Machine Learning

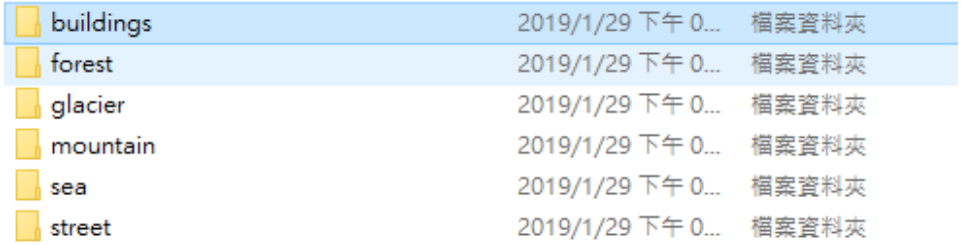
4105931

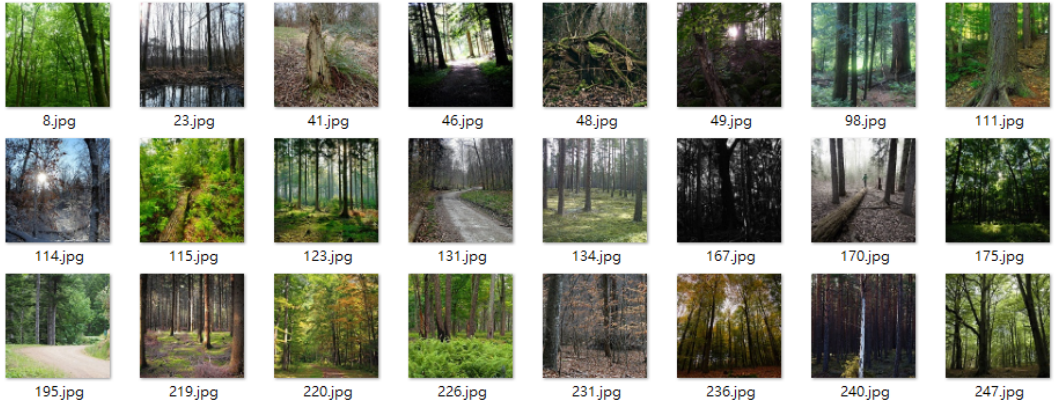
Assignment 1 – Scene Recognition

Dataset Details

- Goal
  - Classify image data of natural scenes around the world
- Task
  - image classification (6 classes)
- Training / Testing
  - around 14k / 3k
- Reference
  - https://www.kaggle.com/puneet6060/intel-image-classification

Intel Image Classification - Image Scene Classification of Multiclass





範例程式簡介

- models.py
  - VGG16模型架構
- dataset.py
  - Dataset 讀取方式/方法
- train.py
  - 訓練模型架構之程式
- test.py
  - 測試訓練完模型架構之程式

參數調整- train.py

1. Reproducibility – 初始權重產生
   - torch.manual_seed()
2. Batch size 每次丟多少張
   - 批次訓練資料量大小
   - 1~128 (1, 20, 50…)
3. Epoch 重複丟的次數
   - 訓練次數 (1, 20, 50…) 最佳化演算法
4. Optimizer(SGD, Adam,…等等) ref: https://pytorch.org/docs/stable/optim.html
5. Learning rate 調整
   - 0 ~ 1
   - 建議從0.1開始往下試

```
12  ##REPRODUCIBILITY
13  torch.manual_seed(0)
14  torch.backends.cudnn.deterministic = True
15  torch.backends.cudnn.benchmark = False
```

```
29      train_set = IMAGE_Dataset(Path(DATASET_ROOT), data_transform)
30      data_loader = DataLoader(dataset=train_set, batch_size=32, shuffle=True,
31      #print(train_set.num_classes)
```

```
34      best_model_params = copy.deepcopy(model.state_dict())
35      best_acc = 0.0
36      num_epochs = 50
37      criterion = nn.CrossEntropyLoss()
38      optimizer = torch.optim.SGD(params=model.parameters(), lr=0.001, momentum=0.9)
```

