Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/46/20241243

Performance comparison of deep learning-based image classification algorithms on ImageNet

Zhelin Liu

College of Liberal Arts, University of Minnesota-Twin Cities, Minneapolis, Minnesota, 55455, United States

Liu02133@umn.edu

Abstract. The rapid evolution of Artificial Intelligence (AI) technology has propelled image recognition to the forefront of computational advancements. Since the inception of Convolutional Neural Networks (CNNs), the field has expanded into a multitude of sophisticated models and their derivatives, each tailored to address specific challenges and applications. Image recognition's landscape encompasses foundational tasks such as object and face detection, extending to more specialized applications like emotion analysis, optical character recognition, and complex interpretation of biological imagery. This domain's historical perspectives trace back to models like AlexNet, which set benchmarks with accuracy rates of around 70%. Fast forward to contemporary times, and advanced algorithms consistently achieve accuracy figures beyond the 90% threshold on benchmark datasets like ImageNet. Moreover, the diversification of AI applications has led to the development of models like MobileNet, which are intricately designed for streamlined efficiency on mobile devices, balancing performance with resource constraints. This discourse will navigate the intricate maze of image recognition, primarily leveraging insights from the ImageNet dataset as a canonical reference. By the end of this exploration, this work will discuss several cost-efficient models. Finally, this work will also cover some complex algorithms with high accuracy. All these algorithms use different approaches and obtain good performance in either cost-efficiency or accuracy. This discourse will provide an overview of these algorithms, detailing their novelty, implementation, and experimental results for accuracy and cost-efficiency.

Keywords: Image Classification, Model Comparison, Deep Learning.

1. Introduction

Artificial intelligence technology is evolving rapidly, and its potential applications are vast. Image recognition is one of its most classic and prevalent domains. Beginning with CNNs, there are now various neural network models and even variants of the same model. The essential applications and technologies in image recognition encompass object detection, face recognition, handwriting detection, optical character recognition, emotion analysis, and scene identification. While this may seem broad, specific instances involve the recognition of traffic violations by vehicles and machine-aided analyses of intricate biological images. Image recognition research commenced quite early and has achieved significant results over the years. From the AlexNet with an accuracy rate of around 70% to contemporary models achieving over 90% accuracy on ImageNet [1]. Furthermore, these algorithms

DOI: 10.54254/2755-2721/46/20241243

possess diverse characteristics. For instance, algorithms like MobileNet are specifically designed for mobile devices, showcasing impressive efficiency and resource usage. Thus, discussing different image recognition algorithms is immensely meaningful. Here, we'll begin the discussion with the classic dataset, ImageNet, exploring the features and performance of various algorithms trained on it. Ultimately, we'll select and compare four representative algorithms from different eras.

2. AlexNet

The first algorithm is AlexNet. This algorithm won the championship in the ImageNet Large Scale Visual Recognition Challenge in 2012. AlexNet also introduced ReLU, dropout, and dual-GPUs training, which are novelty and advanced in 2012. Devices with dual-GPUs will obtain excellent accuracy on OCR compared to other devices in 2012. This architecture contains five convolutional and three fully connected layers. The initial five layers are convolutional, whereas the subsequent three are fully connected. The output from the final fully-connected layer is directed to a 1000-class softmax, generating a distribution for the 1000-class labels. AlexNet aims to optimize the multinomial logistic regression target, which aligns with maximizing the average log probability of the accurate label within the predicted distribution across training samples [1].

2.1. Architectures

AlexNet uses Rectified Linear Units (ReLU) as its activation function [1], also it was the first large-scale model to utilize the ReLU function. Using the terminology of Nair and Hinton [2], neurons with the nonlinearity $f(x) = \max(0; x)$ are termed as ReLU. Compared to networks with tanh units, deep convolutional networks utilizing ReLUs accelerate their training by several multiples.

After ReLU, Local Response Normalization (LRN) is applied. This is a technique that simulates a form of lateral inhibition, a concept borrowed from neuroscience [1]. Lateral inhibition is a process in the nervous system where activated neurons reduce the activity of their neighbors. By mimicking this, LRN will be able to enhance the model's generalization, making the activated neurons stand out more in their local neighborhood. In the network, when a particular kernel, or feature detector, responds strongly to a specific feature, its response will be normalized concerning the responses of neighboring kernels. This normalization helped in stabilizing the activations and reducing the chances of extreme response values.

