Comparing VGG19, ResNet34, and GoogleNet for Image Classification: A Performance Evaluation

Mutiah Apampa
220520511 | ec22181@qmul.ac.uk

Abstract — This paper presents a comparative study of three popular convolutional neural network architectures, namely VGG19 [2], ResNet34 [3], and GoogleNet [4], for image classification tasks. The performance of these models is evaluated on the widely used MNIST and CIFAR-10 datasets, and their accuracy and computational efficiency are reported. The results show that all three models achieve high accuracy on both datasets, with overall better performance on the MNIST dataset. The findings of this study provide insights into the effectiveness of deeper CNN models for image classification tasks and could guide the choice of suitable models for specific tasks.

1. INTRODUCTION

Image classification is a fundamental task in computer vision, with numerous real-world applications such as object recognition, face recognition, and autonomous driving [1]. Convolutional neural networks (CNNs) have shown remarkable performance in image classification, and various CNN architectures have been proposed in recent years. Among these, VGG, ResNet, and GoogleNet are well-known models that have achieved state-of-the-art results on various benchmark datasets [2, 3, 4].

The MNIST dataset consists of grayscale images of handwritten digits [13], while CIFAR-10 is a more complex dataset that includes color images of various objects [14]. The performance of the models in terms of training and validation accuracy is compared, and their ability to generalize to unseen examples is explored. Additionally, the impact of the models' complexity and hyperparameters on their performance is examined.

In this study, the effectiveness of three different CNN architectures, namely VGG19, ResNet34, and GoogleNet, is evaluated for image classification on the MNIST and CIFAR-10 datasets. Their performances in terms of accuracy, efficiency, and complexity are compared to determine which architecture performs best on these datasets.

The aim of this study is to provide useful insights into the suitability of these CNN architectures for image classification tasks with different levels of complexity.

2. RELATED WORK

The field of image classification has witnessed significant progress in recent years, thanks to the development of convolutional neural networks (CNNs) which was first introduced by LeCun et al. [5] in 1998. The Le-Net5 CNN model consists of two convolutional layers, two subsampling layers, and three fully connected layers, and was designed for handwritten digit recognition [5]. Since then, several other CNN architectures have been proposed, each with its own unique characteristics [2, 3, 4].

One of the early and most influential works in CNN was the AlexNet architecture proposed by Krizhevsky et al. [7], which achieved state-of-the-art performance on the ImageNet dataset with a top-5 error rate of 15.3%. This work demonstrated the potential of deep learning for image classification and paved the way for the

development of deeper CNN architectures. The AlexNet architecture consists of five convolutional layers and three fully connected layers [7].

In 2014, the VGG architecture was introduced by Simonyan et al. [2], which showed improved accuracy on the ImageNet dataset by increasing the depth of the network. Later, He et al. proposed the ResNet architecture [3], which utilized residual connections to enable the training of even deeper networks. This architecture achieved significant improvements in accuracy and became the state-of-the-art for image classification on the ImageNet dataset.

Another widely used CNN architecture is GoogleNet, also known as Inception v1, which was proposed by Szegedy et al. [4]. GoogleNet achieved state-of-the-art performance on the ImageNet dataset while using fewer parameters compared to previous architectures. Subsequent versions of the Inception architecture were proposed by the same group, further improving accuracy and efficiency [8].

In recent years, various modifications and improvements to these architectures have been proposed, such as ResNeXt [9], DenseNet [10], and MobileNet [11]. These architectures aim to improve accuracy and efficiency by incorporating different design choices and techniques, such as multi-scale feature extraction, skip connections, and depthwise separable convolutions[9, 10, 11].

Overall, VGG, ResNet, and GoogleNet are widely used and well-known CNN architectures for image classification, and have achieved state-of-the-art results on various benchmark datasets. Several studies have compared the performance of these models on different datasets and found that ResNet achieves the highest accuracy [12] and is relatively faster to train compared to VGG and GoogleNet. The choice of CNN architecture depends on the specific application and the trade-off between efficiency and complexity.

3. METHOD / MODEL DESCRIPTION

In this study, the performance of three CNN architectures for image classification is compared: VGG-19, ResNet-34, and GoogleNet. Specifically evaluating the models' accuracy, efficiency, and complexity on two widely used datasets: MNIST and CIFAR-10.

3.1 Model Architecture

3.11 VGG19

The VGG19 is an extended version of the VGG16 architecture, comprising 19 layers with three additional convolutional layers [2]. The model employs 16 convolutional layers with ReLU activation function, 5 pooling layers, and 3 fully connected layers. It uses a small filter size of 3x3 for the convolutional layers and max pooling for the pooling layers. The VGG19 achieved state-of-the-art performance on the ImageNet dataset in 2014 [2]. However, the model has a large number of parameters, making it computationally expensive and challenging to train.

