

Project 3

Computer Vision(CSI4116-01)

Spring, 2024

Due 20th June, 23:55

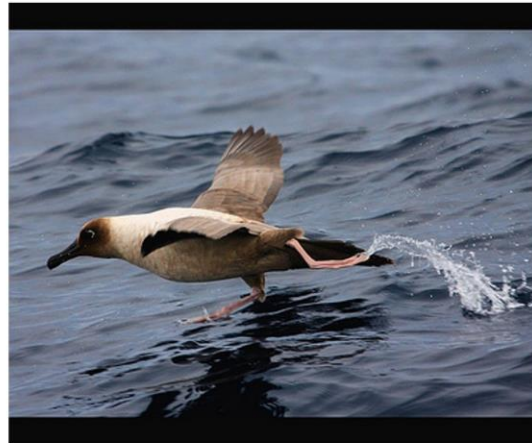
Objective

- Implement deep learning models (ResNet-34) for *bird classification* using PyTorch framework
- Explore data augmentations and hyper-parameter tuning for training
- Explore inference techniques for improving classification accuracy

**Black Footed
Albatross**



**Sooty
Albatross**



?



Summary

<https://kaggle.com/competitions/csi-4116-project-3>

- Please join our Kaggle competition with the above link
 - Change your team name to your student ID on the Team tab
- Submit *test_subm.csv* file with your predicted class labels (*fill in the second column*)
 - The first column, 'filename', should contain the names of each file
 - The second column, 'cls', should contain the predicted class for each file
- How to achieve high accuracy?
 - Find optimal training hyperparameters & strategies
 - Use some inference techniques
 - We provide the codebase for your reference
- Write a report that summarizes your experiments
 - in 4 pages – no cover page
 - Write name and student ID on the top-right of your report

*** There is no interview for this project!

*** We provide codebase for your reference!

Inference Technique

After training your model, you can try some inference techniques to improve the accuracy. Here, we provide some of them, but you can try others.

- Multiple crops at test-time (Multi-view inference)
 - Taking multiple crops of image – e.g., top-left corner, bottom-right corner, and center – then, averaging the output probability

<https://pytorch.org/vision/main/generated/torchvision.transforms.FiveCrop.html>

- Ensemble (based on the multiple models)
 - Hard-voting VS Soft-voting from the multiple models
 - Training multiple models (using the same model architecture) with different random seed or training configurations
 - Please read some parts of paper for the theoretical details:

<https://openreview.net/forum?id=Uuf2q9TfXGA>

Rules

- **20 submissions** are allowed for the competition
 - **-5 points** for extra one submission
 - E.g., if you attempted 22 submissions, -10 points
- Must use **ImageNet-1k pretrained** weights of **ResNet-34** which is available from torchvision
 - Other than the backbone network, you can flexibly add your customized layers and classifier
 - ResNet-34 backbone parameters: 21284672 (~21M)
 - Allowed maximum number of model parameters: 30M

Report (1)

[Model Details]

- Extra layers or classifier (including normalization layer and activation functions)
 - Dimension and other hyperparameters (e.g., not using bias for FC layer)
 - Please do not include unrelated arguments, such as inplace=True
- Number of your model parameters

