# Analysis and Comparison of VGG16 and ResNet-50 for Image Recognition

Nemekhbayar Nomin[1],
[1]Computer Science Engineering Dep. of PNU.

## 1 Introduction

Deep convolutional neural networks (CNNs) have achieved significant breakthroughs in large-scale image recognition tasks. VGG16 is known for its simplicity and uniform layer design, while ResNet-50 introduces residual learning to tackle the optimization challenges associated with deeper networks. This report presents an in-depth analysis and comparison of two prominent CNN architectures. The objective of this analysis is to evaluate and contrast the performance, efficiency, and architectural approaches of these two models by assessing the performance of these models on image classification tasks.

## 2 Server

The practice in this report were conducted using the Google Colab environment, which provides GPU acceleration for efficient training of deep learning models. Using GPU resources in Colab allowed us to train complex architectures like VGG16 and ResNet-50 with improved speed and performance. Transferring model weights and training data to the GPU ensures that all computations are optimized, significantly reducing training time compared to CPU-only processing.

## 3 Dataset

For this report, a subset of the CIFAR-10 dataset was used in the both model, focusing on the first three categories: plane, car, and bird which is visualized in the Figure 1. The training set includes 15,000 images, while the test set contains 3,000 images. Data augmentation techniques, such as random cropping and horizontal flipping, were applied to enhance the training data and improve model generalization.

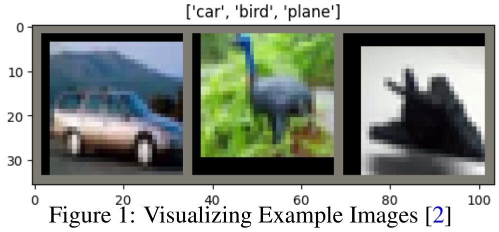

Figure 1: Visualizing Example Images [2]

## 4 VGG16

The VGG16 model implemented in this report follows the Type-D configuration, adapted for a 3-way classification task on CIFAR-10 categories: plane, car, and bird. This architecture utilizes a series of convolutional layers with 3x3 filters, followed by batch normalization and ReLU activation. Max-pooling layers are interspersed to progressively reduce spatial dimensions which are visualized in two sub figures.

For simplicity, dropout is omitted, and batch normalization is added after each convolution to stabilize training. The final fully connected layer maps the extracted features to three output classes. An adaptive pooling layer ensures a consistent output size before classification.

Training was conducted using cross-entropy loss, which measures the difference between the predicted logits and the true labels. The model was optimized with the SGD optimizer, set with a learning rate of 0.01, momentum of 0.9, and weight decay of $5 \times 10^{-4}$ over 20 epochs. The classification accuracy, defined as the percentage of correct predictions, served as the primary evaluation metric to monitor model performance.

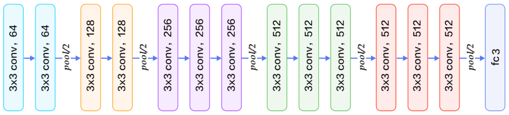

Figure 2: VGG Network Figure [3]

| VGG-16 with (mini-)CIFAR-10 config. | | |
|---|---|---|
| layer name | output size | |
| conv1_x | 32x32 | ( 3x3, 64 ) x2 |
| pool1 | 16x16 | maxpool |
| conv2_x | 16x16 | ( 3x3, 128 ) x2 |
| pool2 | 8x8 | maxpool |
| conv3_x | 8x8 | ( 3x3, 256 ) x3 |
| pool3 | 4x4 | maxpool |
| conv4_x | 4x4 | ( 3x3,512 ) x3 |
| pool4 | 2x2 | maxpool |
| conv5_x | 2x2 | ( 3x3, 512 ) x3 |
| pool5 | 1x1 | maxpool |
| | 1x1 | 3-d fc |

Figure 3: VGG Network Configuration [3]

## 5 ResNet-50

The ResNet-50 model implemented in this report follows a deep architecture consisting of 50 layers, structured with bottleneck blocks. Each bottleneck block contains three convolutional layers (1x1, 3x3, 1x1), with batch normalization and ReLU activation applied after each convolution. This configuration enables the model to achieve substantial depth while maintaining computational efficiency.

For this implementation, the initial convolution layer has been simplified to fit the CIFAR-10 dataset, and max-pooling is replaced by strided convolutions for down-sampling. A 1x1 convolution with a stride of 2 is applied to the residual path when down-sampling, which helps adjust the dimensions without adding computational overhead.

