# ECE 7123 Deep Learning Project 1

https://github.com/Kang346/25SpringDL_Projects

**Ruizhe Huang rh4145** **Junhan Zhang jz7178** **Zikang Zhang zz10463**

## Overview

Our initiative centers on developing a tailored deep learning model built upon the ResNet architecture for image classification tasks using an adjusted version of the CIFAR-10 dataset. The core goal is to achieve the highest possible accuracy on test data while strictly limiting the model's parameter count to under 5 million. To address this, we explored structural adjustments such as applying SGD optimization, optimizing convolutional layer and fully connected layer configurations, experimenting with pooling methods and channel attention. Next, we compare the results obtained on the CIFAR-10 test set with those of our Kaggle-custom test set.Conducting methodical testing and iterative adjustments, our aim is to produce a high-performing model that adheres to the specified computational constraints without compromising classification precision.

## Methodology

### Original structure

In the beginning, the basic neural network is implemented exactly according to the structure in the introduction (figure 1), without any modifications.

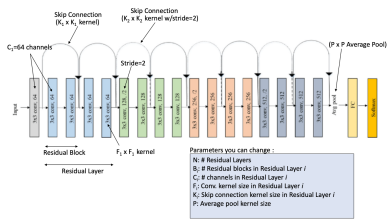


Figure 1: Structure in the introduction

The accuracy of this model is 84.17%. However, the number of parameters is more than 11 million which far exceeds the required 5 million.

### Model Optimization and Data Augmentation

Since the original model achieved an accuracy of 84.17%, but the number of parameters far exceeded the required limit, we first halved the number of channels. This adjustment reduced the number of parameters to one-fourth of the original model. However, this also led to a drop in accuracy to 81.27%. Therefore, further methods were needed to improve accuracy. Since CIFAR-10 is a small dataset, data augmentation is highly effective. To leverage this, we applied the following techniques:

- Random cropping to improve robustness to spatial transformations.
- Horizontal flipping to enhance generalization.
- Color jittering to introduce variation in brightness, contrast, and saturation.
- Normalization to standardize input features.

Additionally, we modified the optimizer and learning rate. The previous model used Adam, but SGD with Momentum typically outperforms Adam on CIFAR-10, especially for small networks. The initial learning rate of 0.001 was too slow, so we increased it to 0.1 for faster convergence. Furthermore, L2 regularization (weight decay) was introduced to prevent overfitting. figure 2 shows the change in the learning rate over 10 cycles.

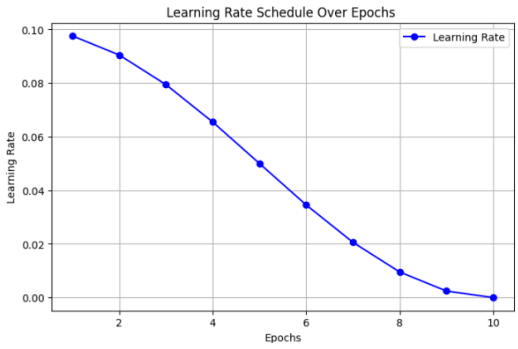


Figure 2: Learning Rate Schedule Over Epochs

### Two-layer Fully Connected Layer

Since the number of parameters is only 2.8 million, this means that the complexity of the neural network can be increased. Inspired by LeNet LeCun et al. (1998), an intermediate layer was added to the fully connected layers. As figure 3 shown, in the fully connected layer, instead of directly

decreasing the dimension from 1024 to 10, a middle layer whose dimension is 800 is added. The progressive reduction from 1024 → 800 helps gradually extract high-level features while contributes to the control of the number of parameters and the maintenance of learning capacity.

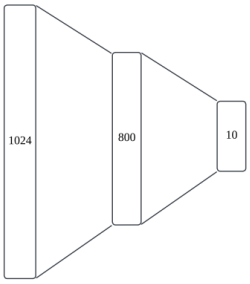


Figure 3: Structure of two-layer fully connected layer

The accuracy of this model is 88.5%, showing further improvement and the number of parameters is 3.7 million, within the limitation.

The drawback of this model is that it sometimes assigns high weights to the background, leading to misclassification. As shown in figure 4, the first set of images assigned very high weights to the bottom of the ship and the padding at the edges, while the second set of images assigned very high weights to the background above the object.

