NYCU Introduction to Machine Learning, Final Project

109612019, 林伯偉

1. **Environment Settings**:

- Python version : 3.9.18 in conda
- Framwork : PyTorch with fastAI package
- Hardware : GPU-1080Ti

2. **Implementation**:

Model Architecture:

For this project, I've experimented with two different pre-trained models: ResNet50 and VGG11. The ResNet50 model outperformed VGG11, providing a better accuracy on our image classification task. ResNet50 is a deep residual network architecture known for its success in image recognition tasks. It consists of 50 layers, including residual blocks, which help mitigate the vanishing gradient problem during training

Hyperparameters:

- Learning rate : dynamic
- Weight Decay : 0.3
- Epoch : 10

These hyperparameters were chosen through a combination of manual tuning and experimentation to find a balance between fast convergence and avoiding overfitting.

Training Strategy:

- **Data preparation:**

Before initiating the training process, I divided my dataset into training and validation sets. Approximately 80% of the data was used for training, while the remaining 20% served as a validation set. This split was chosen to provide a sufficient amount of data for the model to learn from while allowing for robust evaluation.

- **Fine tuning:**

I employed the fine-tuning technique to adapt the pre-trained model to my specific bird classification task. During fine-tuning, the weights of the pre-trained model were updated based on the gradients computed on my dataset. This process helps the model specialize in recognizing features relevant to bird species.

- **Learning rate schedule:**

  I used a one-cycle learning rate schedule during training. This involves gradually increasing the learning rate for the first phase of training, reaching a maximum value, and then gradually decreasing it for the remaining epochs. This approach helps the model converge faster initially and then refine its weights with a smaller learning rate, potentially avoiding overfitting.

- **Hyperparameter tuning:**

  The choice of hyperparameters, including learning rate, weight decay, batch size, and the number of epochs, was based on a combination of empirical experimentation and manual tuning. I performed a systematic search across different hyperparameter values to find a set that provided a good trade-off between rapid convergence and preventing overfitting.

- **Monitoring:**

  To ensure the model was learning effectively, I monitored performance metrics on the validation set during training. Early stopping was implemented to halt training if there was no improvement in validation accuracy after a certain number of epochs. This prevented the model from overfitting to the training data and ensured that I retained a well-generalized model.

3. **Experiment Result**:

Evaluation metrics:

It's difficult to show a confusion matrix with 200 class classification on screen, instead I've created a table showing the top 4 missclassidied relationship between two classes.

RESNET50 :

Original output:

Actual: 072.Pomarine_Jaeger, Predicted: 071.Long_tailed_Jaeger, Occurrences: 6, Confusion Value: 6
Actual: 025.Pelagic_Cormorant, Predicted: 023.Brandt_Cormorant, Occurrences: 5, Confusion Value: 5
Actual: 030.Fish_Crow, Predicted: 029.American_Crow, Occurrences: 5, Confusion Value: 5
Actual: 051.Horned_Grebe, Predicted: 050.Eared_Grebe, Occurrences: 5, Confusion Value: 5

Table:

Actual	Predeicted	Occurences	Confusion Value
72	71	6	6
25	23	5	5
30	29	5	5
51	50	5	5

VGG11 :

Original output:

Actual: 112.Great_Grey_Shrike, Predicted: 111.Loggerhead_Shrike, Occurrences: 6, Confusion Value: 6
Actual: 060.Glaucous_winged_Gull, Predicted: 066.Western_Gull, Occurrences: 5, Confusion Value: 5
Actual: 102.Western_Wood_Pewee, Predicted: 103.Sayornis, Occurrences: 5, Confusion Value: 5
Actual: 137.Cliff_Swallow, Predicted: 136.Barn_Swallow, Occurrences: 5, Confusion Value: 5

