Deep Learning and Practice

Lab 4 309505018 郭俊廷

1. Introduction (20%)

使用ResNet18/ResNet50 來訓練黃斑部病變的classification problem, 有五種 class 0~4,表示病變的嚴重程度, 並比較有pretrained weight和無pretrained weight結果的差異

2. Experiment setups (30%)

A. The details of your model (ResNet)

首先用torchvision import ResNet18和RestNet50, 然後把output layer設成5個 class, 因為RestNet18/50 本身已經是一個設好參數的模型, 所以feature extraction model (先train最後一層layer), 再finetuning整個model

ResNet18:

```
class ResNet18(nn.Module):
    def __init__(self,num_class,pretrained=False):
        super(ResNet18,self).__init__()
        self.model=models.resnet18(pretrained=pretrained)
        if pretrained:
            for param in self.model.parameters():
                param.requires_grad=False
        num_neurons=self.model.fc.in_features
        self.model.fc=nn.Linear(num_neurons,num_class)

    def forward(self,X):
        out=self.model(X)
        return out
```

ResNet50:

```
class ResNet50(nn.Module):
    def __init__(self,num_class,pretrained=False):
        super(ResNet50,self).__init__()
        self.model=models.resnet50(pretrained=pretrained)
        if pretrained:
            for param in self.model.parameters():
                param.requires_grad=False
        num_neurons=self.model.fc.in_features
        self.model.fc=nn.Linear(num_neurons,num_class)

    def forward(self,X):
        out=self.model(X)
        return out
```

B. The details of your Dataloader

要實作自己的dataloader,

__init__(self, img_path, mode):主要是去讀csv檔的圖片名稱, 印出總共幾張圖片, 定義transforms.

__getitem__(self, index):主要是依據圖片名稱用PIL讀取對應圖片,然後做transform, 來實現data augmentation, 並讀圖片的label

```
class RetinopathyDataSet(Dataset):
    def __init__(self, img_path, mode):
        self.img_path = img_path
        self.mode = mode
        self.img_names=np.squeeze(pd.read_csv('train_img.csv' if mode=='train' else
                                              'test_img.csv').values)
        self.labels=np.squeeze(pd.read_csv('train_label.csv' if mode=='train' else
                                           'test_label.csv').values)
        assert len(self.img_names)==len(self.labels),'length not the same'
        self.data_len=len(self.img_names)
        self.transformations=transforms.Compose([transforms.RandomHorizontalFlip(),
        transforms.RandomVerticalFlip(),transforms.ToTensor(),transforms.Normalize
        ((0.3749, 0.2602, 0.1857),(0.2526, 0.1780, 0.1291))])
        print(f'>> Found {self.data_len} images...')

    def __len__(self):
        return self.data_len

    def __getitem__(self, index):
        single_img_name=os.path.join(self.img_path,self.img_names[index]+'.jpeg')
        single_img=Image.open(single_img_name)  # read an PIL image
        img=self.transformations(single_img)
        label=self.labels[index]

        return img, label
```

C. Describing your evaluation through the confusion matrix

用一個5x5的矩陣在testing data時統計數量, 然後對每一列做normalize

```
confusion_matrix=np.zeros((num_class,num_class))

    with torch.set_grad_enabled(False):
        model.eval()
        correct=0
        for images,targets in loader_test:
            images,targets=images.to(device),targets.to(device,dtype=torch.long)
            predict=model(images)
            predict_class=predict.max(dim=1)[1]
            correct+=predict_class.eq(targets).sum().item()
            for i in range(len(targets)):
                confusion_matrix[int(targets[i])][int(predict_class[i])]+=1
        acc=100.*correct/len(loader_test.dataset)

    confusion_matrix=confusion_matrix/confusion_matrix.sum(axis=1).reshape
    (num_class,1)
```

3. Experimental results (30%)

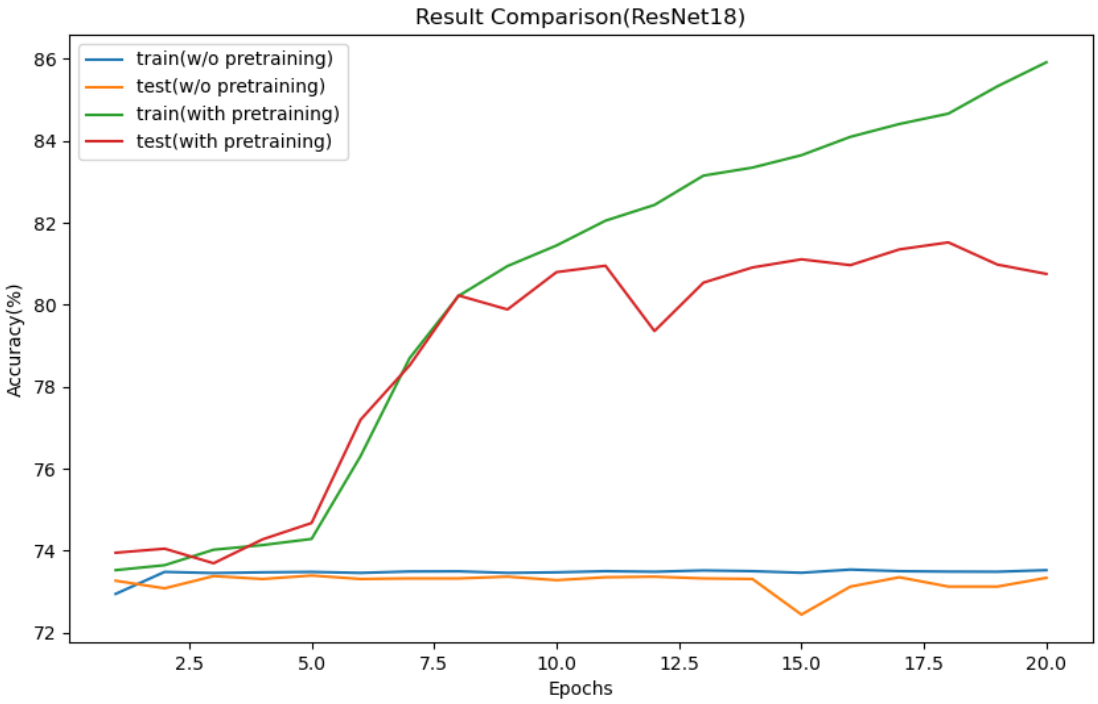
A. The highest testing accuracy
(Screenshot, Anything you want to present)

