

Lab2

中興 黃聖祐 L122502

April 10, 2024

1 Introduction

Lab2 的任務是分類 100 種蝴蝶和蛾的照片，透過自行設計前處理和 data augmentation，並用 VGG19 和 ResNet50 進行訓練和預測。這個資料集共有 12594 個 train data，500 個 validation data 以及 500 個 test data。以下隨機五張資料集的視覺化照片：

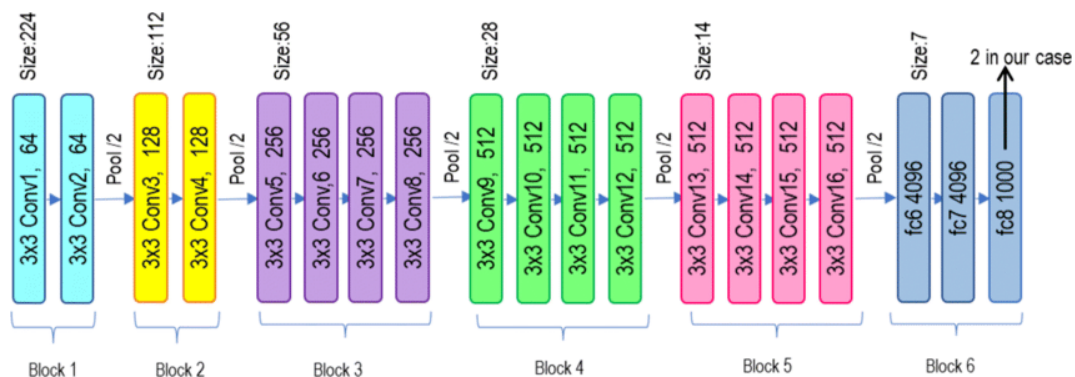


2 Implementation Detail

A The detail of my model (VGG19 and ResNet50)

這次 lab 實作兩個經典的模型 VGG19 以及 ResNet50。

VGG 系列的模型是透過大量的 3*3 卷積層，並在最後接上少量的全聯階層建構而成。它是出自 2014 年的 ILSVRC 比賽上，作者認為 size 較小的 kernel 以及 pooling 可以增加資訊量，以下為我實作的 VGG19 架構：

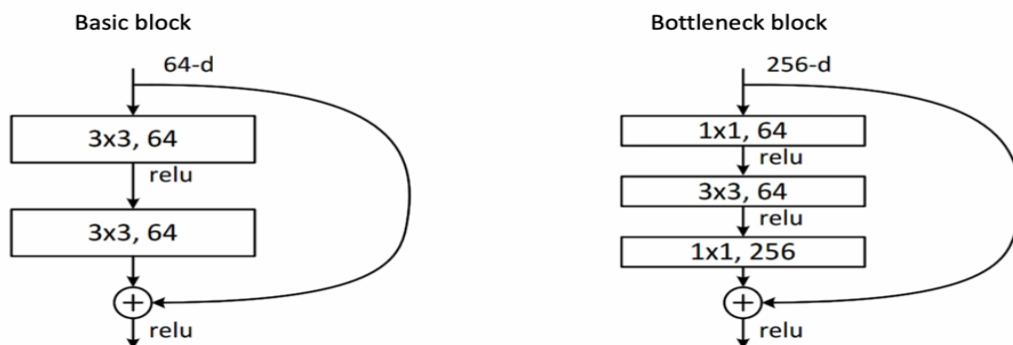


VGG 優/缺點:

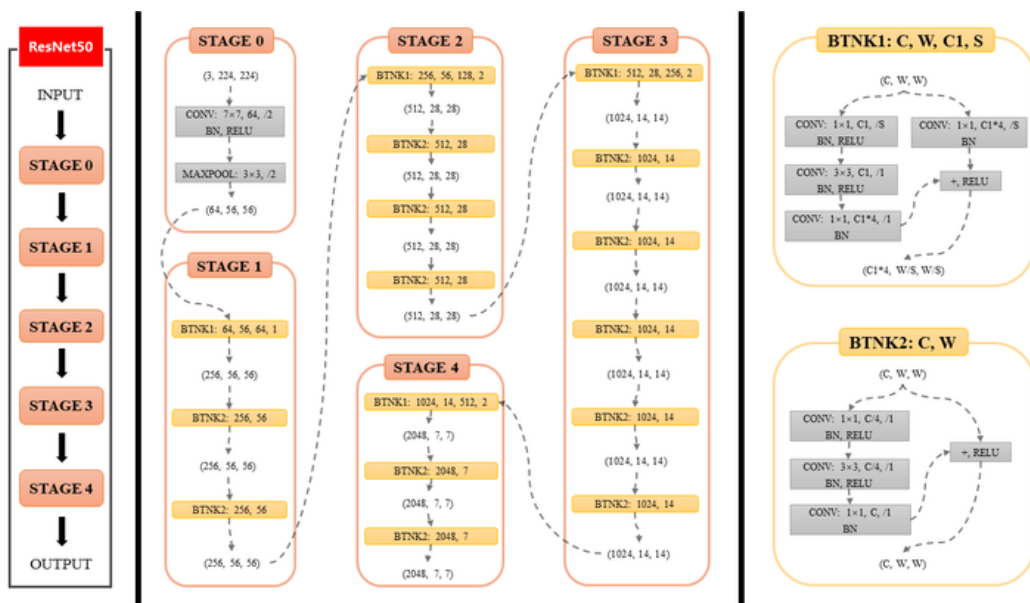
- 架構簡單統一
- 證明較深的層數能提高效能
- 參數量龐大，計算資源需求高
- 訓練時間過長，難以調整參數

VGG19 的參數量十分龐大，超過 1 億 4000 萬個參數，且結構簡單，因此現今較少使用。

ResNet 系列發表於 2015 年，設計的目的是由於神經網路越來越深，導致 gradient vanishing/exploding 的問題，因此使用 basic block 或是 bottleneck block 這種 residual connection 的設計，解決梯度消失或是梯度爆炸的問題，同時也保留了原始的特徵，避免經過太多層卷積導致特徵消失。



下面是 ResNet50 的架構，它分成了四個大區塊，其中每個區塊是由數個 bottleneck block 組成，加上初始的卷積降維和最後的全連接層，一共 50 層。



ResNet 優/缺點:

- 在 VGG 基礎上加入 residual connection，解決梯度消失和梯度爆炸問題
- 若使用 bottleneck block，可以避免參數過多，能夠更好的訓練
- 網路訓練更穩定解順暢，提高精度和泛化能力
- 容易過擬和
- 仍需大量的運算資源計算和推理，且推理速度較慢
- 不能完全解決梯度消失和梯度爆炸問題，只能緩解

B The detail of my DataLoader

我將 DataLoader function 設計成自定義的 Dataset，並在其中對資料集做了以下幾種 data augmentation:

- RandomHorizontalFlip: 隨機水平翻轉，設機率為 0.5。
- RandomRotation: 隨機旋轉，設 degree 為 $[0, 180]$ 。
- RandomCrop: 隨機裁切

因為蝴蝶和蛾的照片常常以不同的角度拍攝，因此使用翻轉和旋轉我認為可以加強資料集角度多樣性。除此之外，每張照片的蝴蝶和蛾的大小不一，裁切可以訓練模型以部分資訊進行判斷。以上僅用於 train data。

3 Data preprocessing

A How I preprocessing my data

對資料做完 augmentation 後，我先將所有資料都轉成 tensor，這可以讓計算平行化，加速運算。而在 `torch.nn.ToTensor` 中，它順便將資料 normalization，因此轉乘 tensor 後，資料皆在 $[0, 1]$ 內。之後，我更將資料做 standardization。透過 normalization 和 standardization，這樣除了可以加速運算，還能避免受到離群值的影響，並將資料變成常態分佈。最後將 train, valid, test data 用 dataloader 包裝成 batch size 為 64 的訓練資料，即完成了 data preprocessing。

B What make my method special

剛開始我並沒有使用那麼多的 data augmentation，也沒有使用 standardization，因此訓練出來的沒那麼好。之後藉由觀察訓練資料集，加上了這幾種資料增強，我認為對模型的泛化訓練有一定提升。但整題來說，我並無做太多資料前處理，因為資料集蠻乾淨的。比較特別的是使用 standardization，我使用的是 ImageNet 的平均值和標準差，但效果十分不錯。

