ISSN 2959-6157

Exploring the Impact of Architectural Variations in ResNet on CIFAR-100 Performance: An Investigation on Fully Connected Layers, Residual Blocks, and Kernel Size

Rensi Deng

Agriculture and Engineering, Faculty of Science, University of Newcastle, upon Tyne, UK

Abstract:

ResNet is one of the leading neutral networks that has been widely applied in image classification. This study built a simple baseline network based on the concept of ResNet and then examines how variations in ResNet's architecture affect its model performance to provide insights for optimizing network design. Firstly, this study investigates the number of fully connected layers, the results show that by reducing the number of fully connected layers significantly decreases the total number of trainable parameters, which in turn reduces the training time. However, this reduction does not lead to a noticeable improvement in the accuracy after convergence. In addition, increasing the number of fully connected layers not only greatly increases the training time but also leads to overfitting on the CIFAR-100 dataset, slightly reducing the training performance. Secondly, this study also analyzes the influence of reducing number of residual basic block. Analysis suggests that reducing the use of Residual Blocks has significantly negatively impacted both accuracy and training time. This may be because the use of Residual Blocks positively affects the network's ability to learn features from the CIFAR-100 dataset. Finally, this study explores the effect of bigger kernel size of convolution layer in Residual Basic Block. The outcome demonstrates increasing the kernel size in the Residual Blocks significantly improves both training time and accuracy. Additionally, it was observed that in this variant experiment, a definitive convergence has not yet been clearly established, leaving the possibility that accuracy might continue to improve with more training epochs.

Keywords: Artificial intelligence; image classification; CIFAR-100; ResNet.

1. Introduction

In recent years, artificial intelligence has made significant progress, particularly in the field of computer vision. For instance, OpenAI's GPT-3 has garnered extensive attention for its remarkable performance in natural language processing. Image classification, as a crucial task in computer vision, plays a vital role in decision-making across various domains. The CIFAR-10 & 100 dataset, a fundamental benchmark in image classification, has been extensively studied [1]. These studies have propelled advancements in image classification techniques and laid a solid foundation for subsequent applications

Convolutional Neural Networks (CNNs) were first introduced by LeCun in 1998 where they demonstrated significant success in handwritten digit recognition through their innovative feature extraction capabilities [2]. Subsequently, Krizhevsky et al. showcased the impressive performance of deep CNNs in 2012 using the ImageNet dataset [3]. Their model achieved a substantial reduction in error rates, marking a breakthrough in deep learning for computer vision. Following that, Simonyan and Zisserman introduced the VGG network in 2014 paper [4]. By increasing the network depth and leveraging the ImageNet dataset, VGG achieved even higher classification accuracy. The deep architecture of VGG set new standards and became a milestone in image classification research

Among the many image classification networks, ResNet stands out as one of the most representative models, introduced by He et al. in 2015 [5]. ResNet addressed the degradation problem in very deep networks by incorporating residual connections. This innovation allowed for the effective training of deeper networks and significantly improved classification performance. It has since become a benchmark model in various image classification tasks. Nowadays, ResNet has been widely applied across different domains. For instance, Gudhe et al. employed ResNet to extend the classical U-Net architecture for biomedical image segmentation [6]. Li et al. proposed an Adaptive Multiscale Deep Fusion Residual Network (AMDF-Res-

Net) to enhance remote sensing image classification performance [7]. Zhang et al. explored a Two-Stream Residual Convolutional Network (TS-RCN) for visual tracking to address the limitations of current deep learning-based trackers when dealing with challenges such as dense distractors, confusing backgrounds, and motion blur [8]. Kocić et al. utilized ResNet to build an end-to-end autonomous driving system, enhancing the vehicle's environmental perception capabilities and demonstrating excellent performance across various driving scenarios [9]. Although these studies highlight ResNet's broad applications, they primarily focus on the model's performance in various contexts without delving into how structural changes within ResNet impact its performance. Therefore, this study aims to explore the effects of structural variations in ResNet on training outcomes using the CIFAR-100 dataset.

This study utilized the CIFAR-100 dataset and built upon the ResNet architecture for the experiments. The developed approach involves modifying ResNet in three key aspects: adjusting the number of residual blocks in the network, altering the convolutional kernel configurations within the residual blocks, and changing the number of fully connected layers at the network's end. By examining these variations, this study aims to understanding how structural changes impact ResNet's training performance, providing new insights for network design optimization.

