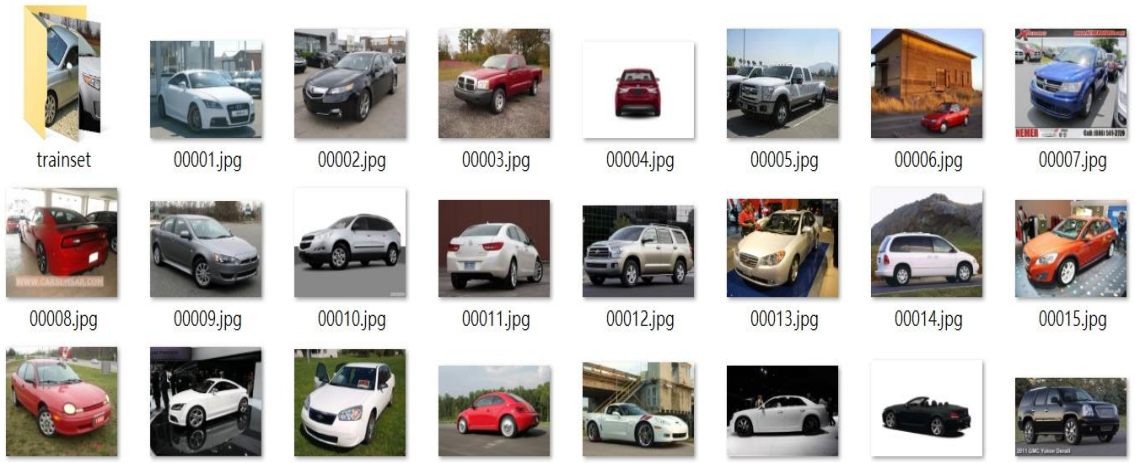
Stanford car classification

照片切割

- 使用**matlab**將cars_test和cars_train裡面的照片分別依據cars_train_annos.mat和cars_test_annos_withlabels.mat裡面的bbox_x1、bbox_y1、bbox_x2、bbox_y2資訊將照片切割



Ex：以cars_train為例

```
for i=1:length(annotations)
    bbox_x1=annotations(i).bbox_x1;
    bbox_y1=annotations(i).bbox_y1;
    fname=annotations(i).fname;
    width=annotations(i).bbox_x2-bbox_x1;
    height=annotations(i).bbox_y2-bbox_y1;
    img=imread(fname);
    result=imcrop(img,[bbox_x1,bbox_y1,width,height]);
    filename=strcat('./trainset',fname);
    imwrite(result,filename);
```

end

程式實作&執行

- train.py、test.py、dataset.py

  train.py-參照作業1的train.py，將模型修改成resnet101，且每執行兩個epoch就將當下model儲存

  test.py-參照作業1的test.py

  dataset.py-從mat檔讀取切好照片路徑以及對應的class

- 分成param_false和param_true資料夾，裡面各自會有三個python和devkit
- 並將切好的照片放進testset和trainset資料夾中

```
wayne1116@gslave03:~/hw2$ ls
param_false  param_false_v2  param_true  param_true_v2  testset  trainset
```

- 執行test.py – python test.py

模型參數

- 隨機初始權重、epoch、SGD、learning rate、batch_size

```
torch.manual_seed(0)
torch.backends.cudnn.deterministic=True
torch.backends.cudnn.benchmark=False
```

```
best_model_params=copy.deepcopy(resnet101.state_dict())
best_acc=0.0
num_epochs=30
criterion = nn.CrossEntropyLoss()
optimizer=torch.optim.SGD(params=resnet101.parameters(), lr=0.01, momentum=0.9)
```

```
data_loader=DataLoader(dataset=train_set, batch_size=32, shuffle=True, num_workers=1)
```

- 建立resnet101模型，num_class=196

```
resnet101=models.resnet101(pretrained=True)
fc_features=resnet101.fc.in_features
resnet101.fc=nn.Linear(fc_features,196)
resnet101=resnet101.cuda(CUDA_DEVICES)
resnet101.train()
```

設定pretrained

- pretrained=False (不使用預訓練模型)

```
resnet101=models.resnet101(pretrained=False)
fc_features=resnet101.fc.in_features
resnet101.fc=nn.Linear(fc_features,196)
```

- pretrained=True (使用預訓練模型)

```
resnet101=models.resnet101(pretrained=True)
fc_features=resnet101.fc.in_features
resnet101.fc=nn.Linear(fc_features,196)
```

