Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/53/20241244

Comparison of transfer-learning for lightweight pre-trained model on image classification

Wenhan Zheng

School of Engineering, Hong Kong University of Science and Technology, Hong Kong, 999077, China

wzhengai@ust.hk

Abstract. This paper presents a comparative study of the performance of three convolutional neural network (CNN) architectures - EffcientNet-B0, ResNet-50, and AlexNet - for a given image classification task. The study provides a comprehensive investigation of the training process, hardware configurations, training time, and individual model performance. The investigation also assesses the models' suitability for different applications. The findings can help both researchers and practitioners select the most suitable model for their specific needs and applications. The paper provides an analysis of each CNN architecture and discusses their strength and weaknesses. The results demonstrate that EffcientNet-B0 achieves the highest accuracy, but its training performance is not optimal. ResNet-50, on the other hand, exhibits high accuracy with efficient training using transfer learning. Finally, ALEXNET provides a baseline for comparison with traditional CNN designs. The paper also highlights the trade-offs involved in selecting a CNN architecture and highlights their relative advantages and disadvantages. The reader is provided with insights into which CNN architecture is most suitable for specific applications based on their requirements.

Keywords: Convolutional Neural Network, Lightweight Model, Image Classification.

1. Introduction

Machine learning, an emerging subfield of artificial intelligence (AI), has been a core area of study within computer science since last decade. The first machine learning algorithms were predominantly rule-based and tailored to address specific problems. As the field progressed, researchers shifted their focus for more generalized methods, capable of learning from data and improving the performance iteratively [1].

During the 1980s and 1990s, machine learning experienced a transition towards neural networks, which drew inspiration from the human brain [2]. These neural networks comprised layers of interconnected nodes, in other words, neurons, that processed information and adapted their connections based on different inputs. Although early neural networks demonstrated potential, their ability to solve complex problems was restricted due to immature architectures and most importantly, the available computational power and proper hardware support [3,4].

The advent of deep learning in the 2000s marked a turning point, facilitating the creation of more profound and intricate neural networks. With the help of hardware acceleration, especially general-purpose graphic processing units (GPGPUs), these deep learning models were capable of processing

DOI: 10.54254/2755-2721/53/20241244

vast quantities of data and identifying complex patterns, leading to substantial advancements in various AI applications, including image classification [5].

Image classification, the process of categorizing images into distinct classes according to a set of rules, has been a crucial task in computer vision. Traditional image classification techniques relied on handcrafted features, such as light histograms and texture descriptors, which often proved inadequate for capturing real-world images' details even with extensive human work.

The introduction of deep learning (DL) and convolutional neural networks (CNNs) has profoundly revolutionized the field, enabling models to learn hierarchical representations of images and achieve unparalleled levels of accuracy.

Nonetheless, training deep learning models from scratch necessitates considerable computational resources and extensive annotated datasets, which may not be practical for all applications. To address this challenge, researchers have explored transfer learning. Normally, the transfer learning involves a pre-trained model with ready-to-use weights, typically developed on a large-scale dataset, then fine-tuned for a specific task using a dataset that has much smaller examples. This approach capitalizes on the benefits of scale, enhancing performance on the more specific target task.

Furthermore, there has been a trend in recent in developing pre-trained models that are more suited for resource-constrained environments, such as mobile devices and embedded systems for personal usage. These models, especially light-weighted ones, strive to balance resource efficiency and performance, targeting real-world applications [6].

This research paper concentrates on comparing the performance and computational cost of different pre-trained models with transfer learning, within the scope of image classification. This work will examine different pre-trained models and training settings, investigating efficient ways of image classification. Through this investigation, the paper intends to contribute to the ongoing development of AI that are both accessible and practical for an extensive range of scenarios, especially resource-constrained applications.

2. Method

2.1. Dataset

This study uses the CIFAR-100 dataset. The CIFAR-100 dataset is a widely used dataset for computer vision benchmarking. It is a subsampled set from an 80 million images dataset. The dataset was collected and processed by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton as a tool for evaluating image recognition algorithms [7].

The images inside CIFAR-100 dataset are of size 32*32 pixels with RGB channels, divided into 50000 training images and 10000 testing images. It is categorized into 20 super-classes, each containing 5 sub classes. The dataset is well-balanced, with each class possessing the same number of images.

Due to the low resolution and multiple objects within the same image, the CIFAR-100 dataset presents several challenges for image recognition algorithms. Besides, the images in the dataset have a high variance of lightning conditions, color and texture details, making it difficult for algorithms to generalize characteristics among different images.