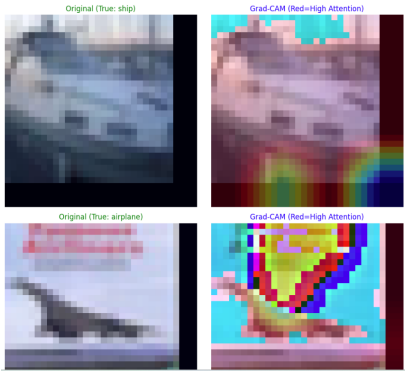


Figure 4: Heat map of this version

## Pooling layer

To allow the model to assign higher weights to the features of objects in the images, a max pooling layers whose kernel size is 2 x 2 and stride is 1 was added after each residual layer (figure 5 ).

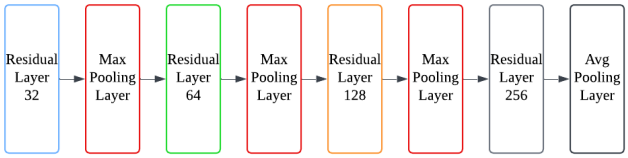


Figure 5: Structure of pooling layer

The accuracy of this model is 93.2%, showing some improvement compared to the previous version. and the number of parameters is 3.7 million, within the limitation.

By comparing figure 4 and figure 6, the issue of excessively high background weights has been partially resolved; the bottom of the ship in the first set of images no longer has high weights. However, the background in the second set of images still has high weights.

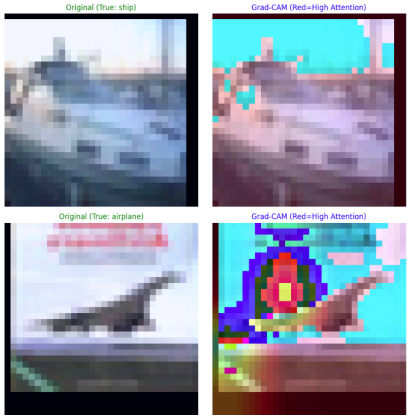


Figure 6: Heat map comparison to previous version

## Channel Attention

Considering that human vision has different sensitivities to the three primary colors, I decided to incorporate an attention mechanism into our network. Specifically, I applied channel attention by utilizing a Squeeze-and-Excitation (SE) block Hu et al. (2017). I modified the original BasicBlock by adding an SE block at the end before summing it with the original skip connection input X. In addition to this, I also introduced a dropout layer within the BasicBlock to reduce overfitting and improve the model's generalization. The updated structure of the BasicBlock is now as shown in the figure.

After adding the SE layer, making appropriate adjustments to the network structure, and fine-tuning some hyperparameters, the accuracy on the CIFAR-10 test set has reached 94.0%.

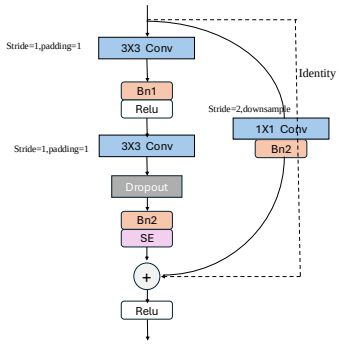


Figure 7: Basic block after adding SE and dropout

# Experiment and Results

## Gap between CIFAR-10 and kaggle

We have previously discussed the training methodologies employed. In this section, we will compare the results obtained on the CIFAR-10 test set with those on our custom test set from Kaggle. Our primary objective is to analyze the performance discrepancies between these datasets, identify potential factors contributing to the differences, and explore strategies to improve the model's generalization ability across varying test distributions.

The accuracy observed on the CIFAR-10 test set often exhibits a substantial gap compared to that on the Kaggle test set, with the difference typically around 11%. For instance, when the local accuracy reaches 93%, the accuracy on the Kaggle test set is only approximately 81%. Likewise, when the local accuracy achieves 94.5%, the online accuracy on Kaggle stands at 83.4%. In order to better understand the root cause of this performance discrepancy, we conducted a series of experiments.