*預訓練模型(pre-trained model)
是前人為了解決類似問題所創造出來的模型，並不用從零開始訓練一個新模型

最佳模型測試結果-best_model.pth

- pretrained=False (不使用預訓練模型)

```
(demo) wayne1116@gslave03:~/hw2/param_false_v2$ python test.py
total: 8041
Accuracy on the ALL test images : 29 %
Accuracy of class   0 : 56 %
Accuracy of class   1 : 40 %
Accuracy of class   2 : 23 %
```

```
test_loss: 4.4324
```

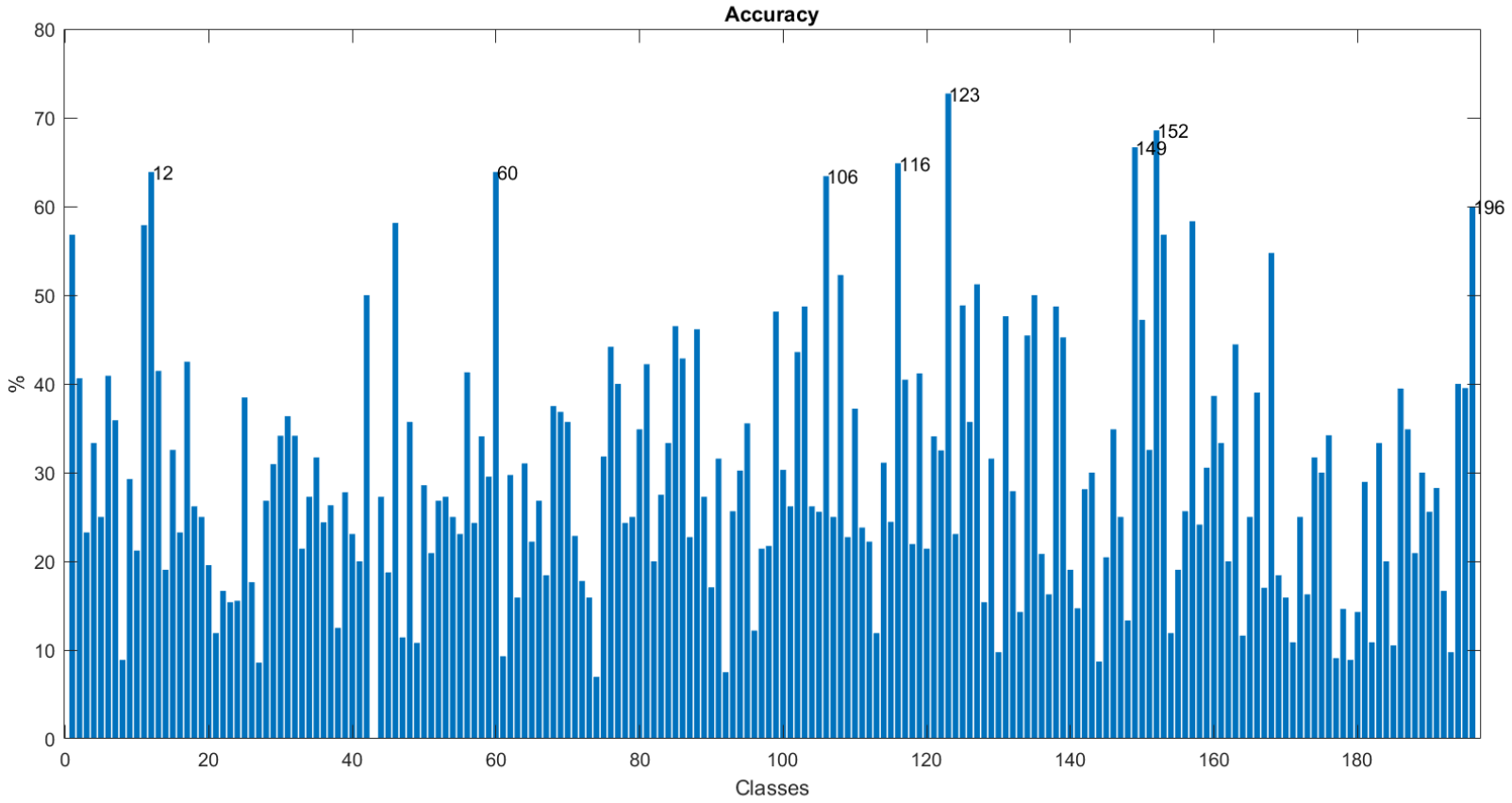
- pretrained=True (使用預訓練模型)

```
(demo) wayne1116@gslave03:~/hw2/param_true_v2$ python test.py
total: 8041
Accuracy on the ALL test images : 91 %
Accuracy of class   0 : 93 %
Accuracy of class   1 : 90 %
Accuracy of class   2 : 95 %
Accuracy of class   3 : 90 %
```

```
test_loss: 0.3578
```