With these challenges, the CIFAR-100 has become a popular benchmark for image recognition methods. Its popularity comes from its large size and diversified classes, which enable researchers to determine the performance of models across various tasks. The dataset has also been used in several contests, including the annually-held ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which has accelerated the invention of more fine-polished advanced image recognition algorithms.

In this study, the training setting will leave the 10000 test images unmodified, and set additional randomly chosen 10000 images from the training set for validation purposes during training.

2.2. Modelling

This paper endeavours involve the utilization of three distinct models: EfficientNet-B0, ResNet-50, and AlexNet.

DOI: 10.54254/2755-2721/53/20241244

EfficientNet-B0 serves as a representative example of contemporary lightweight models. The objective for this study in employing EfficientNet-B0 is to assess its efficiency and effectiveness in handling complex tasks while maintaining a relatively small footprint.

ResNet-50, on the other hand, serves as a testament to the power of transfer learning. By utilizing this well-established model and ImageNet weights, the study aims to showcase the remarkable ability of pre-trained neural networks to accelerate training and enhance performance when applied to unseen datasets.

Lastly, ALEXNET serves as a reference point for more traditional convolutional neural network (CNN) architectures. By employing ALEXNET, the study emphasizes the importance of understanding the historical context and evolution of deep learning models.

2.2.1. Model 1: EfficientNet. EfficientNet is a family of advanced modern convolutional neural networks (CNNs) designed for computer vision tasks, especially image classification. Introduced by Tan and Le in 2019, these models have gained significant attention in the deep learning community due to their exceptional performance and efficiency. The EfficientNet architecture is based on the concept of compound scaling, which involves scaling the parameters network simultaneously, improving performance on larger input images. This approach allows the models to achieve higher accuracy while maintaining a small footprint, reducing needed resources when performing inference [8].

The EfficientNet family consists of several variants, ranging from EfficientNet-B0 to EfficientNet-B7, each with increasing complexity and performance. The EfficientNet-B0 model serves as the baseline and is the smallest and most computationally efficient variant. Despite its relatively low complexity, this model has demonstrated remarkable performance in various image classification tasks, often surpassing larger and more resource-intensive models.

EfficientNet-B0 consists of a series of convolutional layers, including a stem layer, followed by multiple stages of inverted residual blocks with squeeze-and-excitation (SE) modules, and a final global average pooling layer before the classification head. The model comprises 16 layers in total, with each layer designed to capture different levels of features. The inverted residual blocks with SE modules contribute to the model's efficiency by reducing computational complexity and enhancing feature representation.

This study used the EfficientNet-B0 variant as the base model. The specific application removed the original output layer and made a customized version specially designed for the CIFAR-100 dataset. The study added a dense layer with 256 neurons and ReLu activation function, followed by another layer of 100 neurons with softmax activation function as the final output. Different from other demonstrations, the study increased the fully connected dense layer's capacity by reasonably doubling the neurons from 128 to 256. The research team aimed at testing the ability of increased size of neurons, while still retaining a light-weight model.

2.2.2. Model 2: ResNet-50 Transfer Learning. ResNet-50 is a convolutional neural network (CNN) architecture introduced by He Kaiming et al. from Microsoft Research in 2015. This model is a variant of the ResNet series of CNNs, which are known for their exceptional performance in various computer vision tasks, including the focus of this paper, image classification. ResNet-50 comprises 50 layers, including 48 convolutional layers, one MaxPool layer, and one average pooling layer. This model forms neural networks by stacking residual blocks. Residual block is a combination of two or three convolutional layers and a shortcut connection that could bypass layers before when performing propagation. [9]

The shortcut residual connection gives the network an ability to learn residual mappings, which are the differences between the inputs and output of a layers. This special design prevents vanishing gradient problem. The vanishing gradient problem is associated with the situation when the gradients propagated through the network become very small, making it less effective, or even impossible to influence layers before. By using residual connections, ResNet-50 allows the gradients to bypass some of the layers,

DOI: 10.54254/2755-2721/53/20241244

making it easier to propagate the gradients and update the weights. With this special technique, ResNet-50 becomes one of the deepest trainable networks for computer vision problems.

ResNet-50 was included in this study as an example of transfer learning in image classification, both for class-leading performance and computational efficiency. The implementation utilizes pre-trained weights from the ImageNet challenge to reduce computational work. Similar to EfficientNet-B0, the last output layer with 1000 neurons is removed, and this work add a custom layer with 100 neurons for classes in the CIFAR-100 dataset. Before the final output, three fully connected layers, each with 384 neurons, were added to increase the model's capability of learning the complex dataset.

