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

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 explored the field of medical image analysis by classifying histological images of cancerous tissues. The dataset is a 150x150 pixel RGB image, representing six distinctive tissue textures, namely Tumor, Stroma, Complex, Lympho, Debris, and Mucosa, shown in Fig. 1. To achieve this classification task, I employed the pre-trained ResNet18 model from ImageNet-1K, training it to effectively recognize and differentiate between these varied textures.

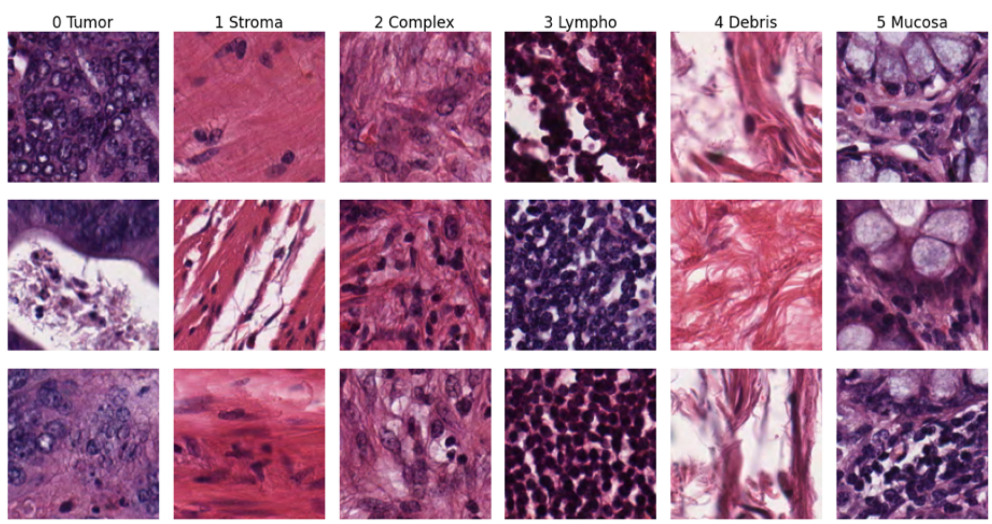

Fig.1 The dataset images of six distinctive tissue textures

2. Discussion and Analysis

I utilized a pre-trained ResNet-18 model from ImageNet-1K, employing the Adam optimizer with a learning rate set to 0.001 and default weight decay. Additionally, I implemented the CosineAnnealingLR learning schedule with an eta_min value of 1e-6. Fig. 2 (a) and (b) show the model accuracy and loss during the fine-tuning of ResNet-18. The training accuracy rapidly converged to 100% by the 5th epoch. However, the validation accuracy plateaued at nearly 93% from the 10th epoch, while the test accuracy reached 93.67%. Therefore, I will start to overcome the problem of overfitting.

In a previous experiment during HW4, I concluded that employing the Fixed Feature Extractor of ResNet-18 yielded better test accuracy than fine-tuning one. Fig. 2 (c) and (d) show the model accuracy and loss of ResNet-18 with the Fixed Feature Extractor. In this case, the train accuracy plateaued at 90%, with the validation accuracy maintaining at nearly 83%. This observation shows that the pre-trained ResNet-18 model from ImageNet-1K may struggle to generalize effectively to our dataset.

To overcome the issue of overfitting, I attempted to fine-tune hyperparameters such as weight decay, learning rate, and eta_min. Moreover, I added the dropout, Batch Normalization, and ReLU in the fully connected layer of ResNet-18. However, there was no significant improvement. Therefore, I think that adjusting the dataset through data augmentation to achieving generalization across diverse features is the way of addressing the issue of overfitting.

