

Computer Vision HW2 Report

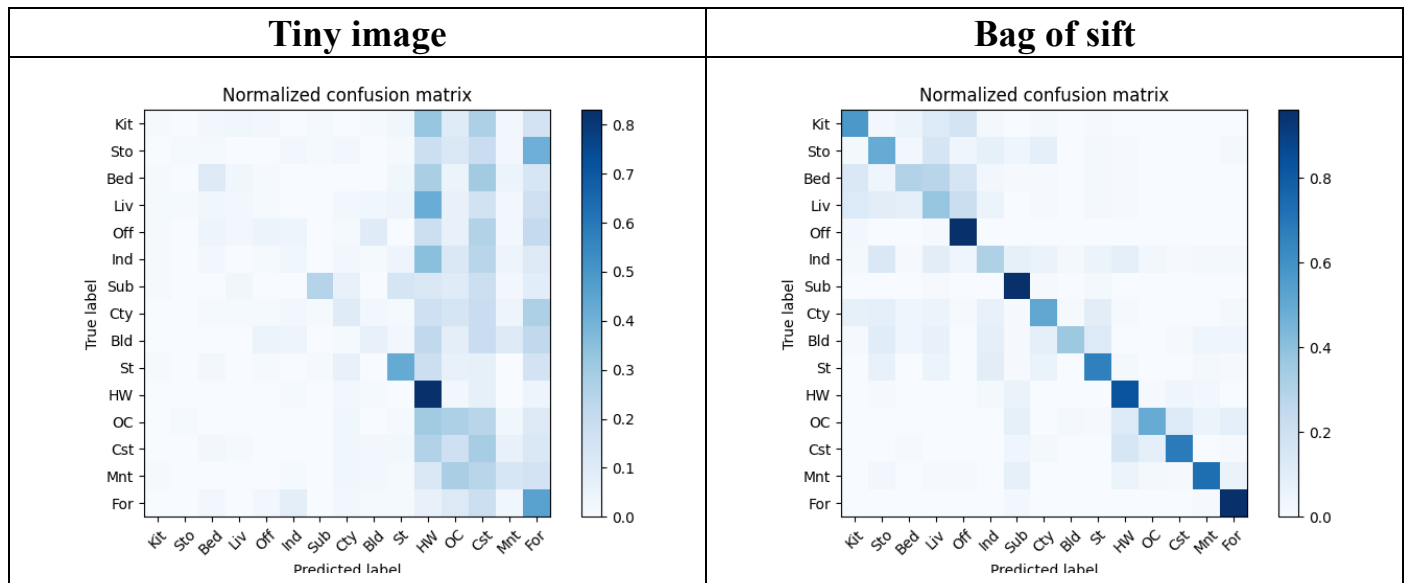
Student ID: R10521802

Name: 李睿蓀

Part 1. (10%)

- Plot confusion matrix of two settings. (i.e. Bag of sift and tiny image representation) (5%)

Ans:



- Compare the results/accuracy of both settings and explain the result. (5%)

Ans:

表 1、Final accuracy

	Tiny image	Bag of sift
accuracy	20.8%	61.3%

如表 1，模型最終達到的精度為 Tiny image 20.8%、Bag of sift 61.3%。

表 2、Accuracy of Tiny image

<div><div></div><div>k</div></div> <div>metric</div>	1	3	5	7	9
L1-norm	20.1%	19.2%	20.8%	20.3%	20.8%
L2-norm	19.1%	18.7%	18.9%	19.3%	18.6%

將原始圖片縮小成 16x16 做為特徵，再以 KNN 為分類器對測試影像分類。如表 2，實驗過程將 k 設定為 1、3、5、7、9，KNN 距離計算指標分為 L1 與 L2 討論。可以觀察到以 L1-norm 計算特徵距離後分類的準確率較 L2-norm 高，此外，模型準確率似乎沒有隨著 K 增加/減少而有顯著的變化，可以看到 K 在等於 5 的時候，模型對這份測試資料的分類準確度最高。

表 3、Accuracy of Bag of sift (K=5, L1-norm)

img_feats dsift	step=[5, 5]	step=[3, 3]	step=[3, 3] [:,3]	step=[3, 3] [:,5]	step=[1, 1] [:,3]	step=[1, 1] [:,5]
vocab dsift						
step=[5, 5]	54.8%	58%	56.1%	53.7%	59.5%	59.2%
step=[3, 3]	54.8%	58%	56.1%	53.7%	59.5%	59.2%

Bag of sift 可分為兩部分，第一部分以 dsift()取得特徵產生 vocab 時，step 設定[5,5]與[3,3]兩種情況，而為了加速 kmeans 流程，以 random()隨機抽樣特徵後再做分群。第二部分在產生訓練與測試影像的特徵(image_feats)，step 設定[5,5]、[3,3]、[1,1]三種情況，[:,3]指的是由 dsift()取得特徵後，每三間隔取一個特徵再與 vocab 計算距離。

