2023 Fall NYCU-EE Deep Learning – Lab02: Image Classification

Name: Kuan-Wei Chen 陳冠瑋 **ID: 310510221 / Institute of Electronics / Oct. 27, 2023**

***Abstract*- In this lab, we conducted food classification with four categories: baked potato, crispy chicken, donut, and fries. In Task 1, I achieved an accuracy of 83.64% (no pre-trained) and 89.70% (pre-trained) using a standard ResNet18. In Task 2, I compared various CNN and Vision Transformer (ViT) models, applying data augmentation, K-Fold cross-validation, dropout, and optimizing with SGD (including momentum and weight decay) to improve model accuracy. The selected ViT-B model achieved an impressive 97.71% accuracy, marking an improvement of 14.07% over ResNet18 (no pre-trained), 8.01% over ResNet18 (pre-trained), and 4.85% over ConvNeXt-L (pre-trained).**

I. Introduction

The objective of this lab is to implement an image classification neural network, such as ResNet, using a food dataset consisting of four categories: baked potato, crispy chicken, donut, and fries, totaling 1646 images. The lab comprises two tasks. Task one is to train ResNet18 to achieve an accuracy of over 75% and compare pre-trained and non-pre-trained models. Task two involves training a model to maximize accuracy and visualizing the learning rate schedule, weights, and gradients using TensorBoard.

Fig 1. Food dataset

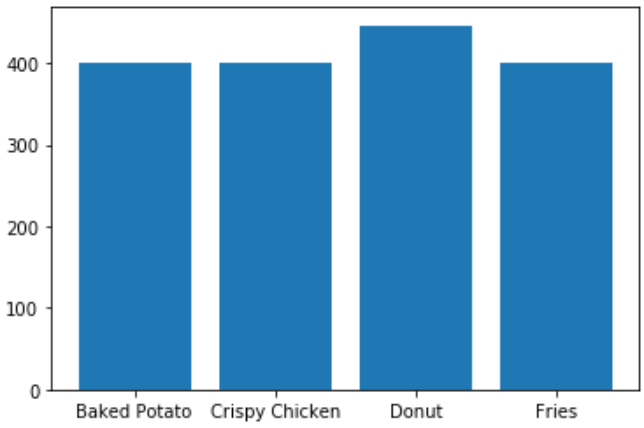

Fig 2. Data distribution

II. Methodology

A. *Residual Network*

Residual Network (ResNet) is a deep neural network architecture that introduced the concept of residual learning. Its primary innovation lies in the utilization of residual blocks, which enable neural networks to become significantly deeper while addressing the vanishing gradient problem.

B. *ConvNeXt*

ConvNeXt is a neural network architecture designed for image classification, known for its hierarchical, multi-path convolutional structure that efficiently extracts accurate features from images.

C. *Vision Transformer*

ViT is a neural network that adapts the transformer, originally for natural language, to image classification. Its key contribution is showing the power of self-attention in computer vision, surpassing traditional convolutional networks, and emphasizing attention-based models in visual recognition.

III. Optimization Method

A. *Pre-trained model*

Pre-trained models are a valuable starting point for machine learning tasks, leveraging extensive dataset knowledge for faster training, and strong generalization even with limited data. The following is the comparison of my results between ResNet18 without pre-trained weights and ResNet18 with pre-trained weights.