The model was trained using cross-entropy loss, which calculates the discrepancy between predicted logits and true labels. To optimize the model, the Stochastic Gradient Descent (SGD) optimizer was used with a learning rate of 0.01, momentum of 0.9, and weight decay of $5 \times 10^{-4}$. The model was trained for 15 epochs without learning rate scheduling.

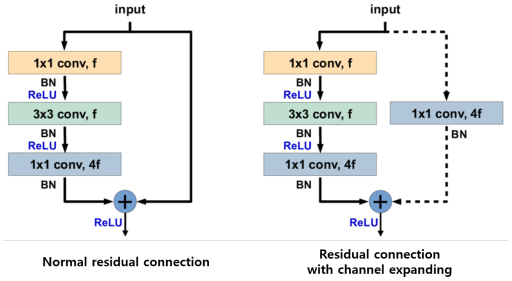

Figure 4: ResNet-50 Network Connection [4]

| ResNet-50 with (mini-)CIFAR-10 config. | | | |
|---|---|---|---|
| layer name | output size | | |
| conv1 | 32x32 | 3x3, 64, stride 1 | |
| conv2_x | 32x32 | 1x1, 64<br>3x3, 64<br>1x1, 256 | x3 |
| conv3_x | 16x16 | 1x1,128<br>3x3, 128<br>1x1, 512 | x4 |
| conv4_x | 8x8 | 1x1,256<br>3x3,256<br>1x1, 1024 | x6 |
| conv5_x | 4x4 | 1x1,512<br>3x3,512<br>1x1, 2048 | x3 |
| | 1x1 | average pool, 3-d fc | |

Figure 5: ResNet-50 Network Configuration [4]

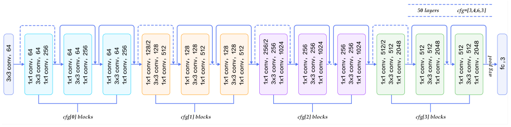

Figure 6: ResNet-50 Network Figure [1]

## 6 Comparison and Evaluation Analysis of the models

Both models were evaluated on classification accuracy and loss during training and testing phases. For VGG16, training concluded after 20 epochs with a final training accuracy of 83.62

Testing results further demonstrate the advantages and limitations of each model. VGG16 achieved a testing accuracy of 83.37

In summary, while both models perform well, ResNet-50 demonstrates superior accuracy but also reflects the potential trade-offs associated with deeper architectures. This comparison underscores the importance of balancing network depth and complexity to achieve robust generalization in image recognition tasks.

Table 1: Training and Testing Accuracy and Loss of VGG16 and ResNet-50 Models

| | **VGG** | **ResNet** |
|---|---|---|
| Train Accuracy (%) | 83.62 | 96.20 |
| Train Loss | 0.422 | 0.1044 |
| Test Accuracy (%) | 83.37 | 89.67 |
| Test Loss | 0.4303 | 0.528 |

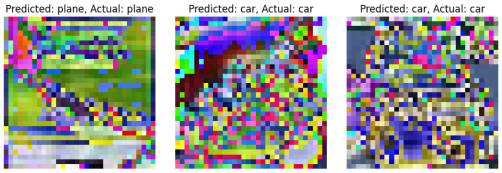
Predicted: plane, Actual: plane
Predicted: car, Actual: car
Predicted: car, Actual: car

Figure 7: Prediction Example [2]

## 7 Conclusion

This practice provides an in-depth comparison of VGG16 and ResNet-50 on a subset of CIFAR-10 image classification tasks. While VGG16 showcases simplicity and effective feature extraction through its deep architecture, it achieved an accuracy of 83.62

The results underscore that ResNet-50 not only outperforms VGG16 in terms of accuracy but also achieves better optimization with fewer epochs, demonstrating superior efficiency. These findings reinforce the impact of residual networks on advancing deep learning models for image recognition tasks, offering a robust approach for future architectures aimed at achieving both depth and efficiency.

[1] CodeProject. Deep learning using python + keras: Chapter review. https://www.codeproject.com/Articles/1248963/Deep-Learning-using-Python-plus-Keras-Chapter-Re. Accessed: 2024-10-08.

[2] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. https://www.cs.toronto.edu/~kriz/cifar.html.

[3] Quora. What is the vgg neural network? https://www.quora.com/What-is-the-VGG-neural-network. Accessed: 2024-10-08.

[4] Eric A. Rezende, Mateus R. Silva, and Rafael J. de Rezende. Signal processing: Image communication. *Signal Processing: Image Communication*, 65:66–80, 2018. doi: 10.1016/j.image.2018.02.008.