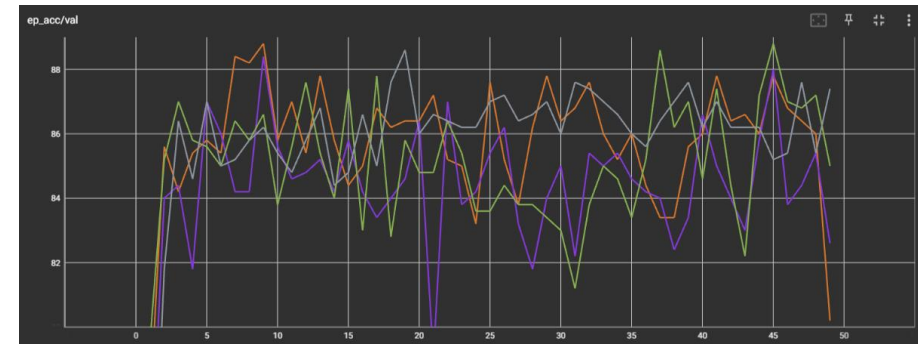
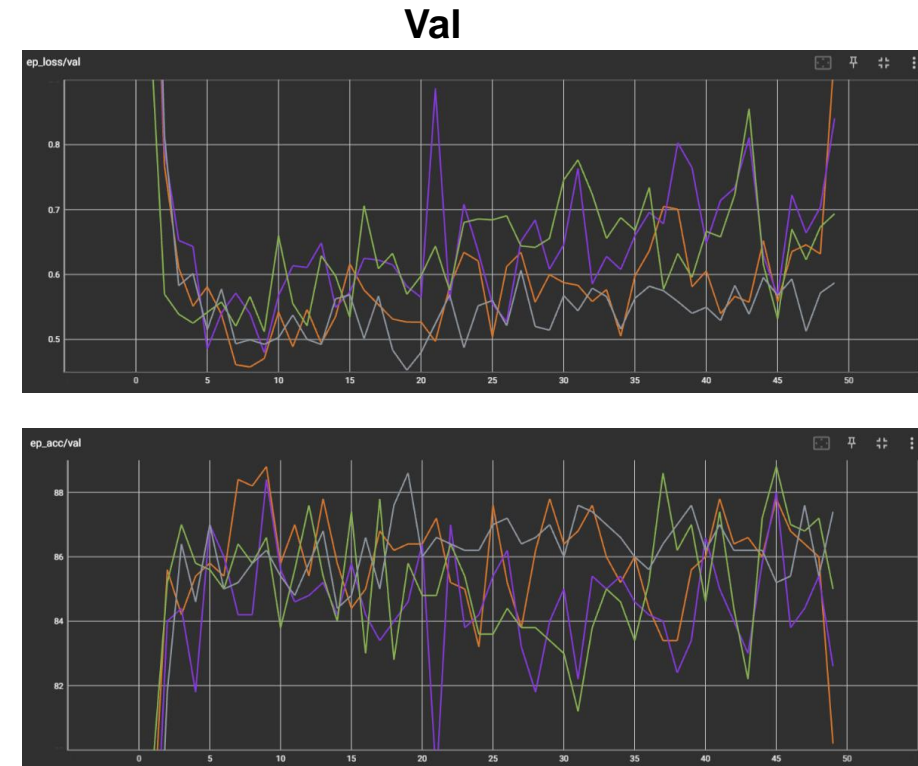
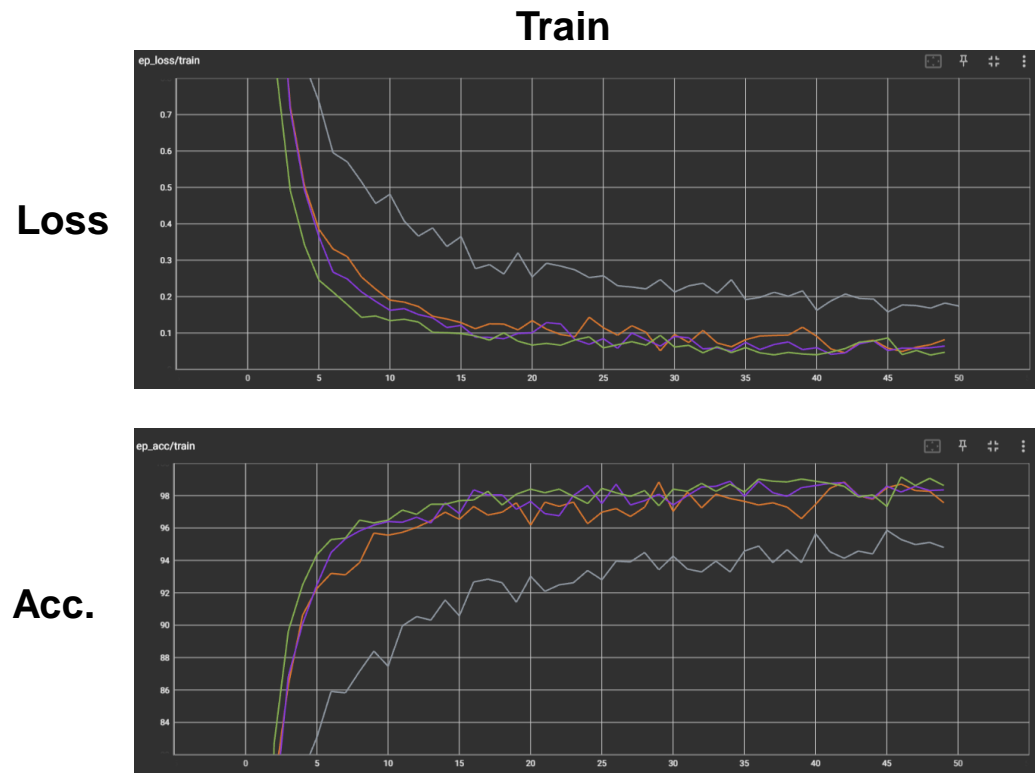
[Training Details]

- Data augmentations (including hyperparameters, such as randomness value)
- Learning rate | num_epoch | batch_size | optimizer | etc.

Report (2)

[Inference Details & Experiment Results]

- Accuracy & loss values of your experimented configs (in train/val phase) – include the graphs from tensorboard (shown below)
- Accuracy and the configs of your submitted version (in test phase)



Grading Scheme

- Coding (70) – Cut-off test accuracy
 - 80% (30)
 - 84% (50)
 - 86% (70)
- Report (30)
 - Model Details (10)
 - Training Details (10)
 - Experiment Results (10)

Due Date

- 20th June, 23:55 KST
- Delay Policy
 - -50% pts for ~21st June, 23:55 KST

Submission

- Submit the **zip** file in **LearnUs**. The file must have the structure below.
 - [Student ID]_project1.zip
 - Your code files (in .py)
 - report.pdf
 - [Student ID]_test_subm.csv

Contact

- Regarding the clarification about the project, please use the Class Q&A of LearnUs
- If you have any issues, contact us at the following email address:
yonsei.2024.csi4116@gmail.com