HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 3 (2024)

Depth vs. Complexity: A Comparative Study of Neural Network Architectures in Image Classification

Mihir Kulgod

The Shri Ram School Aravali, India

ABSTRACT

There is a growing requirement for image classification algorithms in a plethora of fields, including medical imaging, autonomous vehicles, surveillance, etc. To streamline the process of designing such algorithms to accomplish such a task, one must be aware of the strengths and drawbacks of existing models. This paper investigates the performance of various image classification algorithms, focusing on the dynamic between model depth and complexity, and their effect on accuracy. This study utilizes three datasets - MNIST, Fashion MNIST, and CIFAR10 - to conduct a comprehensive analysis of six distinct image classification architectures. There is a discernible accuracy gradient as one traverses model complexities, from the standard Multilayer Perceptrons (MLPs) to a Visual Transformer (ViT). Training a ViT requires large amounts of computational resources, yet the investment is justified by the remarkable accuracy it achieves. However, it is always more efficient to use a model that fits the scale of the data. No model is the best for every dataset, and data complexity plays a vital role in determining the optimal model architecture for any data.

Introduction

Image classification is a transformative technology with multifarious possible uses in a myriad of industries. From healthcare and autonomous vehicles to security systems, image classification empowers machines by allowing them to comprehend, interpret and categorize visual information. The demand for automated systems continues to surge, heightening the importance of image classification.

Selecting the appropriate algorithm is a critical decision when tasked with an image classification problem. This issue is exacerbated by the fact that no two datasets are entirely the same, and similar performances are not guaranteed across different data. Ensuring the same level of performance is inherently challenging. Despite the factor of uncertainty, I aim to provide a general guide as to which model will outperform others, so long as there are sufficient similarities in the data. In order to do this, I have utilized three different datasets and six different models, in an attempt to encompass a wide range of complexity levels in regard to image classification.

The MNIST, Fashion MNIST, and CIFAR-10 [1] datasets have been chosen in an effort to provide three distinct tiers of difficulty. The order in which they have been listed here also corresponds to their respective complexities fairly accurately. In exploring various models, namely the Multilayer Perceptron (MLP), LeNet, VGG16, ResNet, DenseNet, and Vision Transformer (ViT) [2-6], I have scrutinized their architectures and variants. Each model has been experimented upon multiple times, each time introducing new alterations in architecture and hyperparameters.

This process has been instrumental in discerning which models are ideal for each dataset. There are distinct strengths and weaknesses to each architecture. A clear understanding of this can help expedite the process of finding the optimal model for an image classification task. The effect of architectural sophistication as well as model depth on accuracy has been carefully analyzed.

Proposed Methods

Naturally, the most fundamental decision in image classification tasks is which architecture to use. Considering accuracy exclusively, the Visual Transformer (ViT) architecture is a clear favorite, demonstrating outstanding accuracy in all datasets. This is mostly due to its highly sophisticated design. The original transformer algorithm was designed for Natural Language Processing (NLP), a task which demands innovative approaches and complex models. However, the management of computational resources is as vital as accuracy when developing a product. The ViT is far from the best when tackling simpler, less complex data. Similar performances can be attained more efficiently through use of less sophisticated algorithms. For example, on a dataset as straightforward as MNIST, LeNet performs similarly to the ViT, and is far less computationally demanding. Therefore, there is a delicate relationship between the complexity of the optimal architecture and that of its intended task.

Experimental Setup

Datasets

Three distinct datasets were used in the experiment. They were chosen to encompass varying degrees of complexity in the realm of image classification, allowing for an extensive evaluation of the performance of the various models.

Dataset 1: MNIST

MNIST (Modified National Institute of Standards and Technology) is commonly used as a benchmark in the field of computer vision, comprising a collection of 28x28 grayscale images of handwritten digits (0-9). As such, the task given is for an algorithm to correctly identify which digit a given image is of. See *Figure 1* for some example images from the dataset.

