CV HW2 report

姓名: 梁中瀚

學號: r09922a02

系級: 資工 人工智慧 碩一

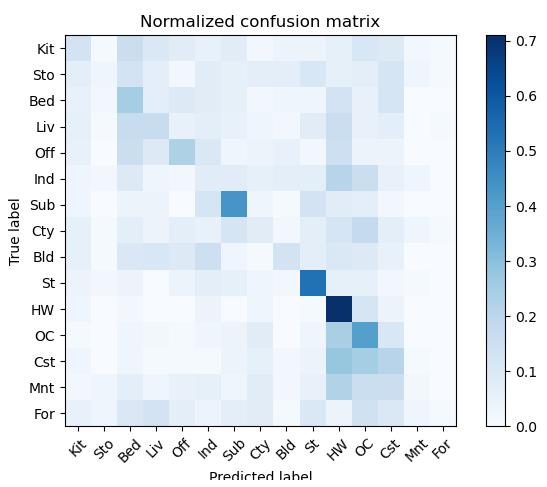
# Part 1

1. Report accuracy of two settings

* Tiny images + KNN: 0.2273
* Bag of sift + KNN: ?

1. Plot confusion matrix of two settings

* Tiny images + KNN:



* Bag of sift + KNN:

1. Compare the results of both settings and   
   explain the result

Tiny images的做法僅僅是將原圖的resolution降低，再pixel-wise的計算兩張圖的Euclidean distance，當作KNN當中的距離求出最接近的training data。

Bag of sift的做法會用sift求出多個descriptors再計算預先建好的vocabulary的出現次數當作feature，相較於Tiny image中pixel-wise的計算方式，這些sift descriptor具有更好的比較意義。

從Confusion matrix中可以看到，使用tiny images方法預測的結果，在HW的類別上有最高的準確度，應該是因為高速公路的圖片，有比較相近的結構，所以就算resolution降低KNN依然很有效。而其它類別的預測準確度都不高，尤其是Mnt與For(森林與高山)的類別式最少被預測到的，可能是因為住兩個類別圖片間的結構相差很大，所以把low resolution的圖片當作feature會讓KNN無法判斷與其他類別的不同。

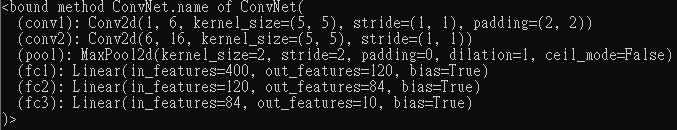
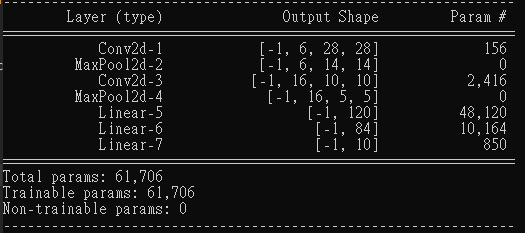
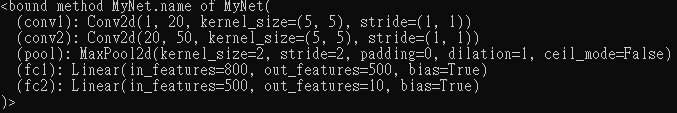
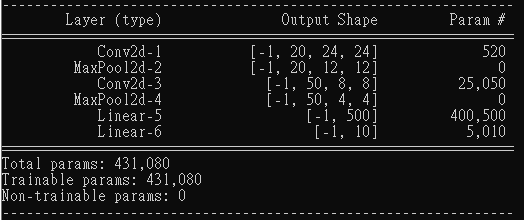
Bag of sift相比於tiny image有較高的準確率。

1. Bag of sift parameters (Best one)
2. Constructing vocabulary
   * Number of sift descriptors sample from each image: 250
   * Sift step: [10, 10]
   * Sift size: [3, 3]
   * Number of vocabulary: 600
3. Calculating histogram of each image
   * Number of sift descriptors sample from each image: 750
   * Sift step: [3, 3]
   * Sift size: [3, 3]
4. KNN
   * K: 250
5. Files

