




HW5

612415013 蕭宥羽

1. How to execute codes.

- 有測試三種不同的神經網路架構 (ResNet EfficientNet DenseNet)

其中分別會由這些架構建立出不同層數的經典模型(如下圖)

 DenseNet.py	2024/5/1 上午 10:02	Python 來源檔案	5 KB
 Efficientnet.py	2024/5/1 上午 10:02	Python 來源檔案	7 KB
 model.py	2024/5/1 上午 10:02	Python 來源檔案	6 KB

```
def DenseNet121():  
    return DenseNet(init_channels=64, growth_rate=32, blocks=[6, 12, 24, 16])  
def DenseNet169():  
    return DenseNet(init_channels=64, growth_rate=32, blocks=[6, 12, 32, 32])  
def DenseNet201():  
    return DenseNet(init_channels=64, growth_rate=32, blocks=[6, 12, 48, 32])  
def DenseNet264():  
    return DenseNet(init_channels=64, growth_rate=32, blocks=[6, 12, 64, 48])
```

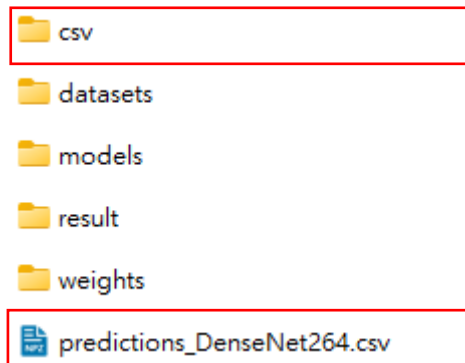
- 訓練的架構的改動只有調整優化器，並將 epochs 設為 50，其他都沒有調動

```
optimizer = torch.optim.Adam(model.parameters(), lr=4e-5, betas=(0.9, 0.999))
```

- 導入所需要的模型

```
from models.Efficientnet import EfficientNet  
from models.DenseNet import DenseNet121, DenseNet264  
from models.model import MyCNN, ResNet50
```

- 繳交的 csv 檔，由測試結果得到使用 DenseNet264 的分數會最高，而其他的結果會放在 csv 這個資料夾



2. Experimental results

ResNet50

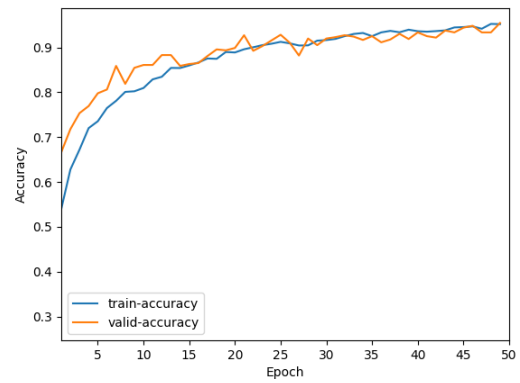
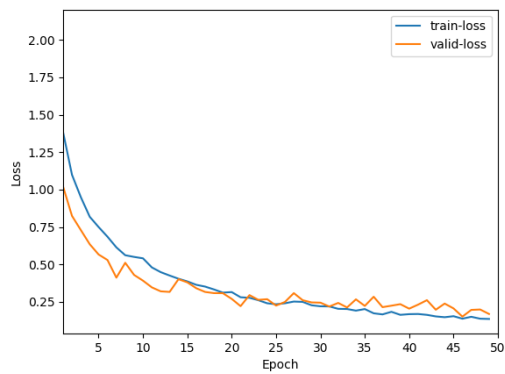