2. Method

2.1 Data Description and Preparation

This study focused on the dataset called CIFAR-100, a classic dataset for computer vision classification task, it is collected and introduced by Krizhevsky et al. [1]. This dataset contains 100 distinct classes, which are evenly organized into 20 super categories. Totally there are 60000 images in this dataset and each image is labeled with a "fine" label indicating its specific class and a "coarse" label representing its super category. Each image has a size of 32 x 32 x 3. Fig. 1 below is dataset demonstration, where 20 random images and their classification have been selected from the CIFAR-100 dataset. To be noted that the CIFAR-100 dataset is already well divided into a training set and a test set, with a split ratio of 5:1.

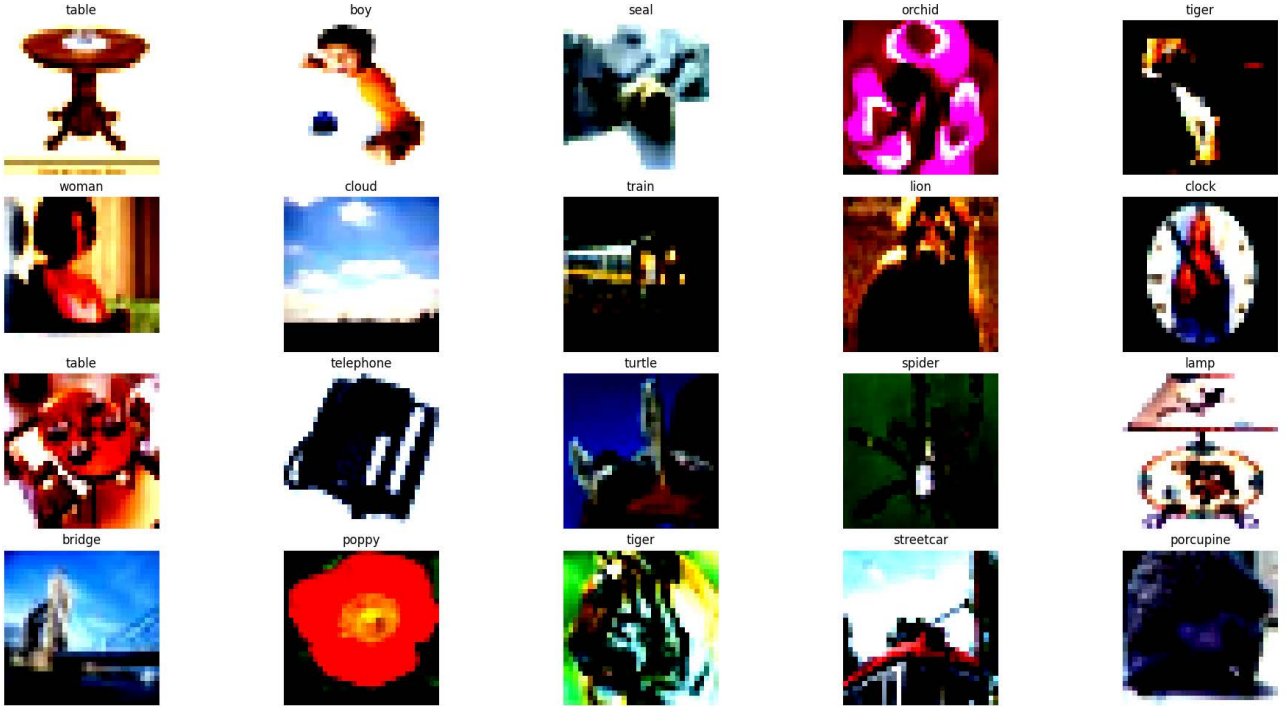

Fig. 1 20 sample images from CIFAR-100 [1].

The data preprocessing used in this study consists of three parts. First, the training set is normalized based on its mean and standard deviation. This normalization helps in stabilizing and accelerating the training process, as it ensures that the input features have similar scales and reduces the risk of numerical instability. Secondly, each data is padded with a border of 4 pixels on each side. This padding increases its overall dimension from 32 x 32 to 40 x 40. After padding, a random 32 x 32 is cropped from the padded image. The random cropping introduces variability into the training data, which helps the model generalize better [10]. Thirdly, each data is randomly flipped horizontally with a probability of 50%. By introducing horizontal flips, this augmentation technique creates vari-

ations of the original image. Augmenting the dataset with horizontally flipped images helps the model learn features that are invariant to horizontal flipping, improving its robustness [10]. Fourth, data is randomly rotated within a range of -15 to 15 degrees. By doing so the model can generalize better, as it encounters different angles during training.

