

Homework2 Report

Task A

1. Training & testing curves are as below (*relative path: ./task_A_curves.png*):

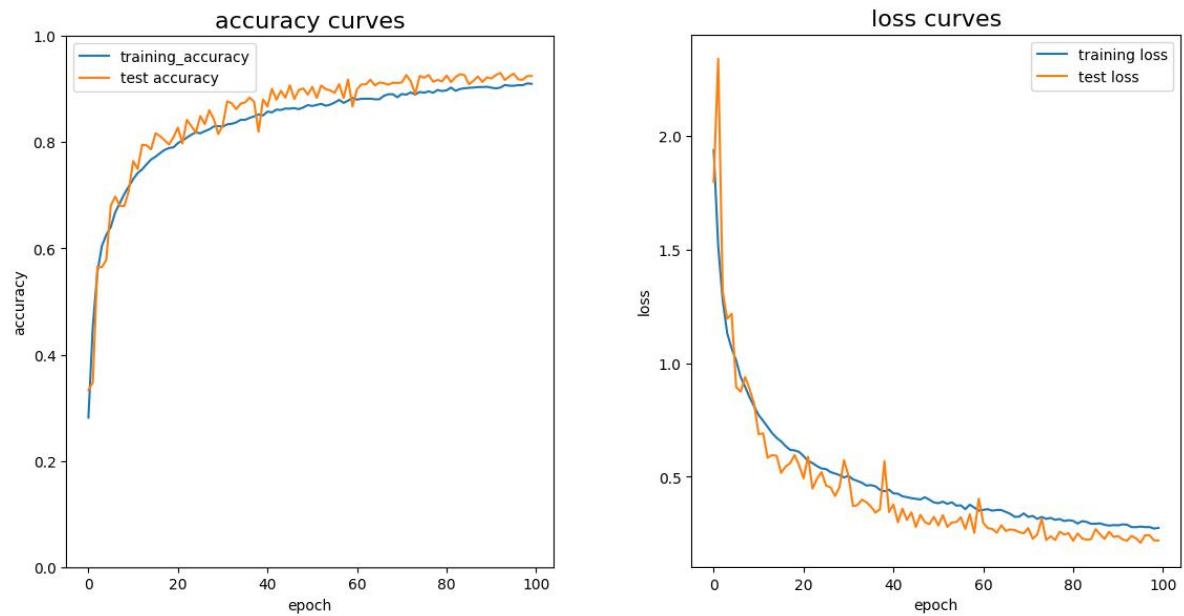


Figure 1 task A curves

The accuracy of best model is 91.3%.

2. T-SNE plot for randomly selected 480 samples (`max_num_step = 15`) of `fc_features` are as below (*relative path: ./tSNE_figure.png*):