predictions.csv

Complete (after deadline) · 3d ago

0.93828

0.93828



EfficientNet b0

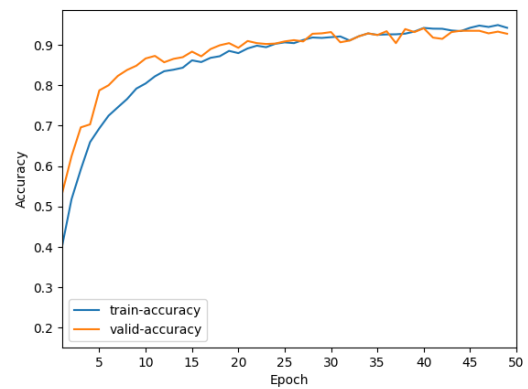
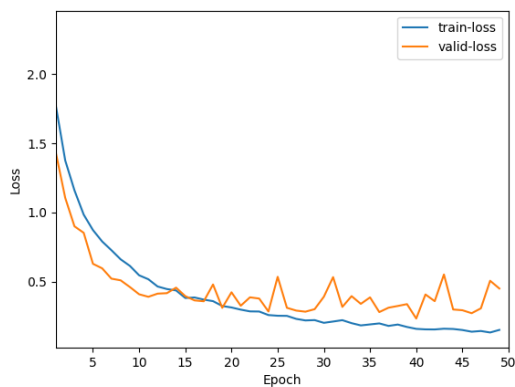


predictions.csv

Complete (after deadline) · 2d ago · EfficientNet() myself

0.94017

0.94017



DenseNet121

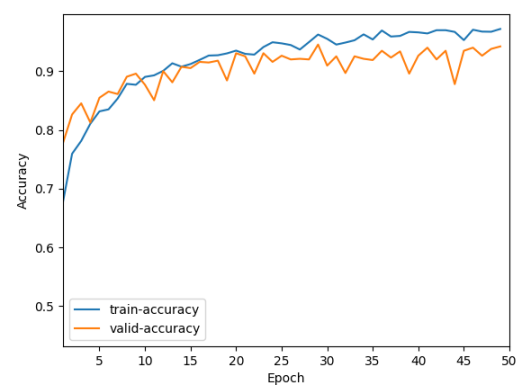
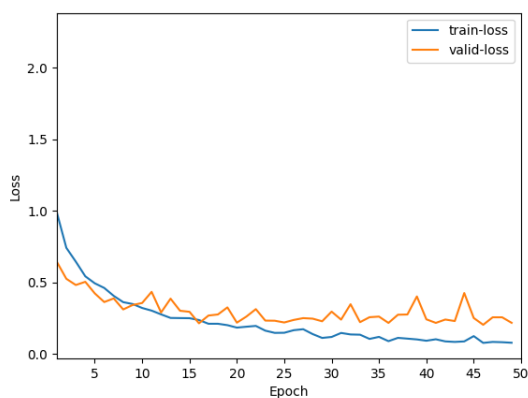


predictions_DenseNet121.csv

Complete (after deadline) · 3h to go

0.94080

0.94080



DenseNet264

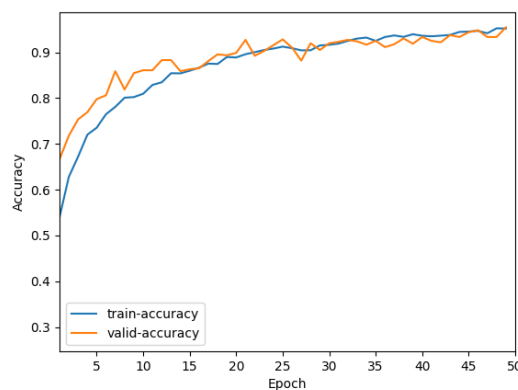
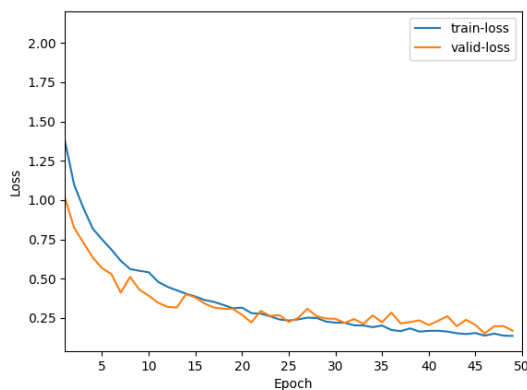