2.2 ResNet-Based Classification

ResNet is a deep learning architecture introduced by Kaiming He et al. in 2015[5]. It is designed to address the challenges associated with training very deep neural networks, particularly the degradation problem where accuracy gets worse as the network depth increases.

The central idea of ResNet is the use of residual connections (or skip connections). These connections allow the output of a previous layer to be added directly to the output of a later layer. Essentially, residual connections enable the network to learn residual mappings, which are the differences between the desired output and the input. This helps in training deeper networks by mitigating the vanishing gradient problem.

It has been demonstrated that, for networks of the same depth, ResNet can better address optimization challenges and achieve higher accuracy compared to plain networks on datasets such as CIFAR-10, ImageNet, and MNIST [5]. In this study, a relatively simple neural network is first constructed based on the principles of residual networks (Fig. 2), serving as the baseline model. Then the number of fully connected layers at the end of the network (Experiment 1), the number of residual blocks in the network (Experiment 2), the configurations of convolutional kernels within the residual blocks (Experiment 3) are then adjusted on this baseline network to create different variant networks. The performance of these variants is recorded to analyze the impact of different hyperparameter combinations.

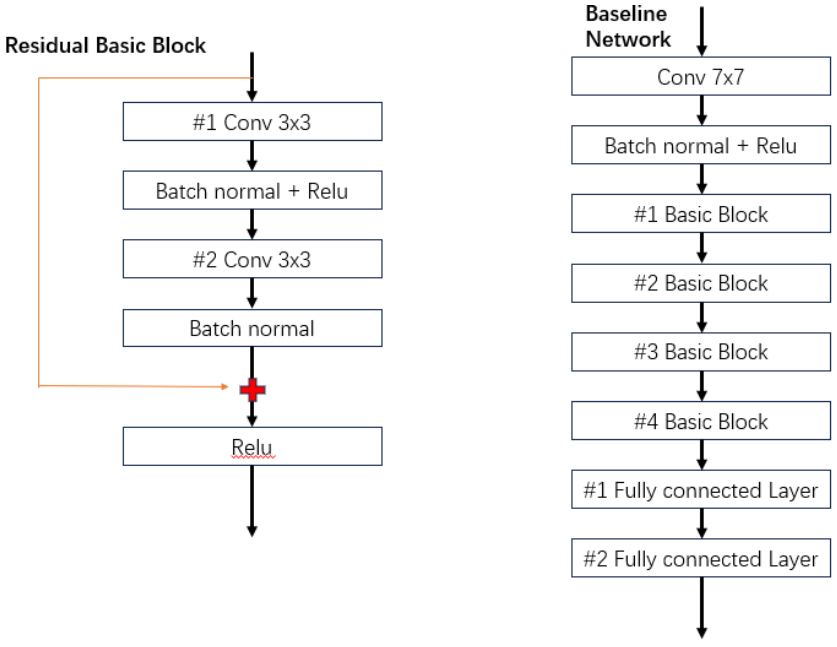

Fig. 2 Basic Residual Block & Baseline Network (Photo/Picture credit: Original).

This Basic Residual Block (Fig. 2) is a fundamental component in the ResNet architecture, designed to facilitate deep residual learning. It consists of two convolutional layers with 3x3 kernel, connected by batch normalization and ReLU activation functions. Specifically, the input tensor x is added to the output of the second convolutional layer (#2 conv). This addition occurs after the second batch normalization (bn2) and before the final ReLU activation. This design allows the network to learn residual mappings, which helps in effectively training very deep networks by mitigating issues such as vanishing gradients and degradation in performance. This Baseline Residual Network (Fig. 2) begins with an initial 7x7 convolutional layer that extracts feature from input images, followed by batch normalization and ReLU activation. Then the output process with 4 Resnet Basic Block introduced in last section. Finally, the output from the residual blocks is flattened and fed into 2 fully connected layer to produce the final classification output.

2.3 Experiment 1

In this experiment, the last fully connected layer is removed from baseline network (Fig. 3) and one more fully connected layer is added at the end of the baseline network (Fig. 4). The impact of the number of fully connect-

ed layer employed in baseline residual network will be discussed later.