* p1.py
* get\_tiny\_images.py
* build\_vocabulary.py
* get\_bags\_of\_sifts.py
* nearest\_neighbor\_classify.py
* vocab.pkl
* train\_image\_feats.pkl
* test\_image\_feats.pkl

# Part 2

1. Print the network architectures &   
   number of parameters of both models

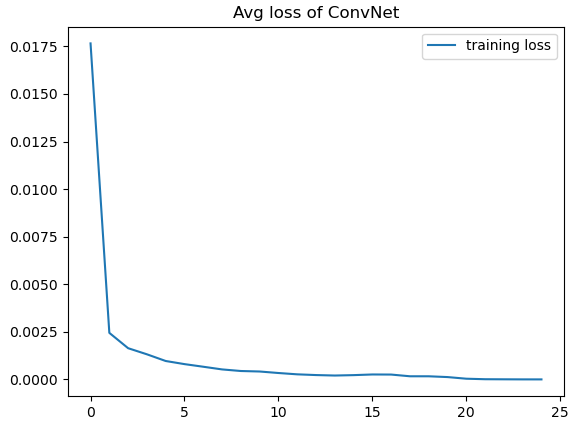
* Baseline model (LeNet-5):
  + Network architecture  
    
  + Number of parameters: 61706  
    
* Improved model (MyNet):
  + Network architecture  
    
  + Number of parameters 431080  
    

1. Plot the learning curve (loss, accuracy) of   
   the training process(train/validation)

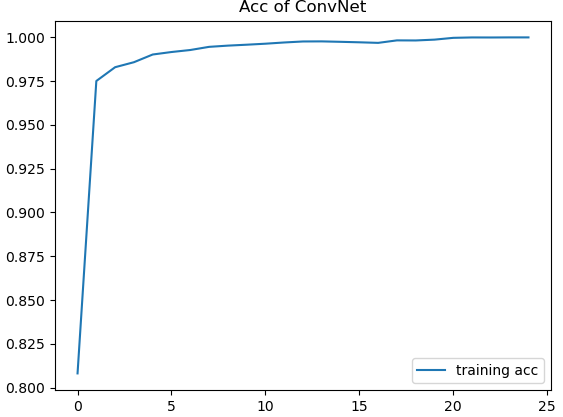
兩個model的訓練過程皆使用25個epoch

* Baseline model (LeNet-5):

