Image classification using pretrained convolutional networks

1.Introduction

The rise of deep learning techniques has sparked a revolution in the field of computer vision, especially in image recognition tasks. With the improvement of hardware performance and the increased availability of large-scale datasets, researchers have proposed a series of advanced deep neural network models to solve problems such as image classification. In this context, this thesis aims to provide an in-depth study and comparison of three representative deep learning image recognition models: the GoogleNet, AlexNet and ResNet.

Deep learning has made breakthroughs in several areas, especially in computer vision. The continuous deepening and expansion of neural network models provides more powerful tools for image recognition tasks, which is crucial for applications facing complex scenes, large-scale data and diverse image types.

As model complexity increases, we face challenges in training and inference efficiency. Different deep learning architectures employ different strategies in model design, such as GoogleNet's Inception module, AlexNet's deep convolutional structure, and ResNet's residual connectivity. Understanding the design principles of these models and comparing their performance in tasks is important for model selection in resource-constrained environments.

2.Experiment

Through an in-depth comparison of these three typical models, we aim to evaluate their classification performance on the CIFAR-10 dataset. Evaluate and analysis the training results of the models on the datasets. Explore their adaptability to different image types, including complex scenes, small-sized objects, etc. This research aims to provide insights into model selection for real-world applications and to facilitate the further development of deep learning in the field of image recognition. The implementation is carried out on Jupyter, utilizing Pytorch library for deep-learning functionalities. The main phases of my implementation include data processing, training procedure and evaluation metrics.

2.1 Pretrained models

GoogleNet

GoogleNet improves the computational efficiency of the model by introducing the Inception module, resulting in excellent performance in image recognition tasks. Its comprehensive performance and ability to adapt to complex image scenarios have made it a highly regarded classical model in the field of deep learning.[1]

AlexNet

AlexNet is a pioneering work in the field of deep learning image recognition, and its remarkable success in the 2012 ImageNet competition led the way for deep learning. By introducing deep convolutional structures and Dropout techniques, AlexNet effectively solves the overfitting problem and lays the foundation for the design of the subsequent models.[2]

ResNet50

The proposal of ResNet solves the gradient problem in deep neural networks by introducing residual blocks, enabling deeper and more stable training of the model. Its excellent performance in the field of image recognition makes it one of the current representatives of deep learning image models.[3]

2.2 Data Processing

The CIFAR-10 dataset is imported from torchvision package, for GoogleNet and ResNet, the input size of image should be 224*224, but for AlexNet should be 227*227. So the transformer of them are separated. The dataset is splited into 40000(randomly split for training), 10000(for validation), 10000(for testing).

2.3Data visualization

All of data are loaded into dataloaders for each pretrained model. When reading the dataloader, meanwhile visualize these images randomly.

Figure 1. Visualization of the images in CIFAR-10

2.4 Training Setup

As there are 3 pretrained models need to fit to CIFAR-10 dataset, the performance of colab isn't enough for doing it, so the project is moved to personal desktop.

For CIFAR-10 dataset, there is only 10 classes, so the number of output of different models are also need to be set to 10

```
Linear(in_features=1024, out_features=10, bias=True)
Linear(in_features=4096, out_features=10, bias=True)
Linear(in_features=2048, out_features=10, bias=True)
```

Figure 2. Configuration of the output of different models

For loss function, as cross entropy function is normally used in image classification task, so also use it here. The learning rate is set to 0.001 for the optimizer of each pretrained model.

```
# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer_googleNet = optim.Adam(googleNet.parameters(), lr= 0.001)
optimizer_resNet = optim.Adam(resNet.parameters(), lr= 0.001)
optimizer_AlexNet = optim.Adam(alexNet.parameters(), lr= 0.001)
```

Figure 3. Set up of loss function and hyperparameters

The training process of the model is built with the support of ChatGPT.

The training epochs of each model is set to 40, as CIFAR-10 is a small dataset, it's enough.

For experiment tracking, the average loss during the epoch is recorded and printed out.