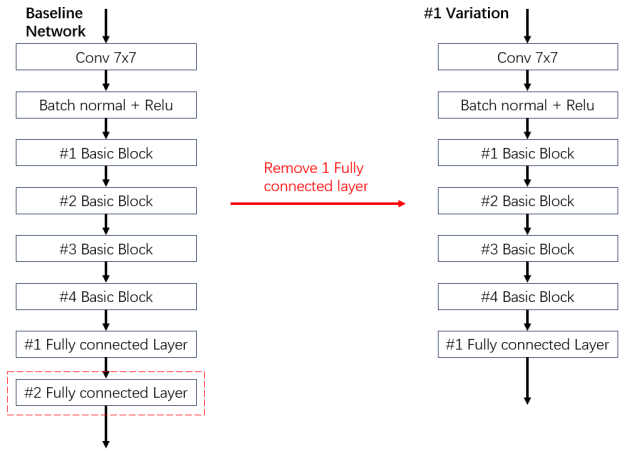

Fig. 3 #1 Variation of the architecture (Photo/ Picture credit: Original).

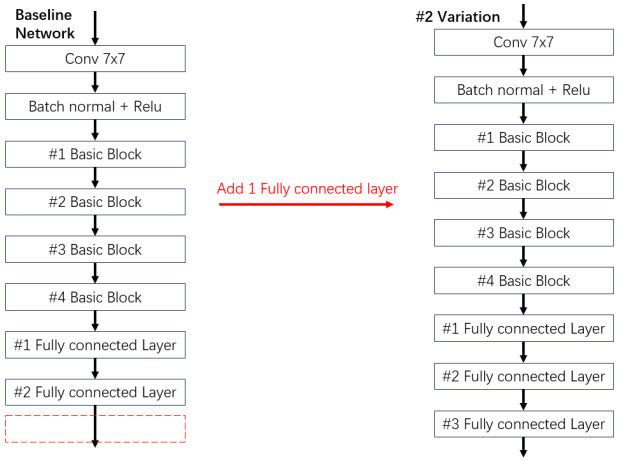

Fig. 4 #2 Variation of the architecture (Photo/ Picture credit: Original).

2.4 Experiment 2

In this experiment, the number of residual basic blocks is reduced from 4 to 2 (Fig. 5) and its impact will be discussed later, below is the variation network architecture.

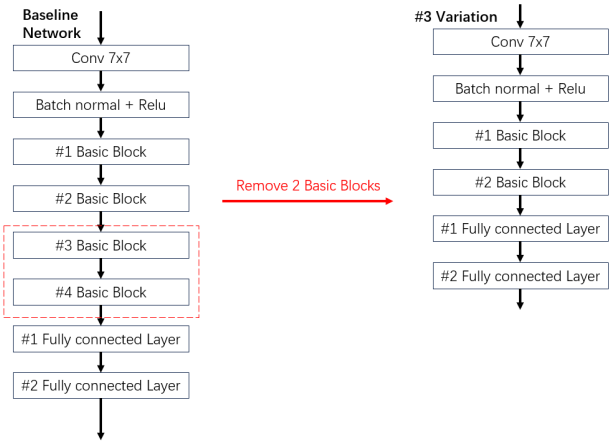

Fig. 5 #3 Variation of the architecture (Photo/ Picture credit: Original).

2.5 Experiment 3

In this experiment, the kernel size of convolutional layer in basic block will be changed from 3x3 to 5x5 (Fig. 6) and its impact will be discussed later, below is the variation network architecture.