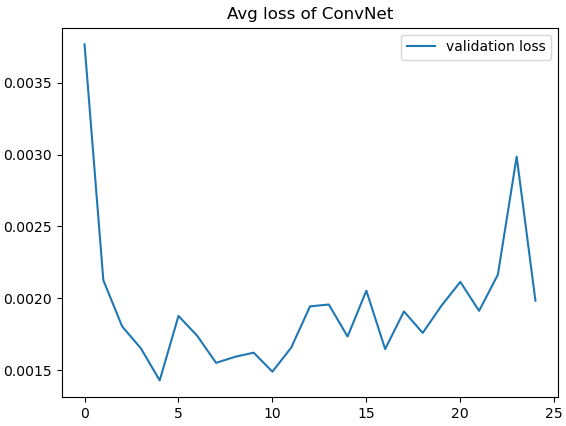
1. Learning curve of training loss



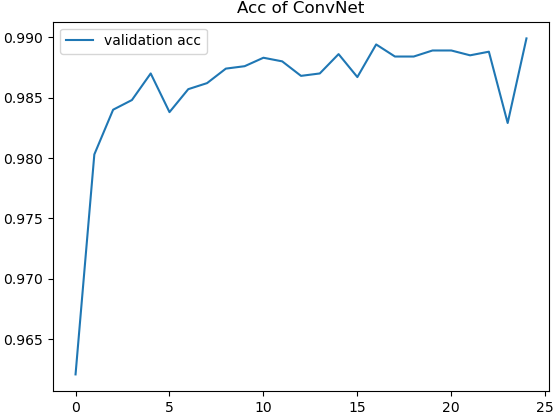
1. Learning curve of training accuracy



1. Learning curve of validation loss

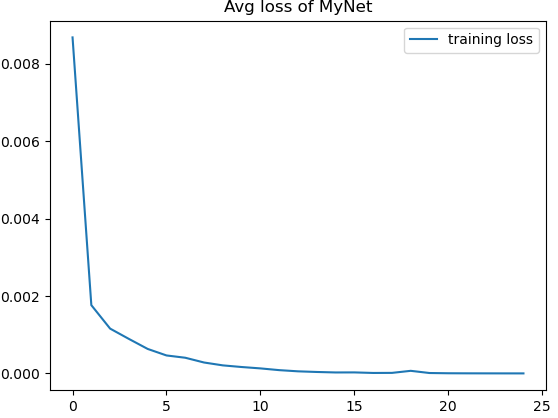


1. Learning curve of validation accuracy

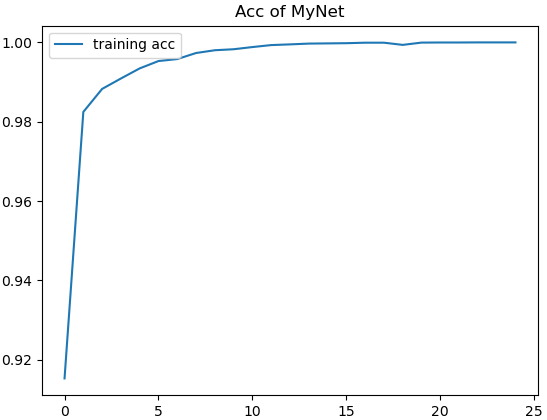


* Improved model (MyNet):

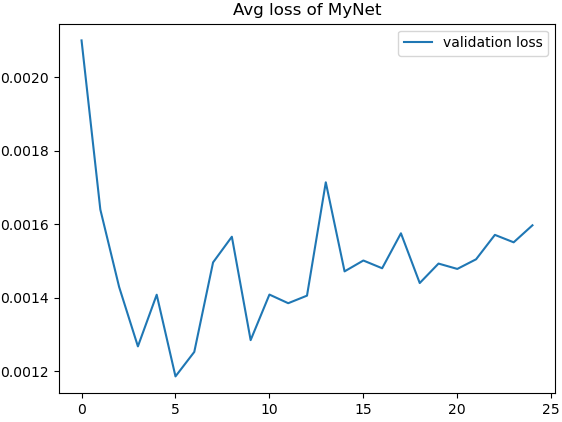
1. Learning curve of training loss



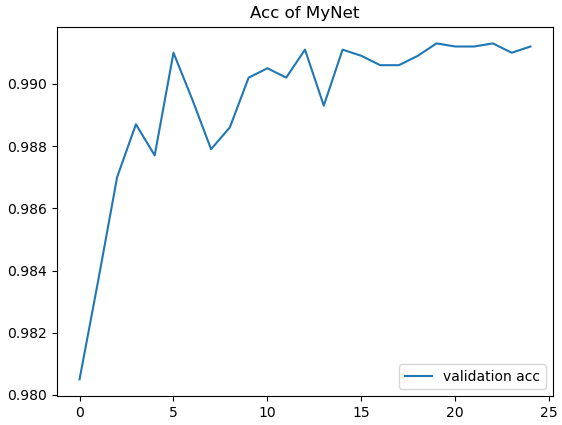
1. Learning curve of training accuracy



1. Learning curve of validation loss



1. Learning curve of validation accuracy



1. Compare the results of both model and explain the result

LeNet在validation set上最好的預測結果為: 98.99%的準確率。網路的架構非常單純，只有三種layer，convolution、pooling、fully connected。MyNet在validation set上最好的預測結果為: 99.12%。MyNet與baseline model不同的地方在於，增加捲積層的kernel數量，所以參數數量才會增加，並且在最後一層加入了log\_softmax，讓所有輸出的總合為1。

1. Files

* model.py
* eval.py
* train.py
* ConvNet.pth: baseline model
* MyNet.pth: improved model