NYCU Visual Recognition using Deep Learning
HW1 report

109612019, 林伯偉

Github Link : https://github.com/Albert5865/NYCU-Computer-Vision-2025-Spring

1. Introduction:

This assignment focuses on building an image classification model capable of recognizing 100 object categories. The objective is to preprocess the dataset, design a robust neural network, apply regularization and learning rate scheduling techniques, and evaluate performance on both validation and test datasets. The challenge emphasizes improving generalization and optimizing accuracy under resource constraints.

2. Method:

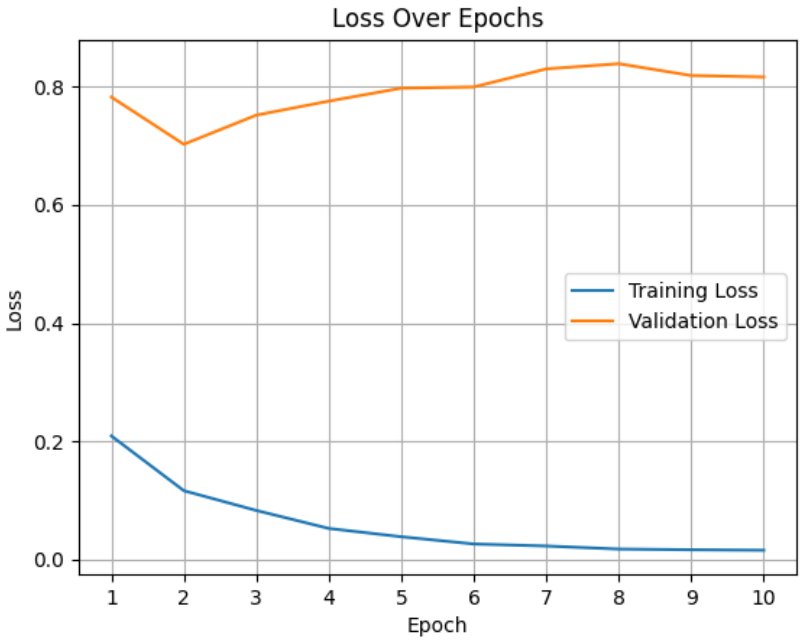
I used the ImageFolder format to load the 100-class dataset, applying standard transformations such as resizing to 224x224, normalization, and Gaussian blur for regularization. For the backbone architecture, I utilized the SEResNeXt101d_32x8d.ah_in1k model from the TIMM library, known for its high performance and efficient channel recalibration.

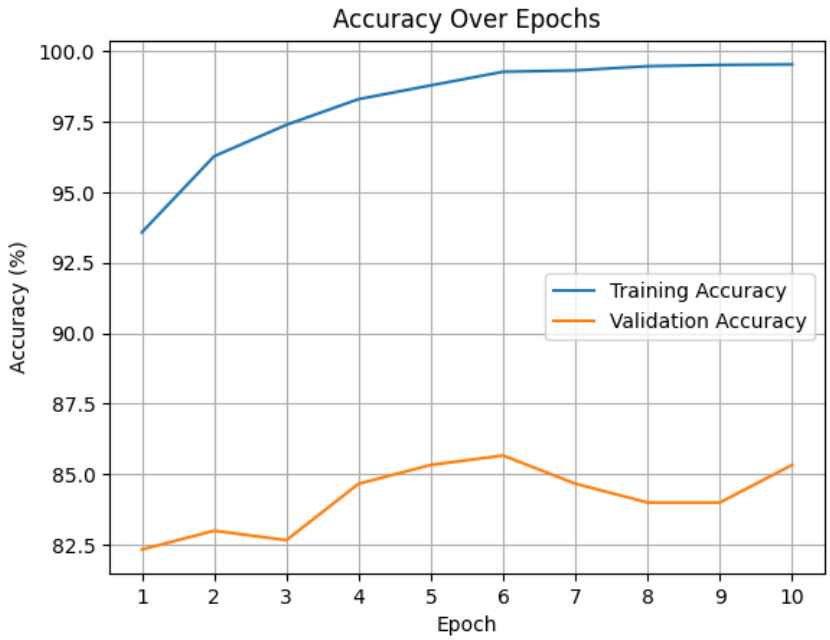
To improve generalization and reduce overfitting, dropout was applied at a rate of 0.5 before the final classification layer. L2 regularization was implemented via weight decay in the AdamW optimizer. Learning rate scheduling was handled using ReduceLROnPlateau, which decays the learning rate when validation loss stagnates.