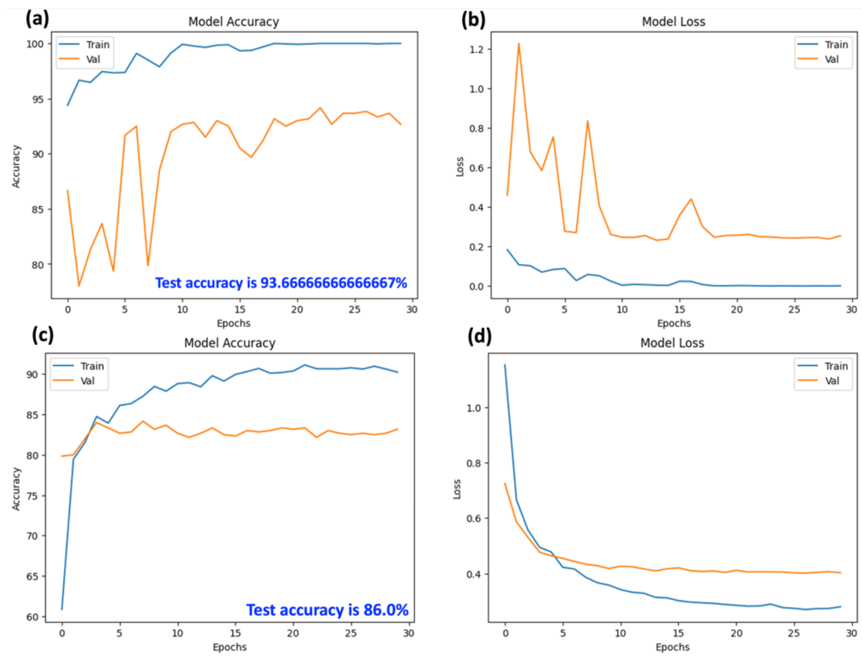

Fig.2 (a) and (b) the model accuracy and loss of fine-tuning pre-trained ResNet-18 model from ImageNet-1K. Fig.2 (c) and (d) the model accuracy and loss of pre-trained ResNet-18 model from ImageNet-1K with the Fixed Feature Extractor.

Fig.3 (a) and (b) illustrate the model accuracy and loss of adding the data augmentation. The code of “transform” is shown in Fig. 4, these transforms are as follows: the first transforms: RandAugment is a form of data augmentation that randomly applies a series of augmentations to the images in the dataset, which means that the specific augmentations and their magnitudes are randomly selected. The second transforms: RandomRotation is that randomly rotates the images by up to 10 degrees in either direction. The third transforms: RandomHorizontalFlip is that randomly flips the image horizontally with a 50% probability. The fourth transforms: ColorJitter is used to randomly adjust the brightness, contrast, saturation, and hue of the image. The specified values indicate the range of possible adjustments. Final ToTensor is to convert the image to a PyTorch tensor. With adding these augmentations, the train accuracy reached 100%, with the validation accuracy maintaining at nearly 88%, and the test accuracy is 90.33% which is worse than without adding the transforms.

I think the result might be due to the overfitting of data augmentation. When too many random transforms are applied, the model may start learning noise in the training data rather than features conducive to generalization, resulting in a decline in performance on the test data. Moreover, excessive color augmentation in certain cases might make it difficult for the model to distinguish between different tissue textures. Therefore, I decided to step by step testing the different transforms to determine which ones are more effective for this dataset and model.

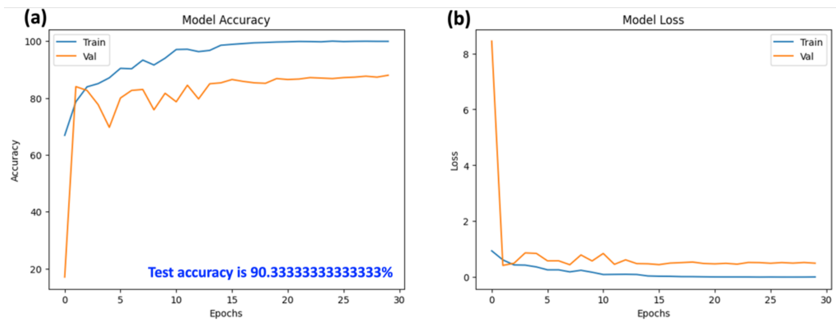

Fig.3 (a) and (b) the model accuracy and loss of fine-tuning pre-trained ResNet-18 model adding data augmentation

```
transforms.RandAugment(),
transforms.RandomRotation(10),
transforms.RandomHorizontalFlip(),
transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.2),
transforms.ToTensor(),
```