3.12 ResNet34

ResNet34 is a CNN architecture consisting of 34 layers, proposed by He et al. in 2016 [3]. The model is an improvement over previous CNNs in addressing the vanishing gradients problem in very deep networks by using residual blocks to allow for the propagation of gradients through shortcuts, even in very deep networks. The architecture includes 33 convolutional layers with varying filter sizes and one fully connected layer [3]. The model uses batch normalization and ReLU activation function after each convolutional layer. The

residual blocks allow for efficient learning of the image features and enable the model to achieve high accuracy on image classification tasks. ResNet34 has fewer parameters than VGG models and is easier to train, but its more complex architecture can make it less efficient.

3.13 GoogleNet

GoogleNet is a CNN architecture also known as Inception-v1, proposed by Szegedy et al. in 2015 [4]. The model includes a unique module called an Inception module, which consists of parallel convolutions with different kernel sizes and pooling operations, allowing the model to capture multi-scale features efficiently [4]. The architecture consists of 22 layers, including 9 Inception modules and 1 average pooling layer at the end. The final layer is a fully connected layer with 1000 units, corresponding to the number of classes in the ImageNet dataset. The model uses a combination of ReLU and softmax activation functions [4].The model has fewer parameters than VGG models and is relatively efficient to train.

3.2 Training Method

The CNN models were implemented using TensorFlow and trained on a workstation equipped with an NVIDIA Quadro RTX 6000 GPU. The training on ResNet34 and GoogleNet was performed using the Adam optimizer and sparse_categorical_crossentropy loss, while VGG19's training was performed using the Stochastic Gradient Descent (SGD) optimizer and sparse_categorical_crossentropy loss. This decision was made after testing, as it was found that VGG19 performed poorly when trained with the Adam optimizer. The batch size was set to 64, and the models were trained for 40 epochs to accommodate memory constraints.

4. EXPERIMENT

In this section, the experiments performed to compare the performance of VGG, ResNet, and GoogleNet models on the MNIST and CIFAR-10 datasets are described.

4.1. Datasets

4.1.1 MNIST:

The MNIST dataset comprises 70,000 handwritten digit images, including 60,000 for training and 10,000 for testing [13]. To test the model's accuracy, the training dataset was split into training and validation sets. The models were trained on the training set and evaluated on the validation set. After tuning the models' hyperparameters, they were evaluated on the test set. Our study includes the training and validation accuracies and training time of all three models.

However, it should be noted that the original CNN models were designed to receive images with dimensions of 224 x 224 x 3, whereas the MNIST images are only 28 x 28. To adapt the input size to the models, the third axis of the images was expanded and repeated three times, resulting in an image size of 28 x 28 x 3. Then, these images were resized to 224 x 224 x 3 at the first layer of the models. This image-resizing process allowed to use of the same CNN architectures for both the MNIST and CIFAR10 datasets.

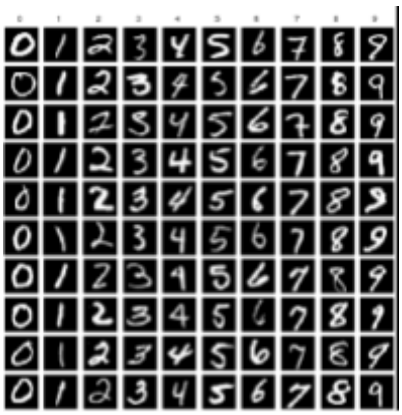


Figure 1: MNIST dataset

4.1.2 CIFAR-10:

The CIFAR-10 dataset consists of 50,000 training images and 10,000 testing images of 10 different classes of objects, such as airplanes, cars, and cats [14]. To test the model's accuracy, the training dataset was split into training and validation sets. The models were trained on the training set and evaluated on the validation set. After tuning the models' hyperparameters, they were evaluated on the test set. Our study includes the training and validation accuracies and training time of all three models.

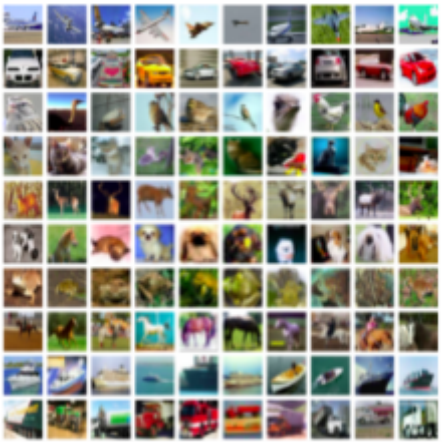


Figure 2: CIFAR10 dataset

4.2 Training Process

During the training process, cross-entropy loss function and Stochastic Gradient Descent (SGD) optimizer were used for VGG19, and Adam optimizer for both ResNet34 and GoogleNet. Each model was trained for 40 epochs with a batch size of 64 for both MNIST and CIFAR-10 datasets. It's worth noting that VGG19 had an initial accuracy of about 10%, and wasn't converging with the Adam optimizer, which led to the decision of using the SGD optimizer for this model.