最佳模型對classes分類準確率

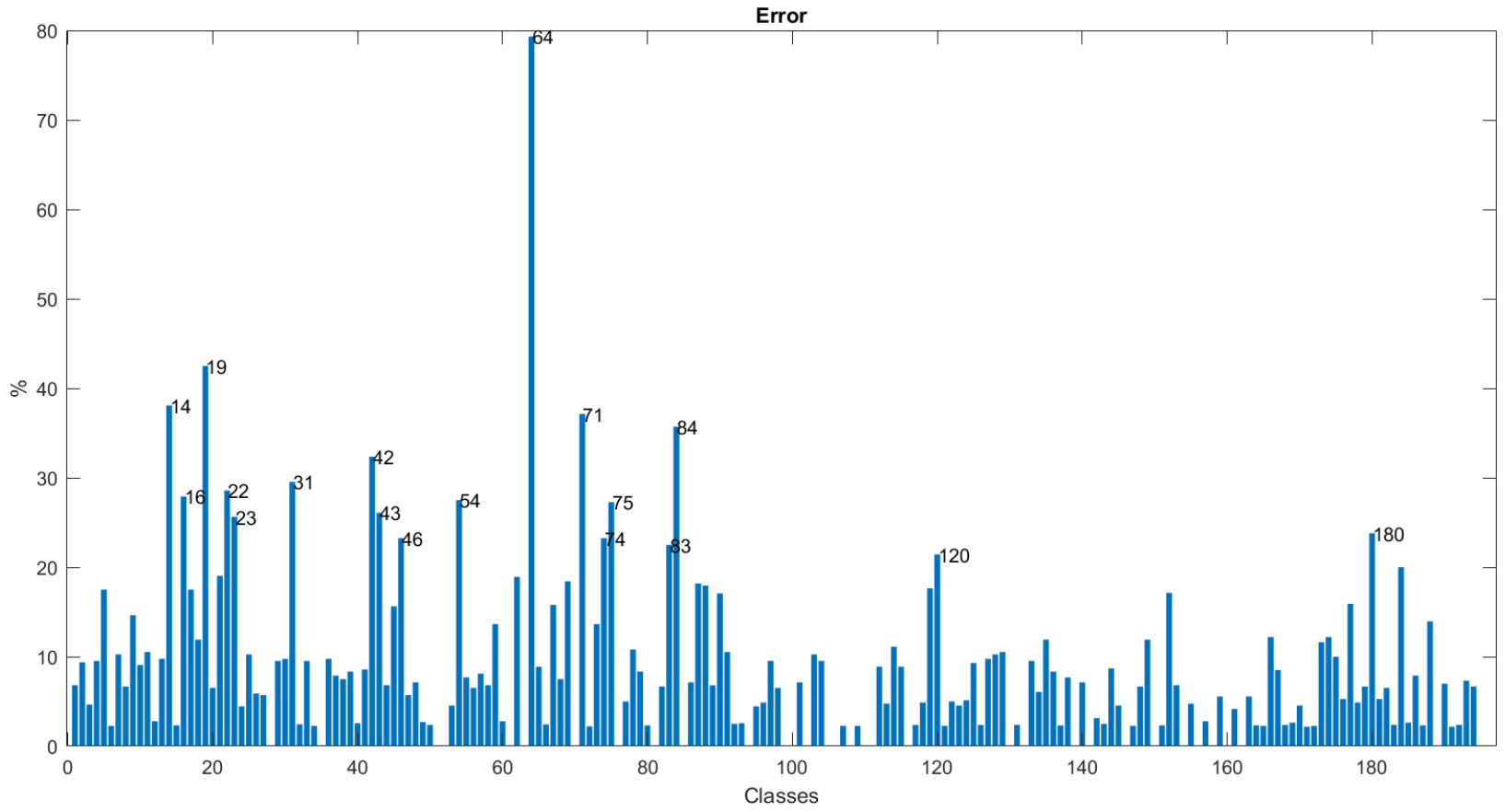
- pretrained=False (不使用預訓練模型)