Fig.4 the code of the data augmentation

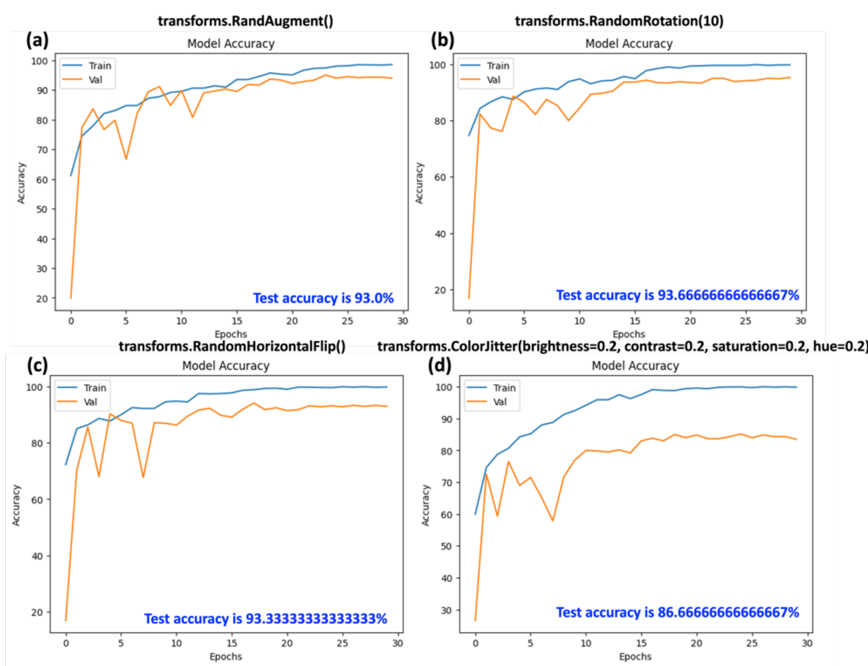

Fig.5 (a) to (d) the model accuracy of fine-tuning pre-trained ResNet-18 model adding data augmentation of transforms.RandAugment, transforms.RandomRotation, transforms.RandomHorizontalFlip and transforms.ColorJitter, respectively.

Fig. 5 show the model accuracy with of fine-tuning pre-trained ResNet-18 model with adding different transforms. The Fig.5 (d) shows the result of adding the ColorJitter, the train accuracy plateaued at 99.88% with the validation accuracy maintaining at nearly 83.5%. The test accuracy, however, was observed to be 86.66%, marking the lowest performance compared to other transforms. Therefore, I also adjusted ColorJitter parameters, specifically setting all values to 0.1 and 0.5, resulting in accuracies of 86.66% and 90.33%, respectively. In this dataset shown in Fig.1, color augmentation may result in the model struggling to differentiate between different textures of tissues, which may impact the performance of the model on test data.

However, we can also observe variations in orientation in the dataset. From Fig.5 (b), it is apparent that RandomRotation in 10 degrees yields the highest test accuracy, and its validation accuracy can reach 95.33%. Consequently, I think that rotating the photos stands out as the most effective data augmentation technique for this particular dataset. Fig. 6 shows the model accuracy and loss of fine-tuning pre-trained ResNet-18 model with adding RandomRotation in 5 degrees. The test accuracy experiences a notable improvement, surpassing 94%.

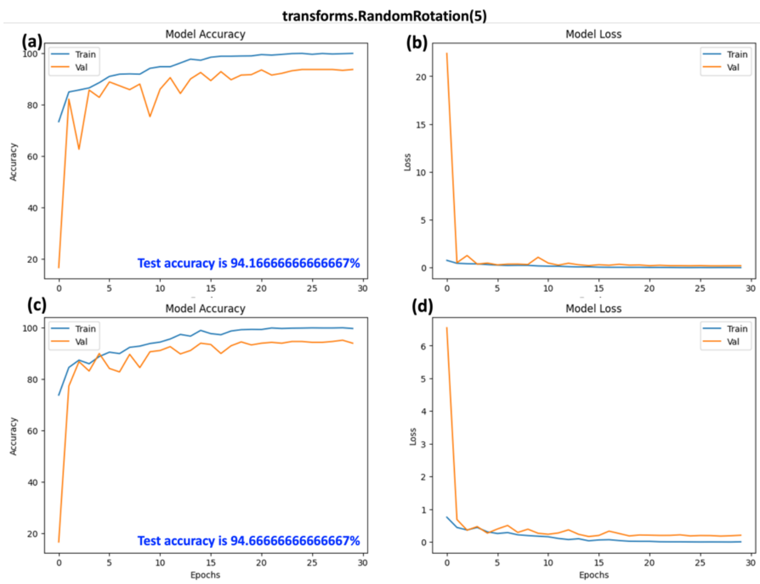

Fig.6 (a), (c) and (b), (d) the model accuracy and loss of fine-tuning pre-trained ResNet-18 model adding data augmentation of transforms.RandomRotation(5)

3. Conclusion

Using the pre-trained ResNet-18 model from ImageNet-1K yields impressive results, test accuracy reaching 93%, without tuning the hyperparameters. However, when analyzing the training and validation accuracy, it becomes apparent that the model is overfitting. Therefore, I attempted to adjust hyperparameters and the results showed no improvement. Consequently, I applied transforms for data augmentation on the dataset. I discovered that adding random transforms might not be sufficient to enable the model learning additional distinctive features and cause the overfitting. The use of inappropriate transforms may also make it challenging for the model to differentiate between various textures, such as when incorporating color variations in this dataset. In conclusion, I found that employing a 5-degree RandomRotation enhances the model's training performance, resulting in a test accuracy exceeding 94%.