

Environment details

Python 版本是 3.8

其他的 packages 都在 requirements.txt 裡，沒有特別註明版本代表直接使用 pip install -r requirements.txt 即可。

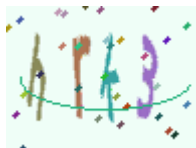
Implementation details

model architecture:

data preprocessing and label vector:

三個 task 我都有針對每張圖片產生對應的灰階圖片，再用 `cv2.connectedComponentsWithStats` 根據我給定的參數去除掉照片中的小點點或是線條，再用 `cv2.connectedComponentsWithStats` 去算出連接在一起的方塊的片機大小，如果太小的就去除，接著就把這個產生出來的 `mask` 乘上 `r,g,b` 就會得到比較乾淨的圖片，如下圖

處理前：



處理後：



把處理完的圖片根據 pytorch 官方給 resnet 設定的 preprocessing ([transforms.Resize(224), transforms.RandomHorizontalFlip(), transforms.ToTensor(), normalize]) 再經過一次處理，最終得到處理好的 training data。

[illegible]

Network:

Network 的部分就是 resnet18 去掉最後一層，再接上一層 dropout layer 和全連接層組成的，全連接層的輸出根據每個 task 的 label vector 大小有所改變，task1 是 10、task2 是 72，而 task3 是 144。

Hyperparameters:

Learning rate 是 $1e-3$ ，和原本的 sample code 一樣，因為在 loss 的折線圖看不出有 lr 太大的跡象，因此維持原本的數值。

Epochs 在 task1 和 task2 都是 100，因為在 loss 的折線圖來看已經收斂的差多了，然後 validation 的 accuracy 也蠻高的(有到 1)，因此沒有做更動，而 task3 因為 validation 的 accuracy 是 0.3214，因此有調高到 200。

used deep learning framework: PyTorch