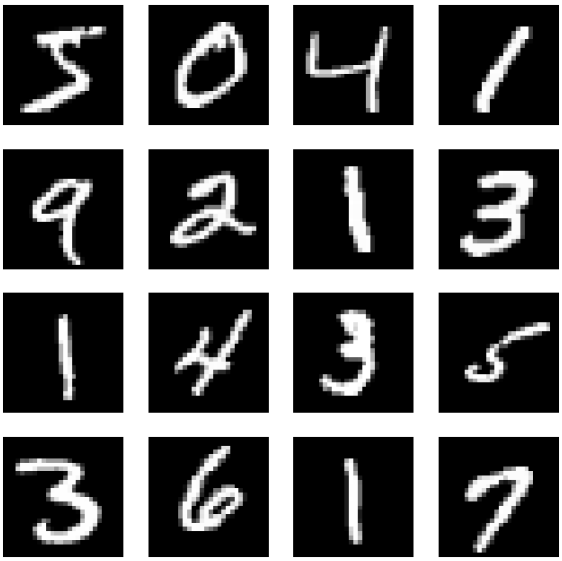

Figure 1. Example images from the MNIST dataset.

Dataset 2: MNIST Fashion

Fashion-MNIST serves as a more challenging alternative to MNIST, consisting of images of various clothing items in the same 28x28 grayscale format. It also maintains the 10-category output for classification. View *Figure 2* for reference.

Figure 2. Example images from the Fashion-MNIST dataset.

Dataset 3: CIFAR-10

The CIFAR-10 dataset consists of 60,000 32x32 pixel color images divided into 10 mutually exclusive classes. It is a subset of the 80 million tiny images dataset. The results make it apparent that this dataset is by far the

most complex of the three used in the experiment. Examples of images and their classes can be viewed in Figure 3.

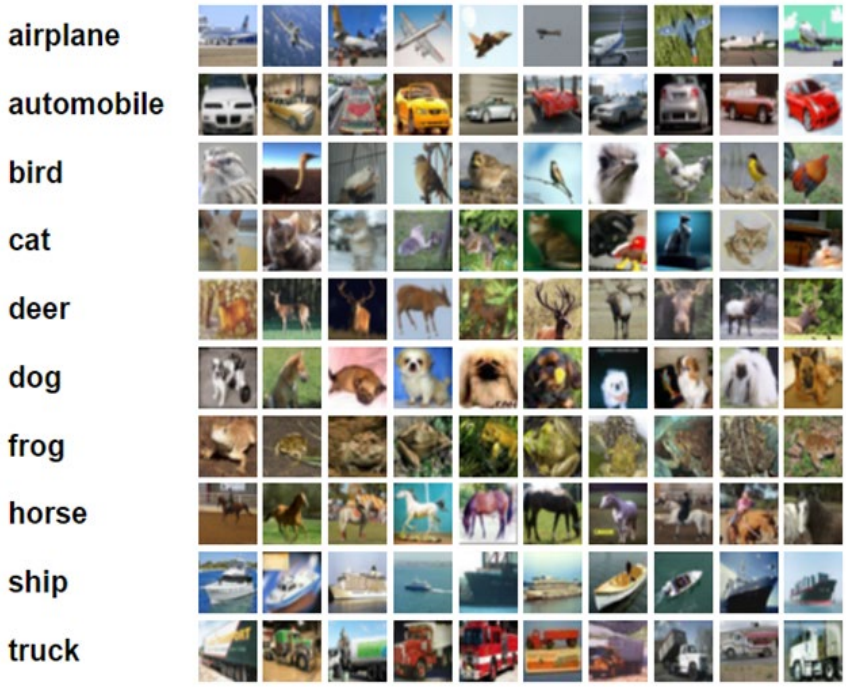

Figure 3. Example images from the CIFAR-10 dataset [1].

Models

There were six image classification algorithms used in the experiment. Up to three variants of each architecture were created, trained, and tested.

Model 1: Multilayer Perceptron (MLP)

An MLP is a type of Artificial Neural Network (ANN) that contains an input layer, one or more hidden layers, and an output layer. The model is loosely inspired by the biology of neuron structures in animal brains. An MLP can serve as a strong foundation for more complex networks. A diagram of the architecture of an MLP with one hidden layer can be seen below in *Figure 4*.

