National Tsing Hua University
Fall 2023 11210IPT 553000
Deep Learning in Biomedical Optical Imaging
Homework 4

CHIH-YING,LIEN[1]

[1] *Institute of Photonics Technologies, National Tsing Hua University, Hsinchu 30013, Taiwan*
Student ID:110066520

1. Introduction

In this report, we will learn Transfer Learning using pre-trained models because they have been trained on large-scale datasets, acquiring valuable general features and knowledge. These features and knowledge can be used to enhance the training of new tasks, enabling us to improve training performance and generalization by existing data resources.

2. Task A: Model Selection

In this report, I choose Resnet-18 and GoogLeNet pre-trained models on our 2 chest x-ray dataset.

Resnet-18 is known for its residual connections, in each residual block, there is a skip connection, also known as an identity mapping, which allows the neural network to easily learn the identity transformation, enabling it to overcome the issue of vanishing gradients during backpropagation. This problem occurs when gradients gradually diminish, making it challenging to effectively update the network's weights. Number 18 means that it consists of 18 layers, which is relatively simple, in our task applying 30 epochs, the computation time is around 3 mins and 14 second.

GoogLeNet, also known as the Inception architecture, employs a feature extraction approach in its Inception module. This module simultaneously utilizes multiple convolutional filters of various sizes within the same layer. It includes 3x3, 5x5, and 1x1 convolutional filters, and then combines their outputs. This allows the model to capture features at different scales without relying solely on a single convolutional filter. In the context of processing chest X-ray images, where anomalies can range from small spots to larger opacities, the use of multi-scale feature extraction enables the model to better comprehend and distinguish these varying-sized abnormalities. While in our task also applying 30 epochs, the computation time is around 3 mins and 55 second.

3. Task B: *Fine-tuning the ConvNet*

In the task B, I conducted six experiments each with ResNet18 and GoogLeNet. Fig. 1 illustrates the model accuracy when using ResNet18, while Fig.2 represents the model accuracy when using GoogLeNet. The results show that when employing ResNet18, the training accuracy rapidly converges to 100%, and the validation accuracy also converges to approximately 98-99%. The reason I didn’t show model loss is because both the training and validation sets remains low when using ResNet18, and the test accuracy is around 70-76%. When using GoogLeNet, the training accuracy exhibits a similar rapid convergence to 100%, and the validation accuracy also stabilizes at around 98-99%. Notably, the test accuracy consistently performs better, reaching approximately 79-80%, which is a significant improvement compared to ResNet18.

I think these results to the architecture of GoogLeNet, which appears to have multiple convolutional filters of different sizes to capture multi-scale image features. This approach is better suited for our task of classifying pneumonia, as it can effectively recognize various physiological structures or anomalies of X-ray images at different scales.

Additionally, I would like to emphasize that when using pre-trained ResNet18 and GoogLeNet models, achieving 100% train accuracy early in the training process is because these models have been pre-trained on a large dataset ImageNet-1K. When fine-tuned for our specific task, they require only minor adjustments to their weights to converge quickly. Their pre-trained features serve as a strong foundation for learning the nuances of our task.