predictions_DenseNet264.csv

Complete (after deadline) · 28s ago · DenseNet264

0.95465

0.95465

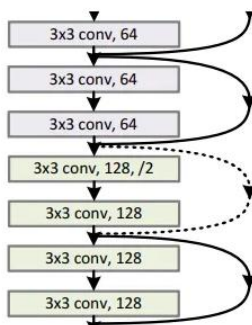


3. Conclusion

本次的實驗根據以下幾個有名的模型進行實作

◆ ResNet50 (Residual Network 50):

是一種深度殘差網絡，由微軟研究人員於 2015 年提出。它採用了殘差學習的概念，通過引入殘差塊 (Residual Blocks) 來解決深度神經網絡中的梯度消失問題。ResNet50 由 50 個層組成，包括捲積層 (convolutional layers)、池化層 (pooling layers) 和全連接層 (fully connected layers)。它在圖像分類、物體檢測和語義分割等任務中表現出色。



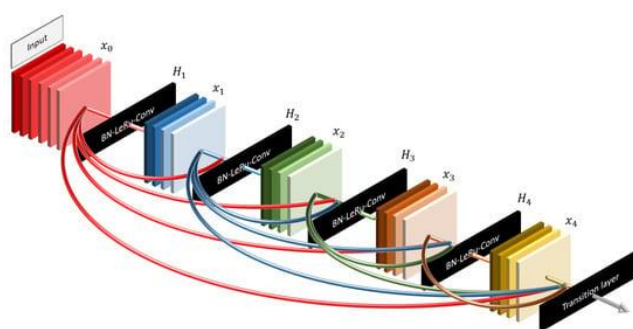
◆ EfficientNet B0:

採用了網絡放大 (Network Scaling) 的思想，通過統一縮放不同維度 (深度、寬度、分辨率) 來提高模型的效率和性能。EfficientNet B0 是其中的一個基於 B0 網絡大小的模型，具有較少的參數量和計算量，同時保持了良好的準確性

◆ DenseNet121 (Densely Connected Convolutional Networks):

是一種密集連接的卷積神經網絡。與傳統的卷積神經網絡不同，DenseNet 中每一層的輸出都與之前所有層的輸出相連接，這種密集連接的結構有助於信息的流動和梯度的傳播，從而減輕了梯度消失問題，並促進了特徵的重用。DenseNet121 是其中一個包含 121 層的模型，廣泛應用於圖像分類、目標

檢測和語義分割等任務中，同時具有較高的準確性和模型效率。



◆ DenseNet264 :

由於 densenet121 有相較前面兩個模型較好的結果，所以使用更深的 densenet264 來做訓練試試看，而最終的結果獲得了目前最佳的成績。

所以根據最終的繳交結果 DenseNet264 得到最高的分數，所以我們的繳交檔就採用 DenseNet264 所生成出的 csv 檔

4. Discussion

在實驗的過程中，我嘗試去復刻經典的分類模型來做為我所使用的模型，一開始我直接使用 pytorch 中內部的模型來訓練(有導入預訓練的權重)，結果也十分得不錯；於是我嘗試著自己刻出這些經典的網路，但由於沒有使用預訓練的權重作為初始權重，所以正確率相較內建的模型差了一點，於是我花了很多時間調整參數和嘗試不同的方法，但效果仍然不如預期。所以我就先以這樣的結果繳交，但在實現的過程中，我覺得我學到了許多，同時也對 CNN 的設計及調整有了更好的了解，而我在查詢資料的過程中有發現如果使用前一學習的方式可以有蠻不錯的結果，且訓練時間也可以減少許多，雖然無法實踐但也讓我學到很多。