**Lab3: Diabetic Retinopathy Detection**

Author: 311553007應耀德

1. **Introduction**

**Lab Objective:**

In this lab, you will need to analysis diabetic retinopathy (糖尿病所引發視網膜病變) in the following three steps. First, you need to write your own custom DataLoader through PyTorch framework. Second, you need to classify diabetic retinopathy grading via the ResNet architecture. Finally, you have to calculate the confusion matrix to evaluate the performance.

**Requirements:**

1. Implement the ResNet18, ResNet50 architecture and load parameters from a pretrained model.
2. Compare and visualize the accuracy trend between the pretrained model and without pretraining in same architectures, you need to plot each epoch accuracy (not loss) during training phase and testing phase.
3. Implement your own custom DataLoader.
4. Calculate the confusion matrix and plotting.

**Data - Diabetic Retinopathy Detection (Kaggle):**

Diabetic retinopathy is the leading cause of blindness in the working-age population of the developed world. It is estimated to affect over 93 million people. This dataset provided with a large set of high-resolution retina images taken under a variety of imaging conditions.

Image format: .jpeg

Image size: 512 x 512

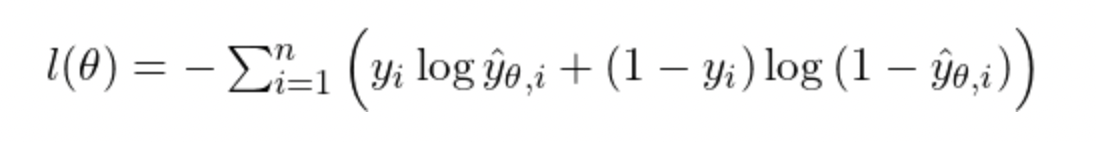
Number of images: 25,124 images

*Reference:* [*https://www.kaggle.com/c/diabetic-retinopathy-detection#description*](https://www.kaggle.com/c/diabetic-retinopathy-detection#description)

* Train data: train\_img.csv, train\_label.csv, 28,009 images
* Test data: test\_img.csv, test\_label.csv, 7,025 images

**Loss function:**

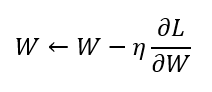
Cross-Entropy Loss



Where

**Optimizer:**

Stochastic Gradient Descent (SGD)



Where

1. **Experiment setups**

**Training settings:**

* Batch Size: 128(ResNet18), 64(ResNet50)
* Epoch Size: 10(ResNet18), 5(ResNet50)
* Optimizer: SGD, learning rate = 1e-3, momentum = 0.9, weight decay = 5e-4
* Loss function: Cross-Entropy Loss

**The details of your model (ResNet18 and ResNet50):**

一張含有 文字 的圖片

自動產生的描述

Figure 2. Implementation of ResNet18 and ResNet50

**The details of your dataloader:**

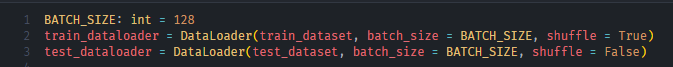
一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

Figure 3. Implementation of dataset



**Describing your evaluation through the confusion matrix:**

x-axis為predicted label (代表A)，y-axis為ground truth (代表B)。斜對角為預測正確之比例。

**從row來解讀**，例如:屬於B0的data分布，普遍預測正確，部分資料被誤判為A2，少部分被誤判為A3, A4。

**從column來解讀**，例如:被預測為A0的data，大部分預測正確，部分資料實際上是屬於其他label。

**合併解讀**，例如:

從A1與B1可知，此model將B1皆誤判為其他label，而預測為A0的最多，代表此model在預測label 1的data上沒有學習到任何關鍵辨別特徵。

也從A0可知，實際上還是有很多data被誤判為A0，代表並沒有完全學習到辨別各label的特徵。

一張含有 文字, 監視器, 電子用品, 螢幕 的圖片

自動產生的描述 **一張含有 文字, 監視器, 螢幕, 電子用品 的圖片

自動產生的描述**

Figure 4. An example of confusion matrix (w/o normalization and with normalization)

1. **Experimental results**

**The highest testing accuracy:**

|  |  |  |
| --- | --- | --- |
| **Network** | **pretrained** | **from scratch** |
| ResNet18 | **77.29%** | 73.36% |
| ResNet50 | **77.45%** | 73.35% |