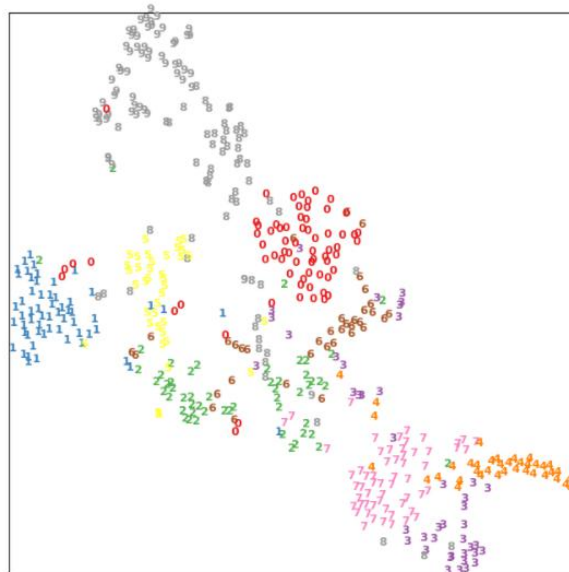
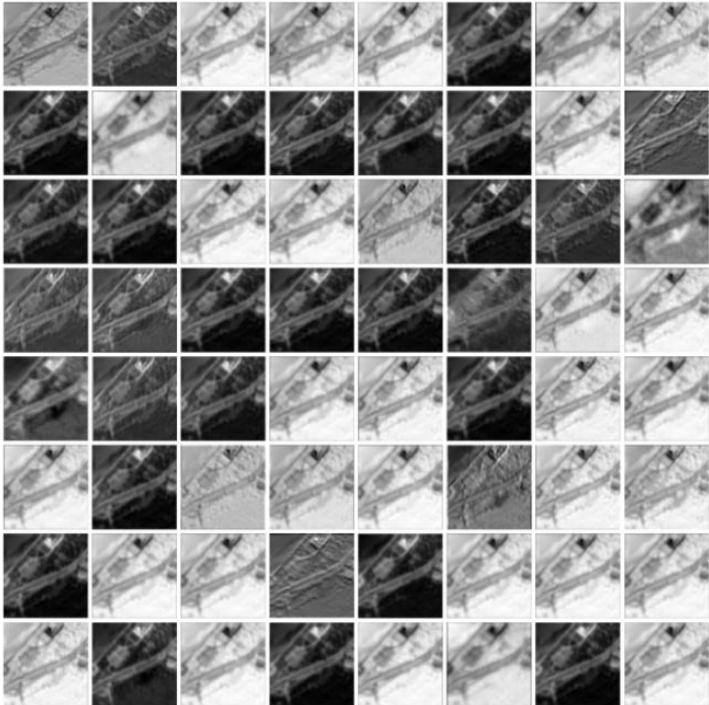
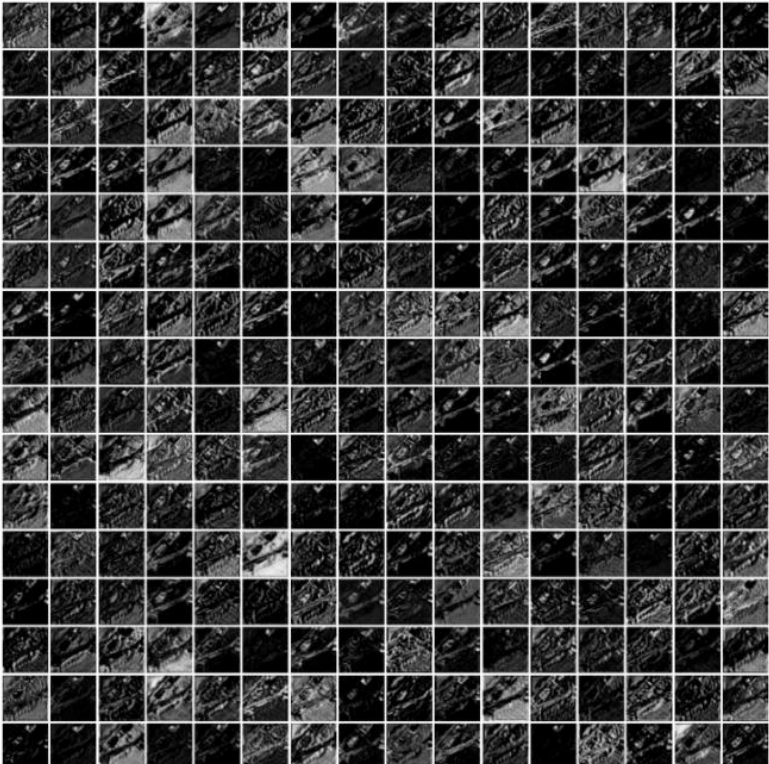