表 3 以 **K=5、L1-norm** 討論：產生 image_feats 時設定越小 step，代表圖片萃取的特徵越細緻，模型準確率也越高，而產生 vocab 時設定越小的 step 似乎對準確率沒有顯著的增加。另外，由於 step[1,1]在我的筆電有點跑不動，因此透過[:,3]、[:,5]減少 image_feats 特徵維度，加速與 vocab 距離計算，卻也流失了部分特徵資訊使得分類的準確率下降。因此在表 3 中，step=[1, 1][:,3]的設定使得模型表現最佳。

表 4、Accuracy of Bag of sift (L1-norm)

k	1	3	5	6	7	8	9
accuracy	59.6%	58.6%	59.5%	60%	59.5%	59.9%	59.8%

而將表 3 中分數最高的 step[1,1][:,3]，以不同 k 進行分類，由表 3 可知 k 為 6 時準確率達到 60%。

表 5、Accuracy of Bag of sift (K=5)

img_feats knn metric	step=[1, 1] [:,3] L1	step=[1, 1] [:,3] L2	step=[3, 3] L1	step=[3, 3] L2
L1-norm	59.5%		58%	61.3%
L2-norm	50.7%		48.9%	51.8%

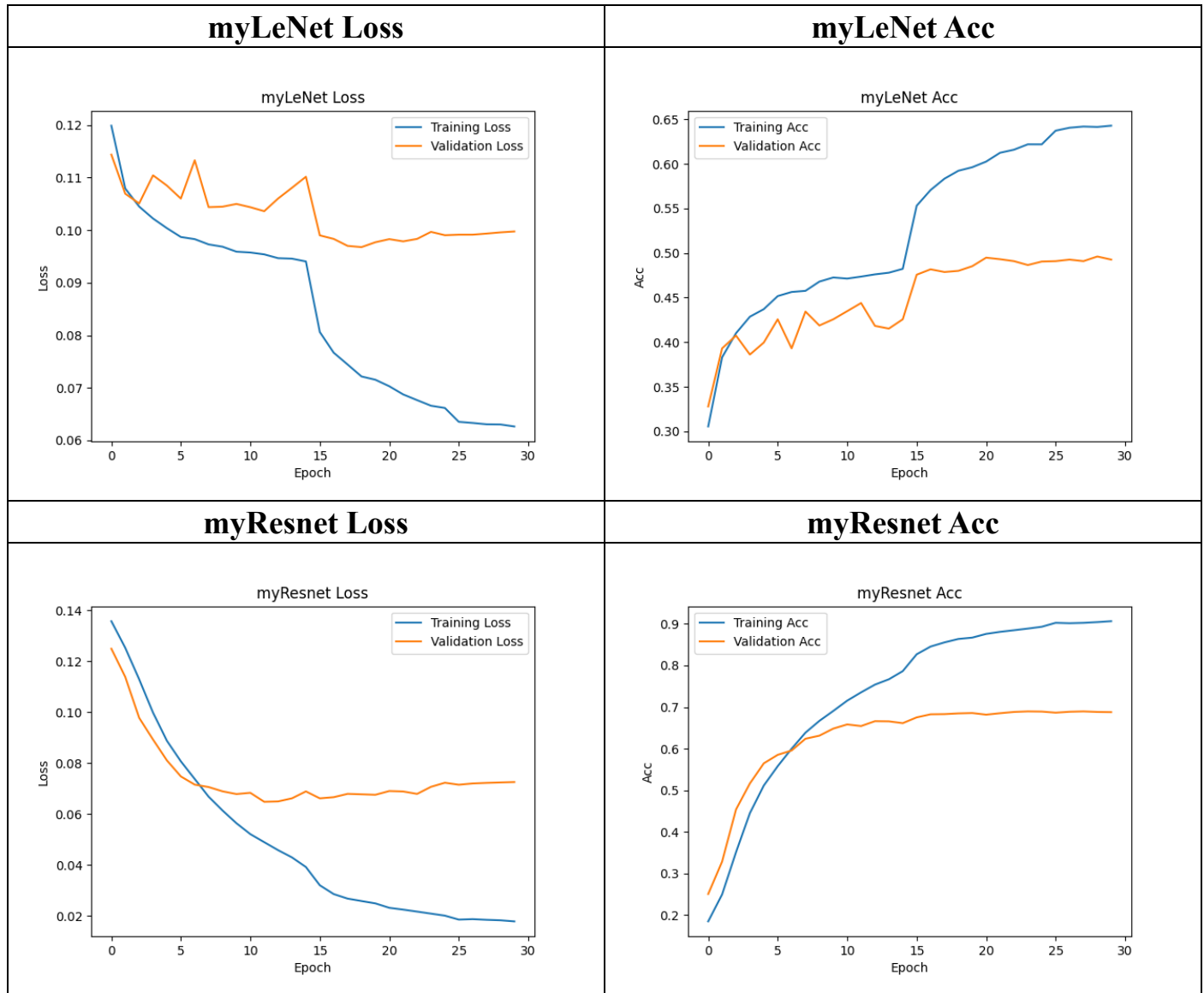
(斜線區代表沒時間做了 QQ)

