

Computer Vision HW2 Report

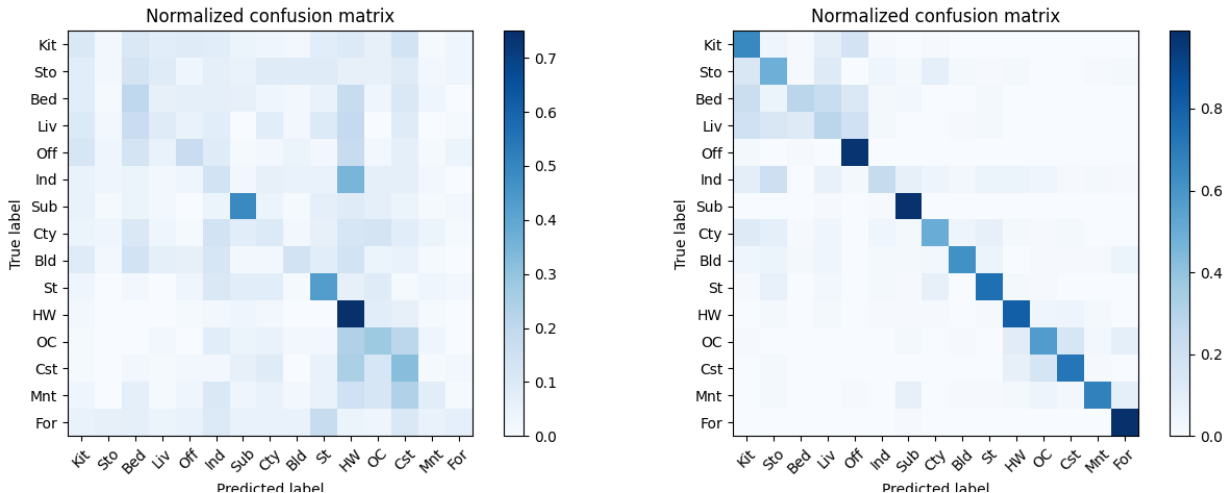
Student ID: R13943124

Name: 施伯儒

Part 1. (10%)

- Plot confusion matrix of two settings. (i.e. Bag of sift and tiny image) (5%)

Ans:



左/右: tiny image/Bag of sift

- Compare the results/accuracy of both settings and explain the result. (5%)

Ans:

Tiny image accuracy: 22%

Bag of Sift accuracy: 63.53%

Tiny image 只是將圖片縮小並拿 pixel 值來當 features，但是 bags of sift 中間通過 SIFT 抽取 features，並且最後 KNN 用來比較的 feature 是每張圖的 vocabulary 分布，自然比完全沒特別處理圖片的 tiny image 還來的高。

Part 2. (25%)

- Report accuracy of both models on the validation set. (2%)

Ans:

	A: MyNet	B: ResNet18
accuracy	0.8448	0.9074

- Print the network architecture & number of parameters of both models. What is the main difference between ResNet and other CNN architectures? (5%)

Ans:

MyNet:

Layer (type:depth-idx)	Param #
MyNet	--
└Sequential: 1-1	--
└Conv2d: 2-1	896
└BatchNorm2d: 2-2	64
└ReLU: 2-3	--
└Conv2d: 2-4	9,248
└BatchNorm2d: 2-5	64
└ReLU: 2-6	--
└MaxPool2d: 2-7	--
└Conv2d: 2-8	18,496
└BatchNorm2d: 2-9	128
└ReLU: 2-10	--
└Conv2d: 2-11	36,928
└BatchNorm2d: 2-12	128
└ReLU: 2-13	--
└MaxPool2d: 2-14	--
└Conv2d: 2-15	73,856
└BatchNorm2d: 2-16	256
└ReLU: 2-17	--
└Conv2d: 2-18	147,584
└BatchNorm2d: 2-19	256
└ReLU: 2-20	--
└MaxPool2d: 2-21	--
└Sequential: 1-2	--
└Linear: 2-22	1,049,088
└BatchNorm1d: 2-23	1,024
└ReLU: 2-24	--
└Linear: 2-25	5,130
Total params: 1,343,146	
Trainable params: 1,343,146	
Non-trainable params: 0	

