311551180_楊佳誠

Introduction

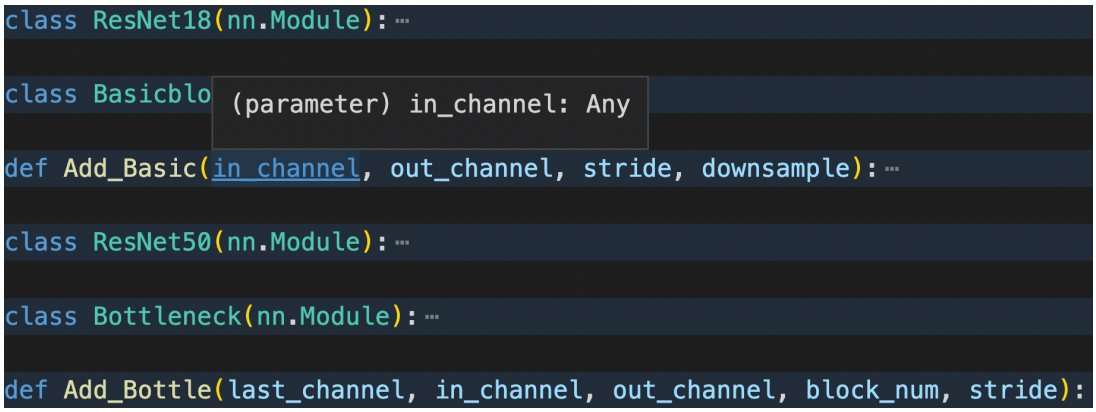
In this lab, we need to analyze diabetic retinopathy through machine learning. By using ResNet18 and ResNet50, we can identify to which class the picture belongs. Additionally, experiments on both pre-trained and non-pretrained models are necessary.

Experiment setups

A. The details of your model (ResNet)

Misunderstood, so I have done my Resnet model by hand. The basic structure of Resnet are designed by a main class, which contain all information of models. The block class are designed to build the Resnet easily by produce some same blocks repeatedly. The hyperparameter follows the TA's recommendation.

- Batch size= 4
- Learning rate = 1e-3
- Epochs = **10** (resnet18), **5** (resnet50)
- Optimizer: SGD Momentum = 0.9 Weight_decay = 5e-4
- Loss function: Cross Entropy Loss



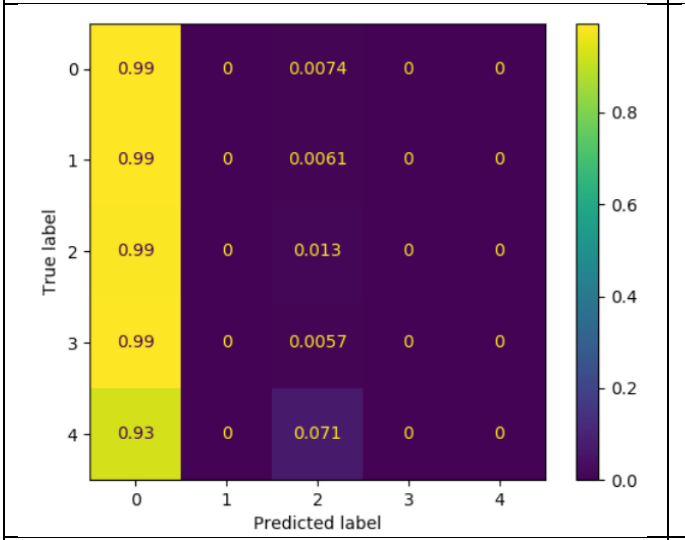
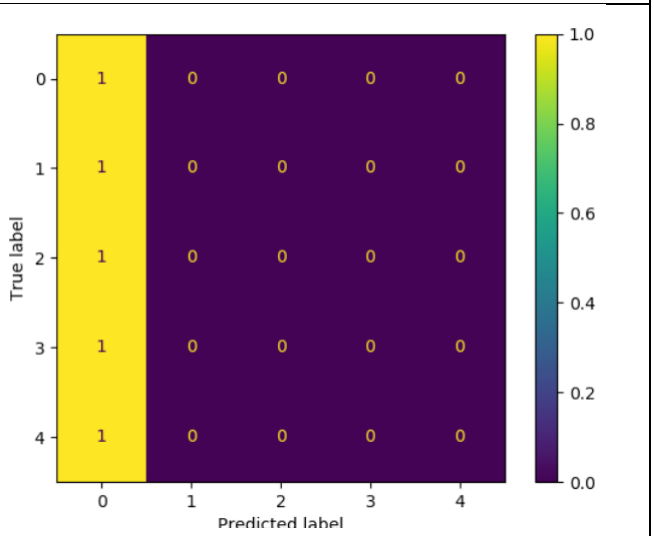
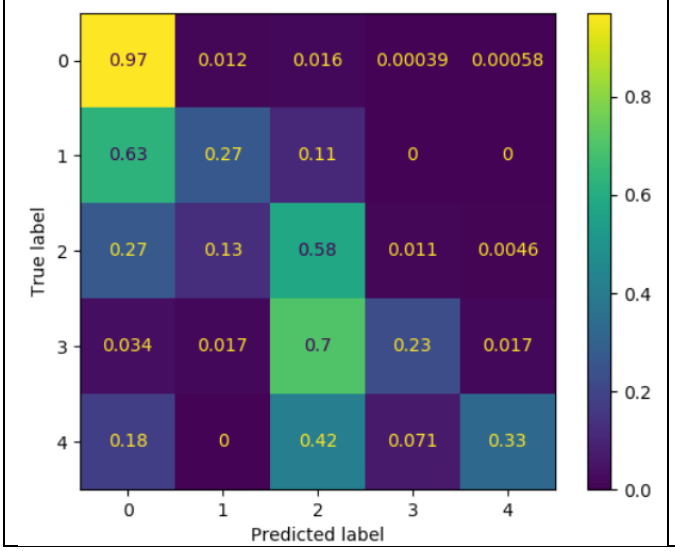
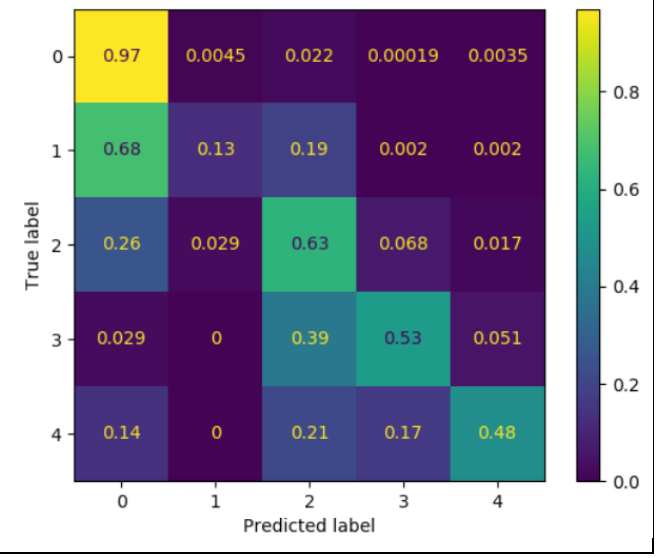
B. The details of your Dataloader