2.2. Dual-GPU Training

AlexNet is designed to be trained in parallel on two NVIDIA GeForce GTX 580 GPUs [1]. This parallel computation approach allowed the model to train on a significantly large dataset and complete the training in a reasonable amount of time. Moreover, the use of GPUs enabled the training process to handle the large-scale matrix and vector operations efficiently, which were prevalent in the convolutional layers and fully connected layers. This strategic use of GPUs not only made the training of such a deep network feasible but also set a precedent for the widespread adoption of GPUs in the training of deep learning models.

2.3. Avoid Overfitting

In reducing overfitting, AlexNet also has 2 excellent solutions.

Data Augmentation is a technique that amplifies the training dataset and bolsters the generalization capability of a model. This technique will artificially create new training examples by making minor transformations to the original images, such as rotation, scaling, flipping, etc. This will infuse diversity into the training data so that the model is more likely to grasp the genuine underlying structure of data rather than specific, potentially noisy patterns. This is similar to studying the process of a student. Most of the time, students who finish more practice of the same chapter (all of these questions have made minor modifications) will have better scores than other students who only finish one practice of the same chapter.

DOI: 10.54254/2755-2721/46/20241243

Dropout is a very efficient version of the model combination. It is a technique of randomly "shutting off" a subset of neurons during the training process, in order to prevent the model from overly relying on any singular neurons during the training process [1].

3. MobileNet

MobileNets is an architecture, which focuses on designing models for mobile devices. It has 70.6% of accuracy. This model is developed by Google, many ARM-base chip such as Qualcomm Snapdragon devices can run this model. This is very amazing since a lot of models on that time requires strong GPUs and CPU. This is a successful and widely used model [2].

3.1. Depthwise Separable Convolution

This will separate a standard convolution into a depthwise convolution and a 1*1 convolution, and this combination can be considered as a pointwise convolution. In MobileNet, this design applies an individual filter to each input channel, then pointwise convolution utilizes a 1*1 convolution to combine the outputs of the depthwise convolution. Differing from traditional standard convolutions, depthwise separable convolution divides this proof into 2 separate layers, one for filtering and another one for combining. This factorization will decrease computational requirements and overall model size [2].

3.2. Width Multiplier and Resolution Multiplier

These two are a feature for making this architecture smaller and using fewer resources of computation. A width multiplier is a hyperparameter, which will shrink or expand the model's size and cost of computation by scaling the number of channels. This will reduce the model's size and requirement of computation. Interestingly, this can be used in any other models, in order to make the model size trade-off. Resolution multiplier is another hyperparameter that adjusts the input image's resolution. This will make the internal representation of each layer in the network proportionally scaled down [2]. Within these two hyperparameters, the computational cost will be reduced significantly.

4. ShuffleNet

ShuffleNet is a neuron network that requires lower device configuration, which means it is also suitable for mobile devices. This has 70.9% accuracy in the ImageNet dataset. Compared to other algorithms designed for mobile devices such as MobileNet which has been discussed previously, ShuffleNet is more efficient with slightly higher accuracy.

4.1. Channel Shuffle

This is one of the key features of ShuffleNet. Channel Shuffle enhances the interplay between input and output channels in group convolution. By drawing data from varied groups, a prior group layer's channels can be divided into distinct clusters. In a setup with "g" groups resulting in g × n channels, the output channels are modified to a (g; n) format, then transposed and flattened for the subsequent layer. The technique is adaptable across varying group counts, and the differentiability of Channel Shuffle supports its integration into holistic network training, enabling the design of intricate architectures with multiple group convolution layers [3].

4.2. Grouped Convolution

Pointwise Group Convolutions are crucial in assessing ShuffleNet models. Models were compared across group numbers from 1 to 8. With a group number of 1, the model lacks pointwise group convolution, mirroring the "Xception-like" structure [4]. According to Table 2, models with group convolutions ($g > 1$) consistently outshine those without ($g = 1$), especially in smaller models like ShuffleNet 0.5× and 0.25×. However, some models, like ShuffleNet 0.5×, experience performance saturation or decline with larger group numbers (e.g., $g = 8$). While larger group numbers offer broader feature maps, the decrease in input channels for each convolution might reduce representation capability. Interestingly, smaller models, e.g., ShuffleNet 0.25×, consistently improve with larger group numbers.