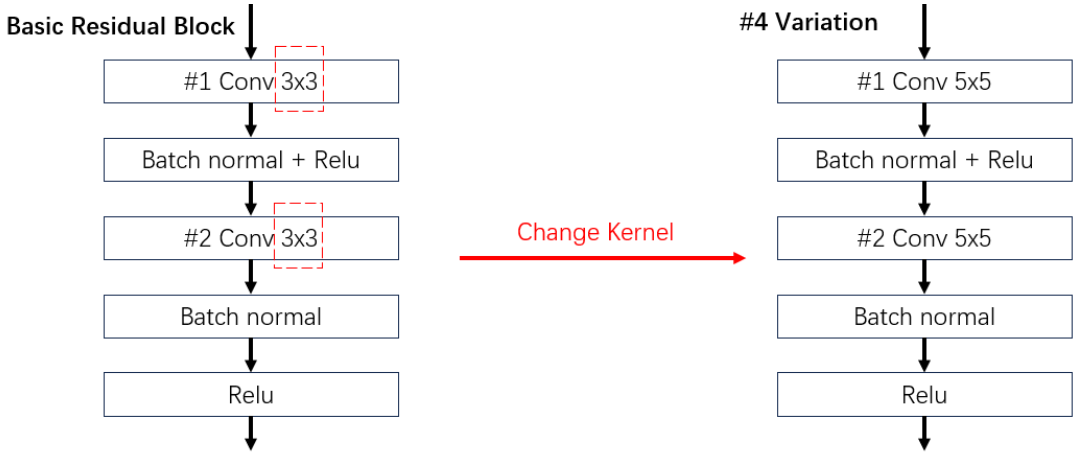

Fig. 6 #4 Variation of the architecture (Photo/Picture credit: Original).

2.6 Implementation Details

The Resnet Baseline Network as well as all the variation network are all implemented using Pytorch from python and Data augmentation were implemented using the torchvision from python. All the network is trained by CPU (13th Gen Intel(R) Core (TM) i5-1340P 1.90 GHz). The following training parameters are used: batch size = 4, number of workers=2, learning rate = 0.0004, optimizer = Adam, loss function = Cross Entropy Loss. The accuracy is employed as the evaluation metrics for all experiments.

3. Results and Discussion

3.1 The Influence of the Number of Fully Connected Layer in Model Performance

This study first examines the impact of changing the number of fully connected layers at the end of the baseline network. It compares the accuracy and training time between the modified versions and the baseline model.

Variation # 1 ResNet with 1 FC layer: In terms of accuracy, it can be observed that after 15 training epochs, the accuracies of both networks converge around 35%, with convergence beginning around the 5th epoch shown in Fig. 7. However, it is noteworthy that during the first 5 epochs, the variant network consistently shows an accuracy that is approximately 5-7% higher than the baseline network. Additionally, the variant network achieves a peak accuracy of 38%, which is slightly higher than the baseline network's 36%. As for training time, the variant network significantly outperforms the original network. The original network takes an average of 57 minutes per training run, while the variant network requires only approximately 7.5 minutes.

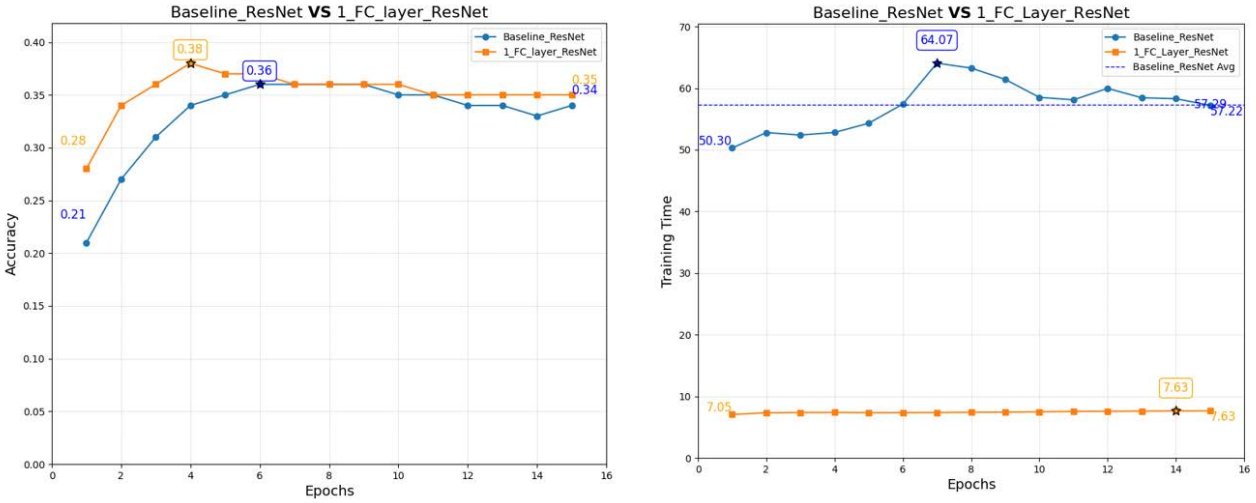

Fig. 7 The Accuracy and Training Time of Baseline ResNet VS ResNet with only 1 FC layer (Photo/Picture credit: Original).