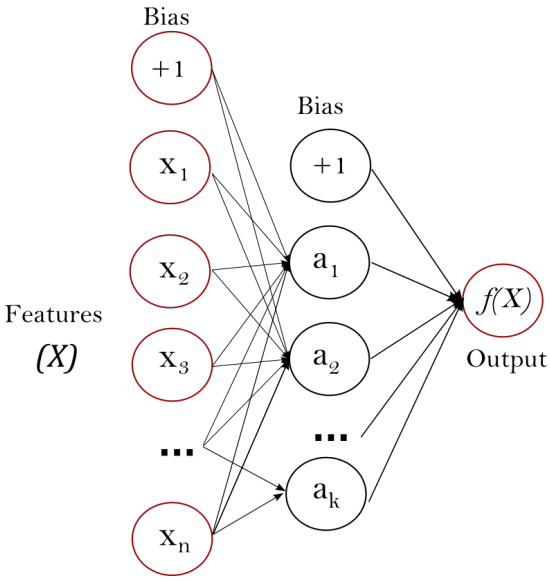

Figure 4. Diagram of MLP architecture with one hidden layer [7].

Model 2: LeNet

LeNet was designed for handwritten and machine-printed character recognition tasks. A CNN is a type of deep learning model designed for analyzing visual data. It can capture spatial relationships in such data, allowing it to accurately detect visual patterns. *Figure 5* details the architecture of a LeNet model.

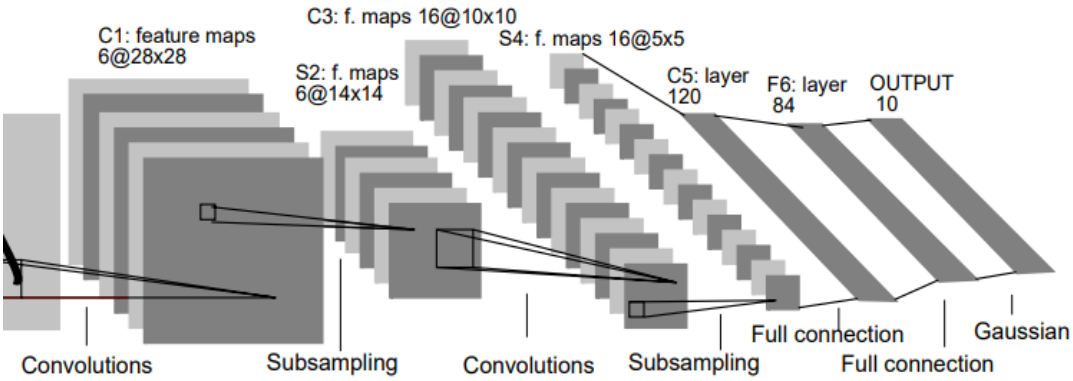

Figure 5. Architecture of LeNet, a CNN, here for digits recognition [2].

Model 3: VGG16

VGG16 is a CNN architecture akin to LeNet, but with increased complexity and depth. It consists of blocks of multiple convolutional layers, each terminating with a max-pooling layer. Due to its remarkable performance, taking the basic concepts of neural networks to the next level, it is a popular choice for image classification tasks in Computer Vision.

Model 4: ResNet

Short for Residual Network, ResNet is a revolutionary neural network developed in 2015 [4]. It introduced the concept of residual learning, facilitating the training of much deeper models. The vanishing gradient problem

is a common one. It occurs when the gap between the algorithm's guesses and the correct answers becomes so small that it stops learning and improving entirely. ResNet alleviated the issue with this concept, allowing one to create networks with far more layers than ever before, and have them continue to improve in accuracy and remain trainable. The architecture of a residual network with 34 parameter layers can be seen in Figure 6.

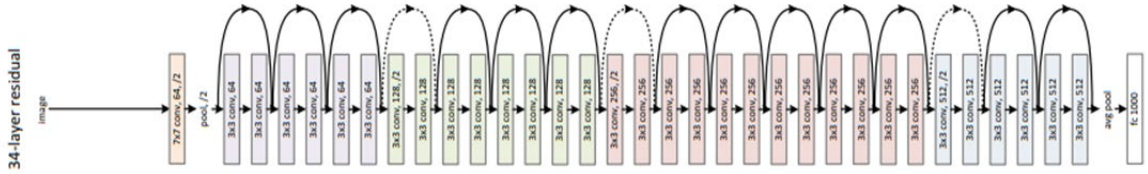

Figure 6. The architecture of a ResNet model [4].

