Exploring Advanced Deep Learning Techniques for Image Classification on CIFAR-10

Yi Li
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yil6@cs.cmu.edu

Abstract

This paper aims to present novel and innovative methods for achieving high accuracy and efficiency in deep learning-based image classification. We target the CIFAR-10 dataset, which is a commonly used benchmark in the field, consisting of 10 distinct classes of 60,000 color images . To do so, we take a standard conventional neural network as the baseline and introduce novel architectures to surpass this baseline. Our architectural designs involve multi-scale features in convolutional layers, residual blocks, dense connectivity blocks and global average pooling to achieve a higher accuracy. We analyze the performances of the two innovated models and show significant improvements in model accuracy compared to conventional approaches. Furthermore, through conducting these experiments, we talk about the robustness issues regarding the complexity of the model, providing some insights to future image classification methods across different application domains.

1 Introduction

In the rapidly evolving field of machine learning, image classification remains an important application, driving advancements in both theoretical and practical aspects of the discipline. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images (1). It provides a precious opportunity to investigate advanced image classification methods. This project focuses on implementing some novel deep learning approaches to improve the accuracy of standard models like CNN, MLP, etc., trained on the CIFAR-10 dataset.

CIFAR-10 is chosen mainly because of its balanced distribution over different classes, making it an excellent benchmark for evaluating new concepts and techniques in image recognition. Although it is a commonly used dataset, there is still considerable potential to enhance both the accuracy and the computational efficiency of models trained with it. This project is based on a baseline using common deep learning techniques and then applies novel architectures and methods to exceed the baseline.

With this investigation, we hope to achieve better state-of-the-art results on CIFAR-10 while also developing a better understanding of current deep learning models' behaviors and boundaries with images at a small scale. These observations are critical for deploying future models that operate effectively across different domains and datasets. As a result, the results of this project will likely be utilized in informing researchers about what their future steps might be in the vector recognition fields and beyond, if feasible.

2 Background

The baseline model we used is a Convolutional Neural Network (CNN), optimized for the peculiarities of the CIFAR-10 dataset. The first convolutional layer contains 32 3x3 filters and includes the Rectified Linear Unit activation function . The subsequent max pooling 2x2 layer reduces feature map spatial dimensions, decreasing the computational burden on the following layers and limiting the risk of overfitting by including abstraction. The second convolutional layer also consists of 3x3 64 filters and provides a more complex feature extraction from the output of the first pooling layer . Following this, the input enters another 2x2 max pooling layer to further reduce dimensionality. The third convolutional layer includes 3x3 64 filters. Thereafter, a flattening layer transforms feature map 3D vectors to a 1D flat set to connect to dense layers . The first dense layer contains 32 ReLU units, responsible for incorporating the extracted features into a form suitable for classification. The model terminates with an output layer initialized with ten units to correspond to the ten classes offered in the CIFAR-10 dataset. The output layer employs the softmax function for probability establishment over the classes .

We ran this CNN for 10 epochs and achieved the result in Figure 1 with a 0.6998 accuracy and 0.9158 loss on the validation set at the end of 10 epochs.

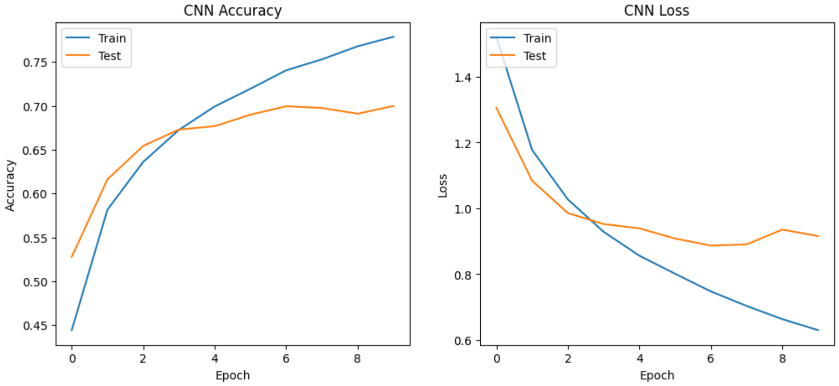

Figure 1: CNN on CIFAR-10.

3 Related Work

The CIFAR-10 serves as a crucial role in testing different image classification algorithms, especially deep learning techniques. For the past decades, a great many researchers have used this dataset to validate their algorithms and introduced novel designs and training approaches that increase the accuracy of identifying images.

The paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman (2), which introduces the VGG network, has a significant influence on image classification tasks, including those using the CIFAR-10 dataset. Although it was primarily used on ImageNet, the ideas from VGG—such as increased network depth with numerous convolutional layers, the utilization of small (3x3) convolution filters, and the stacking of multiple convolution layers before a pooling step—have been important to the enhancement of the CIFAR-10 performance. These innovations on the architecture has enabled deeper networks with more complex feature extraction capabilities while maintaining acceptable computational complexity, thereby setting a new standard for network design in image classification, including CIFAR-10, by promoting better feature representation and generalization.

The paper "Deep Residual Learning for Image Recognition" by He et al. (3) introduces the concept of Residual Networks (ResNets). It has also made a profound impact on deep learning, particularly in image classification. ResNets add skip connections which allow gradients to flow through the network without attenuation to address the vanishing gradient problem in deep neural networks. This innovation makes the training of much deeper networks possible by effectively bypassing layers that do not need training. As a result, it allows for better performance in image classification without increasing computational complexity.

Another influential architecture is the Dense Convolutional Network (DenseNet), introduced by Huang et al. (4) This model is designed based on the idea of feature reuse, which plays a crucial role on learning highly representative features with fewer parameters. DenseNet achieves this by connecting each layer to every other layer in a feed-forward fashion, significantly boosting feature propagation and reducing the number of parameters. Experiments on CIFAR-10 have shown that DenseNets not only enhance performance but also improve model compactness and efficiency.

Besides the architectural innovations discussed above, attention mechanisms have also been incorporated into CNNs for CIFAR-10. For example, the Residual Attention Network introduced by Wang et al. (5) uses attention modules stacked at multiple stages of the network to refine the feature representation. This method has proved to improve the performance by allowing the model to focus more on salient parts of the input image.

The preceding studies only represent a fraction of the vast work done by the machine learning community to improve the performance of image classification on CIFAR-10. They are not only a strong start for developing more complex models but represent the work and innovation required to adapt to more complicated image datasets. They also provide ideas and insights for me to develop my method in this project.

4 Methods

4.1 Failed method: Resnet50

We also tried to run ResNet50 for 50 epochs but achieved a much worser result than CNN in Figure 2 with a 0.3491 accuracy and 1.8723 loss on the validation set at the end of 50 epochs.

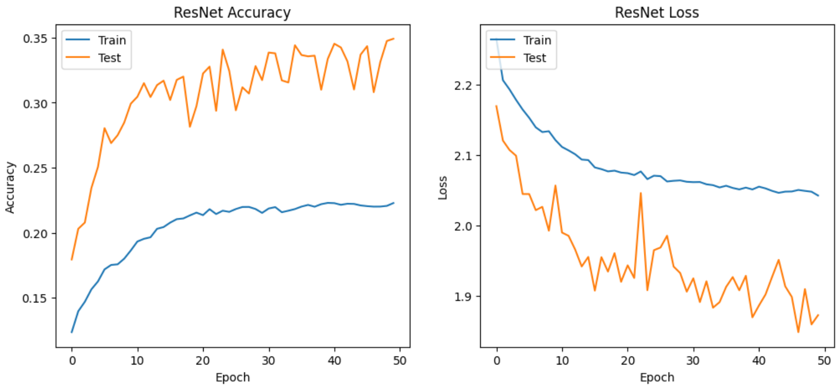

Figure 2: ResNet50 on CIFAR-10.

4.2 Successful Method 1: multi-scale convolutional features + dense connectivity + global average pooling

4.2.1 Multi-scale Convolutional Features

In this model, we use three different sizes (1x1, 3x3, and 5x5) of convolutional filters simultaneously. Motivation: The use of multi-scale features is motivated by the need to capture spatial hierarchies effectively of the images. The neural network can process diverse aspects of the input image by combining different filter sizes. This idea is inspired by the VGG's (2) success with small filters and multi-scale processing. We believe that this approach would capture a broad range of features from fine details to broader contextual information compared to a single sized convolutional filter.

4.2.2 Dense Connectivity

After the three different convolutional filters, we use a dense block to dense the connectivity of the features, which is similar to the DenseNet architecture (4). This method connects each layer to every subsequent layer within a block, which facilitates robust feature propagation and reuse across the network. Motivation: As discussed in the related work, dense connectivity tends to address the vanishing gradient problem and reduces number of parameters in the model, significantly enhancing training efficiency and model compactness. This idea is inspired by DenseNet's success in creating highly efficient and effective models for tasks in image recognition.

4.2.3 Global Average Pooling

Finally, the output of the dense's block is passed through a global average pooling and a final classification layer using a softmax activation. The aim of this design is to avoid overfitting by simplifying the final stages of the network with a relative small amount of trainable parameters.The motivation is that global average pooling simplifies the output stage of the model by reducing each feature map to a single summary value to enhance generalization.