Variation # 2 ResNet with 3 FC layers: In terms of accuracy shown in Fig. 8, it can be observed that during the first four rounds of training, there is no significant difference between the original network and the variant network. However, the variant network starts to converge after the fourth round of training, with a final accuracy of 31% and a peak accuracy of only 33%. In contrast, the original network begins to converge later (around the sixth round), achieving a higher peak accuracy of 36% and a final accuracy of 34%. Regarding training time, the original network performs significantly better than the variant network, with an average training time of 57 minutes per run, while the variant network takes nearly 90 minutes per run on average.

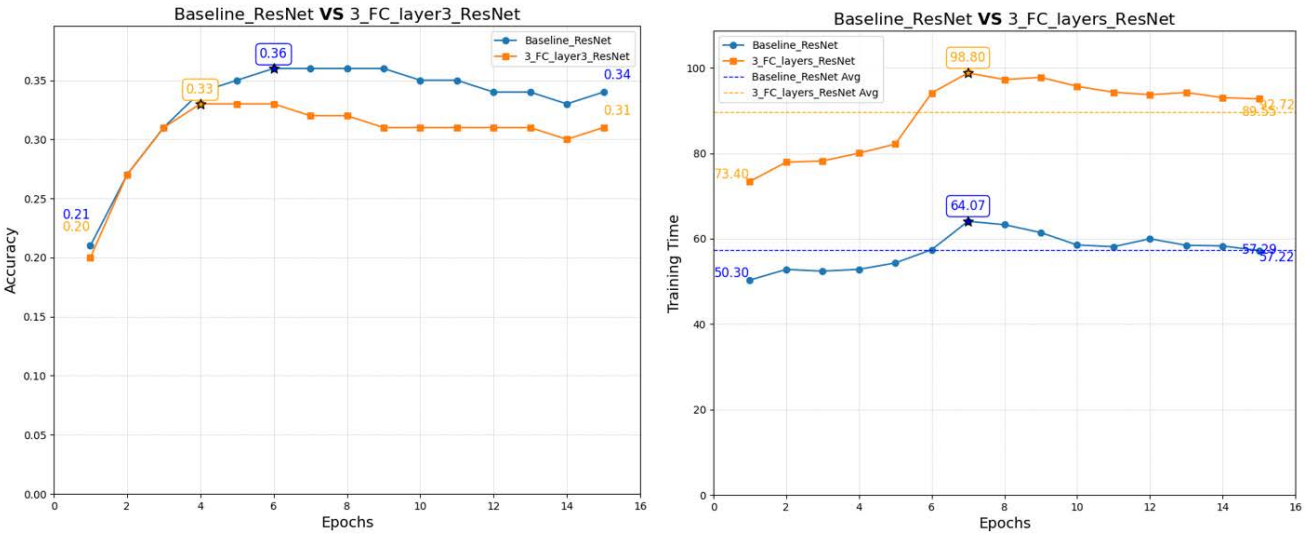

Fig. 8 The Accuracy and Training Time of Baseline ResNet VS ResNet with 3 FC layers (Photo/ Picture credit: Original).

Table 1. Number of Trainable Hyperparameters in variation #1 & variation #2

Network	# of Trainable Hypereparameters
Original(Baseline Residual Network)	8.9M
#1 Variation(ResNet with 1 FC)	1.5M
#2 Variation(ResNet with 3 FC)	18.3M

From the results above, it can be concluded that, compared to the baseline network, on one hand, Variation #1 shows higher accuracy in the early stages and achieves a higher peak accuracy. This is likely because the overall network has fewer trainable parameters (Table 1), leading to more efficient learning in the initial phases. Additionally, due to the fewer parameters, the computational overhead is significantly lower than that of the baseline network. On the other hand, Variation #2, with its higher number of parameters, may suffer from overfitting or be overly complex for the CIFAR-100 dataset. This results in the lowest training efficiency and the lowest peak accuracy among the variants.

3.2 The Influence of Reducing the Number of Residual Basic Block

Experiment 2 examines the impact of reducing the number of Residual Basic Block employed the baseline network from 4 to 2. It compares the accuracy and training time between the modified version and the baseline model.

Variation #3 ResNet with 2 Basic Block: In terms of accuracy shown in Fig. 9, the variant network shows consistently lower results in each training round compared to the original network, with a peak accuracy of only 31%, which is below the original network's 34%. In terms of training time, the variant network takes an average of 90 minutes per round, which is significantly higher than the original network's 57 minutes per round.