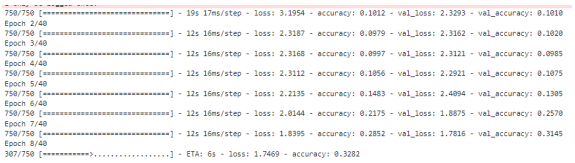


Figure 3: Runtime screenshot of the VGG19 model training with CIFAR10 dataset

4.3 Testing Process

During the testing process, the performance of the trained models was evaluated using the accuracy metric on the validation and test datasets. A separate validation set was used to tune the hyperparameters of the models, and the test set was used to evaluate their final performance. The models were evaluated on both MNIST and CIFAR-10 datasets.

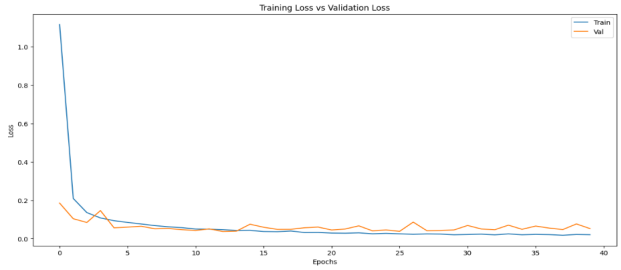


Figure 4: Training and Validation Loss vs Number of Epochs Plot for GoogleNet MNIST training

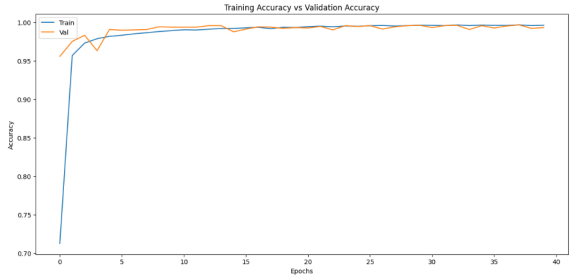


Figure 5: Training and Validation Accuracy vs Number of Epochs Plot for GoogleNet training with MNIST dataset

Figure 6: Training and Validation Loss vs Number of Epochs Plot for ResNet34 training with CIFAR10 dataset

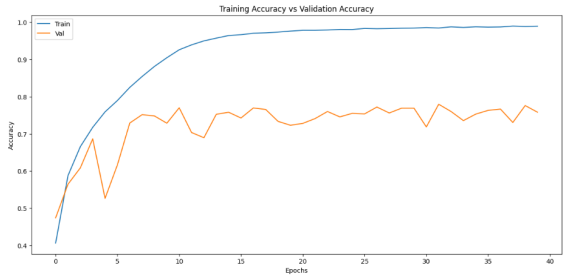


Figure 7: Training and Validation Accuracy vs Number of Epochs Plot for ResNet34 training with MNIST dataset

4.4 Results.

The training and validation accuracies, as well as training time, were recorded for each model on each dataset.

On the MNIST dataset, all models achieved high accuracy, with VGG-19 achieving the highest training accuracy of 99.98% and ResNet-34 achieving the highest validation accuracy of 99.60%. The training time for ResNet-34 was the shortest, at only 9 minutes, while GoogleNet required the longest training time, at 28 minutes.

Dataset	Model	Training accuracy(%)	Validation Accuracy (%)	Training time (minutes)
MNIST	VGG-19	99.98	99.50	21
	ResNet-34	99.87	99.60	9
	GoogleNet	99.60	99.30	28
CIFAR10	VGG-19	95.54	73.55	17
	ResNet-34	98.94	75.80	9
	GoogleNet	87.82	79.05	21

Table 1: Model performances on MNIST & CIFAR10 dataset

The CIFAR10 dataset had generally lower accuracy than MNIST. VGG-19 achieved the highest training accuracy of 95.54%, while ResNet-34 achieved the highest validation accuracy of 75.80%. GoogleNet had the lowest accuracy on this dataset, with a training accuracy of 87.82% and a validation accuracy of 79.05%. The training time for ResNet-34 was once again the shortest, at 9 minutes.

Overall, the results show that:

- there is poor performance on the validation dataset for CIFAR-10 but a good performance for MNIST using the same hyperparameters which may indicate that the model is overfitting on the CIFAR-10 dataset.
- ResNet-34 performs well on both datasets, achieving high accuracy in a shorter training time compared to the other models. VGG-19 also achieved high accuracy but required longer training times.
- GoogleNet had the lowest accuracy on both datasets even after hyperparameter tuning, indicating that it may need further evaluations impossible on this paper due to memory constraints.