Model 5: DenseNet

DenseNet is an abbreviation of Densely Connected Convolutional Networks. Like ResNet, DenseNet aims to mitigate the vanishing gradient problem. DenseNet also employs a similar method to ResNet, except each layer receives input not only from the previous layer and its input, but from all preceding layers. Figure 7 showcases a 5-layer dense block with the feature-maps being input into the succeeding layers.

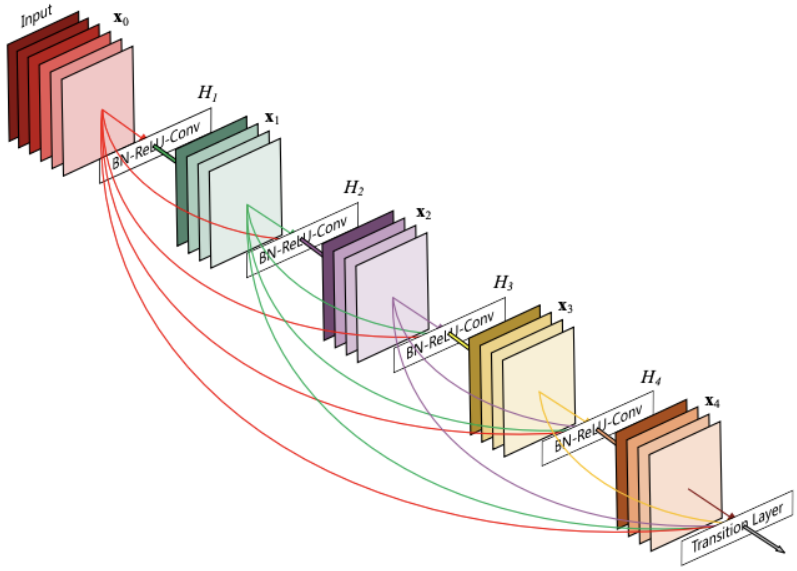

Figure 7. 5-layer dense block with feature-maps being input into the succeeding layers [5].

Model 6: Vision Transformer (ViT)

ViT utilizes a transformer designed for computer vision. A transformer operates on a multi-head attention system. Transformer models can learn context by tracking relationships in data arranged sequentially. This feature is vital for tasks such as Natural Language Processing (NLP). A ViT breaks an input image down into several patches, which it tokenizes - meaning it converts each patch into a string of numbers - before processing them through the transformer mechanism.

