

1. 請說明你實作的 **CNN 模型(best model)**，其模型架構、訓練參數量和準確率為何？(1%)

採用基礎的 VGG16 架構，請參見圖 1 (於報告尾頁)，訓練參數量為：57,727,051，準確率於 training dataset 達 0.99618，於 kaggle 成績為 0.83622。

Epoch = 100, SGD lr = 0.001 with momentum = 0.9

2. 請實作與第一題接近的參數量，但 **CNN 深度 (CNN 層數)** 減半的模型，並說明其模型架構、訓練參數量和準確率為何？(1%)

模型示意圖參見圖二。

將 VGG16 每層包含之 conv2d 層砍一或兩層，並將最後之 FC 層砍半，神經元數量也砍半，共 15,266,059 參數量，訓練於 train+val 資料集準確率達 0.995412，kaggle 成績為 0.82785。

超參設定同 1

3. 請實作與第一題接近的參數量，簡單的 **DNN 模型**，同時也說明其模型架構、訓練參數和準確率為何？(1%)

模型示意圖參見圖三。

將 VGG16 最後數層刪除，只保留前面層數，並改為 FC 層。參數量共 518,595,922，訓練於 train+val 資料集準確率為 0.959161，kaggle 成績為 0.47579。

超參設定同 1

4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

可以看到 1 2 之間的差距並不大，可見模型的深度並非持續疊加就可得到線性的效果成長，要在合理的硬體範圍內得到更大的成長，需要的是更好的架構。而從 3 中可以看到，由於 DNN 的設計，我們需要將圖片拉成一條直線以配合 FC 的輸入，這使得整張圖片「變得不再是一張圖片」。使得神經網路不好找到 2D 圖片的特徵，而是轉而尋找 1D 向量的特徵。

5. 請嘗試 **data normalization** 及 **data augmentation**，說明實作方法並且說明實行前後對準確率有什麼樣的影響？(1%)

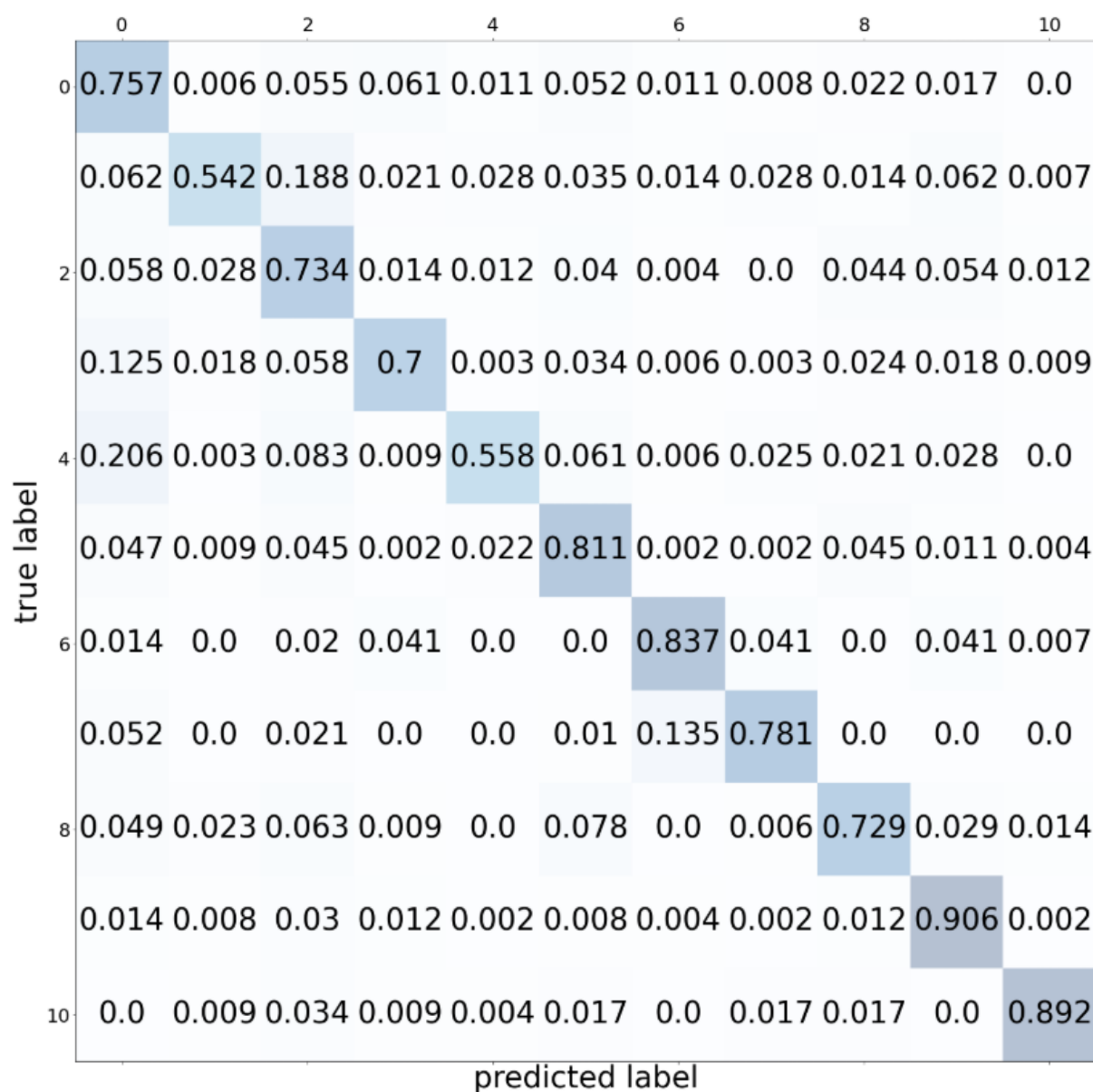
使用了與 1 一樣的模型以及參數，但取消了 normalization 以及 augmentation，在 train+val 上得到了 0.995262，但 kaggle 卻只有 0.52241。可見取消 normalization 後資料的 range 過大使得訓練上遇到了更為崎嶇的參數空間，且取消了 augmentation 使得模型