在表 3、表 4 討論中，image_feats 與群心距離計算以及 knn 距離計算皆採用 L1-norm。在表 5 中可以得知，KNN 採用 L1-norm 分類的準確度較 L2-norm 高，而在 step=[3, 3]、image_feats 與群心距離採 L2-norm 計算的案例中，再以 KNN L1-norm 分類準確度達 61.3%，若有更多時間也許可以將 image_feats-L2、KNN-L1 應用到其他案例上，驗證這種組合的準確率是否較高。

Part 2. (35%)

• Compare the performance on residual networks and LeNet. Plot the learning curve (loss and accuracy) on both training and validation sets for both 2 schemes. 8 plots in total. (20%)

Ans:



myResnet 在卷積層間加入了 residual block，避免在做 backpropagation 時因神經網路太深產生梯度消失，使得訓練效果不佳的問題，與 myLeNet 相比準確度提升不少。由 myLeNet Loss 圖中可見在 epoch 15~30 模型訓練集的 loss 持續下降，驗證資料 loss 則是些微上升，而訓練集的準確度最終約為 65%，顯示出模型不僅在訓練集上表現不佳，在驗證集上的誤差卻是越來越大，也許可以提高模型複雜度或是設定更大迭代次數再觀察模型表現。而在 myResnet Loss 圖中可見在 epoch 6 以後模型訓練集的 loss 持續下降，驗證資料 loss 則是些微上升，而訓練集的準確度最終約為 90%，顯示出模型仍具有 overfitting 的問題，也許可以將清洗數據剔除錯誤的影像、設定 early-stopping、K-fold 驗證等等，以提高模型的準確度。

- Attach basic information of the model you use including model architecture and number of the parameters. (5%)

Ans:

myResnet		
Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 32, 32]	1,792
Conv2d-2	[-1, 64, 32, 32]	36,928
BatchNorm2d-3	[-1, 64, 32, 32]	128
ReLU-4	[-1, 64, 32, 32]	0
residual_block-5	[-1, 64, 32, 32]	0
Conv2d-6	[-1, 64, 32, 32]	36,928
MaxPool2d-7	[-1, 64, 16, 16]	0
BatchNorm2d-8	[-1, 64, 16, 16]	128
ReLU-9	[-1, 64, 16, 16]	0
conv-10	[-1, 64, 16, 16]	0
Conv2d-11	[-1, 64, 16, 16]	36,928
BatchNorm2d-12	[-1, 64, 16, 16]	128
ReLU-13	[-1, 64, 16, 16]	0
residual_block-14	[-1, 64, 16, 16]	0
Conv2d-15	[-1, 128, 16, 16]	73,856
MaxPool2d-16	[-1, 128, 8, 8]	0
BatchNorm2d-17	[-1, 128, 8, 8]	256
ReLU-18	[-1, 128, 8, 8]	0
conv-19	[-1, 128, 8, 8]	0
Conv2d-20	[-1, 128, 8, 8]	147,584
BatchNorm2d-21	[-1, 128, 8, 8]	256
ReLU-22	[-1, 128, 8, 8]	0
residual_block-23	[-1, 128, 8, 8]	0
Linear-24	[-1, 512]	4,194,816
Dropout-25	[-1, 512]	0
ReLU-26	[-1, 512]	0
Dropout-27	[-1, 512]	0
Linear-28	[-1, 10]	5,130
Total params: 4,534,858		
Trainable params: 4,534,858		
Non-trainable params: 0		

Input size (MB): 0.01		
Forward/backward pass size (MB): 4.77		
Params size (MB): 17.30		
Estimated Total Size (MB): 22.08		

- Briefly describe what method do you apply? (e.g. data augmentation, model architecture, loss function, semi-supervised etc.) (10%)

Ans:

在資料增強的部分，主要針對訓練資料集採取隨機水平翻轉 RandomHorizontalFlip() 以及調整影像銳利度 RandomAdjustSharpness() 來增加影像的變異程度。損失函數則採用分類問題常見的 CrossEntropyLoss。在模型架構的部分，主要實作了 3 個 residual block，中間兩層 conv layer 加了 MaxPool、BatchNorm 和 ReLU，最後接了兩層全連接層以輸出結果。第一次訓練的模型有很嚴重的過擬合，因此在全連接層與 ReLU 中間加了兩個 dropout(0.3)，減少過擬合問題。最後優化後的 myResnet 模型提高了準確度，在 public test 也達到了 strong baseline 的要求(78.8%>77.92%)。