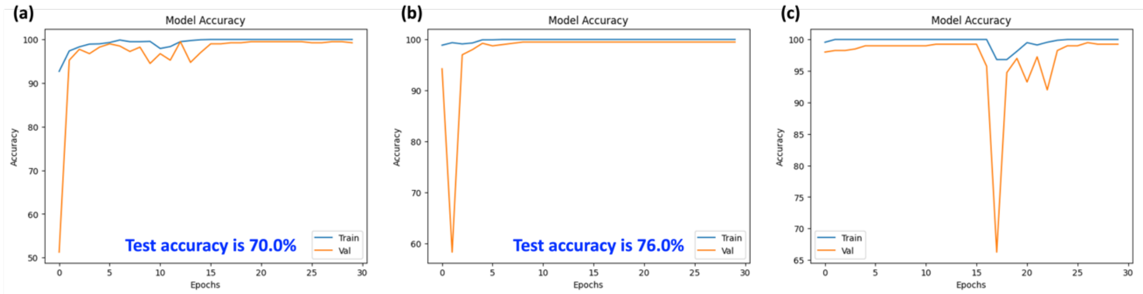

Fig. 1 Model accuracy when using ResNet18

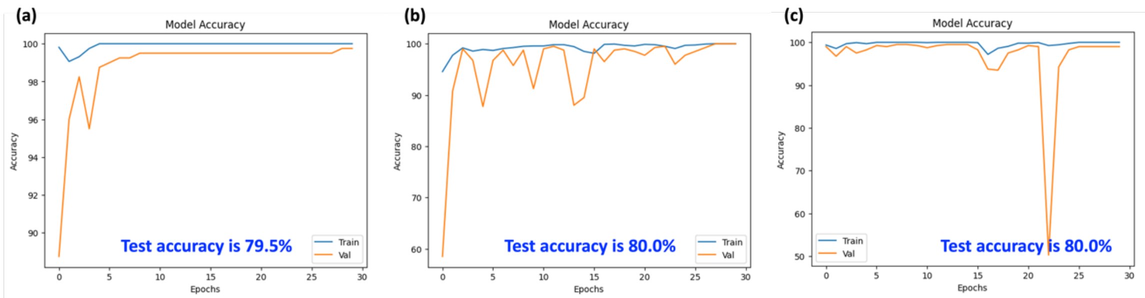

Fig. 2 Model accuracy when using GoogLeNet

4. Task C: *ConvNet as Fixed Feature Extractor*

In the task C, I also conducted six experiments each with ResNet18 and GoogLeNet with freezing fixed feature extractors, shown in Fig. 3 and Fig. 4, respectively. Both models exhibited high train and validation accuracy right from the start, reaching approximately 93% or higher. This is because they directly utilized the pre-trained ResNet18 and GoogLeNet models from ImageNet-1K, with the exception of the final fully connected layer. For test accuracy, the six experiments with ResNet18 consistently yield around 83%, while GoogLeNet performs even better, with test accuracy ranging from 86% to 88%. These results represent a significant improvement. I will further analyze these outcomes in Task D.

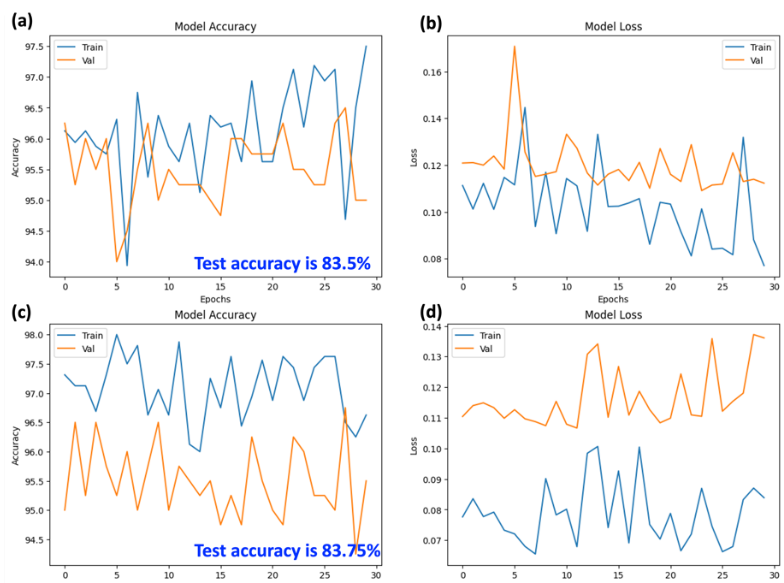

Fig. 3 Model accuracy and loss when using ResNet18 with freezing fixed feature extractors

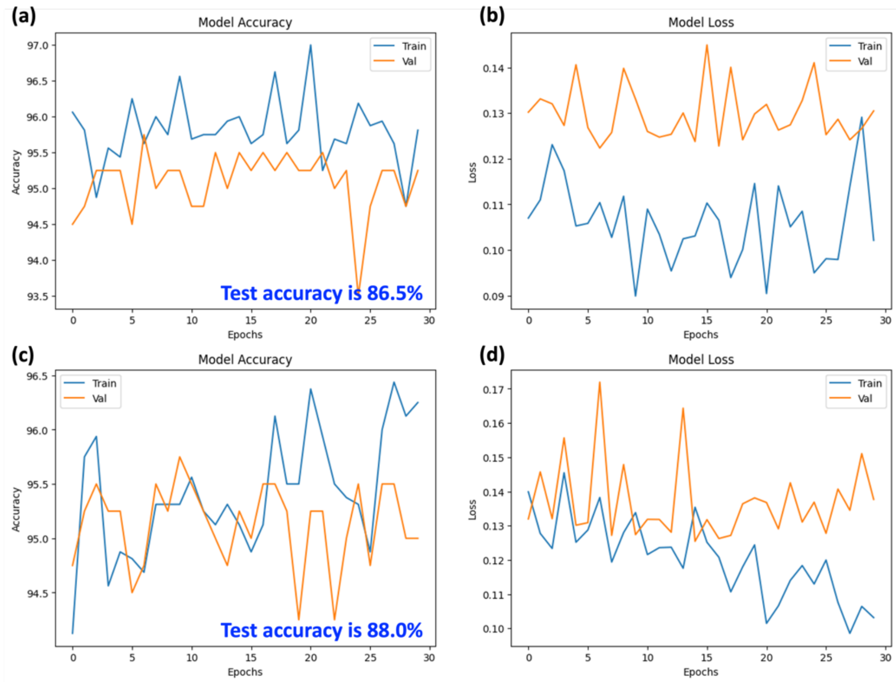

Fig. 4 Model accuracy and loss when using ResNet18 with freezing fixed feature extractors