The training process includes learning rate deacay, which dynamically decreases the learning rate if validation loss did not decrease and validation accuracy did not increase for two consecutive epochs. Training and validation metrics were logged using TensorBoard for visual analysis.

- Model : seresnext101d_32x8d.ah_in1k
- Learning rate : dynamic
- Data augmentation :
  - GaussianBlur (kernel_size=3)
  - RandomHorizontalFlip
  - RandomRotation
- Weight decay : 1e-3
- Drop rate : 0.5
- Epoch : 10

3. Results:





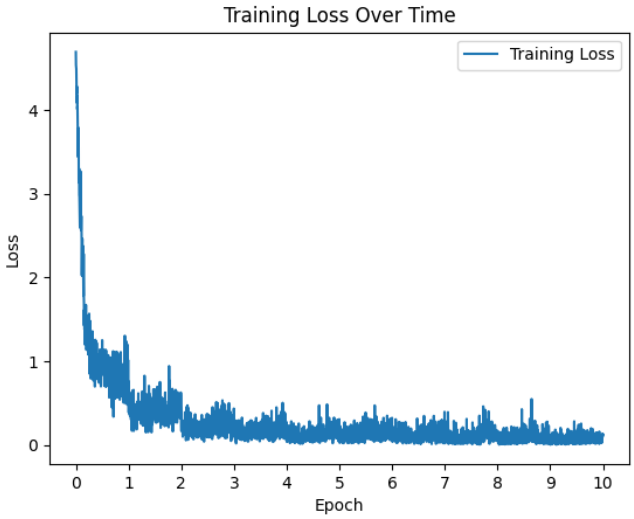
- Best Validation Loss : 0.70277
- Best Validation Accuracy : 85.67%
- Final Test Accuracy (Public) : 91%

This indicates that the SEResNext101d_32x8d.ah_in1k model was able to accurately classify the majority of unseen data points after training. The best validation loss was 85.67%, suggesting that, while the model was able to correctly classify many samples, there were still some challenges in minimizing the loss function, possibly due to overfitting or inherent difficulty in the task.

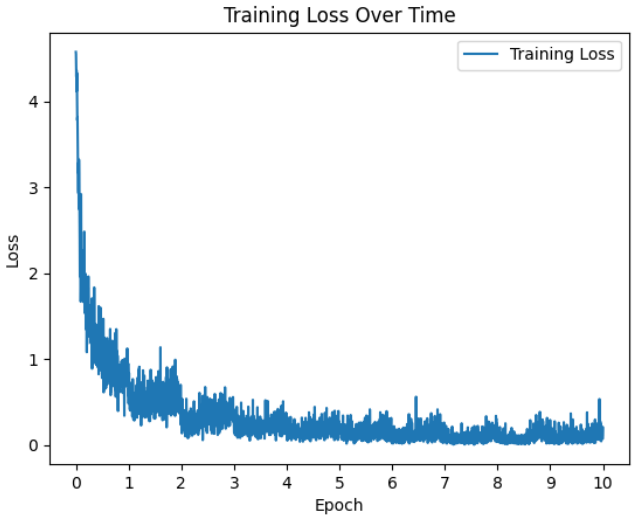
4. Additional experiments:

1. Different model architecture:

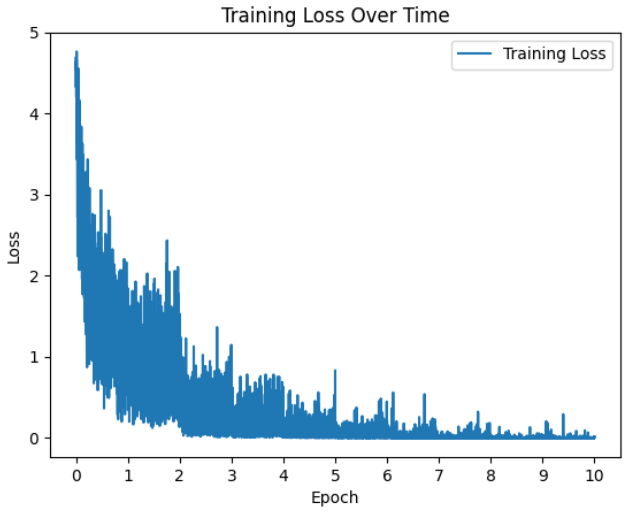
ResNext50 : Accuracy = 0.86



ResNextRS50: Accuracy = 0.9



SE-ResNext101 : Accuracy = 0.91



The ResNext50 model achieved an accuracy of 86% on the validation set, with a learning curve that shows relatively quick convergence. In comparison, the ResNextRS50 architecture performed slightly better, achieving a 90% accuracy. Its training loss curve mirrors that of ResNext50 but demonstrates better convergence, benefiting from the residual structure and refined architecture. The SE-ResNext101 model, also achieving a 91% validation accuracy, by incorporating Squeeze-and-Excitation blocks, which enhanced its ability to learn more informative feature representations. The learning curve for SE-ResNext101 displayed smoother convergence and better control over overfitting, making it the most effective model among the three.

2. Add additional drop out layer :

- **Hypothesis**: Adding dropout would reduce overfitting by preventing the model from relying too heavily on specific neurons.
- **How it works**: Dropout randomly sets a fraction of activations to zero during training, introducing noise and encouraging the network to develop more robust features.

3. Data augmentation - Gaussian Blur, RandomRotation, RandomHorizontalFlip:

- **Hypothesis**: Incorporating multiple augmentation techniques would encourage the model to generalize better by simulating more diverse training conditions.
- **How it works**: Gaussian blur smooths image textures, random rotation alters the orientation, and random horizontal flip simulates mirrored views. These augmentations prevent the model from overfitting to specific spatial patterns.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

4. Implement weight deacy (L2 Regularization) :

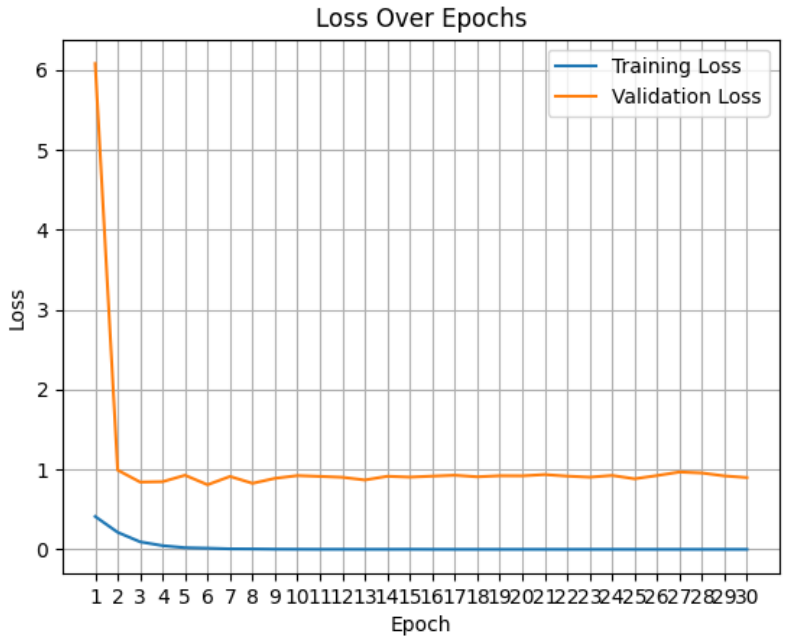
- **Hypothesis**: Applying L2 regularization would discourage the model from learning overly complex or large weights that overfit the training data.
- **How it works**: L2 regularization adds a penalty to the loss based on the magnitude of weights. This constrains the model to learn simpler, more generalizable solutions.