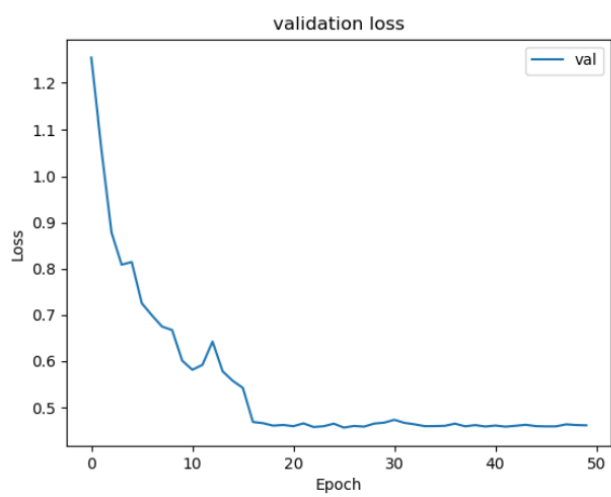
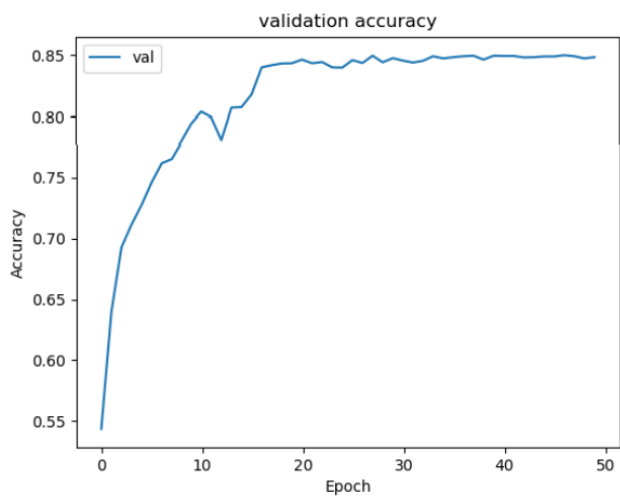
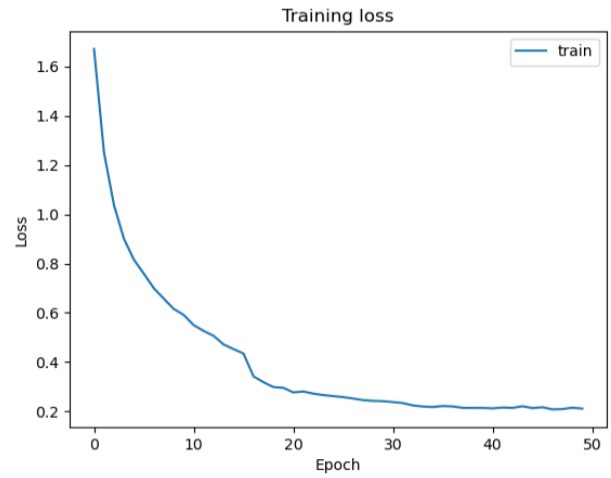
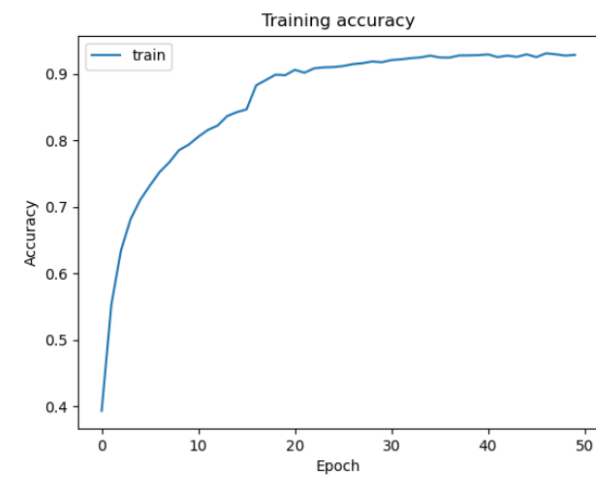
ResNet18:

```
=====
Layer (type:depth-idx)                               Param #
=====
ResNet18                                              --
├─ResNet: 1-1                                          --
│   └─Conv2d: 2-1                                     1,728
│       └─BatchNorm2d: 2-2                           128
│           └─ReLU: 2-3                               --
│               └─Identity: 2-4                       --
│                   └─Sequential: 2-5                 --
│                       └─BasicBlock: 3-1             73,984
│                           └─BasicBlock: 3-2         73,984
│                               └─Sequential: 2-6      --
│                                   └─BasicBlock: 3-3   230,144
│                                       └─BasicBlock: 3-4 295,424
│                                           └─Sequential: 2-7 --
│                                               └─BasicBlock: 3-5 919,040
│                                                   └─BasicBlock: 3-6 1,180,672
│                                                       └─Sequential: 2-8 --
│                                                           └─BasicBlock: 3-7 3,673,088
│                                                               └─BasicBlock: 3-8 4,720,640
│                                                                   └─AdaptiveAvgPool2d: 2-9 --
│                                                                       └─Linear: 2-10 5,130
=====
Total params: 11,173,962
Trainable params: 11,173,962
Non-trainable params: 0
=====
```