4.2.4 Implementation Challenges and Reflections

This model consists of 117610 parameters with 116842 trainable. We trained this model using the Adam optimizer and cross-entropy loss function over 50 epochs. During the experiments, we encountered challenges such as optimizing the number and arrangement of dense blocks and controlling the computational complexity.

4.3 Successful Method 2: convolutional layer + residual blocks + dense connectivity blocks + attention modules + global average pooling

Method 2 is a more complicated and hybrid method. The model begins with an initial convolutional layer, followed by a series of residual blocks, a dense connectivity block, an attention module, a global average pooling layer and a softmax classification layer. Compared to method 1, method 2 is motivated by all the ideas in the related work.

4.3.1 Initial Convolutional Layer

Similar to the multi-scale convolutional layers in method 1, this idea is inspired by the VGG (2) network's use of very small (3x3) convolution filters. This layer aims to capture the details within the images efficiently.

4.3.2 Residual Blocks

Following the VGG-style initial convolution, the model employs ResNet-like residual blocks with skip connections. This idea is motivated by the success of ResNet (3). These blocks are designed to help the network learn deeper features by avoiding vanishing gradient problem.

4.3.3 Dense Connectivity Blocks

The use of dense connectivity blocks is similar to that in method 1. These layers connected in a feed-forward way, where each layer receives inputs from all previous layers and passes its output to every subsequent layer. The advantages of the dense connectivity blocks are the facilitation of feature reuse and reduction of the number of parameters in the network.

4.3.4 Attention Module

After the dense connectivity blocks, an attention module is used to allow the network to focus adaptively on the most informative features, potentially improving classification accuracy. This idea is inspired by the success of Residual Attention Networks (5).

4.3.5 Global Average Pooling and Softmax Output

The model concludes with a global average pooling layer and a softmax output as in method 1.

4.3.6 Implementation and Challenges

This model consists of 368594 parameters with 367378 trainable. The model was implemented using TensorFlow and trained on the CIFAR-10 dataset using the Adam optimizer and cross-entropy loss. We ran 50 epochs to adjust parameters and stabilize the learning process. Challenges encountered included balancing the complexity of the model with the computational resources available and tuning the model to prevent overfitting.

5 Results

5.1 Method 1

Method 1 was evaluated on the CIFAR-10 dataset over several training epochs. We used accuracy and cross-entropy loss as metrics. These metrics are critical for understanding how well the model generalizes to previously unseen data.

5.1.1 Performance Plot

The performance of method 1 is presented in Figure 3 with a 0.7206 accuracy and 0.8120 loss on the validation set at the end of the 50 epochs.

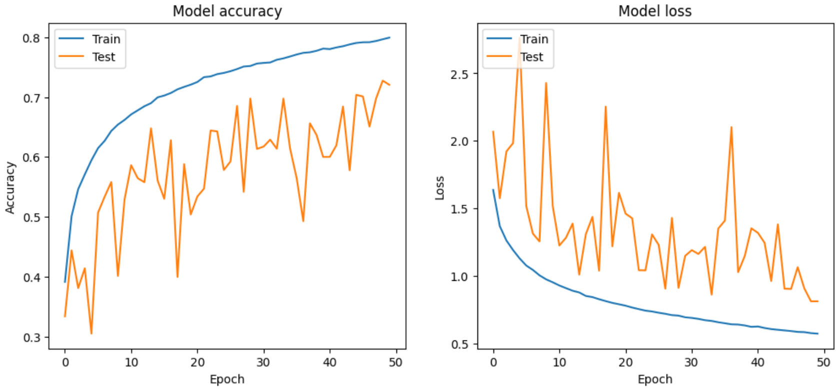

Figure 3: Method 1 on CIFAR-10.

Accuracy Plot: From the plot, we see that the model shows a consistent improvement in the training accuracy, starting from about 0.39 and reaching up to 0.8 by the end of the 50 epochs. Validation accuracy also improves but with some fluctuations. The highest validation accuracy achieved was 0.7275.

Loss Plot: The training loss decreased from 1.6368 to 0.5723, which shows effective learning. However, the validation loss experienced significant variability. This pattern suggests some overfitting or instability in model performance on the validation set.

5.1.2 Interpretation and Alignment with Expectations

From the performance plot above, we see that method 1 improve consistently in terms of training accuracy and loss, which aligns with the expectations for a model combining multi-scale layers and dense connectivity blocks. However, the validation results seem less stable, indicating some overfitting or sensitivity to certain types of image variability in the validation set.

5.1.3 Comparison to Prior Work

Compared to traditional CNNs (baseline), this model likely shows improved feature extraction capabilities, leading to a higher validation accuracy.

Compared to advanced architectures like ResNet or DenseNet, this model combines some of their best advantages which likely contributes to its strong training performance but also lead to some of the observed overfitting.