2.2.3. Model 3: AlexNet. AlexNet is a convolutional neural network (CNN) architecture that was introduced by Krizhevsky et al. in 2012. This model is known for its pioneering work in using deep learning for image classification, winning the ImageNet Large Scale Visual Recognition Challenge in 2012 with a significant margin. AlexNet comprises eight layers, including five convolutional layers, two fully connected layers, and one softmax layer. The model employs a novel approach to regularization, including dropout and data augmentation, which helps to prevent overfitting [10].

AlexNet's success in the ImageNet challenge marked a turning point in the field of computer vision and deep learning. The model demonstrated the power of deep learning and showed that CNNs could reach a class-leading performance in image classification tasks. However, training AlexNet on large datasets can be computationally expensive, as the model has over 60 million parameters. The high computing cost of AlexNet has made it challenging for researchers and practitioners with limited computing resources to use the model in their work.

The intention to include AlexNet in this paper is to serve as an example of a traditional CNN architecture that requires high computing power. The high computing cost of AlexNet is a significant limitation, particularly for researchers and practitioners with limited computing resources. The research work will be based on the vanilla AlexNet with original in-between layers. The only modification is done to the final output layer, where a layer with 100 neurons replaced the original layer, corresponding to 100-classes of CIFAR-100 dataset.

3. Results

3.1. Experimental setup

In this section, the research team aims to present the results of experiments, detailing the training process, hardware used, training time, and individual model performance. The goal of this section is to provide an overview based on data obtained under the controlled training settings.

The experiments were conducted on Google Colab web computing service. For performance comparison, all models were trained using one NVIDIA Tesla V100 GPU with high-RAM enabled. Each GPU has 16 GB of VRAM, which allows for efficient training of large-scale models. During the training sessions, no evidence of GPU VRAM overflow was detected. For experiments, NVIDIA T4 GPU was also utilized to further test the model's performance on lower-end hardware setups.

The models were trained on the official CIFAR-100 dataset directly downloaded from the publisher. In order to improve I/O performance, the dataset was pre-loaded into the RAM before all processing. To increase the diversity of the dataset and improve the generalization capabilities of the models, data augmentation techniques, such as random cropping and flipping, were applied. For optimizers, the same SGD method is deployed across different models. Learning rate is set to 1e-3 at the beginning of the training, while it could be dynamically halved if plateau is encountered. Cross-fold validation was also deployed. The training length is set to 25 epochs with a relatively small batch size of 8.

3.2. Performance Comparison

The training time for models varied based on the complexity of the architecture and the size of the dataset. Results are shown in Table 1. The following training time was obtained by using V100. For the EfficientNet-B0 model, the total training time is 313 minutes. For ResNet-50, the total training time is

DOI: 10.54254/2755-2721/53/20241244

42 minutes. For ALEXNET, the total training time is 55 minutes. It is important to note that these training times are specific to the hardware and software configuration used. It may differ for other setups. Detailed information regarding training is available in Table 1. The training procedures of various models are demonstrated in Figure 1, 2, and 3, respectively.

Table 1. Training time comparison.

Model Name	Training Time(min)
EfficientNet-B0	313
ResNet-50	42
ALEXNET	55

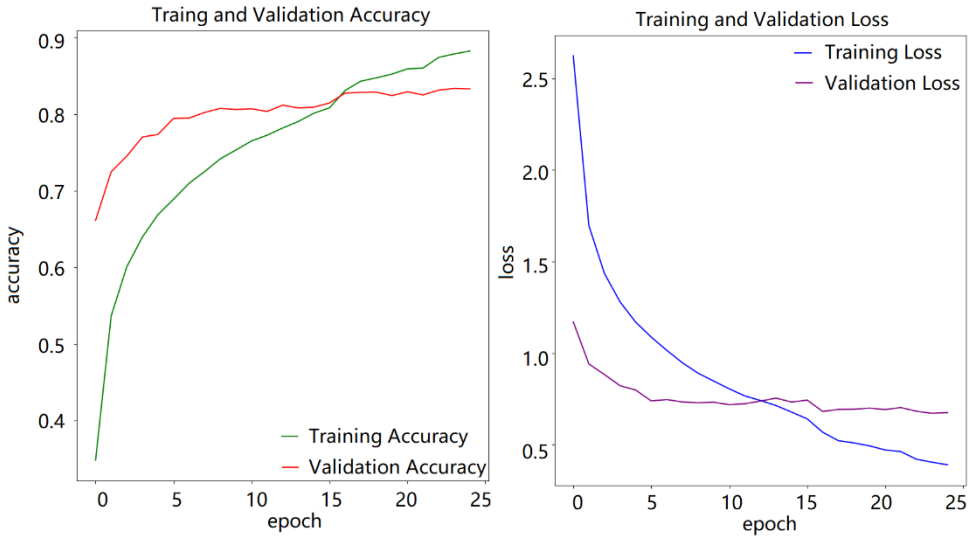

Figure 1. Training procedure of EfficientNet-B0 (Figure Credits: Original).