一張含有 文字 的圖片

自動產生的描述一張含有 文字 的圖片

自動產生的描述

一張含有 文字, 監視器, 電子用品, 螢幕 的圖片

自動產生的描述一張含有 文字, 電子用品, 顯示 的圖片

自動產生的描述

一張含有 文字, 監視器, 螢幕, 電子用品 的圖片

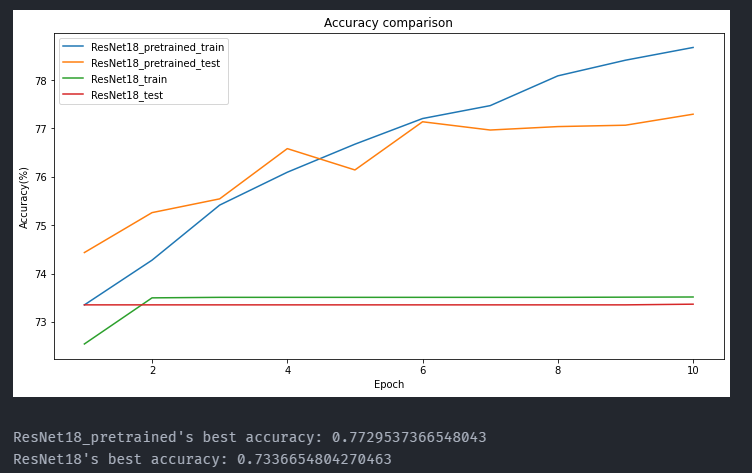
自動產生的描述一張含有 文字, 監視器, 螢幕, 電子用品 的圖片

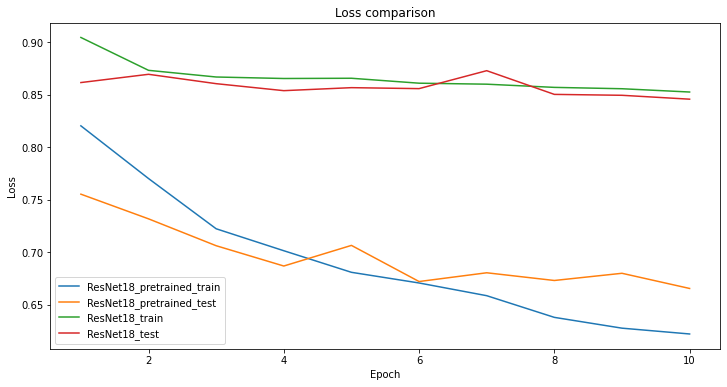
自動產生的描述

Figure 5. Confusion matrix of ResNet18(pretrained) Figure 6. Confusion matrix of ResNet50(pretrained)

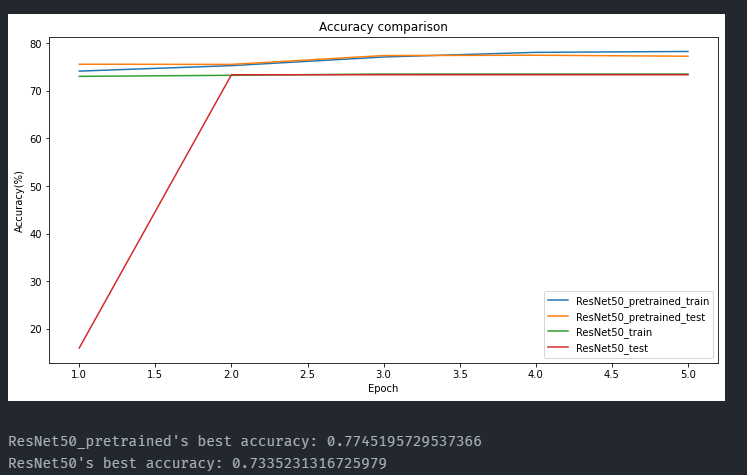
**Comparison figures:**

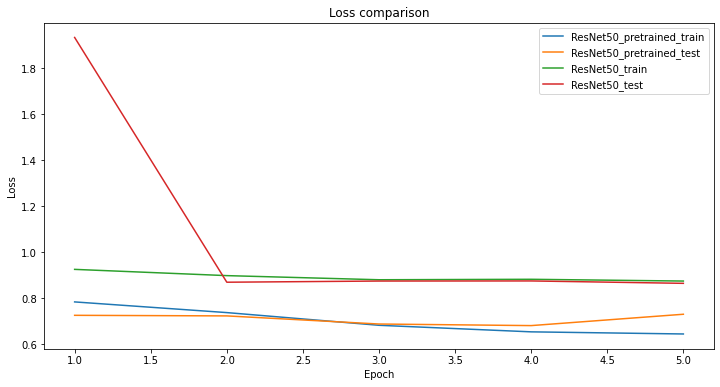
* ResNet18





* ResNet50





1. **Discussion**

**Why is the highest accuracy always around 77%?**

Possible problems

* From the confusion matrix result and the data distribution, the dataset is like long-tailed dataset. The data is imbalanced. From the left to the right entry, it is in the order of labels.



* Use OpenCV to analyze images (training data). About 15% images are too dark (brightness under 25%). Need more data augmentations to learn the contours of image.