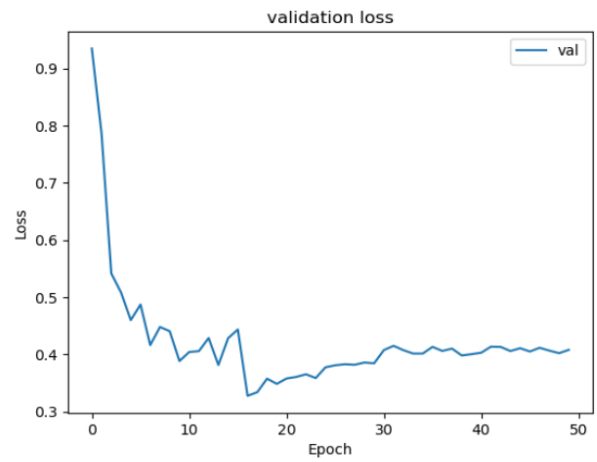
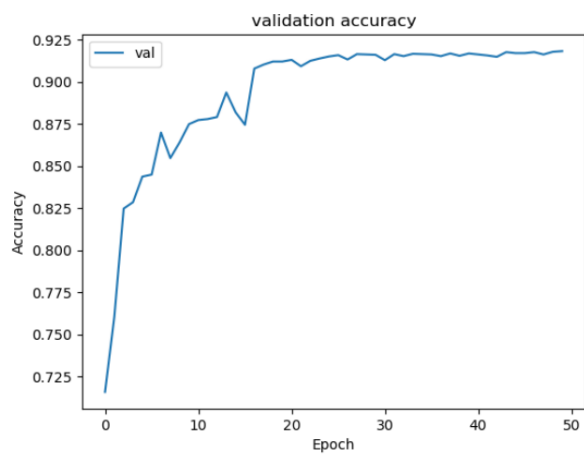
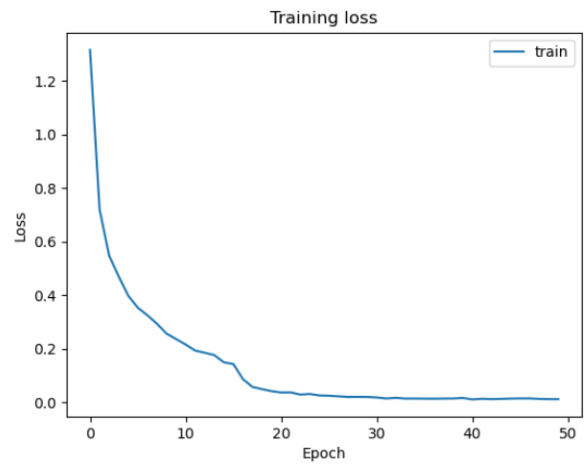
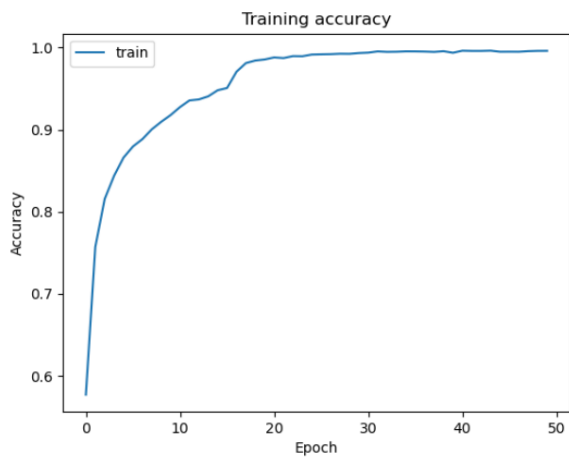
• Plot four learning curves (loss & accuracy) of the training process (train/validation) for both models. Total 8 plots. (8%)

Ans:

MyNet:



ResNet18:



• Briefly describe what method do you apply on your best model? (e.g. data augmentation, model architecture, loss function, etc) (10%)

Ans:

我的 best model 是 resnet18，去除了 maxpool layer 和縮小預設 kernel 大小

如圖：

```
# If pretrained=True, it loads the weights that were pretrained on ImageNet.
# ResNet18 model with pretrained weights
self.resnet = models.resnet18(pretrained=True)

# Modify the first convolutional layer to accept 32x32 images (rather than the typical 224x224 images for ImageNet).
# The original ResNet18 uses a 7x7 kernel with a stride of 2. Here, we're using a 3x3 kernel with stride=1.
# This is to better handle smaller input images like CIFAR-10, which are 32x32.
# We're setting bias=False because BatchNorm will handle the bias term.
self.resnet.conv1 = nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=1, bias=False)

# Remove the first max pooling layer and replace it with Identity().
# The maxpool layer in ResNet18 operates on 224x224 images, but CIFAR-10 images are 32x32.
# So we remove it to avoid reducing spatial dimensions too much.
# Identity() is a no-operation layer, which will effectively skip this layer.
self.resnet.maxpool = nn.Identity()

# Modify the final fully connected layer to output 10 classes instead of 1000.
# ResNet18 originally outputs 1000 classes (for ImageNet), but CIFAR-10 has only 10 classes.
self.resnet.fc = nn.Linear(self.resnet.fc.in_features, 10)
```

原始 resnet18 的 kernel size 是 7x7，但因為 input image 是 32x32，所以採用比較小的 kernel
另外，原始的 resnet18 的 maxpool layer 是作用在 224x224，但是 input image 是 32x32，於是這邊採用 Identity()這個 no-operation layer，直接跳過 maxpool，避免損失太多的 spatial dimensions。