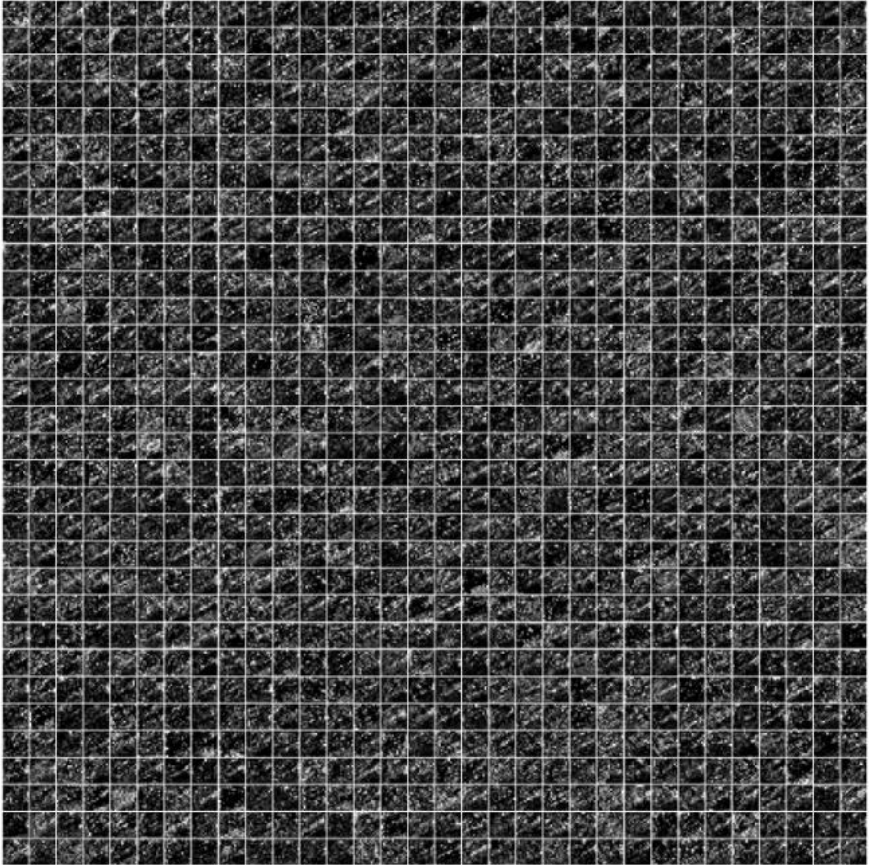
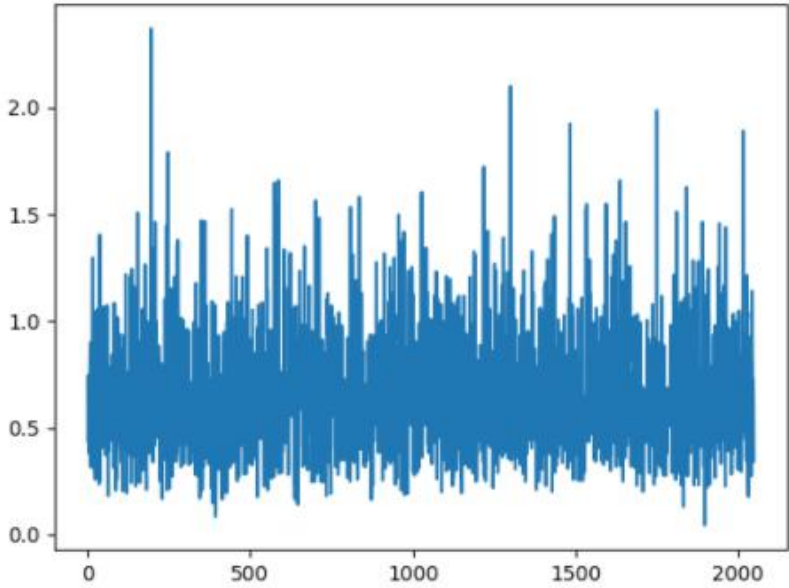