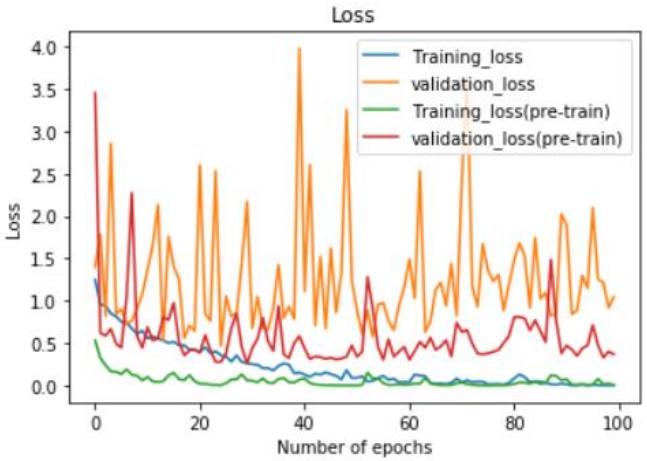

Fig 3. Training and validation loss

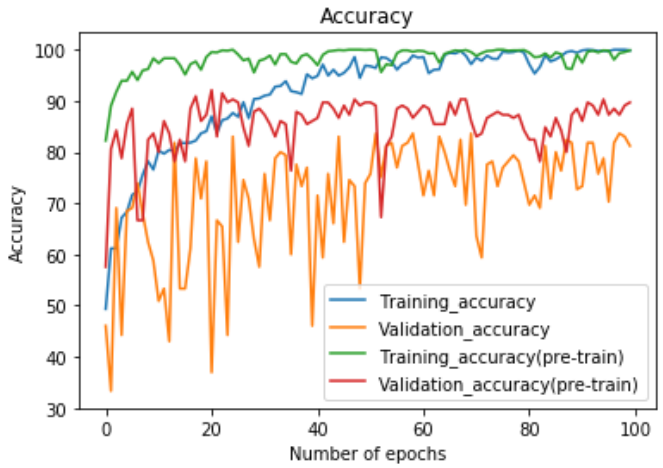

Fig 4. Training and validation accuracy

```
-------------------------------------
epoch: 99/100
Train Acc: 100.0000%, Train Loss: 0.0024
Valid Acc: 83.0303%, Valid Loss: 0.9190
-------------------------------------
epoch: 100/100
Train Acc: 99.8650%, Train Loss: 0.0061
Valid Acc: 81.2121%, Valid Loss: 1.0457
-------------------------------------
```

```
1 print(f"Best Validation Acc: {acc_best.cpu():.2f}%")
```

```
Best Validation Acc: 83.64%
```

Fig 5. Screenshot of validation accuracy of ResNet18 (w/o pre-trained)

```
-------------------------------------
epoch: 99/100
Train Acc: 99.5273%, Train Loss: 0.0212
Valid Acc: 89.0909%, Valid Loss: 0.3996
model saved
-------------------------------------
epoch: 100/100
Train Acc: 99.8650%, Train Loss: 0.0035
Valid Acc: 89.6970%, Valid Loss: 0.3686
model saved
-------------------------------------
```

```
1 print(f"Best Validation Acc (pre-train): {pre_acc_best.cpu():.2f}%")
```

```
Best Validation Acc (pre-train): 89.70%
```

Fig 6. Screenshot of validation accuracy of ResNet18 (w/ pre-trained)

Table1. Pre-trained model

Method	Acc.
ResNet18 (w/o pre-trained)	83.64 %
ResNet18 (w/ pre-trained)	89.70 %

B. *K-Fold Cross-Validation*

I used K-Fold cross-validation with K=5 due to initial model overfitting, where the training loss converged rapidly to near zero, hindering effective updates and resulting in poor testing performance.

Table2. K-fold cross-validation (K=5)

Method	Acc.
ResNet18 (w/o K-fold)	89.70 %
ResNet18 (w/ K-fold)	91.14 %

C. *Data Augmentation*

To reduce overfitting, I used data augmentation to diversify images, primarily applying RandomHorizontalFlip. Excessive augmentation techniques like ColorJitter, RandomRotation, RandomVerticalFlip, and Normalize were found to adversely affect model accuracy in predicting test data for this task.

D. *Stochastic Gradient Descent*

In optimizer selection, I discovered that using SGD with momentum=0.8 and weight_decay=1e-5 outperformed the choice of Adam during my experiments. On the ViT-B model, I achieved a 3.43% increase in accuracy.

Table3. Optimizers

Method	Acc.
ViT-B (Adam)	93.14 %
ViT-B (SGD)	96.57 %

E. Dropout

To combat overfitting, I added dropout (p=0.5) to the final Fully Connected Layer, improving testing accuracy. On the ViT-B model, I achieved a 1.14% increase in accuracy.