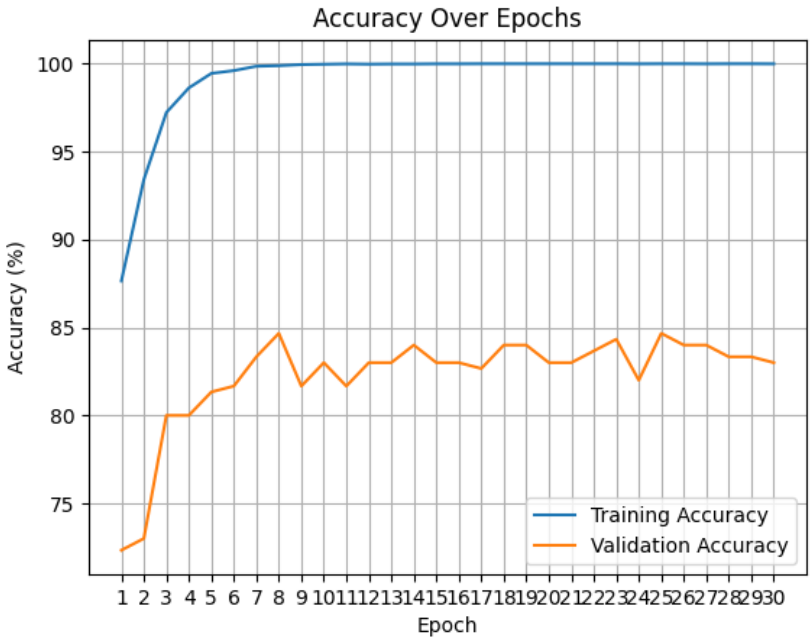
L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

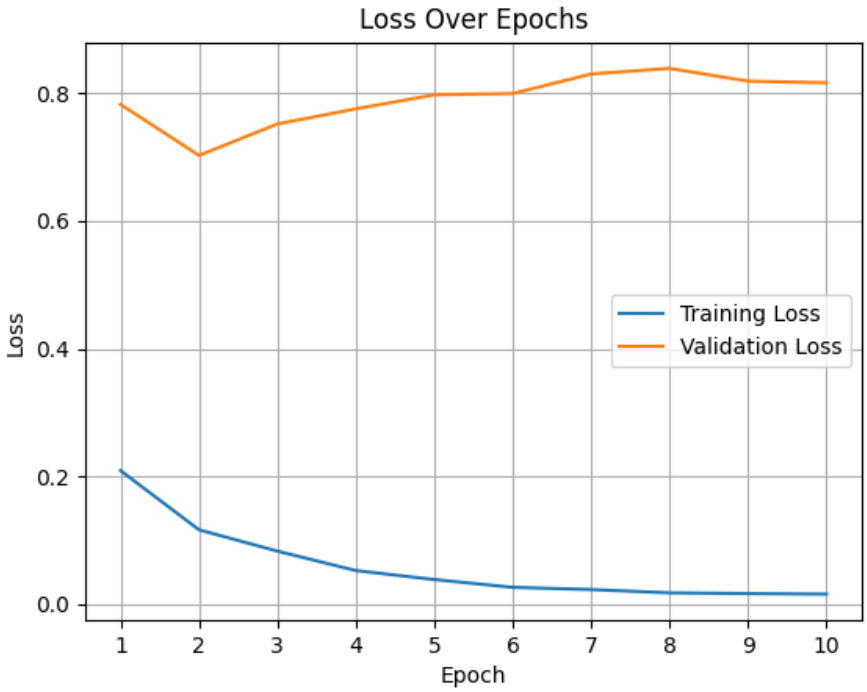
Additional experiment results:

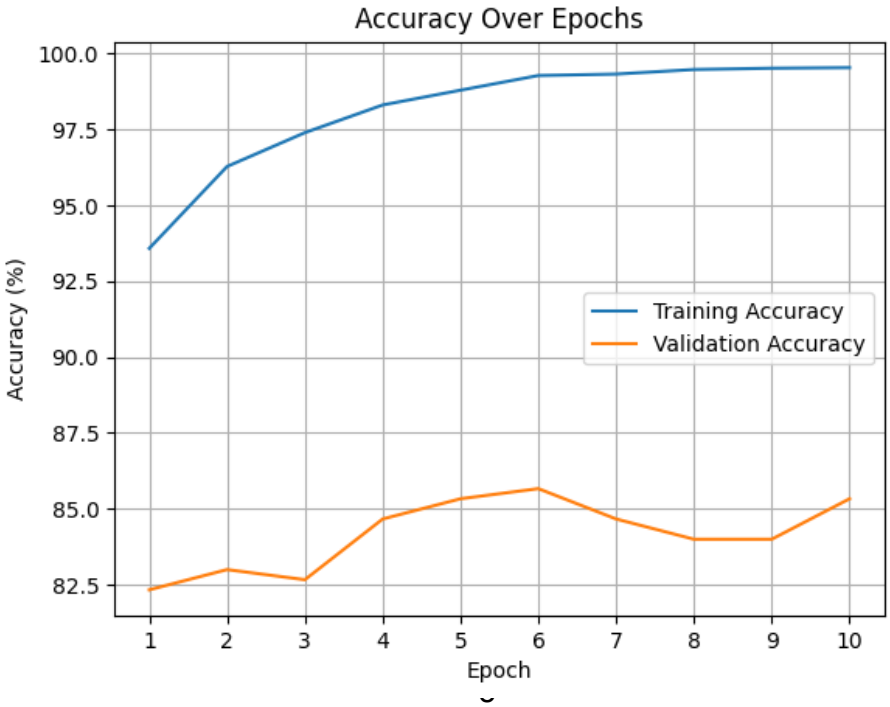
Learning curve using model seresnext101d_32x8d.ah_in1k "without" applying any of the techniques mentioned above reveals a clear and strong indication of overfitting. Specifically, while the model performs well on the training dataset with steadily decreasing training loss and increasing accuracy, the validation loss stagnates, and the validation accuracy remains significantly lower. This divergence between the training and validation metrics suggests that the model is memorizing the training data rather than generalizing to unseen data, which is a classic sign of overfitting





The learning curve obtained using the SEResNext101d_32x8d.ah_in1k model—enhanced "with" all the techniques including dropout, data augmentation (Gaussian blur, random rotation, and horizontal flipping), weight decay, and dynamic learning rate scheduling—demonstrates steady and consistent accuracy improvement across epochs and we can run more epochs to futher enhance the performane of the model. Nonetheless, the plots shows a trend of reduced overfitting, and improved generalization, as compared with the plot from the previos page. This confirms the effectiveness of the applied techniques in enhancing the model's stability and performance





References:

1. https://huggingface.co/timm/seresnext101d_32x8d.ah_in1k
2. https://paperswithcode.com/model/seresnext?variant=seresnext50-32x4d
3. https://arxiv.org/abs/1709.01507v4
4. https://medium.com/analytics-vidhya/l1-vs-l2-regularization-which-is-better-d01068e6658c
5. https://d2l.ai/chapter_convolutional-modern/resnet.html
6. https://blog.csdn.net/weixin_44638957/article/details/101110053