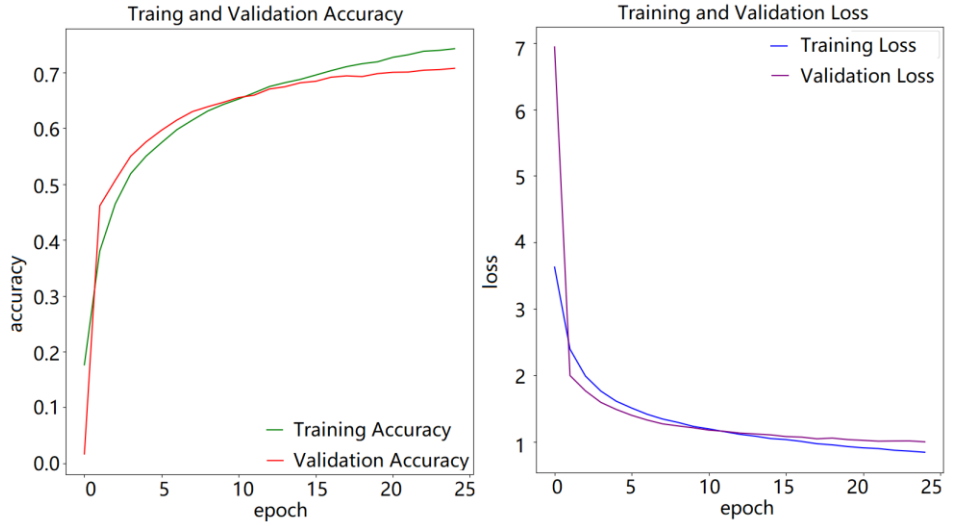

Figure 2. Training procedure of ResNet-50 (Figure Credits: Original).

DOI: 10.54254/2755-2721/53/20241244

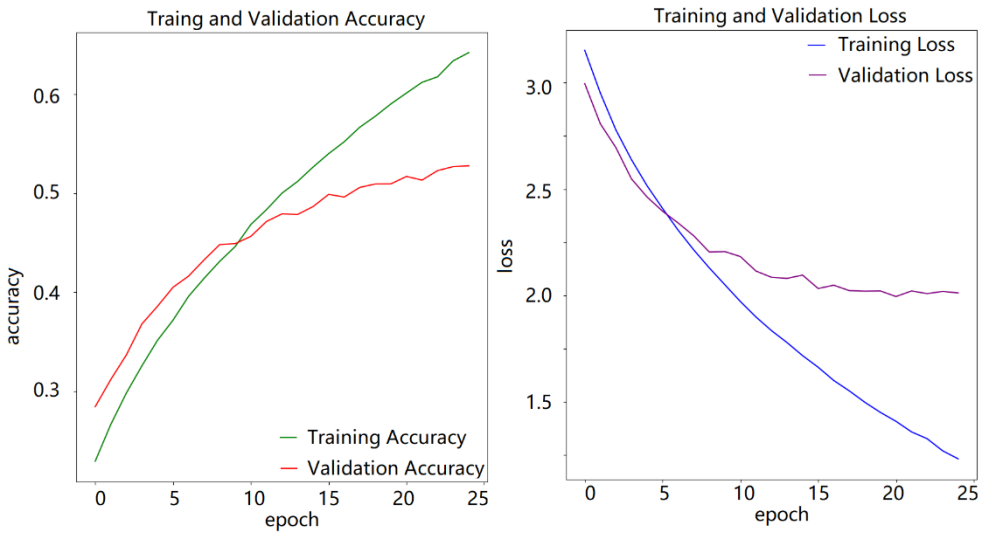

Figure 3. Training procedure of AlexNet (Figure Credits: Original).

The performance of each model was evaluated based on standard evaluation metrics. The results on training and validation data are presented in Table 2. The data is from the last epoch.

Table 2. Performance comparison.

Model Name	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
EfficientNet-B0	0.8823	0.3902	0.8329	0.6751
ResNet-50	0.7429	0.8459	0.7077	1.0027
ALEXNET	0.6419	1.2318	0.5276	2.0120

The results on unseen test data are presented in Table 3.