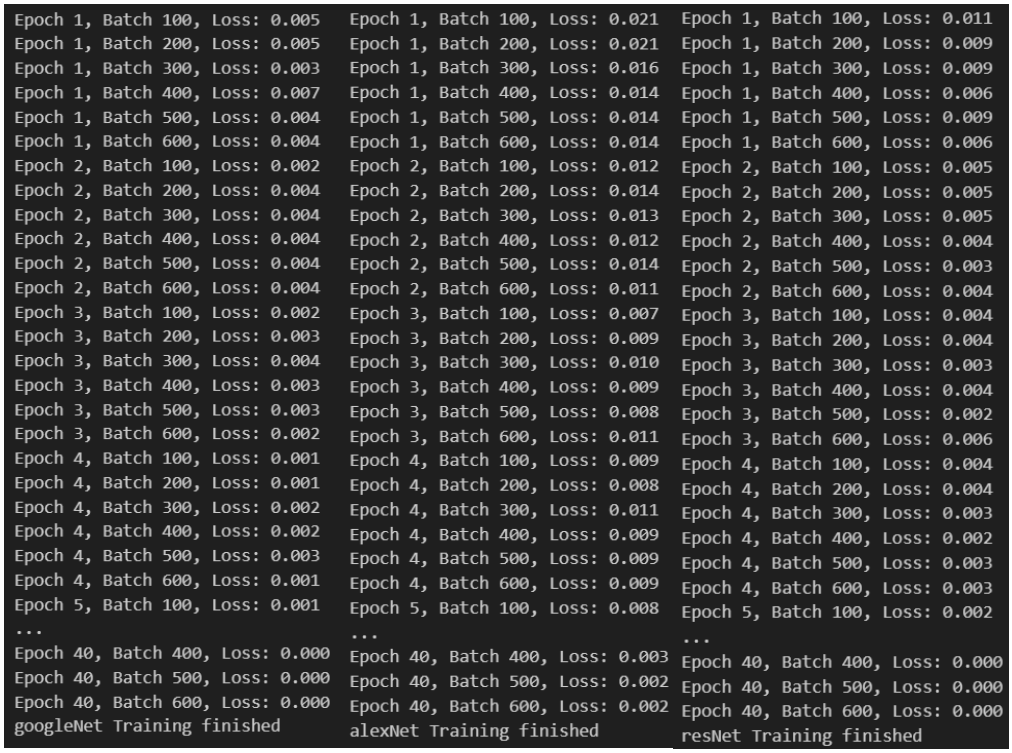

Figure 4. The training process with loss value recorded

3.Evaluation

The evaluation of the training effect of the model can be judged in conjunction with the changes in the loss values from the training in the previous section. In the Training section, it can be seen that the loss values of all three models gradually decrease with the number of training sessions, among which the loss value of AlexNet decreases sharply in the early stage of training, which indicates that the training is effective. However, as the training progresses, the decrease of the loss value gradually becomes slower, and finally only GoogleNet and ResNet50 reach an average loss value of 0.00 in the training set.

The performance of the model after fitting to the dataset is evaluated by accuracy, F1-score, Precision Score, Recall Score and Confusion Matrix with the inference results.

Table 1: **Evaluation of GoogleNet, AlexNet, ResNet50 after training**

Model	**Accuracy**	**F1-Score**	**Precision Score**	**Recall Score**
GoogleNet	91.55%	0.7071	0.84375	0.8333
AlexNet	75.14%	0.6063	0.75	0.8148
ResNet50	88.06%	0.75	0.875	0.875

In Table1, it can be seen that ResNet50 is slightly less accurate than GoogleNet, while AlexNet's accuracy is pulled apart. Despite this, we cannot conclude that GoogleNet outperforms ResNet50, their accuracy is not significantly different and the F1-Score, Precision Score and Recall Score values are better than GoogleNet, while looking at AlexNet's data, the results of each aspect are are significantly worse than the other two Models.

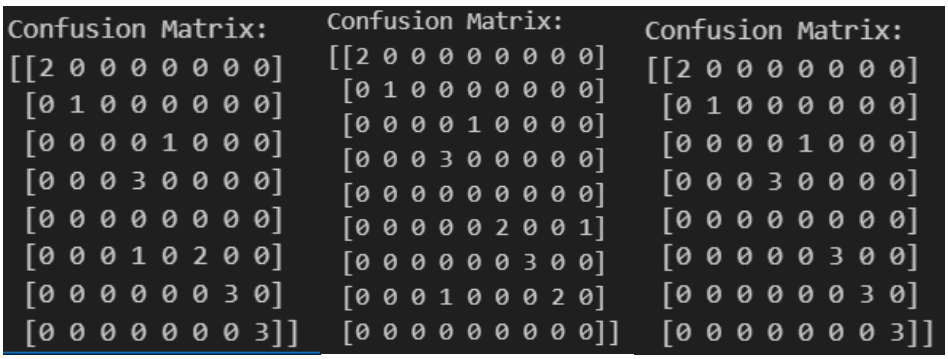

Figure 5. Confusion matrix of GoogleNet, AlexNet and ResNet50

Looking at the confusion matrices of the three models, the confusion matrix of ResNet50 has better non-zero values outside the off-diagonal compared to GoogleNet and AlexNet, which means that ResNet50 will be better at classifying specific categories.

4. Conclusion

Thus combining the above results, ResNet50 will perform better relative to GoogleNet and AlexNet in several ways, especially in terms of focusing on the detailed performance of each category and the balance between precision and recall. This may be important for some tasks.

The project on the comparison of this image classification model served as a baseline project for learning purposes and allowed me to carry out replication and exploration of different image recognition models, allowing me to learn many valuable lessons. Although the post-training evaluation did not always achieve the desired results, the process enhanced my knowledge and skills in the field of image recognition and benefited me greatly.

References

1. Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
2. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).
3. He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.