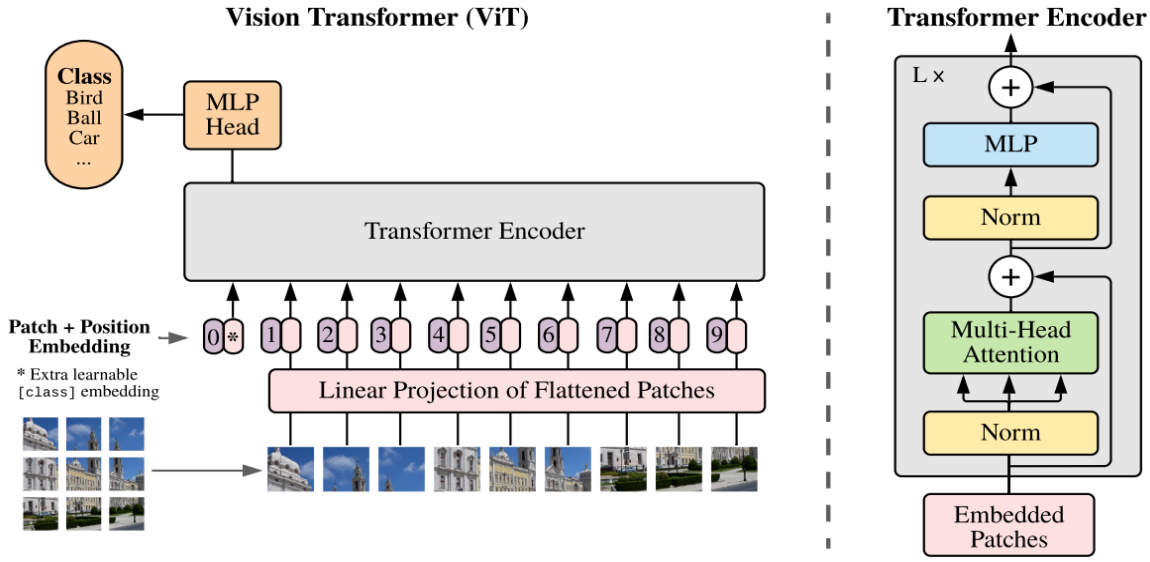

Figure 8. Structure of a ViT [6]. As shown on the left, the image is converted to patches which are then flattened into linear projections, which have the positions of the original patches embedded. The structure of the transformer encoder which they are processed by is shown on the right.

Results and Discussion

Table 1. List of Observations (The Params. row mentions the number of trainable parameters of each model)

	MNIST			**Fashion-MNIST**			**CIFAR10**		
MLP	3L	4L	5L	3L	4L	5L	3L	4L	5L
Params.	243K	575K	2M	243K	575K	2M	828K	2M	4M
Accuracy	0.9723	0.9738	0.9712	0.8662	0.8625	0.8538	0.4375	0.4526	0.4802
LeNet	C[6], C[16], 2D	C[6], C[32], 1D	C[6], C[32], 2D	C[6], C[16], 2D	C[6], C[32], 1D	C[6], C[32], 2D	C[6], C[16], 2D	C[6], C[32], 1D	C[6], C[32], 2D
Params.	44K	49K	78K	44K	49K	78K	62K	73K	112K
Accuracy	0.9818	0.9853	0.9831	0.8664	0.8683	0.8681	0.5206	0.5822	0.5769
VGG16	2D1D	2B2D	3B2D	2D1D	2B2D	3B2D	2D1D	2B2D	3B2D
Params.	11M	28M	27M	11M	28M	27M	34M	51M	35M
Accuracy	0.9917	0.9866	0.9865	0.8909	0.887	0.8749	0.6923	0.6798	0.6616
ResNet	3C	4C	5C	3C	4C	5C	3C	4C	5C
Params.	2M	8M	23M	2M	8M	23M	2M	8M	23M
Accuracy	0.9916	0.9894	0.9287	0.8977	0.8803	0.8176	0.6698	0.5581	0.3303

DenseNet	-	CDT	CDT, DT	-	CDT	CDT, DT	-	CDT	CDT, DT
Params.	-	378K	1M	-	378K	1M	-	384K	1M
Accuracy	-	0.9916	0.9876	-	0.869	0.8829	-	0.55	0.6796
ViT	7x7; 5 Transf.	7x7; 10 Transf.	10x10; 10 Transf.	7x7; 5 Transf.	7x7; 10 Transf.	10x10; 10 Transf.	7x7; 5 Transf.	7x7; 10 Transf.	10x10; 10 Transf.
Params.	18M	18M	11M	18M	18M	11M	18M	18M	11M
Accuracy	0.9867	0.9864	0.9936	0.9141	0.9143	0.9188	0.8212	0.8252	0.8103

In the leftmost column of Table 1, the architecture name has been provided. Each architecture was modified to create three total variants, which are described in the same row. For the MLP, xL has been used to denote the architecture used. Here, x is the number of hidden layers utilized, and L is an abbreviation for 'Layer'. Under LeNet, C has been used to denote a convolutional layer, with the number of filters in square brackets []. Additionally, the Dense layers have been denoted by a D prefixed with the number used. Under VGG16, xB has been used to denote the number of blocks used, where each block contains multiple convolutional layers, capped by a max-pooling layer. The same notation as LeNet has been used for the Dense layers again here. Due to the complex nature of the DenseNet algorithm, only two variants of it were used. Increasing the model's depth led to a rapid decline in its performance, and a simpler variant cannot be created due to the inherent structure of the DenseNet algorithm. For ResNet, xC has been used to denote the number of Residual Blocks involved. The notation used in the DenseNet column is as follows: C, D, and T represent convolutional layers, dense layers, and transition layers respectively. In ViT, the patch size has been mentioned first, followed by the number of transformer layers. Additionally, below each model name, the number of trainable parameters and its test accuracy have been mentioned.