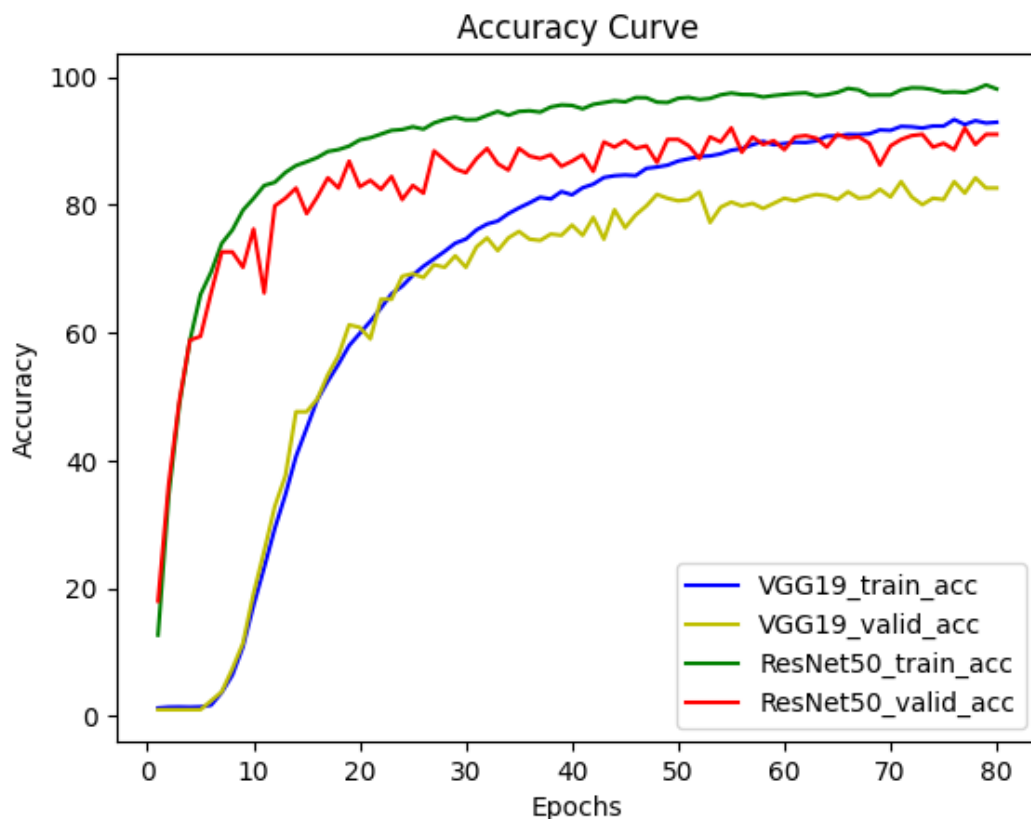
4 Experimental Results

A The highest testing accuracy

VGG19		
VGG19	Train accuracy: 97.19%	Test accuracy: 86.20%
ResNet50		
ResNet50	Train accuracy: 98.05%	Test accuracy: 92.80%

我花在訓練 VGG19 的時間比 ResNet50 還要多不少，這對應了前面提到的 VGG19 參數過多導致訓練時間加長的問題。除此之外，ResNet50 在前期 valid accuracy 上升很快，在 10 epoch 就可以將近 80% 上下，但 VGG19 在同時期還不到 20%。

B Compare figure



VGG19 不論是 train/valid accuracy 都上升的較 ResNet50 慢，雖然 train accuracy 最後也可以追上 ResNet50，但與 valid accuracy 差距越來越多。

5 Discussion

剛開始對於要使用哪些 data augmentation，以及超參數的設計都沒有方向，後來透過一些測試以及對資料集的觀察，找出有效的資料增強方法和超參數。除此之

外，我在訓練過程中發現蠻容易 overfitting，所以我加入了 learning rate scheduling: ReduceLROnPlateau，它是不靠 epoch 更新 learning rate 的方法，而是在學習停滯數個 epoch 時，則會縮小 learning rate。加入後解決了前期容易訓練停滯，後期容易 overfitting 的問題。

一開始因為訓練速度過慢，我將訓練資料都先 resize 至 (64, 64) 的大小，雖然訓練速度加快了不少，但是 train accuracy 停滯於 78% 左右，而 valid accuracy 為 70%。後來恢復用 (224, 224) 大小訓練，讓 train accuracy 可以輕鬆的突破 90%，valid accuracy 也可以保持 85% 以上。經過了 lab2 的學習，熟悉了 VGG 以及 ResNet 的模型架構。兩者最為經典的 CNN 架構，都是以深層著稱，而 ResNet 似乎比 VGG 更好訓練，效果也更好。