5.2 Method 2

Method 1 was evaluated on the CIFAR-10 dataset over several training epochs. Similarly, we used accuracy and cross-entropy loss as metrics.

5.2.1 Performance Plot

The performance of method 2 is presented in Figure 4 with a 0.7500 accuracy and 1.5475 loss on the validation set at the end of the 50 epochs.

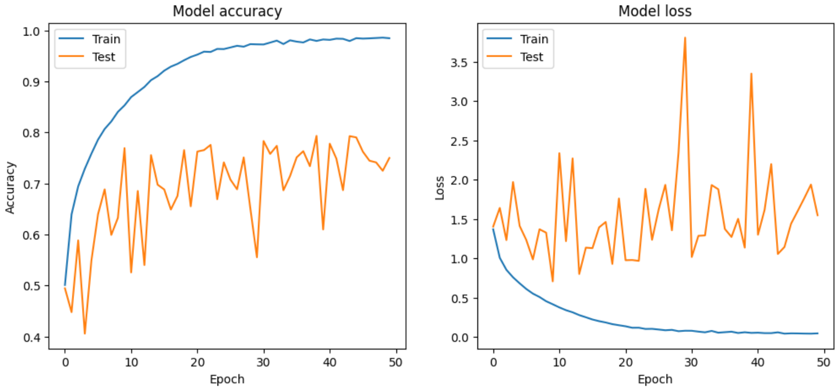

Figure 4: Method 2 on CIFAR-10.

Accuracy Plot: From the plot, we see that the model shows a consistent improvement in the training accuracy, starting from about 0.5011 and reaching up to 0.9847 by the end of the 50 epochs. Validation accuracy also improves but with some fluctuations. The highest validation accuracy achieved was 0.7933.

Loss Plot: The training loss decreased from 1.3687 to 0.0451, which shows effective learning. However, similar to method 1, the validation loss experienced significant variability. This pattern suggests some overfitting or instability in model performance on the validation set.

5.2.2 Interpretation and Alignment with Expectations

The training results met the expectations for a model incorporating advanced features like residual blocks, dense connectivity, and attention mechanisms. As the model gets more complicated than method 1 to enhance the ability to learn from complex datasets, the training and the validation performance improved. However, the fluctuation on the validation performance also indicates that this complex model might result in overfitting and instability.

5.2.3 Comparison to Prior Work

Compared to traditional CNN architectures, this model shows superior training and validation accuracy due to its sophisticated architecture.

Compared to advanced architectures like ResNet or DenseNet, this model aims to combine the best features of those models. However, the variability in validation accuracy and loss suggests that there is still room for improvement.

Compared to method 1, this model incorporates more complex architectures to capture the complicated nature of the images. Method 2 outperforms method 1 significantly in the training and validation accuracy. Both models show some overfitting and instability.

6 Discussion and Analysis

6.1 Analysis of Models and Results

Both models use a hybrid deep learning methods from previous work. These two models incorporate advanced architectural features such as residual blocks, dense connectivity, and attention mechanisms to combine their advantages to improve image recognition.

6.2 Limitations of the Current Approach

6.2.1 Overfitting

Although both models achieve a higher training and validation accuracy than the baseline, the fluctuating validation performances of both models suggest that they might overfit the training data. This overfitting could be caused by the models' complexity.

6.2.2 Model Complexity and Computational Demand

Although the models' advanced architecture have the advantage to capture detailed features within the data, it also might increase the computational complexity. Therefore, the complicated neural networks might take more time to train by consuming more resource, which may raise concerns about its practical usability on large-scaled datasets.

6.2.3 Generalization to Other Datasets

While the models perform well on the CIFAR-10 dataset, their architectures might not generalize to other types of image classification tasks without significant tuning and adaptation.

6.3 Insights into Machine Learning Models and the Environment

The models' performance on CIFAR-10 show the potential of combining several advanced architectures to improve image recognition. However, the overfitting problem also points out the challenges in machine learning about model generalization and the trade-off between model complexity and practical applicability. These results provide precious insights into training deep learning models on datasets. They also remind us the importance of balancing depth and simplicity in the design of deep learning models.

6.4 Improvement

6.4.1 Regularization

We could implement dropout layers or increase the weight decay to reduce the risk of overfitting of the models.

6.4.2 Data Augmentation

To improve generalization, we could employ a more robust data augmentation strategy such as transformations of the images.

6.4.3 Cross-Validation

Applying cross-validation during training could provide a more robust estimate of the model's performance on previously unseen data.

7 Code

The code of this project can be found at this github link.

References

[1] A. Krizhevsky, "CIFAR-10 and CIFAR-100 Datasets." `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009.

[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*, pp. 770–778, 2016.

[4] 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*, pp. 4700–4708, 2017.

[5] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2017.