DOI: 10.54254/2755-2721/46/20241243

4.3. Pointwise Grouped Convolution

Evaluating pointwise group convolutions, ShuffleNet models with varied group numbers (1 to 8) were compared. A group number of 1 makes ShuffleNet resemble an "Xception-like" structure [4]. Models with group convolutions (g > 1) consistently outperform those without (g = 1). Notably, smaller models gain more from groups, with performance differences widening as the model size decreases. For some models, like ShuffleNet 0.5×, larger group numbers, e.g., g=8, could lead to stagnating or even declining classification scores. This is potentially due to reduced input channels for each convolution filter impacting representational capacity. However, for smaller models like ShuffleNet 0.25×, larger group numbers consistently improve results, indicating wider feature maps are particularly beneficial for these [4].

5. EfficientNet

EfficientNet introduced a new way of scaling deep learning models. EfficientNet-B7 is the most advanced model and it has 84.4% accuracy in the ImageNet dataset. Scaling dimensions are not independent in neuron networks. For high-resolution images, both network depth and width are supposed to increase in order to capture more detailed information. This will avoid saturated accuracy. This is a new method. φ is the user-specified coefficient, which indicates available resources for model scaling. α, β, and γ are constants determined by grid search. The expressions of depth, width, and resolution are: $d(depth) = \alpha^{\phi}$; $w(width) = \beta^{\phi}$; $r(resolution) = r^{\phi}$. Furthermore, because this architecture constrains $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$, so that the total Floating-Point Operations Per Second (FLOPS) will only increase about 2^{ϕ} for all ϕ. Thus, the computation is still efficient with this method. This model has 8 generations. It starts in EfficientNet-B0, which is considered as baseline model. EfficientNet-B7 is the latest generation [5].

6. MultiGrain

MultiGrain algorithm is designed for both image classification and instance retrieval. A typical approach for image classification involves training networks to predict class labels, while instance-level recognition or retrieval typically employs embeddings to distinguish between individual images or items. But in MultiGrain, the approach is to bridge these two tasks, thus benefiting both [6].

6.1. Spatial Pooling Operator

Typically, convolutional networks employ local pooling methods, such as max-pooling, to maintain stability against minor image shifts. Global spatial pooling, however, aims to simplify a 3D activation tensor into a singular vector [6]. Early models, taking AlexNet for example, making them sensitive to positional shifts. In contrast, modern architectures, e.g., ResNet, use average pooling for better positional invariance. However, image retrieval tasks demand more precise spatial information, leading to the development of the Generalized Mean Pooling operator (GeM) [6]. GeM is a tunable pooling approach focusing on an image's salient features. It's a broader representation of average and max-pooling. This paper pioneers the application of GeM in image classification, underscoring its efficacy, especially for high-resolution images [7].

6.2. Training Objective

To holistically address classification and retrieval, a joint objective function is necessary to discuss. This function bifurcates into a classification loss and a retrieval loss. For classification, the widely-used cross-entropy loss is employed [6]. However, retrieval is more nuanced. Two methods dominate: the contrastive loss, which distinguishes positive from negative image pairs using a set threshold, and the triplet loss, emphasizing the relational attributes of image triplets [8]. However, adjusting parameters for these methods can be cumbersome. To combat this, Wu et al. introduced a method that re-normalizes image embeddings and uses a modified contrastive loss, termed the margin loss [9]. Computation of this loss utilizes distance-weighted sampling, ensuring efficiency in the joint training environment. Notably,

DOI: 10.54254/2755-2721/46/20241243

this method's adaptability enables work with smaller batch sizes, alleviating the need for intricate parameter tuning [6].

6.3. Preprocessing

PCA whitening is a step applied in this model for transferring features learned from data argumentation to standard retrieval datasets. The effect of PCA whitening could be erased in the parameters of the classification layers, which means the whitened embeddings can be used not only for classification but also for instance retrieval. In this model, standard 224*224 resolution will be trained in this architecture because of p*, for proxy task for cross-validation [6].

7. Meta Pseudo Labels

This is a semi-supervised learning algorithm. Pseudo-label or self-training methods now have been successfully applied in the improvement of state-of-the-art models in many computer vision tasks [10]. However, meta pseudo labels are a more advanced technology to discuss. Meta pseudo labels have a systematic mechanism in which the teacher observes how its pseudo labels would affect the student, and then corrects the bias. In meta pseudo labels, the student and teacher learn parallelly [11]. The difference between pseudo labels and meta pseudo labels can be presented in Figure 1.