Table4. Dropout

Method	Acc.
ViT-B (w/o dropout)	96.57 %
ViT-B (w/ dropout)	97.71 %

IV. Experimental Result

In my experiments, I tried various models including ResNet18, ResNet50, MobileNet_v2, ConvNeXt-B, ConvNeXt-L, ViT-B, and ViT-L. Ultimately, I found that models with more parameters were not necessarily better, as overly complex models could lead to overfitting and yield unexpected testing results. Furthermore, I observed that ViT performed exceptionally well in this task, significantly boosting the performance. In the end, I chose ViT-B as my base model and fine-tuned it.

Table5. Experiment Results

Method	Acc.
ResNet18	83.64 %
ResNet18 (pre-trained)	89.70 %
ResNet50	85.43 %
ResNet50 (pre-trained)	86.23 %
MobileNet_v2 (pre-trained)	74.86 %
ConvNeXt-L (pre-trained)	92.86 %
ConvNeXt-B (pre-trained)	91.43 %
ViT-L (pre-trained, Adam)	90.86 %
ViT-B (pre-trained, Adam)	93.14 %
ViT-B (pre-trained, SGD)	96.57 %
ViT-B (pre-trained, SGD, dropout)	97.71 %

Below are the results of visualizing the learning rate schedule, weights, and gradients in TensorBoard. For weight and gradients, I chose to display the last layer of the fully connected layer.

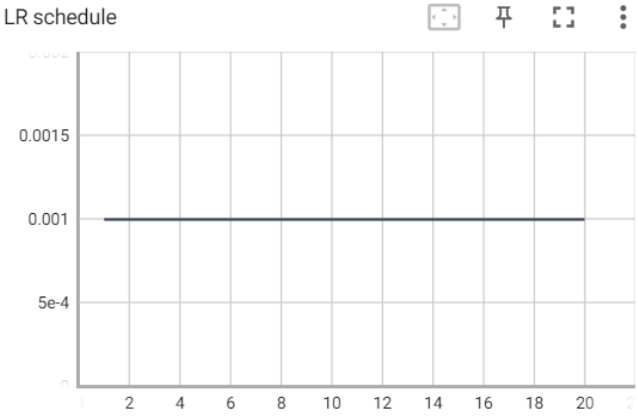

Fig 6. TensorBoard – Learning Rate Schedule

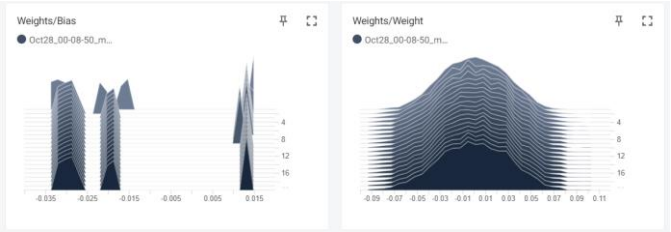

Fig 7. TensorBoard – Weights and Bias

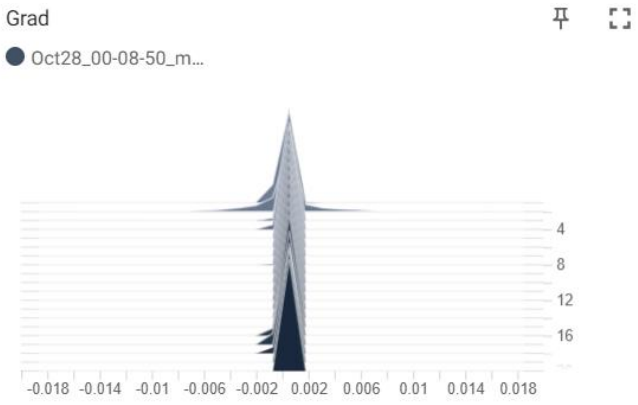

Fig 8. TensorBoard – Gradient

V. Conclusion

This lab involved food classification. In Task 1, I trained a ResNet18 and achieved 83.64% accuracy (without pre-trained) and 89.70% (with pre-trained). In Task 2, various models were compared using many optimization techniques. ViT-B outperformed other models with an impressive accuracy of 97.71%.