*數字標示的class即為準確率高於**60%**

最佳模型對classes分類<u>失誤率</u>

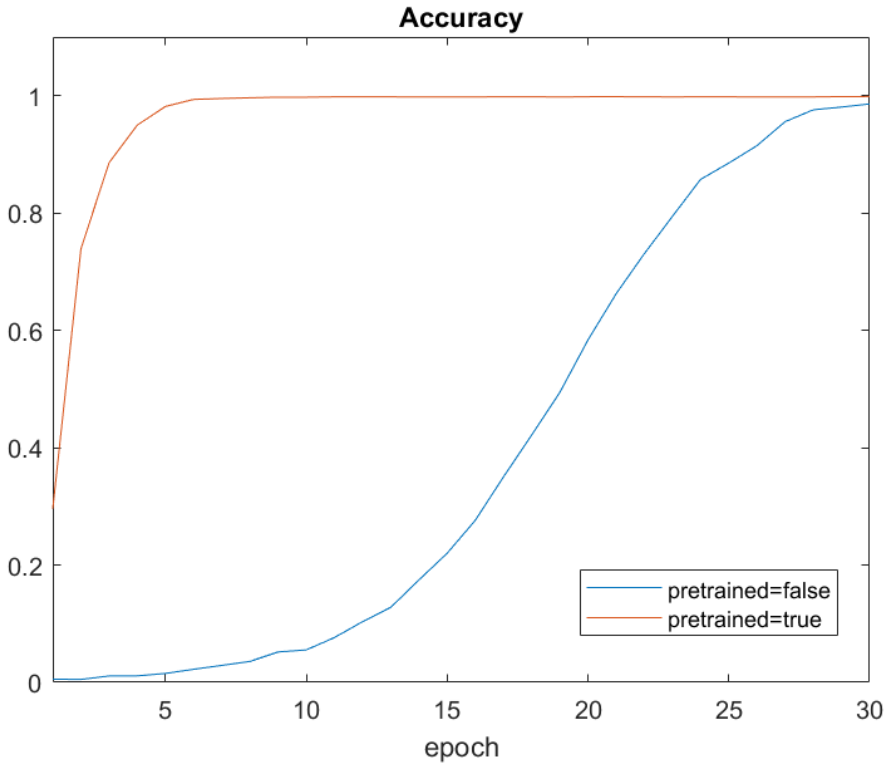
- pretrained=True (使用預訓練模型)



