深度學習 Deep Learning

Homework2

N16120145 機械所 陳祈均

Task: Designing a Convolution Module for Variable Input Channels

本題是由智慧運算學程 鄭九彰同學所授

透過將輸入 channel 拆分為 RGB、RG、GB、RB、R、G、B 四種，再依據輸入之 channel 進行不同模塊執行，達到可以接收不同 input channel 的效果。

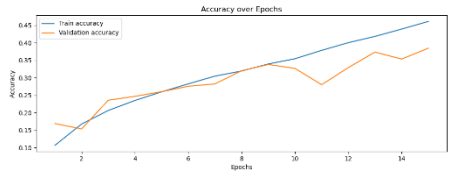
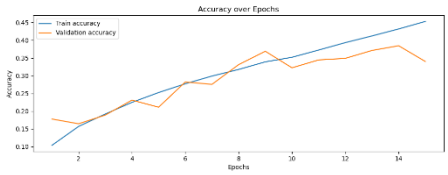
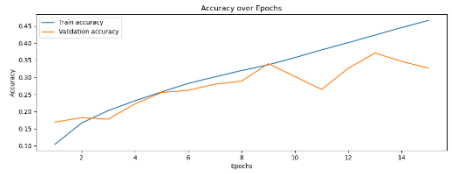
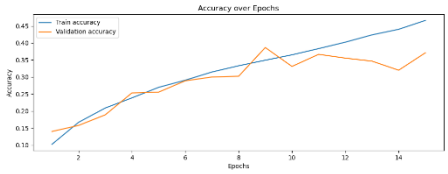
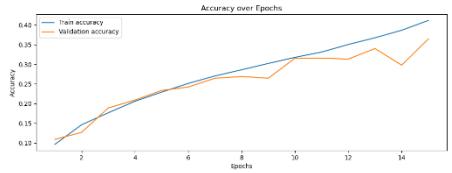
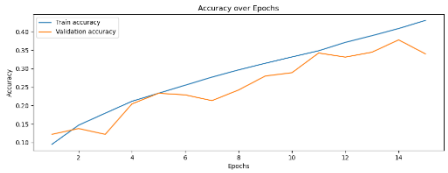
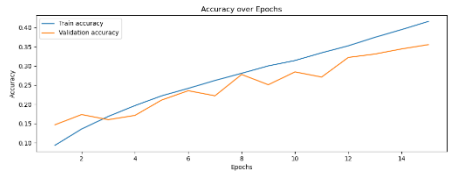
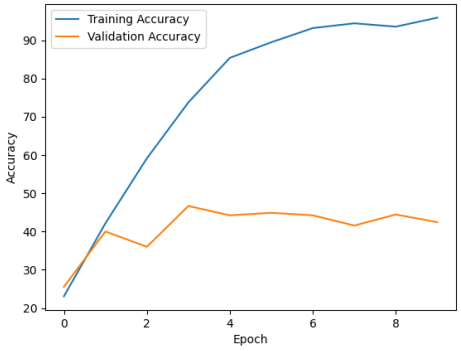
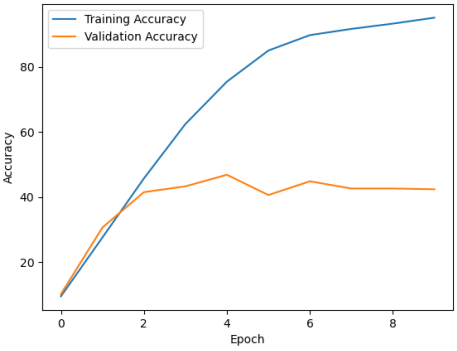
Naive model output