執行步驟

1. 設定參數
2. 執行 train.py (執行完畢後會有自動儲存最佳model .pth檔)
3. 在test.py 修改model path
4.

```
 9   CUDA_DEVICES = 0
10   DATASET_ROOT = './seg_train'
11   PATH_TO_WEIGHTS = './model-0.90-best_train_acc.pth'
```

實驗結果觀察

- train.py 執行結果

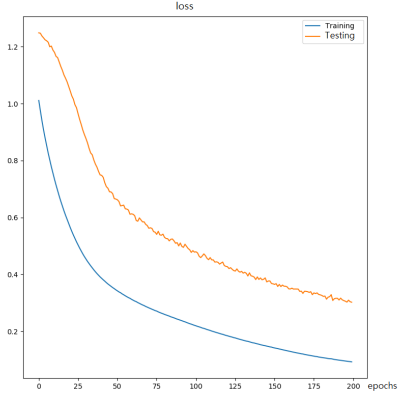
```
Epoch: 1/50
----------
Training loss: 1.2757   accuracy: 0.4823

Epoch: 2/50
----------
Training loss: 0.9526   accuracy: 0.6219

Epoch: 3/50
----------
Training loss: 0.8242   accuracy: 0.6789

Epoch: 4/50
----------
Training loss: 0.7252   accuracy: 0.7258

Epoch: 5/50
----------
```

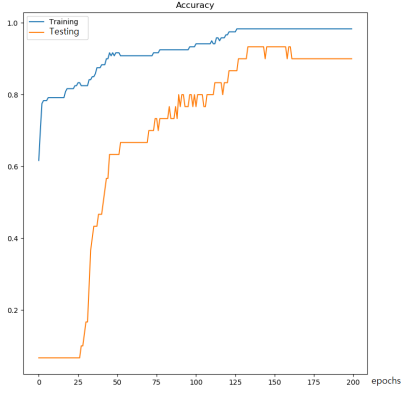
```
Accuracy on the ALL test images: 90 %
Accuracy of buildings : 82 %
Accuracy of forest : 98 %
Accuracy of glacier : 80 %
Accuracy of mountain : 89 %
Accuracy of   sea : 95 %
Accuracy of street : 95 %
```

作業要求

- 訓練過程之 traing loss,acc 曲線圖
- 測試結果之截圖

```
Accuracy on the ALL test images: 90 %
Accuracy of buildings : 82 %
Accuracy of forest : 98 %
Accuracy of glacier : 80 %
Accuracy of mountain : 89 %
Accuracy of   sea : 95 %
Accuracy of street : 95 %
```





作業要求

- 架設PyTorch、Python執行環境
- 利用範例程式、Dataset與下面參數設定方式，完成兩次分類結果
  - 使用一組給定參數值:
    - seed = 123, epochs = 50, batch_size=32, SGD, lr = 0.01
  - 調整參數設定，使分類結果高於給定參數結果
- 兩次訓練過程，分別產生下列兩張結果圖 (Matlab繪圖)
  - Training Accuracy(每一個epoch的訓練結果) + Testing Accuracy (每十個epoch跑一次測試)
  - Training Loss
- 最終最佳的測試結果準確率與執行截圖

x 2

2x2 = 4

Assignment #1 – Scene Classification

- You need to hand in your source code and report
- The report should cover:
  - Method description – What is your strategy for parameter selection?
  - Experimental results
    - Two figures of accuracy and loss curves
    - Testing accuracy
  - Discussion
  - Problem and difficulties
- Upload assignment #1 before 3/27 11:59 pm (Wed)
- File format – zip all your files into a single file: studentID_hw1_version, ex: 602410143_hw1_v1

Assignment Rules

- Late policy
  - You will get 20% deduction of your scores per day.
  - It means if the assignment is delayed one day for 80%, two days for 60%,…, five days for 0% .
- No-copy policy
  - Copying is strictly forbidden in our class.
  - Once the assignment is confirmed by TA as COPY, the score will be 0%.