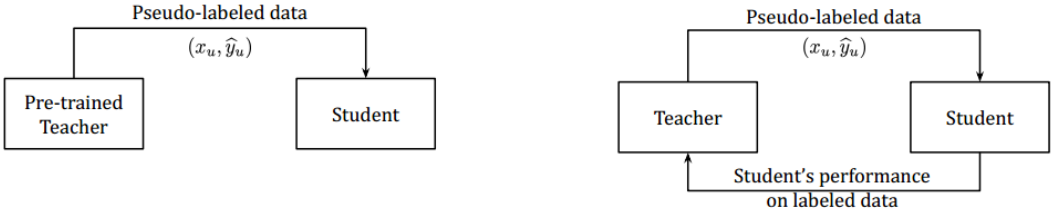

Figure 1. Pseudo label is presented on the left and meta pseudo label is presented on the right [11].

7.1. Pseudo Labels as an Optimization Problem

Two networks will be discussed here: teacher(T) and student(S). T has a parameter θ_T and S has parameter θ_S. Labeled data is represented as (xl; yl), which includes images and their corresponding labels. Unlabeled images are represented as xu. T(xu; θ_T) represents the soft predictions of the teacher network on the batch xu of unlabeled images, with a similar representation for the student. The cross-entropy loss between 2 distributions q, p will be given by CE(q;p) [11].

Pseudo-labels train the student model in order to minimize the cross-entropy loss on unlabeled data. However, this method teaches the "student" network using images without any labels. The goal is to get the student to make predictions that are as close as possible to predictions made by a well-trained "teacher" network. The performance of the student is strongly related to the guidance from teacher. The most unique part of this part is trying to adjust the teacher's guidance based on how well the student performs, aiming to fine-tune the learning [11].

7.2. Practical Approximation

To make Meta Pseudo Labels manageable, techniques from prior meta-learning research are adopted [12]. Instead of a multi-step approach, a one-step gradient update simplifies the student's optimization. This leads to a more hands-on objective for the teacher's role. While soft pseudo labels allow for traditional back-propagation, this study opts for hard pseudo labels for computational benefits in large-scale experiments. Both label types perform similarly, but hard labels require an adjusted gradient approximation approach. As a twist, the teacher's parameters evolve based on the student's optimization, fostering an iterative update process between the two [11].

DOI: 10.54254/2755-2721/46/20241243

7.3. Teacher's Auxiliary Losses

Meta Pseudo Labels show significant standalone efficacy. However, its performance is enhanced when the teacher is trained with added auxiliary objectives. In methodology, the teacher undergoes training with both supervised and semi-supervised objectives. While the former involves labeled data training, the latter incorporates the UDA objective [13]. A comprehensive pseudo code, detailing the integration of supervised and UDA objectives with Meta Pseudo Labels, is provided in Appendix B, Algorithm 1. Further, the student model, initially trained with unlabeled data and the teacher's pseudo labels, can be refined with labeled data for heightened accuracy, as showcased in the experimental section [11].

8. Results

ImagaNet is a large-scale visual database. This is designed for use in visual object recognition research. It contains over 14 million annotated images and has over 20000 categories. One of the primary goals of ImageNet was to advance the field of computer vision, particularly in large-scale image recognition. To this end, ImageNet launched an annual competition [14]. This challenge has drawn global attention from researchers in the field. In 2012, a deep learning model called AlexNet achieved a breakthrough performance in this competition, marking the rise of deep learning in the field of computer vision.

In this paper, the performance of each model will be discussed particularly on how they performed in the ImageNet database.

Table 1. Performance comparison on ImageNet dataset.

Model	Top 1 Accuracy	Top 5 Accuracy
AlexNet (Using ImageNet REAL dataset)	62.88%	-
MobileNet	70.6%	-
ShuffleNet	70.9%	91.5%
EfficientNet (FixEfficientNet- B0)	80.2%	95.4%
MultiGrain (NASNet-A-Mobile (350px))	75.1%	92.5%
MetaPsudo Label (EfficientNet-L2)	90.2%	98.8%

9. Discussion

Based on the different model design, these models are varying not only in performance but also at the cost of computation. In this section, the author will first discuss the difference in computation cost between AlexNet and MobileNet, and why the MobileNet have higher efficiency while obtaining even more accurate result. The model differences are compared.