ResNet 18: 81.28%
ResNet 50: 82.70%

```
(dlp) jackkuo@lab708-Default-string:~/DLPhw4$ python3 hw4.py
1.8.0
True
>> Found 7025 images...
ResNet18 Testing accuracy: 81.28113879003558
>> Found 7025 images...
ResNet50 Testing accuracy: 82.70462633451957
(dlp) jackkuo@lab708-Default-string:~/DLPhw4$
```

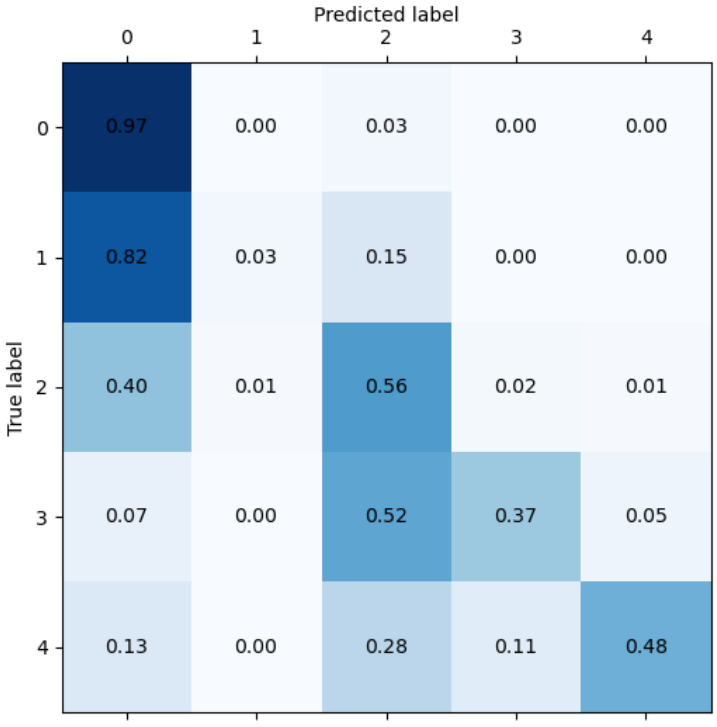
B. Comparison figures
(Plotting the comparison figures)(RseNet18/50, with/without pretraining)

ResNet18:

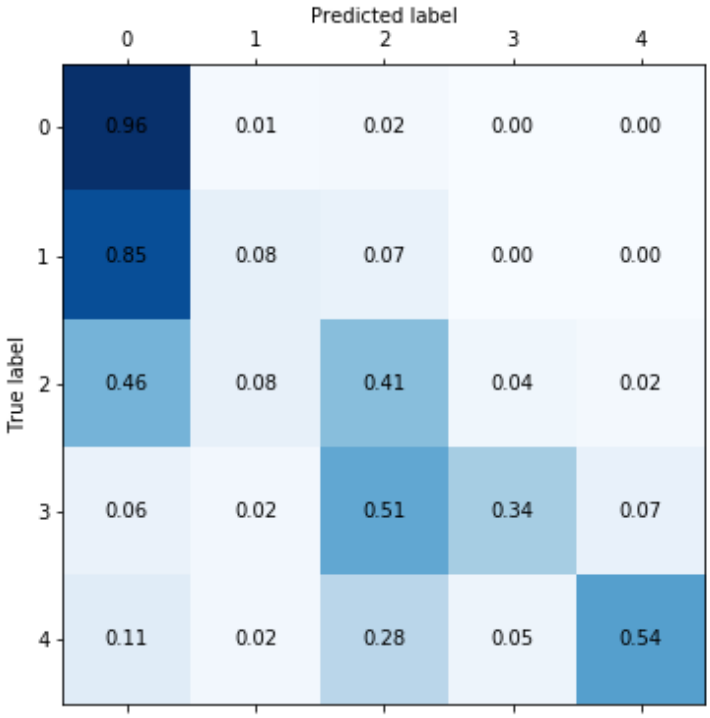


ResNet50:

ResNet 18:
(with pretrained weights)



ResNet 50:
(with pretrained weights)



4. Discussion (20%)

一開始沒有用pretrained的時候, 正確率大概73%左右(以ResNet18為例),
其實如果全部猜class 0正確率也有70%, 所以以73%來說表現算差的,
加了pretrained後正確率提升到82%左右, 算還行