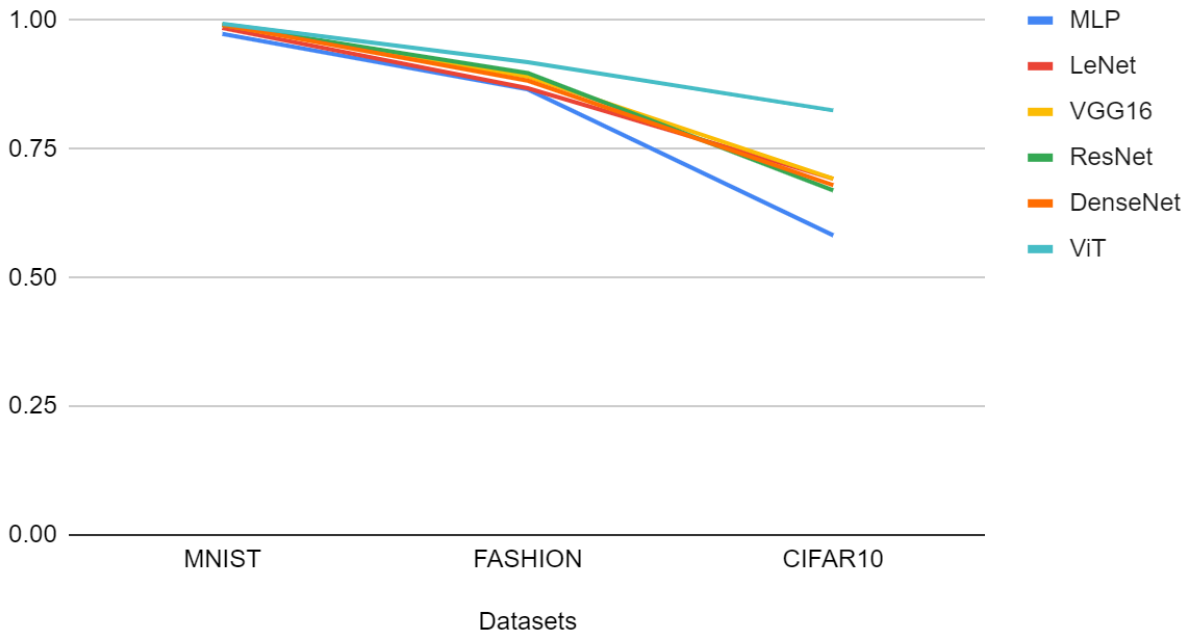

Figure 9. Best test accuracies of each architecture across datasets. Dataset is given on the X-axis, while test accuracy on a scale of 0 to 1 is given on the Y-axis, where 1.00 refers to 100% accuracy.

The chart above provides a clearer, visual representation of the experimental results. It is immediately apparent that there is a clear decreasing accuracy gradient from the MNIST dataset to the CIFAR-10 dataset. This outcome is anticipated, as it reflects the complexity of the datasets used in the experiment. Such results are in line with the performances of similar algorithms on the same data. The accuracy of all the models is near-perfect for MNIST, the least complex dataset, and the gap between model accuracy increases with the data complexity. The ViT emerges with the greatest accuracy in all three datasets. LeNet, VGG16, ResNet, and DenseNet exhibit comparable performance, despite the perception of the latter two employing more sophisticated algorithms. This may be due to the datasets being too simple to leverage their advanced capabilities. The accuracy of the MLP rapidly declines due to its inherently limited capabilities. It lacks the ability to capture spatial relationships and intricate patterns and is fairly susceptible to overfitting.