5. Task D: *Comparison and Analysis*

In the task D, I will further compare the ResNet18 and GoogLeNet with and without the freezing fixed feature extractors. First, we can find that in task B without the freezing fixed feature extractors, both training accuracy exhibits a rapid convergence to 100%; however, in the task C with the freezing fixed feature extractors, the train and validation accuracy is remain around 94-96% for both cases; this is because that we depends on the pre-trained weight except the final fully connected layer; therefore, the earned features are retained in the lower layers of the network and only the final fully connected layer is the only part that is adapted to our specific task.

Second, the test accuracy for both model with the freezing fixed feature extractors results shows a significant improvement. I believe this is because it helps prevent the models from overfitting. Both pre-trained models, trained on the large ImageNet-1K dataset, already possess a strong foundation. When they are fine-tuned on our specific training and validation datasets, they adapt their weights to better align with our task. However, when faced with test datasets that may differ slightly from our training and validation datasets, they may struggle to generalize effectively. Regarding why I think that test datasets that may differ slightly from our training and validation datasets may explain in the task E.

6. Task E: Test Dataset Analysis

6.1 Introduction

In recent homework, I've noticed that regardless of how we optimize the model's training and validation accuracy and loss, the test accuracy remains consistently low, staying below 85%. From my perspective, I believe this is due to slight differences between the test dataset and our training and validation datasets. Therefore, I think the most effective way to solve this issue is to find a method to enhance the model's ability to generalize during training. To prove my point of view, in this task, I will be using data augmentation to enhance the test accuracy.

6.2 Results and Analysis

To add the data augmentation, I applied a series of image transformations to the data using the code "transform", shown in Fig. 5, these transformations are as follows: first transformation is that randomly resizes and crops the images to a size of 224x224 pixels, second transformation is that randomly rotates the images by up to 10 degrees in either direction, and the third transformation is to add random variations to the brightness and contrast of the images. It can be useful for simulating different lighting conditions and enhancing the model's ability to handle variations in image quality.

```
transform = transforms.Compose([
    transforms.RandomResizedCrop(224),
    transforms.RandomRotation(10),
    transforms.ColorJitter(brightness=0.2, contrast=0.2),
])
```

Fig. 5 The code of "transform" to do the data augmentation

I conducted three experiments each with ResNet18 and GoogLeNet, additionally, I tested ResNet18 and GoogLeNet with fixed feature extractors frozen, shown in Fig. 6 and Fig.7, respectively. Although there was a slight decrease in both training and validation accuracy in all scenarios, the test accuracy generally improved.

Notably, when employing ResNet18 and GoogLeNet without freezing the fixed feature extractors, the test accuracy saw a significant boost. For ResNet18, the test accuracy increased from the initial 70-76% to a range of 82-84%. Similarly, GoogLeNet's test accuracy improved from 79-80% to 80-82% when data augmentation was applied. On the other hand, for ResNet18 and GoogLeNet with freeze fixed feature extractors, the test accuracy has improvement of approximately 1%. This outcome is reasonable, given that models with freeze fixed feature extractors primarily rely on pre-trained model weights. Consequently, the impact of data augmentation is somewhat limited in this case.

Therefore, we can utilize data augmentation to enhance the model's generalization capabilities and prevent the overfitting. My results demonstrate that regardless of the model used, there is an improvement in test accuracy. This highlights the test dataset need to have more effective model training to achieve better generalization.