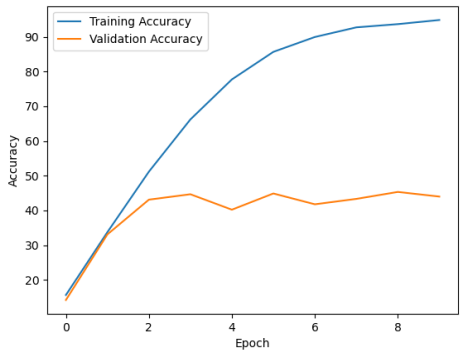
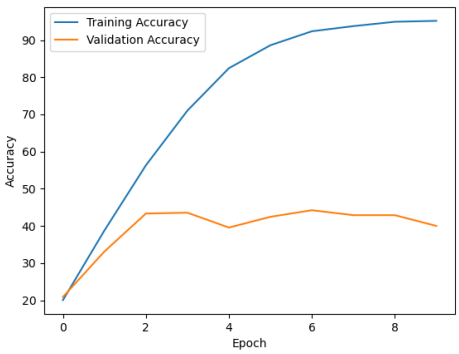
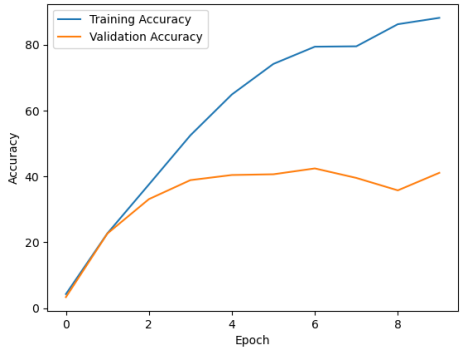
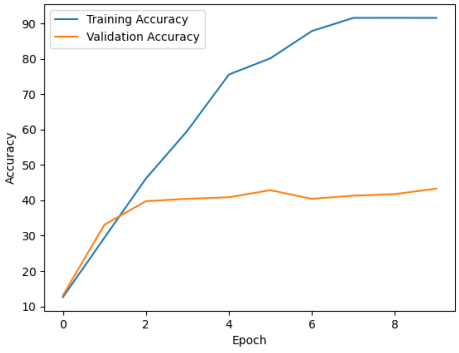
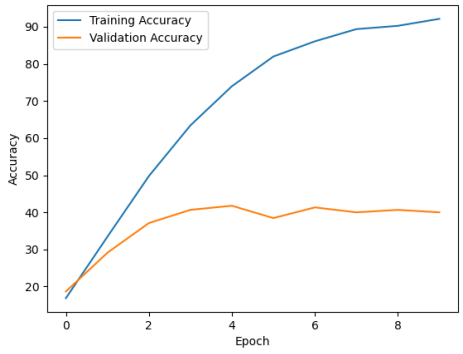
channel	Train accuracy	Val accuracy	Test accuracy	Train Loss	Val Loss	Test Loss
RGB	0.4610	0.3844	0.4	1.8086	2.1797	2.0048
RG	0.4318	0.3844	0.3467	1.9171	2.1624	2.0247
GB	0.4235	0.3711	0.3667	1.9501	2.0979	2.1402
RB	0.3494	0.3867	0.4200	2.2422	2.0402	2.1319
R	0.4116	0.3644	0.3822	2.0049	2.0584	2.1605
G	0.4085	0.3778	0.3511	1.9968	2.1471	2.1295
B	0.4160	0.3556	0.3533	1.9870	2.1095	2.2322

Designed model output

channel	Train accuracy	Val accuracy	Test accuracy
RGB	0.7375	0.4667	0.4533
RG	0.7551	0.4689	0.3978
GB	0.9364	0.4533	0.4156
RB	0.9238	0.4422	0.4200
R	0.7943	0.4244	0.3689
G	0.9166	0.4333	0.3711
B	0.7398	0.4178	0.3756