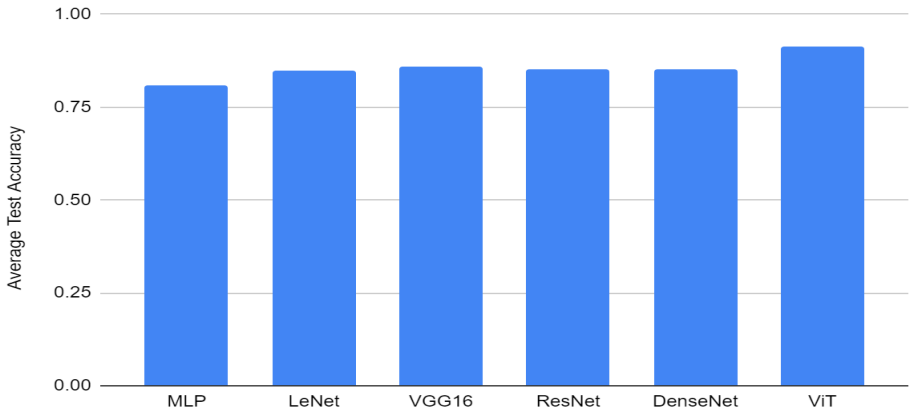

Figure 10. Average test accuracy of each architecture. Architectures on the X-axis, test accuracy on the Y-axis ranging from 0 to 1, where 1.00 represents 100% accuracy.

This figure displays the average test accuracy of each architecture in the form of a bar graph. The average was calculated using the arithmetic mean of the best test accuracies from each dataset. Diving into the distinctions between each model, there is a general increase in accuracy scores from the left of the figure to the right. This aligns with the increasing complexity of the models; however, an intriguing pattern emerges when one observes the accuracies of each variant of each model.

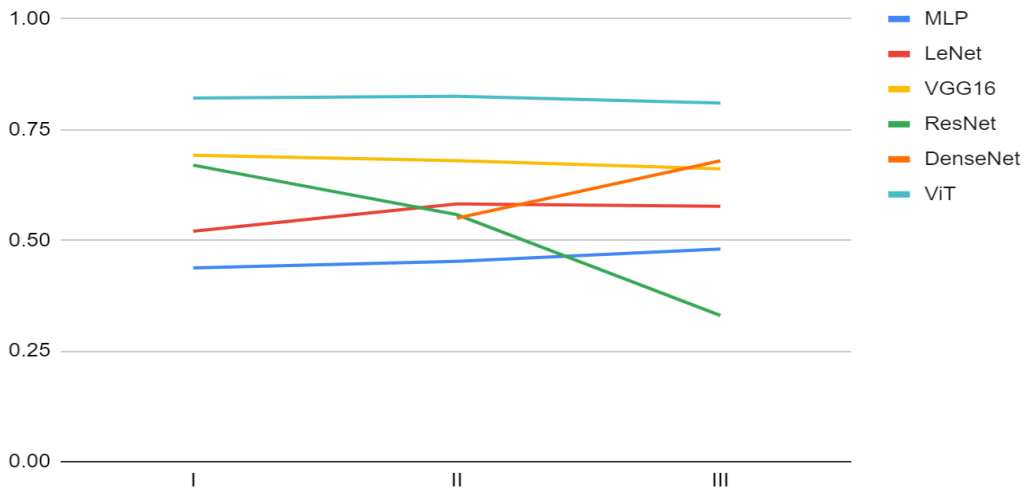

Figure 11. Effect of deepening models on accuracy, where model III is the deepest version of the model.

This figure showcases how deepening a model affects its accuracy in this setup. I, II, and III correspond to the models in Table 1, where III is the deepest model. CIFAR10's complexity makes it the best candidate for distinguishing between various models. Consequently, only the CIFAR10 test accuracies were used here.

The traditional CNNs, LeNet and VGG16, run the risk of overfitting, and of vanishing gradients. Effectively, this means that beyond a point, deepening a model will be detrimental to its performance. How far one can go without passing this point is entirely dependent on the complexity of the data in question. This issue is seen to affect ResNet the most, as it is far better suited to more complex data. However, as one can see in the chart, this problem does not affect DenseNet and ViT. DenseNet utilizes its dense connectivity feature to cross the hurdle. In ViT, the self-attention mechanisms also allow it to do the same. The distinctions among these architectures clearly reflect their abilities.

In general, the ViT seems to shine, demonstrating incredible accuracy across all the datasets. However, there are other factors to take into consideration. Such a complex architecture requires more computational resources, leading to higher financial costs.