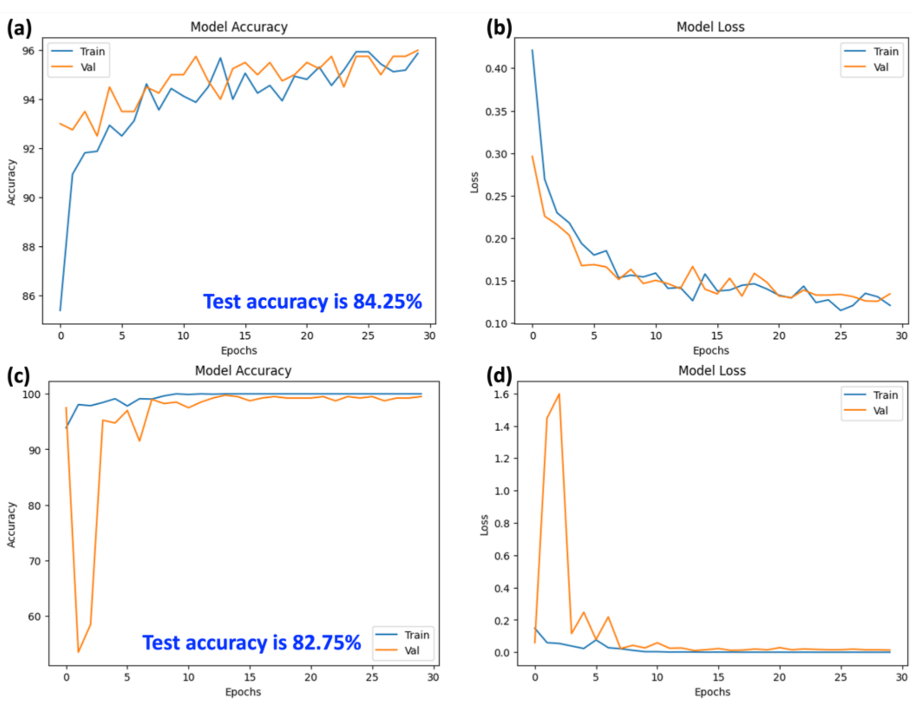

Fig. 6 Model accuracy and loss when using data augmentation (a) Resnet-18 (b) GoogLeNet

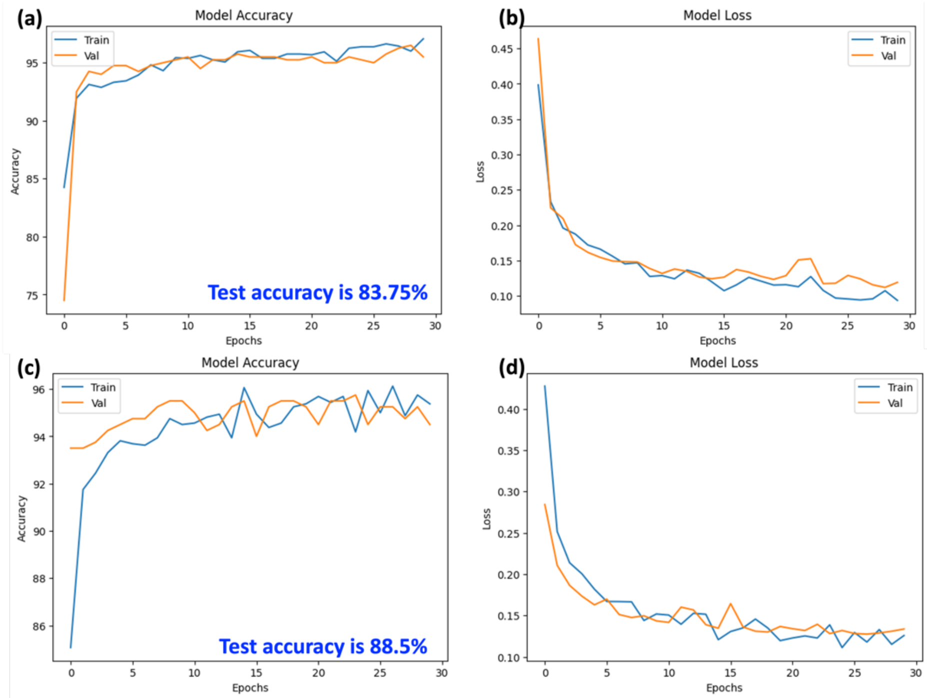

Fig. 7 Model accuracy and loss when using data augmentation (a) Resnet-18 (b) GoogLeNet freezing fixed feature extractors

7. Conclusion

In this report, I used pre-trained models Resnet-18 and GoogLeNet, which have been trained on large-scale datasets, acquiring valuable general features and knowledge. In our results, I first found that the performance of GoogLeNet is better than Resnet-18, I think it is because GoogLeNet appears to capture multi-scale image features, which is better suited for our task of classifying pneumonia. Furthermore, we have tried to freeze the fixed feature extractors expect final layer for both models, and the results that the test accuracy become higher, I think this is because it helps prevent the models from overfitting. This assumption is because I think that test datasets that may differ slightly from our training and validation, pre-trained models without the freezing the fixed feature will fine-tuned on our specific training and validation datasets caused overfitting. The last, I tried to add data augmentation to enhance the test accuracy by generalizing the dataset during training, the results all show that the every test accuracy have improved.