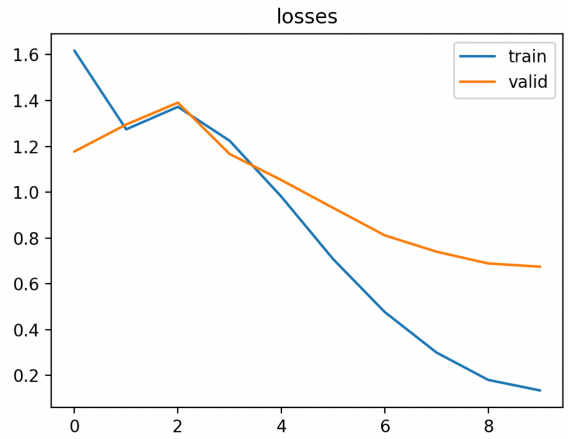
Table:

Actual	Predeicted	Occurences	Confusion Value
112	111	6	6
60	66	5	5
102	103	5	5
137	136	5	5

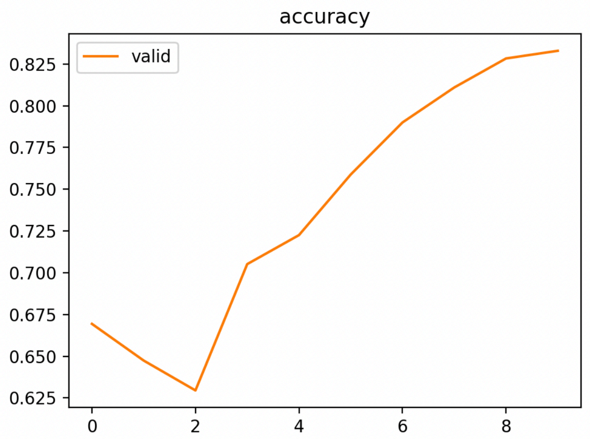
Learning Curve:

RESNET50

Train & Valid loss:

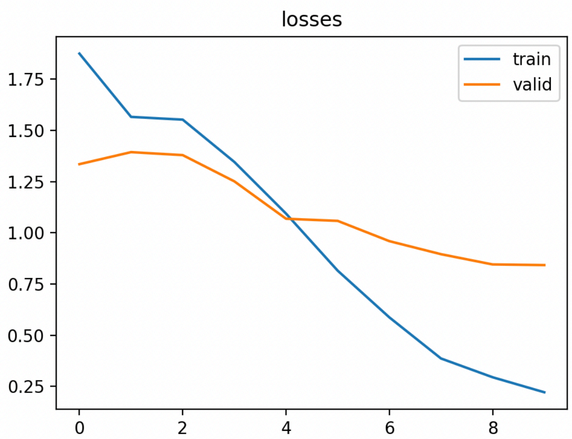


Accuracy:

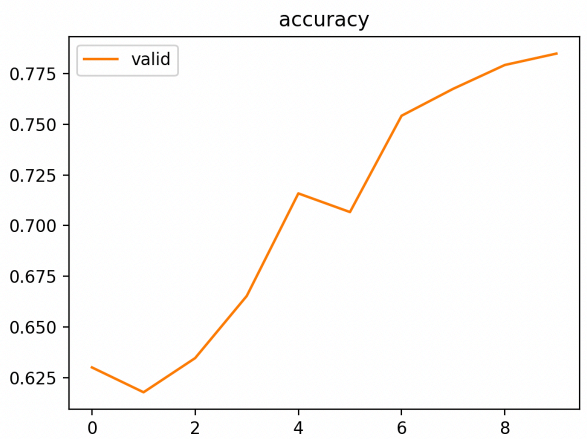


VGG11:

Train & Valid loss:



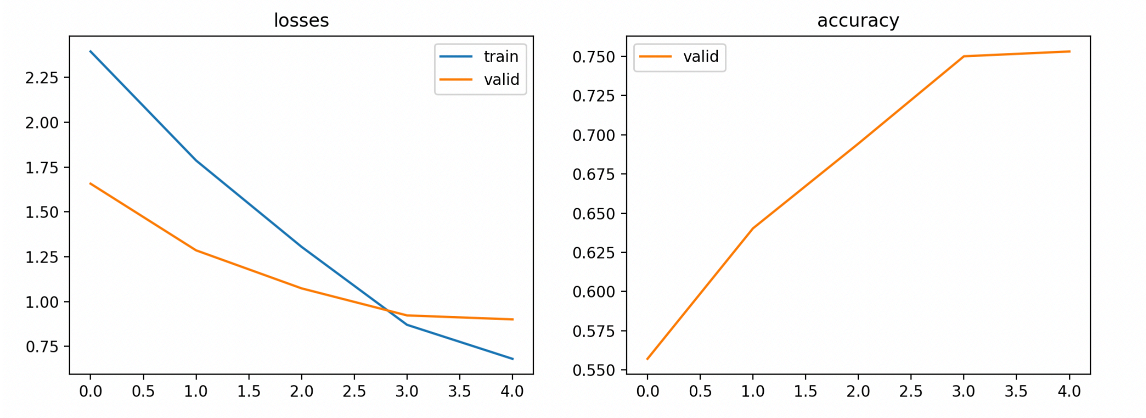
Accuracy:



Ablation study:

For the ablation study, I conducted experiments with two variants of the ResNet34 architecture: one with pre-training on ImageNet and the other without pre-training (randomly initialized weights). The goal was to investigate the impact of pre-training on the model's performance

Pre-trained Resnet34:

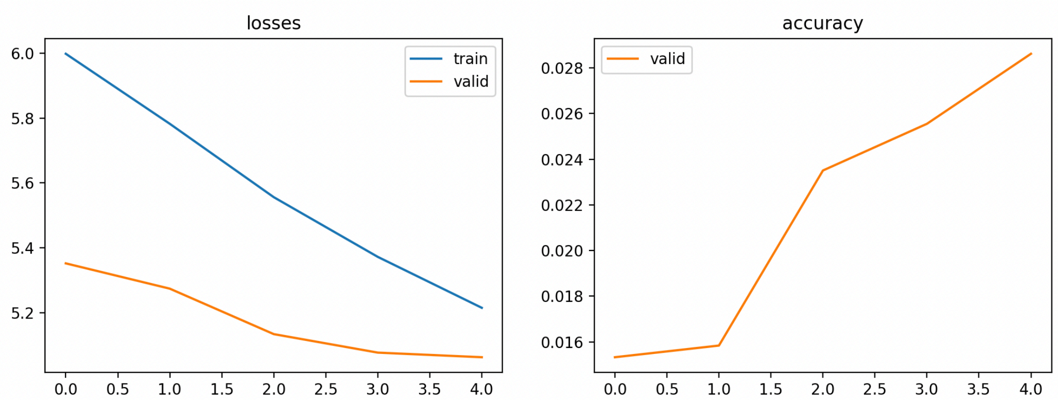


- **Acccuracy:**

  The pre-trained ResNet34 achieved a higher accuracy compared to its randomly initialized counterpart.
- **Convergence:**

  The pre-trained model converged faster during training. It reached a certain level of accuracy sooner, indicating that the knowledge gained from ImageNet pre-training facilitated quicker learning of task-specific features.

Not Pre-trained Resnet34:



- **Acccuracy:**

  The randomly initialized ResNet34 exhibited lower accuracy compared to the pre-trained model.

- **Convergence:**

  The model will definetly took more epochs to reach a comparable level of accuracy. Without the knowledge transfer from pre-training, the model required more training iterations to have the same accuracy.

4. Bonus

- **Discussion:**

  Comparing the performance of different architectures revealed that deeper networks like ResNet50 tend to perform better on complex tasks. While VGG11 performed reasonably well, it lagged behind ResNet50. This observation aligns with existing literature suggesting that the choice of architecture depends on the complexity of the task and available data.

- **Paper Review:**

  I've reviewed relevant papers on image classification, transfer learning, and neural network architectures. Notably, the original ResNet paper by He et al. (2016) provided valuable insights into the advantages of residual networks. This informed our decision to experiment with ResNet50.

- **Future work:**

  Future work could involve exploring additional data augmentation techniques, experimenting with different learning rate schedules, and incorporating ensembling methods to further improve model performance.

5. Conclusion:

In conclusion, the implementation of ResNet50 with carefully chosen hyperparameters and training strategies resulted in a robust image classification model. The ablation study and comparison with VGG11 provided valuable insights into the suitability of different models for our specific task. The report demonstrates the importance of empirical experimentation in choosing the right model for a given problem.

Instruction : where to put the "model.pkl" file