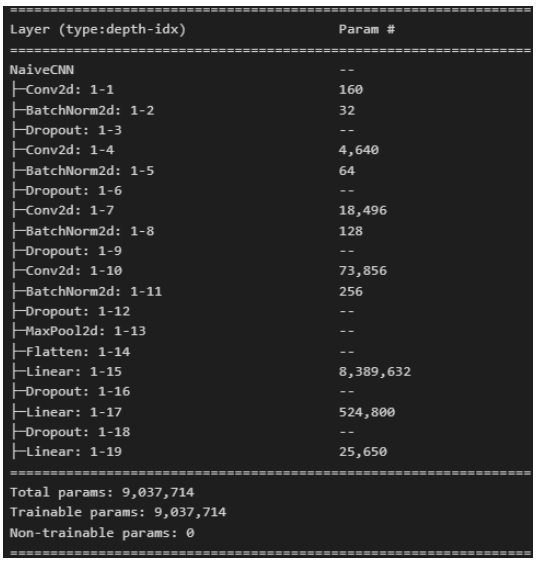
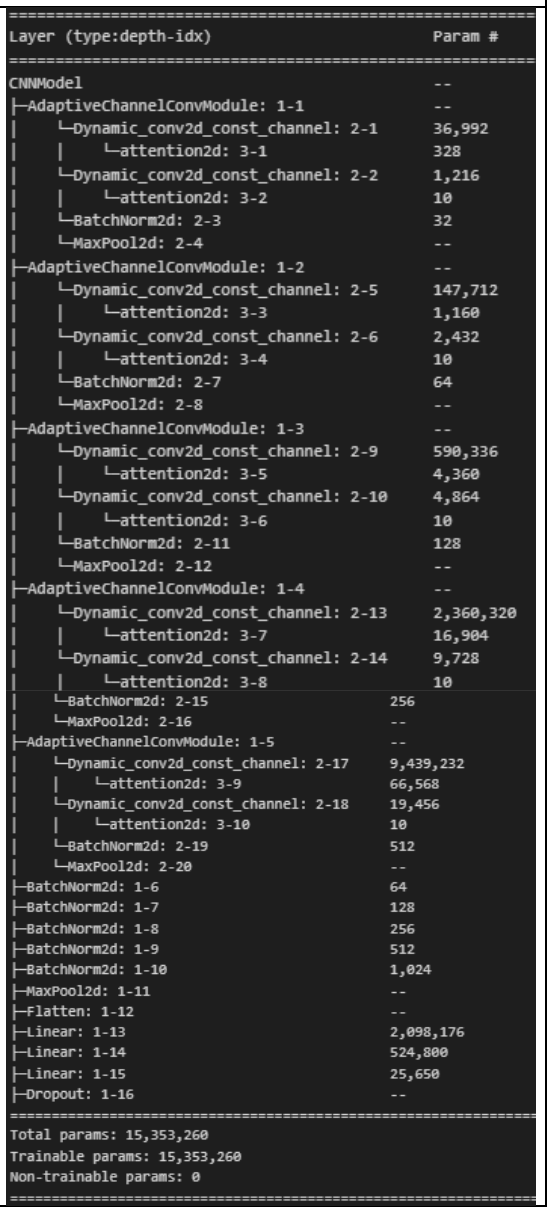
由表格中的結果可以看出：加入自己設計的可接受不同 input channel 的模塊後，整體輸出結果都變好，而 train accuracy 的提升最為巨大，雖然 validation 與 test 的表現有變好，但與 train 相比則相當之小

Naïve RGB	Naïve RG
	
Naïve GB	**Naïve RB**
	
Naïve R	**Naïve G**
	
Naïve B	
	
Design RGB	**Design RG**
	

Design GB	Design RB
	
Design R	**Design G**
	
Design B	
	

由上方的結果圖可以發現雖然準確率有提升，訓練資料準確性大幅提升，但是驗證集準確率提升不大，推測是發生過擬合現象

模型架構：

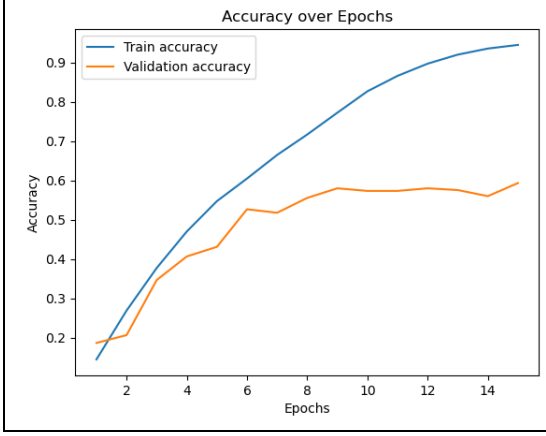
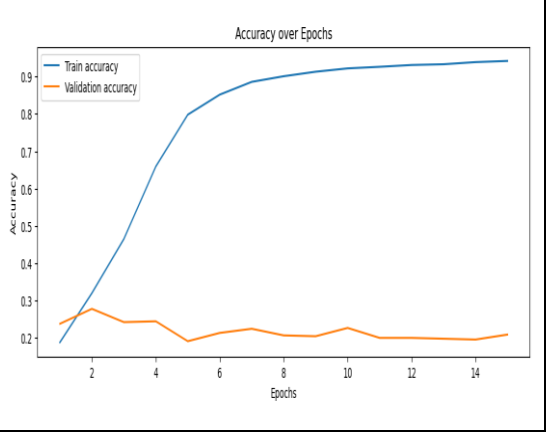
Naive	Design
	