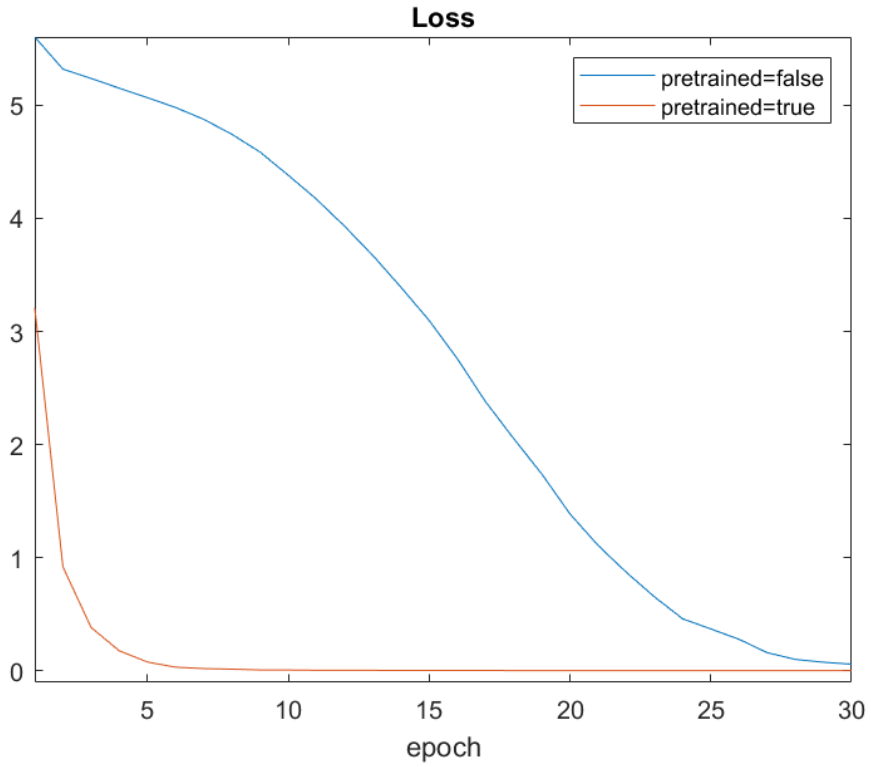
*數字標示的class即為失誤率<u>高於</u>**20%**

模型訓練結果

- training accuracy

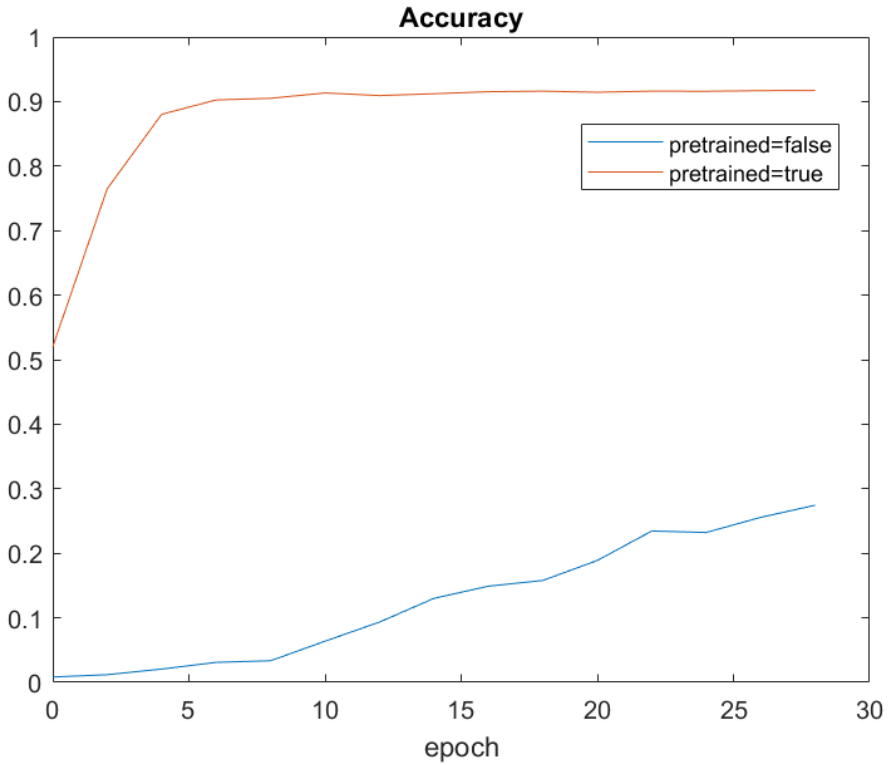


- training loss

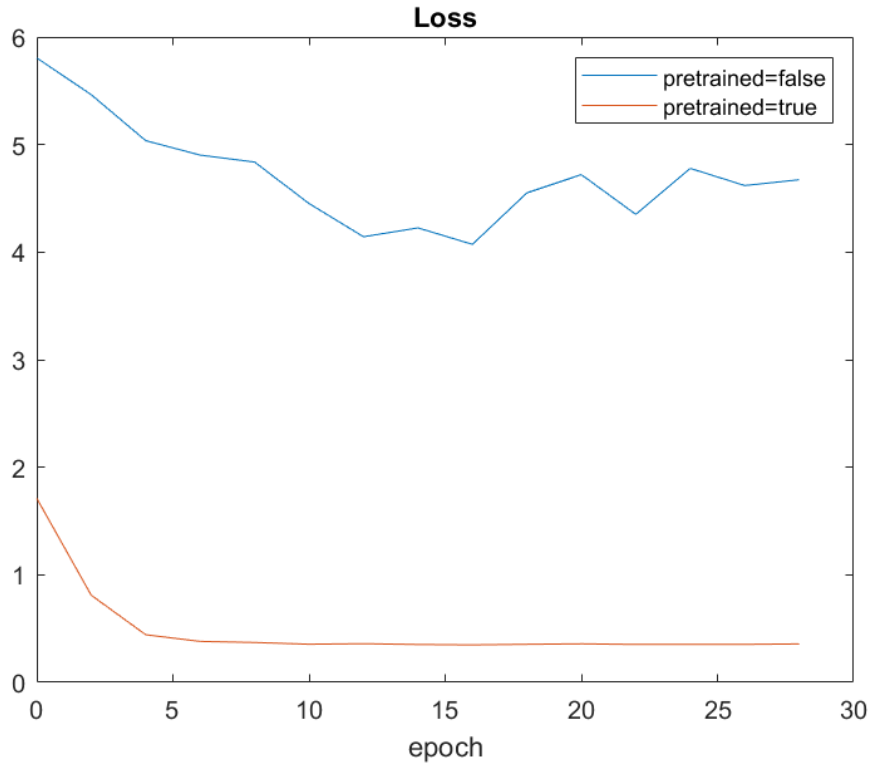


模型測試結果

- testing accuracy



- testing loss



*每兩個epoch就將當下的model存下來，並執行test.py計算model的準確率和loss

結論Discussion

1. 對於pretrained=False/True的模型訓練(training)結果顯示，pretrained=True的訓練準確率在epoch=4的時候達到90%以上，而pretrained=False的訓練準確率epoch=26才能達到90%以上，所以對預先載入預訓練模型會比較快有較高的訓練準確率

2. 對於pretrained=False/True的模型訓練(training)結果顯示，pretrained=False/True的loss都隨著epoch的增長而下降，但是pretrained=True明顯下降快速

3. 對於pretrained=False/True的模型測試(testing)結果顯示，pretrained=True/False的測試準確率與訓練準確率都有明顯呈正相關性，而pretrained=True在epoch=4以後都維持在90%

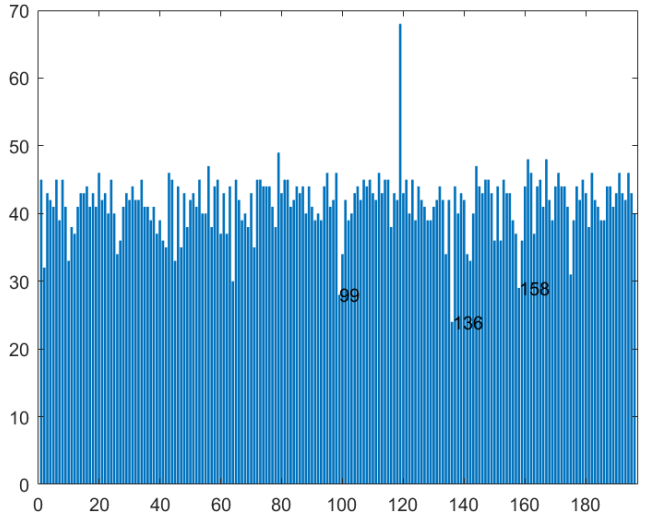
4. 對於pretrained=False/True的模型測試(testing)結果顯示，pretrained=True的loss是隨著epoch的增長而下降，但是pretrained=False的loss卻是有高低起伏的明顯變化