看過的資料不夠多元，變相的削減了總樣本數量，皆使得模型容易 overfitting。

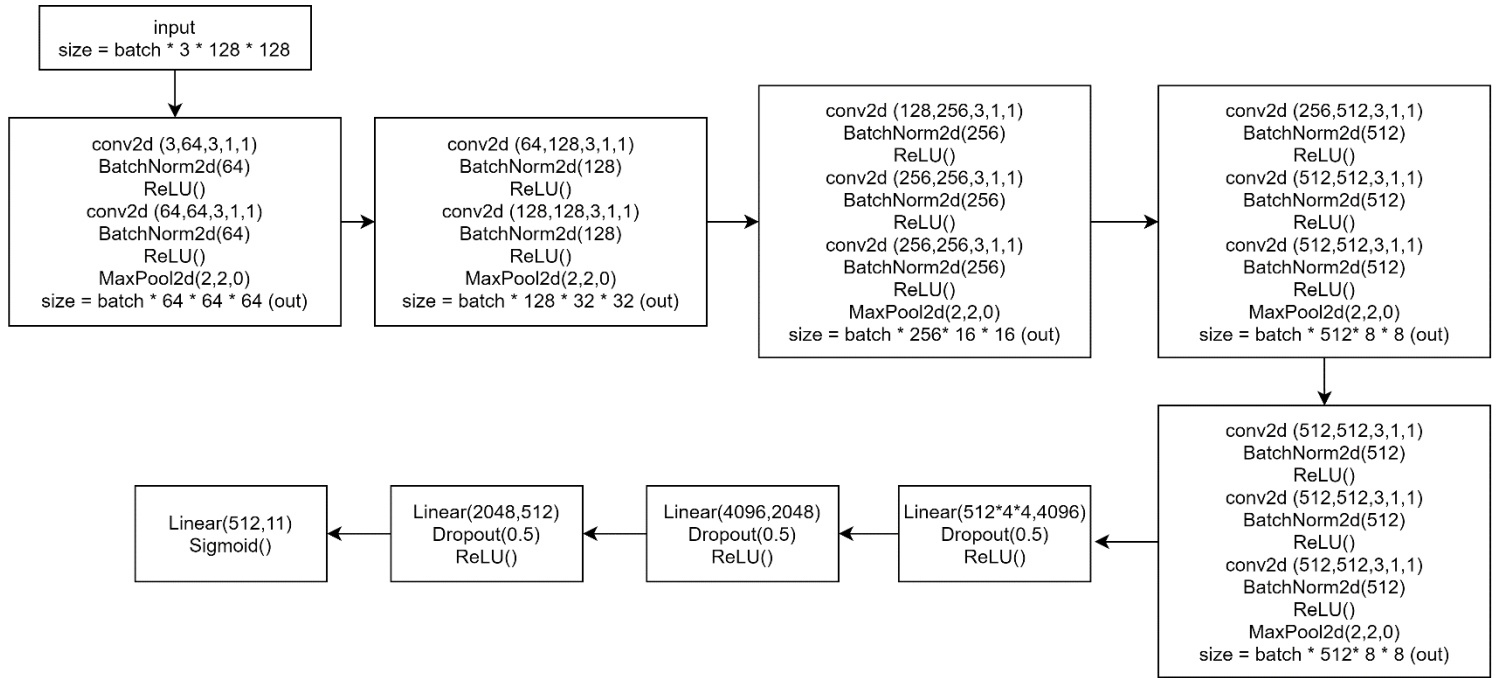
實作方法為取消 tutorial 中的 transforms，由於 ToTensor 會自動 normalize 至[0,1]區間，取消而後於 ImgDataset 中於 transform=None 的分支中額外使用 view 更改為原先的 shape，以及轉換為 Tensor。