從上圖可以看到 Naïve 的 CNN model 其參數量為 9,037,714 個，而增加了自注意機制與動態卷積層後，參數增加到 15,353,260 個。雖然參數有所增加，但是準確率也上升不少。訓練時間的部分，Naïve CNN model 一個 epoch 約 30 秒左右，設計的 model 則是 1 分鐘左右，所以如果需要進行快速訓練的話，可能就只能選擇 Naïve CNN model。

Design a Two-Layer Network for Image Classification

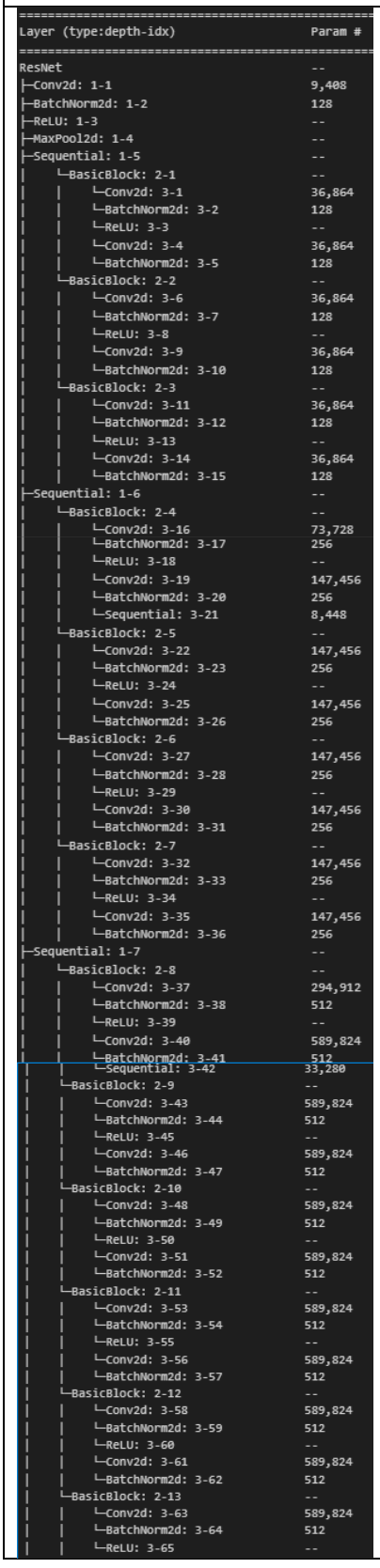
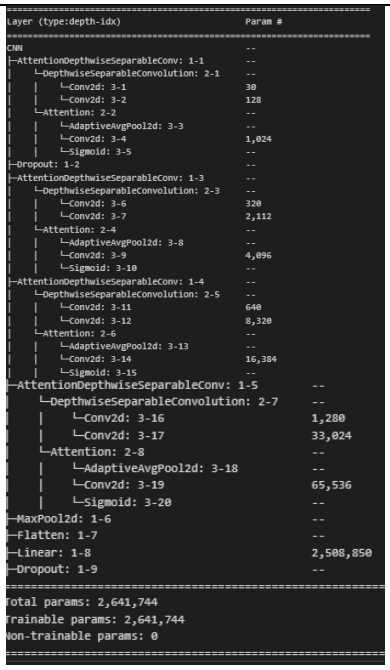
model	Train accuracy	Val accuracy	Test accuracy	Train Loss	Val Loss	Test Loss
ResNet34	0.7722	0.58	0.5933	0.00	1.6323	2.2150
Designed model	0.32	0.2778	0.2178	2.5409	2.8854	4.4544