9.1. Difference in Efficiency – AlexNet and MobileNet

This section will provide examples of AlexNet and MobileNet. These 2 are typical examples of CNNs, which makes 2 of them are worth to discuss. AlexNet contains many convolutional layers, maximum pooling layers, and full connect layer. Also, this model uses ReLU as its activation function. But about the MobileNet, this uses depthwise separable convolution, which can significantly reduce the computational cost [2]. Also, this might be a minor factor for that AlexNet runs slowly, LRN [1]. This technology wasn't widely applied in other models.

9.2. Difference in Approaching – EfficientNet and MultiGrain

It's hard to say which algorithm has higher performance, but these 2 algorithms have quite different approaches, which is worth discussing. Firstly, the design philosophy is quite different. EfficientNet is featured by its compound scaling method, where the network's depth, width, and input resolution are scaled in a balanced manner. This was based on the observation that scaling only one dimension of a network (depth, width, or resolution) can lead to suboptimal performance. Through its compound scaling

DOI: 10.54254/2755-2721/46/20241243

approach, EfficientNet aims to strike a balance between accuracy and efficiency, especially for large datasets. Due to its strategy, EfficientNet can be widely used, from mobile devices to cloud servers [5].

The primary strategy behind MultiGrain focuses on learning image features corresponding to multiple resolutions. By training a single model to handle images of varied resolutions, it seeks to improve the model's generalization capability and accuracy. Due to its operations at multiple resolutions, MultiGrain might have slightly higher computational costs while aiming to increase accuracy. MultiGrain might excel in tasks where strong model generalization is required, like when test data resolution varies from training data [6].

In general, EfficientNet emphasizes scaling the network in a balanced manner to improve both efficiency and accuracy, MultiGrain stresses enhancing the model's generalization capability by handling images of different resolutions. Both approaches have their strengths, and the choice between them would depend on the specific application and requirements.

10. Conclusion

This paper has delineated six distinct and representative image classification models, shedding light on their underlying principles and unique attributes. The first is AlexNet, which is one of the earliest networks with significant improvement in accuracy. Then two cost-efficient networks, MobileNet and ShuffleNet, are discussed. Eventually, three complicated networks EfficientNet, MultiGrain, and Pseudo Labels are discussed. A noticeable segment of these models evolves from the foundational constructs of CNNs, while others break the meld with inventive and novel approaches. An evident trend underscores that more recently published algorithms tend to either elevate accuracy levels or significantly optimize computational costs. This progression illustrates the dynamism of the field and offers a gamut of choices tailored to varied application scenarios. As the horizons of current computational paradigms, like CNNs, are continuously pushed, it paves the way for inspiration from alternative domains. Such cross-disciplinary integration holds the promise of unveiling groundbreaking approaches that achieve simultaneous leaps in both accuracy and computational efficiency. The evolving landscape of image recognition, thus, stands testament to the unending quest for excellence, hinting at a future where current boundaries are transcended in pursuit of more holistic and effective solutions.

References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 60(6), 84-90.

[2] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[3] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition, 6848-6856.

[4] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, 1251-1258.

[5] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, 6105-6114.

[6] Berman, M., Jégou, H., Vedaldi, A., Kokkinos, I., & Douze, M. (2019). Multigrain: a unified image embedding for classes and instances. arXiv preprint arXiv:1902.05509.

[7] Radenović, F., Tolias, G., & Chum, O. (2018). Fine-tuning CNN image retrieval with no human annotation. IEEE transactions on pattern analysis and machine intelligence, 41(7), 1655-1668.

[8] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, 815-823.

DOI: 10.54254/2755-2721/46/20241243

[9] Wu, C. Y., Manmatha, R., Smola, A. J., & Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. In Proceedings of the IEEE international conference on computer vision, 2840-2848.
[10] Lee, D. H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, 3(2), 896.
[11] Pham, H., Dai, Z., Xie, Q., & Le, Q. V. (2021). Meta pseudo labels. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 11557-11568.
[12] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning, 1126-1135.
[13] Xie, Q., Dai, Z., Hovy, E., Luong, T., & Le, Q. (2020). Unsupervised data augmentation for consistency training. Advances in neural information processing systems, 33, 6256-6268.
[14] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, 248-255.