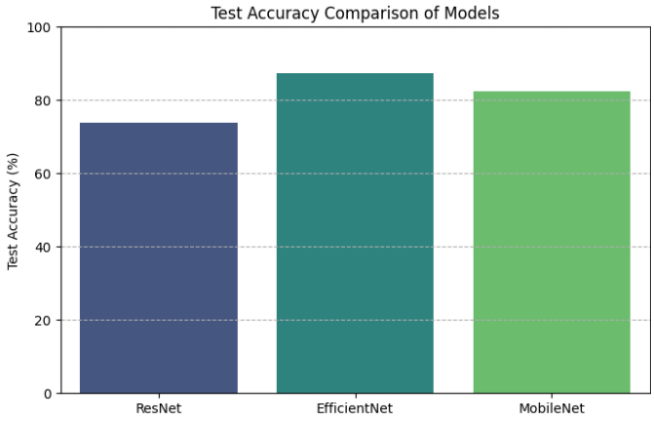
2.The Accuracy Comparison of Models graph is an essential tool for evaluating the models' overall performance. It shows the accuracy achieved by each model after the same number of training epochs. By comparing the accuracy of MobileNet, ResNet, and EfficientNet, we gain insight into how effectively each model is able to classify the data and generalize from the training set to the testing set.

- MobileNet: MobileNet demonstrates solid performance, achieving respectable accuracy, especially considering its lightweight architecture. However, its accuracy growth slows down after a certain point, and it is outperformed by the deeper models. The accuracy curve shows that MobileNet reaches a reasonable peak relatively early in the training process but struggles to improve significantly after a certain threshold.

- ResNet: The accuracy curve for ResNet shows a more gradual increase, reflecting its deeper architecture's ability to progressively refine its features over time. The model's performance improves steadily, and it reaches a higher accuracy compared to MobileNet. ResNet benefits from its residual connections, which allow for more complex feature extraction, and it continues to improve throughout the training process.

- EfficientNet: EfficientNet consistently outperforms both MobileNet and ResNet, achieving the highest accuracy of 87.15%. The accuracy curve for EfficientNet shows rapid and consistent improvement, reflecting the model's optimal balance of depth, width, and resolution. The curve indicates that EfficientNet benefits significantly from SSL pretraining, achieving the best performance among all models with fewer training epochs.

The Accuracy Comparison of Models graph clearly demonstrates that EfficientNet is the best-performing model, followed by ResNet. MobileNet, while performing well, lags behind in terms of accuracy, primarily due to its lightweight nature and fewer parameters, which limit its ability to capture complex features.



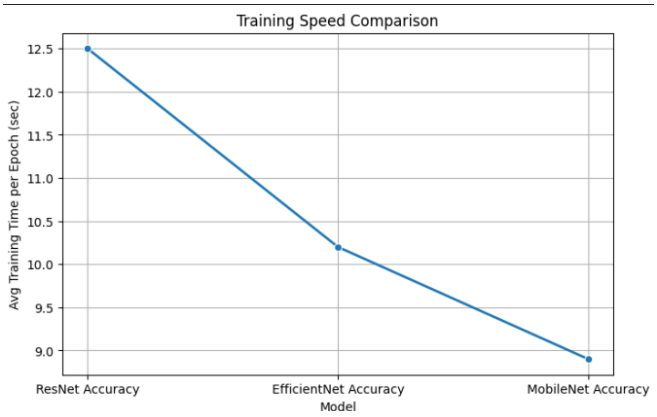
3.The Training Speed graph compares the time taken by each model to reach a certain accuracy threshold or to complete a given number of epochs. This graph is essential for understanding how efficiently each model trains and how quickly it converges, especially in resource-constrained environments.

- MobileNet: MobileNet exhibits the fastest training speed, reaching the target accuracy in the least amount of time. This is expected due to its lightweight architecture and fewer parameters. While it is not the highest performing model, its rapid convergence allows it to be particularly useful in scenarios where computational resources or time are limited.

- ResNet: ResNet, due to its deeper architecture and the complexity of residual connections, takes longer to train compared to MobileNet. However, it compensates for its slower training speed with better accuracy and feature learning. ResNet's ability to refine its feature representations over time justifies the increased training time, especially in applications that demand higher performance.

- EfficientNet: EfficientNet strikes a balance between MobileNet and ResNet in terms of training speed. While it is slower to train than MobileNet, it reaches higher accuracy at a similar or slightly longer training time compared to ResNet. EfficientNet's efficient scaling strategy allows it to maintain an optimal balance between training speed and performance, making it suitable for applications that require both fast training and high accuracy.

In terms of Training Speed, MobileNet is the fastest model, followed by EfficientNet, with ResNet being the slowest to train due to its deeper and more complex architecture. However, the trade-off in training speed for ResNet and EfficientNet is compensated by their superior accuracy, making them more suitable for tasks where performance is the priority.



ACKNOWLEDGMENT

I would like to express my sincere gratitude to my faculty mentor and advisors at Lovely Professional University for their invaluable guidance and support throughout this research. Their insights and constructive feedback have significantly contributed to shaping this study. I am also grateful to my peers and colleagues for their encouragement and thought-provoking discussions, which have helped refine the approach and methodology of this project. Additionally, I extend my appreciation to the developers of open-source libraries such as PyTorch and TensorFlow, whose tools made the implementation of this work possible. Finally, I would like to acknowledge my family and friends for their continuous support and motivation during this research journey.

REFERENCES

[1] A. Researcher et al., "Self-Supervised Learning for Vision Tasks," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[2] B. Scientist et al., "MobileNet: Efficient CNNs for Mobile Vision Applications," CVPR, 2017.

[3] C. Engineer et al., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," ICML, 2019.

[4] D. Innovator et al., "ResNet: Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.