6. 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析](1%)

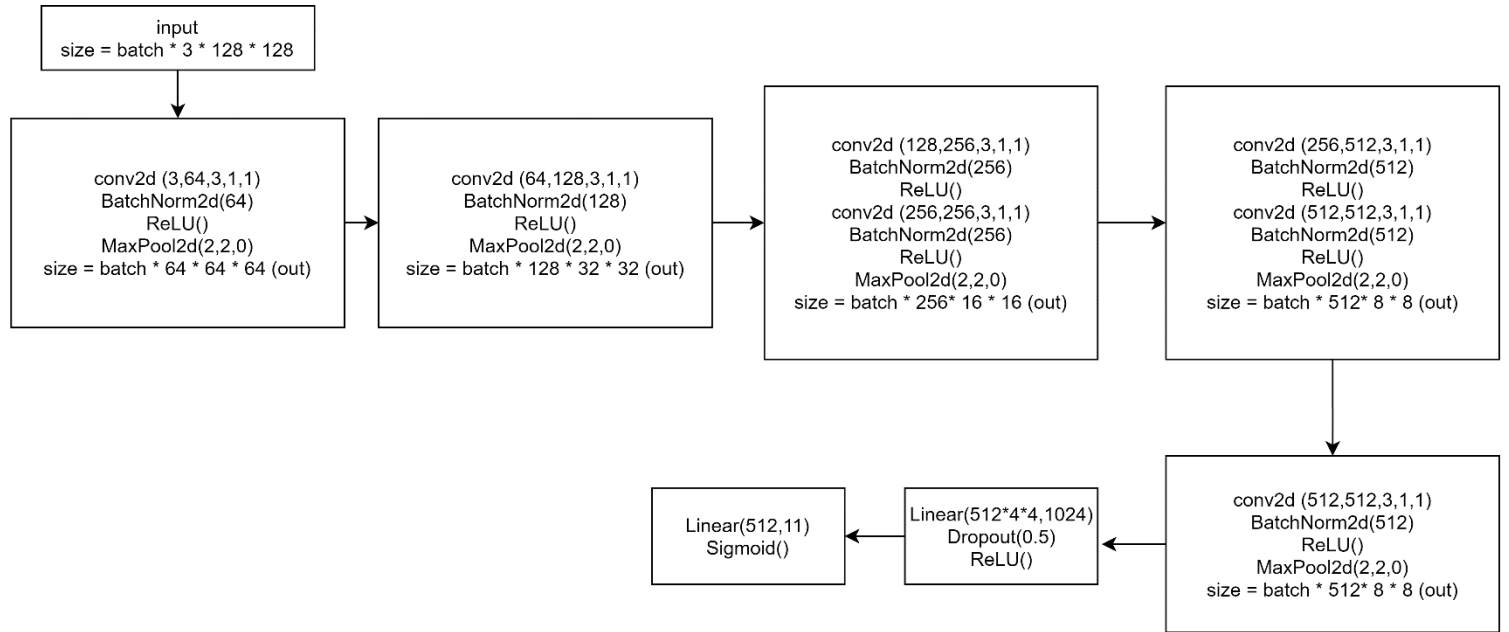
以下矩陣使用 valid 資料集作為 test。並以 true label 作為標準化基準。可以看到達 0.1 以上的有：true=3,pred=0；true=4,pred=0；true1,pred=2; true=7,pred=6；可以看到 3 以及 4 的最容易被此模型分辨錯成 0



圖一



圖二



圖三