## Confusion Matrix

We tested the confusion matrix on the CIFAR-10 test set to analyze which classes the model performs poorly on and which classes are easily confused. The results are as follows:

| Actual \ Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 947 | 1 | 13 | 5 | 2 | 2 | 0 | 0 | 24 | 6 |
| 1 | 5 | 965 | 1 | 1 | 1 | 0 | 0 | 0 | 3 | 24 |
| 2 | 16 | 0 | 927 | 13 | 15 | 16 | 6 | 6 | 1 | 0 |
| 3 | 6 | 0 | 24 | 856 | 17 | 67 | 17 | 6 | 4 | 3 |
| 4 | 3 | 0 | 7 | 16 | 955 | 5 | 5 | 7 | 1 | 1 |
| 5 | 3 | 0 | 7 | 58 | 10 | 907 | 3 | 11 | 0 | 1 |
| 6 | 4 | 0 | 18 | 28 | 7 | 2 | 939 | 1 | 1 | 0 |
| 7 | 2 | 0 | 5 | 9 | 14 | 9 | 0 | 958 | 3 | 0 |
| 8 | 12 | 4 | 2 | 4 | 0 | 0 | 0 | 0 | 969 | 9 |
| 9 | 5 | 18 | 1 | 3 | 0 | 0 | 0 | 0 | 9 | 964 |

Table 1: Confusion Matrix on CIFAR-10 Test Set

Upon further analysis, we identified that class 3 corresponds to cats and class 5 to dogs. Given the low resolution of 32×32 pixels, distinguishing between these two categories presents a significant challenge.

## Improvment on specefic classes

To enhance the model's ability to capture distinguishing features between cats and dogs, we employed oversampling to increase the representation of these classes in the training set. However, experimental results indicate that this approach did not lead to a noticeable improvement in overall model performance. While there was a slight increase in classification accuracy for these two classes, the global performance remained largely unaffected.

Additionally, we experimented with CutMix Sangdoo Yun (2019). This method yielded a marginal improvement, with the model's accuracy increasing to 94.3%.

We also explored the approach of applying different weights to errors during training, with the aim of directing the model's focus toward the more challenging classes. Specifically, we assigned a higher weight to errors in categories 3 and 5 (representing cats and dogs) compared to errors in other categories. This was done with the expectation that the model would prioritize learning the distinguishing features of these difficult-to-classify categories, thereby improving its overall classification performance for these particular classes.

After applying weighted loss and CutMix, the model achieved an accuracy of 94.7% on the test set. Additionally, the accuracy on the Kaggle test set improved to 84.1%. These adjustments helped enhance the model's performance both in local testing and on the Kaggle platform.

## Weakness

Due to time constraints, the performance of our model on the Kaggle test set still exhibits a significant disparity compared to the CIFAR-10 test set. This discrepancy may be attributed to differences in the distribution of the test sets between Kaggle and CIFAR-10, coupled with the model's insufficient generalization capability. Alternatively, it can also be suggested that our model may have overfitted to the CIFAR-10 dataset, leading to reduced performance on the Kaggle test set. The following figure illustrates the variation in the training accuracy curve over the course of a training session, providing support for this hypothesis.

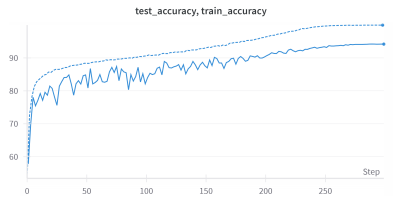


Figure 8: Trian and test accuracies

The train accuracy reached 99.96%, while test accuracy is only 94.19%, which means the model may have overfitted.

## Conclusion

In this work, we proposed a modified version of the basic ResNet-18 architecture, incorporating several key modifications aimed at enhancing the model's performance. Specifically, we introduced pooling layers between every two convolutional layers, allowing the model to shift its focus away from background features and concentrate on more discriminative information. Additionally, we incorporated residual blocks with Squeeze-and-Excitation (SE) modules, enabling the model to assign varying weights to different channels and thereby improve its ability to capture complex feature representations.

To further bolster the model's generalization ability, we employed various data augmentation techniques during training, enhancing the model's robustness to variations in input data. We also analyzed the model's performance across different classes using confusion matrices, which provided valuable insights into the specific categories where the model struggled. To address these challenges, we introduced weighted loss functions, assigning higher weights to errors from difficult-to-classify classes (such as categories 3 and 5) in an effort to guide the model's attention to these hard cases.

As a result of these improvements, our model achieved an accuracy of nearly 95% on the CIFAR-10 test set. Additionally, on the Kaggle custom test set, the model's accuracy reached 84%. These findings demonstrate the effectiveness of our approach in improving both local and real-world model performance, highlighting the importance of architectural modifications, data augmentation, and error weighting for optimizing generalization.

## References


Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Wu, E. 2017. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*. Journal version accepted by TPAMI.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradientbased learning applied to document recognition. *PROC OF THE IEEE*.

Sangdoo Yun, Dongyoon Han, S. J. O. S. C. J. C. Y. Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. *arXiv*.