結論Discussion

5. 對於pretrained=False的最佳模型分類結果分布，其準確率普遍在60%以下

6. 對於pretrained=True的最佳模型分類結果分布，其準確率普遍高於90%，其中對於class=64的分類準確率明顯較低，原本可能原因是class=64訓練資料太少，但是經matlab畫出數據分布後發現比class=64少的class=99、136、158其準確率也很高，可能是class=64與其它車種的特徵不明顯

✓ Class64



✓ Class訓練資料數量分布

問題與困難Problem and difficulties

1. 原本不太清楚要如何取得mat檔的相關資訊，最後經由網路查詢之後才了解

2. 原本在作業1中的train.py是用IMAGE_Dataset函式將照片路徑及labels讀進訓練模型。

當只有修改num_class以及使用for file in _dir.glob('*')讀取檔案並標記對應的class時，最後訓練完後的準確率都很奇怪，後來發現當執行for file in _dir.glob('*')，不一定是從0001.jpg開始讀取，所以在對應的class時就會出錯，後來改正以後明顯的正常許多

參考資料Reference

1. matlab圖像裁剪imcrop()

   https://blog.csdn.net/SMF0504/article/details/51811130

2. 預訓練模型解釋

   https://kknews.cc/zh-tw/news/q2zpa5y.html

3. Pytorch model

   https://www.itread01.com/content/1542034103.html