The variant network performs worse than the original network in both accuracy and training time. This is likely because reducing the number of Residual Basic Blocks impairs the network's feature extraction capabilities, leading to a decline in both training performance and efficiency.

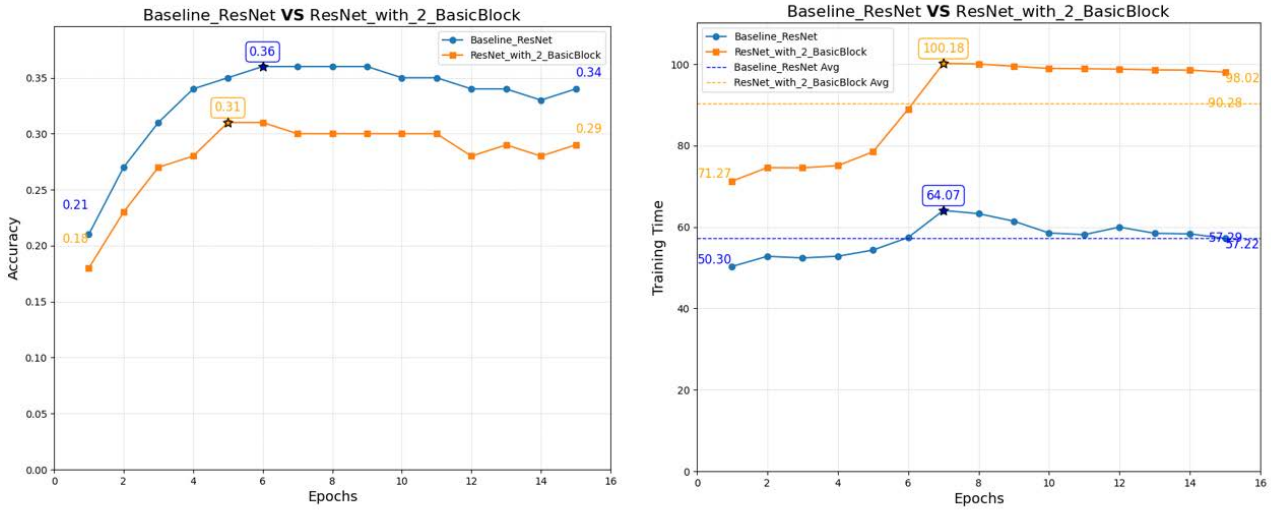

Fig. 9 The Accuracy and Training Time of Baseline ResNet VS ResNet with 2 BasicBlock (Photo/Picture credit: Original).

3.3 The Influence of Change the Kernel Size of Convolution Layer in Residual Basic Block

Experiment 3 examines the impact of changing the Kernel size of convolution layer in Residual Basic Block from 3x3 to 5x5. It compares the accuracy and training time between the modified version and the baseline model.

Variation #4 ResNet with 5x5 Kernel of convolution layer in Basic Block: In terms of accuracy shown in Fig. 10, the original network consistently outperforms the variant network in the first 10 training epochs. However, after the 10th epoch, the original network has already entered a stage of convergence and degradation, while the variant network's performance continues to improve. By the 15th epoch, the variant network shows potential convergence, reaching 38% accuracy (due to limited research time, this experiment only runs up to the 15th epoch; if the training period were extended, the variant network's accuracy might improve further). In terms of training time, the variant network takes an average of 12 minutes per run, which is significantly better than the original network's average of 57 minutes per run.

From the results of the experiment, it can be observed that although the variant network initially performs worse than the original network in terms of accuracy, it surpasses the original network's accuracy starting from the 10th epoch and might continue to improve beyond the 15th epoch. This phenomenon could be due to the increased kernel size, which allows the network to capture features over a larger range, enhancing its ability to learn from the CIFAR-100 dataset. In terms of training time, the variant network shows a significant reduction, likely related to the decreased number of trainable parameters. Although increasing the kernel size from 3x3 to 5x5 may increase the computation per convolutional layer, the overall parameter count in the variant network is much lower (1.7M vs 8.9M) (Table 2), resulting in shorter training times. This indicates that despite the increased computational complexity per convolutional layer, the reduced number of parameters leads to a significant reduction in training time.