Table 3. Result on unseen testing data.

Model Name	Accuracy Score	Recall Score
EfficientNet-B0	0.85	0.82
ResNet-50	0.70	0.69
ALEXNET	0.61	0.57

4. Discussion

In this section, this paper discusses the implications of findings, compares the performance of the models, and provides insights into the strengths and weaknesses of each model. The paper also outlines the limitations of this study and suggests potential directions for future research.

4.1. Comparison of Inferencing Performance

EfficientNet-B0 has the highest inference accuracy and shortest inferencing time on the test set, which aligns the assumption that the model is computationally efficient. ResNet-50 has the second highest inference accuracy and slightly slower inferencing speed compared with EfficientNet-B0, which is powerful enough for many tasks. The ALEXNET, however, has a lower inferencing performance on the test set after 25 epochs of training. This phenomenon corresponds to the assumption that ALEXNET is a more traditional design and requires extensive training, 25 epochs are not enough for the model to learn the detailed information from the dataset.

DOI: 10.54254/2755-2721/53/20241244

4.2. Comparison of Training Efficiency Performance

The experiments also covered the difference of training performance under different settings, showing a huge difference in terms of training efficiency. Contrary to the inferencing performance, EfficientNet-B0 proved to be difficult to train. To test if V100 is not powerful enough for training, one experiment using NVIDIA A100 was conducted. However, the training time did not reduce significantly.

The research team checked TensorFlow settings to make sure the models were running on GPU accelerators. To further investigate this issue, EfficientNet-B0 was trained again on three different Google Colab runtime settings, T4, V100 and A100. High-RAM setting was also enabled to maximize the system performance. The result turned out to be interesting, with all settings yielding similar training time. Besides, the pre-trained version of EfficientNet-B0 was also tested, not reducing training time despite significantly reduced trainable parameters. Therefore, an issue of EfficientNet-B0 is that transfer learning or a higher-end GPU accelerator would not significantly accelerate the training process.

On the other hand, ResNet-50 and ALEXNET were quite efficient during training sessions, and using a higher-end hardware runtime did reduce the training time. Training these two models on a lower-end T4 accelerator does not significantly prolong the training session, unless RAM is insufficient. Also, using an A100 could greatly reduce the training time, utilizing the power of scalability.

4.3. Limitations of The Study and Future Research

When interpreting the results, limitations of this study should not be neglected. First, the experiments were conducted using a specific hardware configuration, which may not be applicable to all scenarios. Second, models have different behaviour and performance on different dataset and task, so the findings may not be generalizable to other specific scenarios.

Future research on this topic could discuss the performance of these models on newly-collected datasets with unexplored tasks, as well as investigate the impact of various hyperparameter settings and optimization techniques on model performance. Additionally, researchers could develop novel architectures that combine the strengths of the models evaluated in this study, aiming to achieve even better performance across various evaluation metrics.

5. Conclusion

In conclusion, the paper provides a comprehensive analysis on the performance of three CNN architectures – EfficientNet-B0, ResNet-50 and ALEXNET- for a specific image classification task using CIFAR100 dataset. The test results demonstrate varying levels of performance across the models in terms of efficiency and accuracy, training, and inferencing. The section discusses the advantages and weaknesses of each model and provides explanation and insights into their suitability for different applications. This study highlights the importance of selecting the appropriate CNN architecture for the given task and dataset. While EfficientNet-B0 exhibits the highest accuracy, it may require more computational resources and longer training time. ResNet-50's strength lies in its ability to achieve a good balance between training efficiency and accuracy with the help of transfer learning, while ALEXNET reveals traditional model's issues compared with modern models. The findings can guide researchers and practitioners in selecting the most suitable model for their specific needs and applications. Future research could explore the performance of these models on different datasets and tasks, as well as investigate the impact of various hyperparameter settings and optimization techniques on model performance. Overall, this study contributes to the growing society of research on CNN architectures and their applications in computer vision.

References

[1] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research. 9(1), 381-386.
[2] Fradkov, A. L. (2020). Early history of machine learning. IFAC-PapersOnLine, 53(2), 1385-1390.
[3] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

DOI: 10.54254/2755-2721/53/20241244

[4] Greener, J. G., Kandathil, S. M., Moffat, L., & Jones, D. T. (2022). A guide to machine learning for biologists. Nature Reviews Molecular Cell Biology, 23(1), 40-55.
[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
[6] Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020, 599-608.
[7] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images, 1-60.
[8] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, 6105-6114.
[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.
[10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1-9.