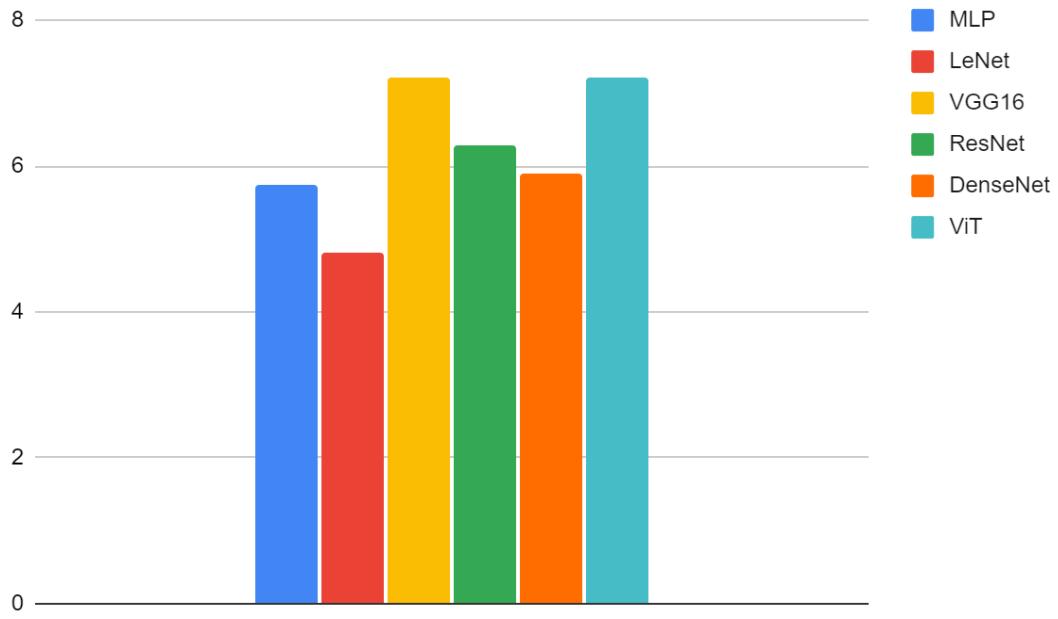

Figure 12. Accuracy-Parameter Ratio displayed on a logarithmic scale.

This figure provides a measure of how well a model performed with respect to the number of trainable parameters it contained. The arithmetic mean of the ratios between the best test accuracies of a model for a dataset and the corresponding number of trainable parameters was taken. The figure displays the negative logarithm of this ratio. Essentially, the lower the final value, the more efficient the model was. LeNet provided the greatest performance for the least resources, while the ViT was the most resource intensive option, with VGG16 in a close second. This clearly demonstrates how straightforward, simple architectures are far more efficient for less complex data, and the correlation between accuracy and the amount of resources required.

Conclusion

In summary, there is a clear margin between traditional image classification algorithms and more sophisticated approaches which each have their own unique features. Therefore, one of the first steps one can take in choosing an image classification algorithm should be that of picking one out of the two categories. This decision should be firmly based on what data the algorithm aims to analyze. For example, for a dataset as simple as MNIST, LeNet or another traditional CNN is far more appropriate for the job than a DenseNet, for example. This is due to the sheer difference in suitability and computational efficiency between the two. A more sophisticated architecture would attain marginally better accuracy than a simpler one, despite requiring computational resources that may be orders of magnitude above that of the latter. Nonetheless, the reverse logic applies; a simpler algorithm would just not be enough for more complex data.

Overall, the ViT appears to be the best. It attained outstanding accuracy in all datasets and had a reasonable number of trainable parameters. However, it is crucial to use models that match the complexity and scale of the data, as mentioned before. No two datasets are the same and should not be treated so.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

1. Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

2. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

3. Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv (Cornell University)*. https://arxiv.org/pdf/1409.1556

4. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1512.03385

5. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1608.06993

6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv (Cornell University)*. https://arxiv.org/pdf/2010.11929

7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M.,

Perrot, M., & Duchesnay, É. (2011). SciKit-Learn: Machine Learning in Python. *HAL (Le Centre Pour La Communication Scientifique Directe)*. https://hal.inria.fr/hal-00650905