4.5 Discussion & Further Evaluation

Based on the results obtained, several conclusions can be drawn about the performance of the different deep CNN models for image classification tasks on the MNIST and CIFAR10 datasets.

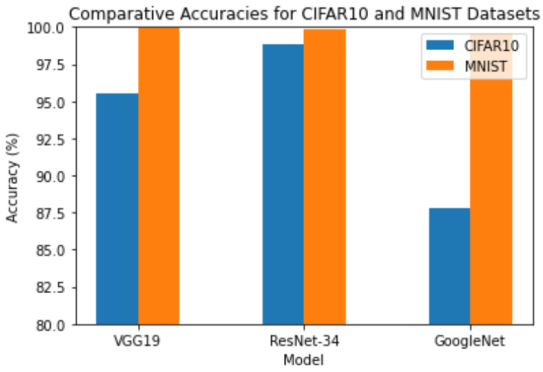
Firstly, it is evident that all models achieved exceptionally high training accuracy on both datasets. Specifically, VGG-19 and ResNet-34 achieved nearly 100% training accuracy on MNIST, while ResNet-34 achieved almost 99% on CIFAR-10. On the other hand, GoogleNet demonstrated relatively lower training accuracy, achieving approximately 99% and 88% on MNIST and CIFAR-10, respectively.

However, there was a significant difference in the models' performance on validation accuracy on the two datasets. For MNIST, all models achieved very high

validation accuracy, indicating that the models could generalize well on this dataset and accurately classify unseen examples. In contrast, the models exhibited poor performance on the CIFAR-10 dataset, suggesting that the models were overfitting on the training data and were unable to generalize well on unseen examples.

Due to memory constraints, we recommend the below to further investigate this issue:

- Conduct experiments with different regularization techniques such as dropout and weight decay, and explore different data augmentation strategies such as random cropping and flipping.
- Additionally, investigating the performance of other deep CNN models on these datasets to see if they exhibit similar trends.



5. CONCLUSION

In a nutshell, the effectiveness of deeper CNN models for image classification was evaluated using the MNIST and CIFAR-10 datasets. The results showed that all three models, ResNet34, VGG19, and GoogleNet, performed well on the MNIST dataset with an accuracy of more than 99%. However, on the CIFAR-10 dataset, ResNet34 and VGG19 outperformed GoogleNet in terms of accuracy on both training and validation sets. It can be concluded that ResNet34 generally has a higher efficiency than both VGG19 and GoogleNet.

When considering model complexity, the VGG19 model is the simplest but most parameter-heavy architecture, while ResNet34 is more complex than VGG19 but has fewer parameters. GoogleNet is the most complex architecture of the three but has a relatively small number of trainable parameters.

The CIFAR10 dataset is complex and diverse, with more varied and intricate images, which may make it more challenging to generalize and could have been the cause of the poor performance on the validation set.

The findings of this study suggest that deeper CNN models can improve accuracy in image classification tasks. However, a deeper model does not always guarantee better performance. Instead, it is essential to balance model complexity and performance. Overall, based on the results of this study, all three models are effective for image classification on MNIST, but their performance on CIFAR-10 is less impressive.

To improve the performance of these models on CIFAR-10, it may be necessary to adjust the hyperparameters or consider alternative models that are better suited to handling the challenges of this dataset. Nonetheless, the results of this study contribute to understanding the strengths and limitations of these deeper CNN models for image classification tasks.

REFERENCES

[1] S. Javaid, "Clarifying Image Recognition Vs. Classification in 2023," AIMultiple, 2023. [Online]. Available: https://research.aimultiple.com/image-recognition-vs-classification/. [Accessed: May 10, 2023].

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on computer vision and pattern recognition, 2016, pp. 770–778.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[6] M. T. Hossain, A. Al Mahmud and A. A. Al Mamun, "A comparative study of deep learning models for text classification," Journal of Big Data, vol. 8, no. 1, pp. 1-22, 2021. doi: 10.1186/s40537-021-00444-8.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet

classification with deep convolutional neural networks," in

NIPS, 2012, pp. 1097-1105.

[8] K. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 4278-4284.

[9] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated Residual Transformations for Deep Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 5987-5995, doi: 10.1109/CVPR.2017.634.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700-4708, doi: 10.1109/CVPR.2017.243.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[12] S. Sharma and K. Guleria, "Deep Learning Models for Image Classification: Comparison and Applications," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1733-1738, doi: 10.1109/ICACITE53722.2022.9823516.

[13] Y. LeCun, C. Cortes and C. J. C. Burges, "The MNIST database of handwritten digits," in IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141-142, Nov. 2012, doi: 10.1109/MSP.2012.2205597.

[14] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009, [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. [Accessed: May 11, 2023].