# Machine Learning Final Project Report

110550085 房天越

## *Important!!!*

If you are running my inference code on Colab, please make sure that the data has the following route:
The route to training dataset is /content/drive/MyDrive/data/train
The route to test dataset is /content/drive/MyDrive/data/test
The route to the trained model weight is /content/drive/MyDrive/trained_model_fin.pth
I am sure that it could be run once the route is correct, the route is determined when uploading them to the drive. If the dataset and the weight are uploaded directly to the drive, the route should be the same.

## 1. Environmental Details

Python Version: 3.10.12.
Framework: PyTorch, version: 2.0.0.
Hardware: Tesla P100-PCIE-16GB x1, running on Kaggle.

## 2. Implementation Details

Model Architecture:

The model I used for this project is ResNet-50, which is a deep convolutional neural network architecture that belongs to ResNet family. The key innovation of ResNet is the introduction of residual learning, which addresses the vanishing gradient problem in deep neural networks. Residual blocks, the building blocks of ResNet, include shortcut connections that allow the network to learn residual functions, making it easier to train very deep networks.

ResNet50 specifically has 50 layers, consisting of convolutional layers, batch normalization, activation functions, and residual blocks.

Hyperparameters:

The hyperparameters used during training include a learning rate of 0.0005, batch size of 64, and the Adam optimizer.

Training Strategies:

There are a lot of things to notice in this project:

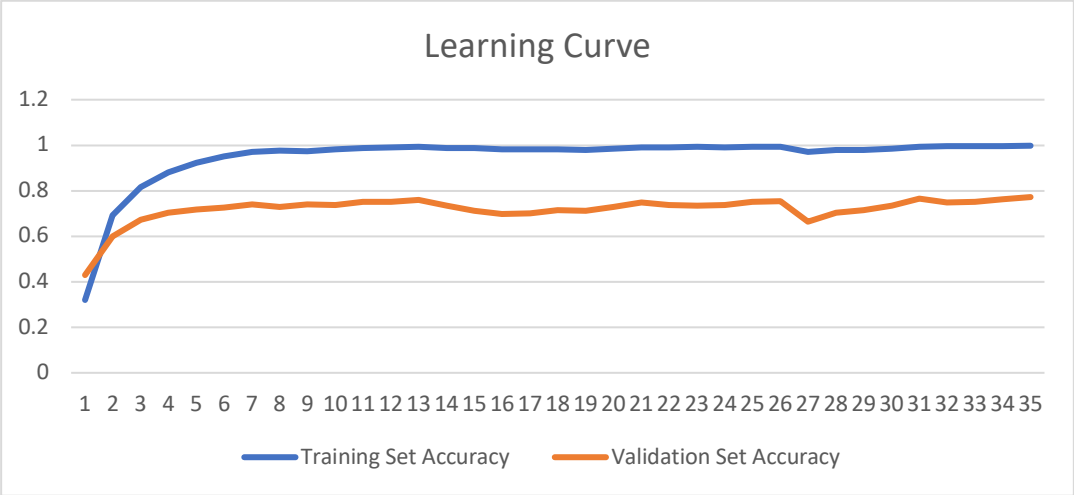
1. Data Augmentation
   This is written in data_transform in the training code, including random horizontal flipping, color jitter, and random rotation.
2. Pretrained model
   I use a pretrained ResNet50 model by using models.resnet50(weights=models.ResNet50_Weights.DEFAULT), this would automatically install a pretrained weight and use it, so we don't need to find it on the Internet and upload it.
3. Splitting into training part and validation part
   In my code, I split the training dataset into two parts, which are training part and validation part, it has a proportion of 8:2, there are mainly two benefits using this method. First, I can observe the current correct percentage by validating the validation part, I do not need to upload it to Kaggle and waste one of my only 5 chances in a day. Second, we can use loss function to check it and prevent it from getting overfitted.
4. Label Smoothing Cross Entropy
   By using this loss function, we can prevent it from overfitting because it could avoid overconfidence in training. It penalizes the model not only for incorrect predictions but also for being too confident. This helps to encourage the model to have more calibrated and less overconfident predictions, leading to improved generalization on unseen data. We can observe that the loss would suddenly grow up when it is likely to be overfitted, and then go back to a performance that is even better than before.
5. Setting checkpoints
   I save the state of every epoch, when Kaggle shut down or something bad happens, I can go back training with the saved .pth file.

## 3. Experimental Results

Evaluation Metrics:

I split the test data into 2 parts, which are training part and validation part, and has a proportion of 8:2, I use the validation part to check the correct percentage. The following learning curve is shown by calculating accuracy with both training dataset and validation dataset.

Learning Curve (for ResNet-50, using weight from Net, 35 epochs trained):




We can see that there is a sudden down, after that, it performs even better.

Ablation Study:

There are a lot of things that would affect the model's performance.

1. Pretrained model

   There are two situations, one is to use pretrained = True, this would not only have a worse performance, but also would generate a warning that indicates that it is not stable, please use weight instead. I ignored this warning for many days and did not know how to improve the performance of the model, until one day I change it to models.resnet50(weights=models.ResNet50_Weights.DEFAULT), the accuracy grew up like 8%, which really astonished me.

2. Data Augmentation

   Data augmentation also plays an important role in improving model's performance, using random flipping, random rotation and color jitter all increase the accuracy, and they could also make use in preventing the model from overfitting.

3. Model to choose

   I tried three kinds of models, which are ResNet-18, ResNet-50 and Inception V4. ResNet-18 has the worst performance, but the training time of it is the least. ResNet-50 and Inception V4 has almost the same performance, but the training time of ResNet-50 is less, so I choose it to implement in the later period.

   I wonder why Inception V4 has almost the same performance with ResNet-50, maybe it's because I use Inception V4 with inception_v4(weight = 'imagenet'), which is maybe not a good usage.

## 4. Something important to mention

It is quite important to mention, because Colab is really difficult to use, and good GPU on it costs a lot, I chose to run it on Kaggle, which has free 30 hours P100 to use every week once I get my phone verified. After training, I found that the inference code should be run on Colab, but my trained model was run on Kaggle, which has different way of mapping from index to class. Not knowing how to solve that problem, I chose to spend hours mapping the indices back to classes manually by observing the relationship between the file I generated on Kaggle and the indices generated in Colab, so you would see a mapping from indices to classes, I had no choice but doing that qq. The code can also be run if you remove the mapping part, modify the route to training and testing data and run it on Kaggle.

## 5. Discussion

During the process of mapping like what I mentioned above, I found that it is very likely that the model would not be able to distinguish between some similar species, like the many kinds of sparrows. Most of the failures in distinguishing happens in situations like that. So, if we can take these similar species out and do specific distinguish between them, the accuracy would definitely become larger. It is such a pity that I have no time to do it this time.

## 5. References


Some details about ResNet family

Introduction to Inception models

Introduction to DCNN