**What’s the performance between models with transforms and baseline model?**

實驗配置皆為前面設定，model使用ResNet18 with pretrained weights，除了transforms有更動，baseline model的配置為Resize與ToTensor。

Finetuning的感想:當時在進行實驗時，並沒有與baseline model比較、accuracy提升程度、confusion matrix預測情況，花費大部分的時間在依照data特性組合多個transforms與training上面。雖然有分析data distribution與圖片亮度占比，就以結果來說，許多transforms配置並沒有改善accuracy，反而還比baseline model還差。下圖為測試過的組合，後面括號為best accuracy

一張含有 文字 的圖片

自動產生的描述

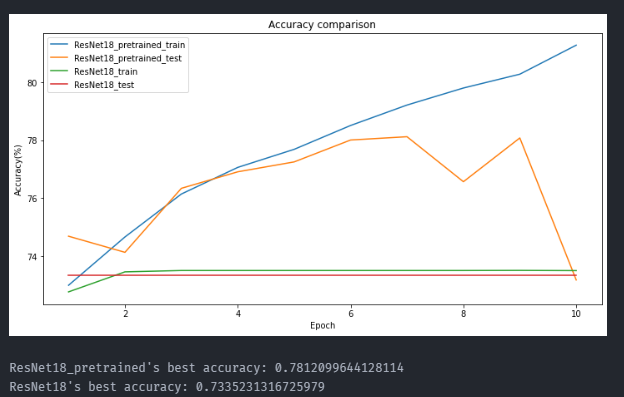


Figure 7. Accuracy comparison of baseline ResNet18(pretrained)

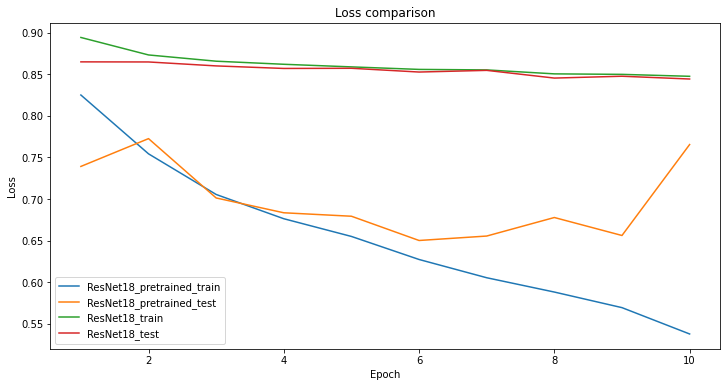


Figure 8. Loss comparison of baseline ResNet18(pretrained)

**The data is imbalanced. How can I solve it?**

Based on the experiment setups

* Test 1: Use cross-entropy loss with class weights

此方法在loss function中加入各個class權重，預設每項class權重為1。

資料量越少的class，該class權重越大 (loss占比較多，gradient較多)。

使得模型也能較關注資料量較少的class上，避免都偏向學習資料量較多的class。

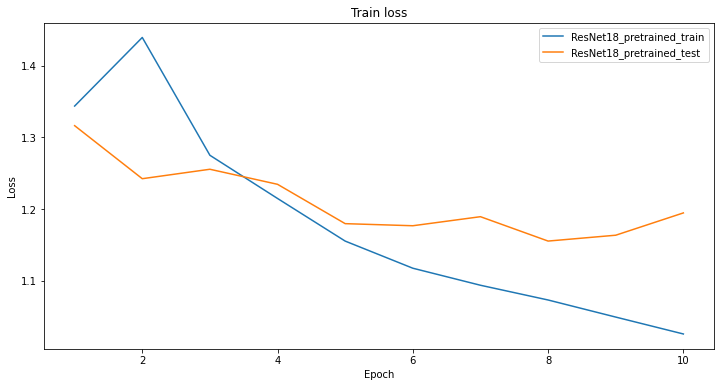
一張含有 文字, 手錶 的圖片

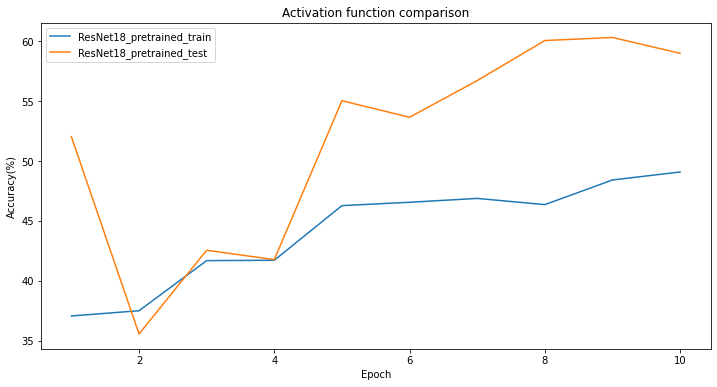
自動產生的描述

Figure 3. The formula of cross-entropy loss function, where , , ,

1. ResNet18 with pretrained weights

從comparison loss圖中可知(請忽略圖中標題)，training loss持續收斂，但礙於訓練時間過長，只能先跑前面的訓練情況。





* Test 2:直接對Data做augmentation，upsampling較少的label類別資料

WIP