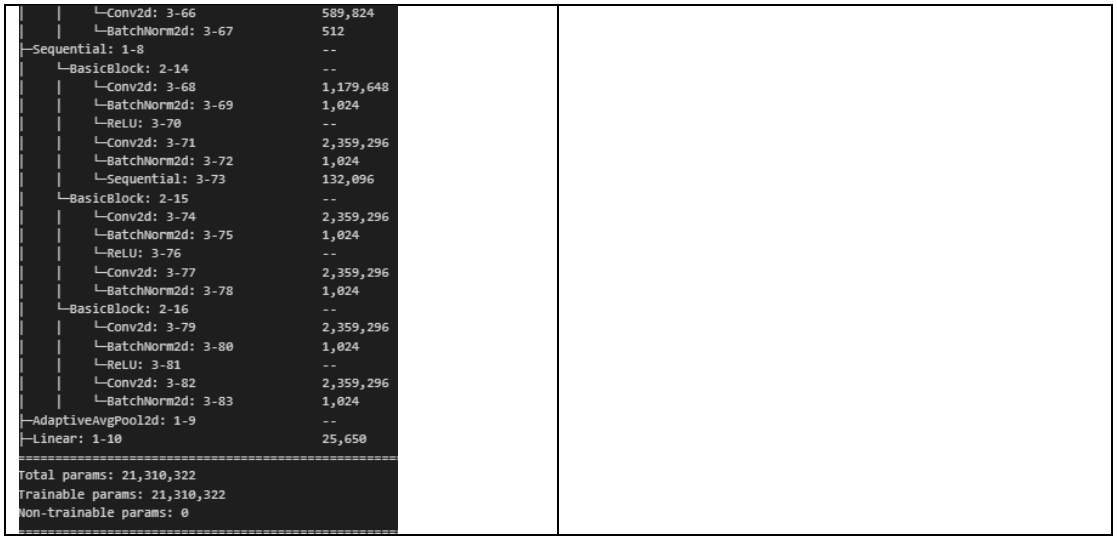
題目目標為希望透過設計一個二到四層 CNN model 來達到 ResNet 90%的表現，本來是希望透過簡單的 DepthwiseSeparableConvolution 模型結合 Self Atttetion 機制來進行，但結果看起來比 ResNet 糟上許多，而且一樣有過擬合的情形發生。

ResNet34	Designed model
	

由上方訓練集與驗證集準確率的圖可以看到：ResNet34 的訓練集要到 epoch=8 才能讓準確率達到 0.7 左右(0.71)，而設計的模型在 epoch=5 時就已經達到準確率約 0.8 左右(0.798)。或許設計的模型真的有機會表現比 ResNet 佳，但過擬合的問題得不到解決，已經嘗試許多方法，例如：在 CNN 中加入 Dropout 層等等，仍無法改善。

模型架構：

ResNet34	Designed model
	



```
│    │    └─Conv2d: 3-66                 589,824
│    │    └─BatchNorm2d: 3-67            512
├─Sequential: 1-8                        --
│    └─BasicBlock: 2-14                  --
│    │    └─Conv2d: 3-68                 1,179,648
│    │    └─BatchNorm2d: 3-69            1,024
│    │    └─ReLU: 3-70                   --
│    │    └─Conv2d: 3-71                 2,359,296
│    │    └─BatchNorm2d: 3-72            1,024
│    │    └─Sequential: 3-73             132,096
│    └─BasicBlock: 2-15                  --
│    │    └─Conv2d: 3-74                 2,359,296
│    │    └─BatchNorm2d: 3-75            1,024
│    │    └─ReLU: 3-76                   --
│    │    └─Conv2d: 3-77                 2,359,296
│    │    └─BatchNorm2d: 3-78            1,024
│    └─BasicBlock: 2-16                  --
│    │    └─Conv2d: 3-79                 2,359,296
│    │    └─BatchNorm2d: 3-80            1,024
│    │    └─ReLU: 3-81                   --
│    │    └─Conv2d: 3-82                 2,359,296
│    │    └─BatchNorm2d: 3-83            1,024
├─AdaptiveAvgPool2d: 1-9                 --
├─Linear: 1-10                           25,650
==========================================================
Total params: 21,310,322
Trainable params: 21,310,322
Non-trainable params: 0
==========================================================
```

由模型架構可以看出，設計的模型架構較 ResNet34 簡單，使用參數也少很多。ResNet34 所使用的參數總量為 21,310,322 個，設計的模型則僅有 2,641,744 個，參數的部分精簡了許多，可惜表現的部分不如預期。

Github 網址：https://github.com/lalala5487/NCKU_DeepLearning-2024_hw2.git