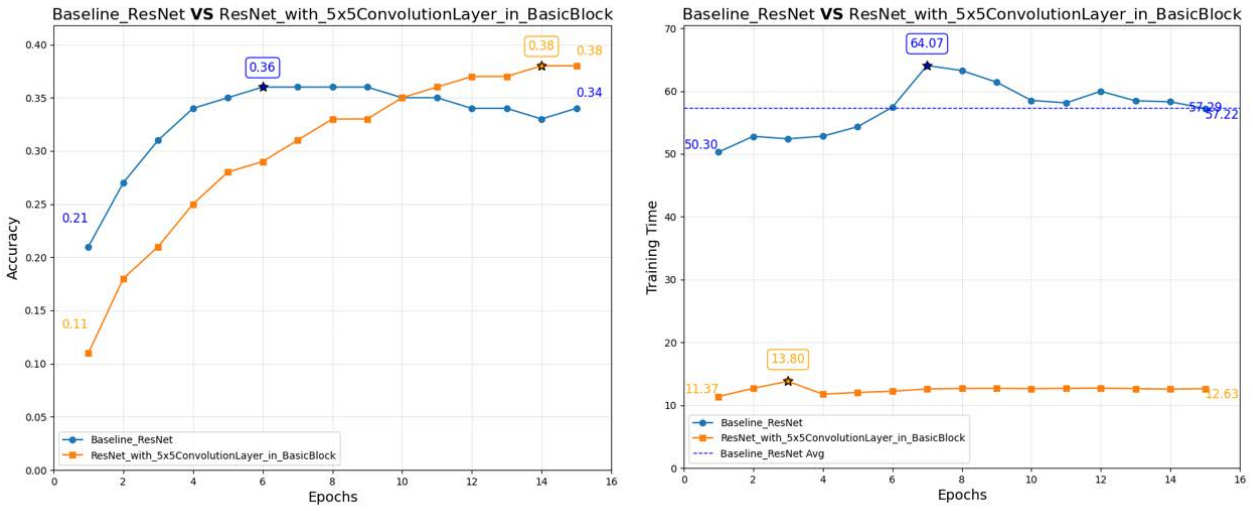

Fig. 10 The Accuracy and Training Time of Baseline ResNet VS ResNet_ with_5x5ConvolutionLayer_in_BasickBlock (Photo/Picture credit: Original).

Table 2. Number of Trainable Hyperparameters in variation #4

Network	# of Trainable Hypereparameters
Original(Baseline Residual Network)	8.9M
#4 Variation(ResNet_with_5x5ConvolutionLayer_in_BasickBlock)	1.7M

4. Conclusion

In this study, first a simple residual network based on the residual shortcut conception is established as the baseline network. Then, by varying three parameters in the baseline network, the impact on the CIFAR-100 dataset is discussed regarding to training accuracy and training time. These three parameters are: the number of fully connected layers at the end of the baseline network, the number of Residual Basic Blocks, and the kernel size in the Residual Basic Blocks. Experimental results showed that firstly reducing fully connected layers in the baseline network decreases trainable parameters and training time but doesn't improve accuracy. More layers increase training time and cause overfitting on CIFAR-100, slightly lowering performance. Secondly reducing Residual Blocks negatively impacts accuracy and training time, likely because these blocks enhance feature learning from CIFAR-100. Thirdly increasing kernel size in Residual Blocks improves both training time and accuracy, with potential for further accuracy gains if training continues. In the future it is planned to explore how other structural changes affect RseNet's performance such as adjusting the learning rate schedules and incorporating attention mechanisms

References

[1] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images, cs.utoronto.ca, 2009.

[2] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998 Nov;86(11):2278-324.

[3] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012;25.

[4] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014 Sep 4.

[5] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).

[6] Gudhe NR, Behravan H, Sudah M, Okuma H, Vanninen R, Kosma VM, Mannermaa A. Multi-level dilated residual network for biomedical image segmentation. Scientific Reports. 2021 Jul 8;11(1):14105.

[7] Li G, Li L, Zhu H, Liu X, Jiao L. Adaptive multiscale deep fusion residual network for remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing. 2019 Jun 26;57(11):8506-21.

[8] Zhang N, Liu J, Wang K, Zeng D, Mei T. Robust visual

object tracking with two-stream residual convolutional networks. In2020 25th International Conference on Pattern Recognition (ICPR) 2021 Jan 10 (pp. 4123-4130). IEEE.

[9] Kocić J, Jovičić N, Drndarević V. An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. Sensors. 2019 May 3;19(9):2064.

[10] Mumuni A, Mumuni F. Data augmentation: A comprehensive survey of modern approaches. Array. 2022 Dec 1;16:100258.