The dataloader is composes of several part.

- getdata(mode) : Given mode, and return the csv information to get image and the correspond label easily.
- __getitem__(self,index) : get the path from the correspond index, do the transformation to the data, and return the modified data and correspond label.
- __len__(self) : return the size of dataset.

C. Describing your evaluation through the confusion matrix

As you can see, the main different between pretrained and not pretrained model is that, the pretrained model has greater value along the diagonal, which means the classifier can identify the difference between classes. Whereas, the model without pretrained tend to predict all the picture as the “0” label. In my opinion, this is because in this dataset, most of the data are labeled as “0”, “0” must be the more safe choice for the model. More action must be taken in this situation, such as using different sampling methods.

Resnet18 w/o pretrained	Resnet50 w/o pretrained
	
Resnet18 with pretrained	**Resnet50 with pretrained**
	

Data Preprocessing

A. How you preprocessed your data?

- To exclude bad data, we need to remove the broken JPEG files in the dataset. It is crucial to exclude them using the 'getData' function at the beginning. There are 3 jpeg files are excluded.

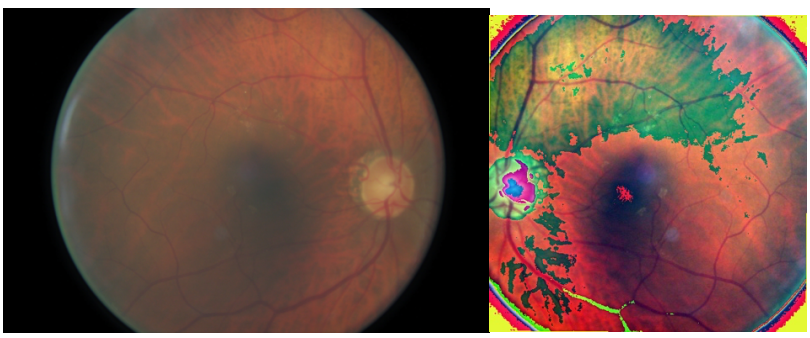
```
try:
    with Image.open(path) as img:
        try:
            img = img.convert('RGB')
            # Do something with the image here
        except PIL.UnidentifiedImageError:
            print('Error: Unidentified image at', path,index)
        except Exception:
            print('Error: Failed to convert image at', path,index)
except OSError:
    print('Error: Failed to open file at', path,index)
except Exception:
    print('Error: Unknown error occurred while processing', path,index)
img = np.squeeze(img.values)
img = np.delete(img, [15663,15970,17616])
label = np.squeeze(label.values)
label = np.delete(label, [15663,15970,17616])
```

- The transform method is shown as bellowed. I use “CenterCrop” to exclude most of the black regions, and the crop size I choose is based on the height of the jpg file. Also, some data augmentation are used recommended by TAs and Chatgpt.

```
transform=transforms.Compose([
    transforms.CenterCrop(height),
    transforms.Resize(512),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(degrees=15),
    transforms.ToTensor(),
    transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))
])
```

B. What makes your method special?

- Exclude broken file make the program work correctly.
- Using CenterCrop can remove much useless pixels, and the parameter is set as height to ensure most of the eye can be covered after crop. The picture remains square so that is can be resized and rotate easily.
- Using data augmentation make the model more robust and able to capture more features.
- Using normalize technique can enhance the speed of training process.
- Here are the results after the transform process.



