**Computer Vision HW2 Report**

Student ID: R12K41025

Name: 杜冠廷

**Part 1. (10%)**

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

**Ans:**

|  |  |
| --- | --- |
| **Tiny Image** | **Bag of SIFT** |
|  |  |

**• Compare the results/accuracy of both settings and explain the result. (5%)Ans:**

|  |  |
| --- | --- |
| **Accuracy** | |
| Tiny Image | 0.23 |
| Bag of SIFT | 0.616 |

As expected, the performance of Tiny Image is worse than that of Bag of SIFT because the Tiny Image approach can only capture very basic image features, such as color distribution and brightness. In contrast, Bag of SIFT can extract more detailed and discriminative features, such as textures, edges, and local shapes. The process of building the vocabulary and generating the vocab.pkl file takes a significant amount of time. To reduce this, we limited the vocabulary construction to only 1/10 of the images from the training set and extracted a maximum of 80 feature points per image.

**Part 2. (25%)**

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

**Ans:**

|  |  |
| --- | --- |
| **Accuracy** | |
| MyNet | 0.7464 |
| ResNet18 | 0.9234 |

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

**Ans:**

|  |  |
| --- | --- |
| **MyNet** | |
|  | |
| **ResNet18** | |
|  |  |

**• 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:**

Since ResNet18 is designed for classifying the large-scale dataset, I resized the images on CIFAR-10 to 224×224 to prevent excessive information loss caused by the Max Pooling layer. Additionally, due to the limited amount of training data, I applied several data augmentation techniques to prevent overfitting:

1. transforms.RandomResizedCrop(224, scale=(256/480, 1.0))
2. transforms.RandomHorizontalFlip()
3. transforms.RandomRotation(10)
4. transforms.RandomGrayscale()`