Experimental results

A. The highest testing accuracy

The highest testing accuracy comes from the ResNet18 pretrained model. It can reach 84.09%.

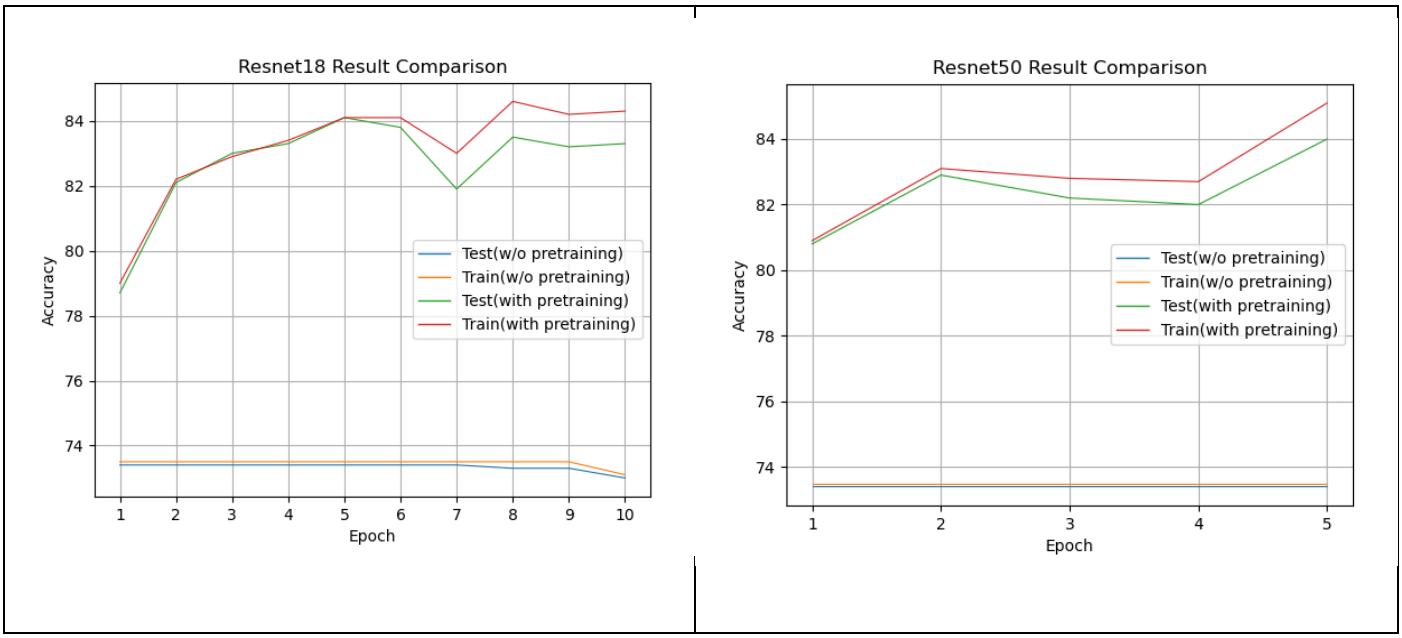
```
Train Epoch: 5 [28000/28097 (99.64%)]   Loss: 1.069524
Train Epoch: 5 [28040/28097 (99.79%)]   Loss: 0.649504
Train Epoch: 5 [28080/28097 (99.93%)]   Loss: 0.205061

Train set: Average loss: 0.1240505920, Accuracy: 23631/28097 (84%)

Test set: Average loss: 0.0157, Accuracy: 5908/7026 (84.09%)

Train Epoch: 6 [0/28097 (0.00%)]        Loss: 1.032969
Train Epoch: 6 [40/28097 (0.14%)]       Loss: 0.322965
Train Epoch: 6 [80/28097 (0.28%)]       Loss: 0.799759
```

B. Comparison figures



Discussion

Data-driven machine learning needs much of dataset to let the model learn the feature, preprocessing data becomes an important task. In this homework, some broken data needed to be removed. Maybe in the future, there will exist much more problems when dealing with the raw data. It is crucial to handle all kinds of situation carefully.

No matter in without pretrained's Resnet18 or Resnet50, the performance stuck in a certain accuracy. I think that the model doesn't learn well at all, and predict all the image to a certain class. It might take a long time to reach the better performance to be competitive to the pretrained model.