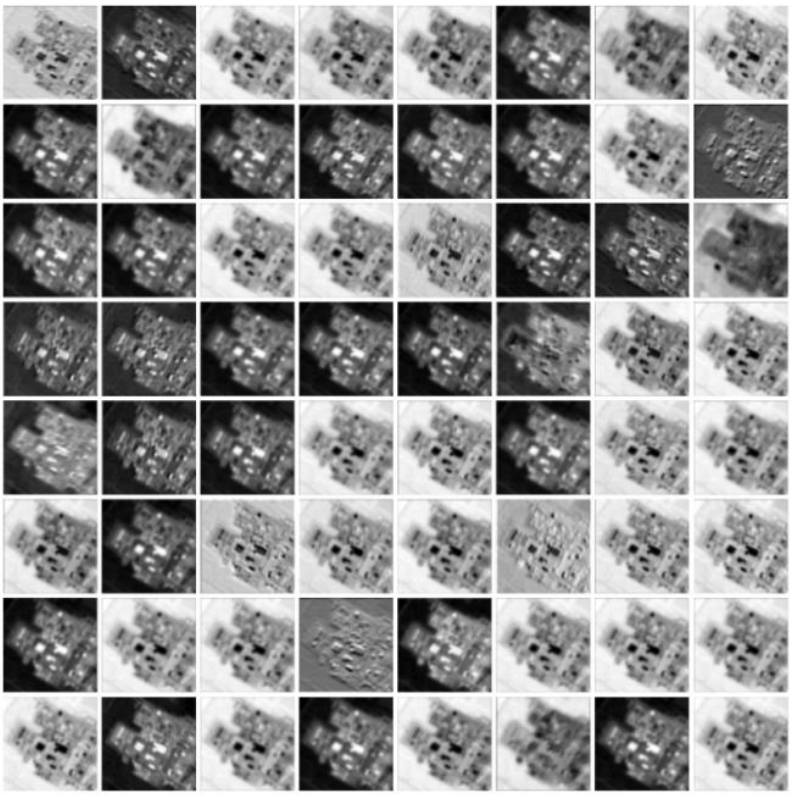
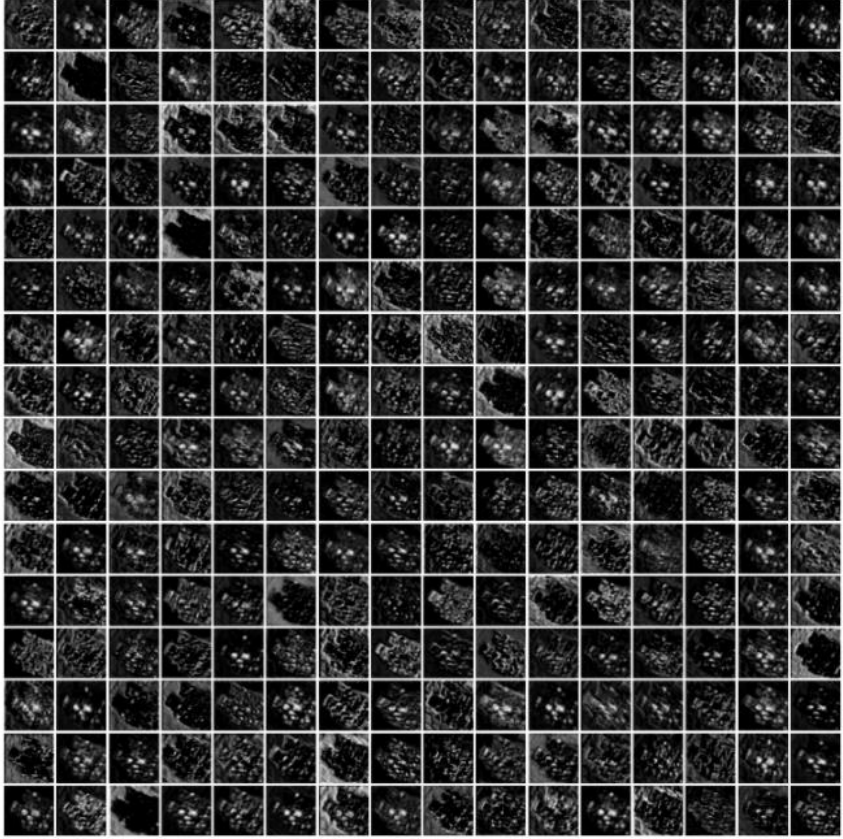
Figure 2 task A t-SNE graph

Note different figures represent different categories of images.

3. Visualization of specific layers of some images are as below (*relative path: ./resnet_vis*):

layer	Image: Highway
F1_conv1	
F5_layer1	

F7_layer3	
F9_avgpool	

layer	Image: Highway
F1_conv1	
F5_layer1	

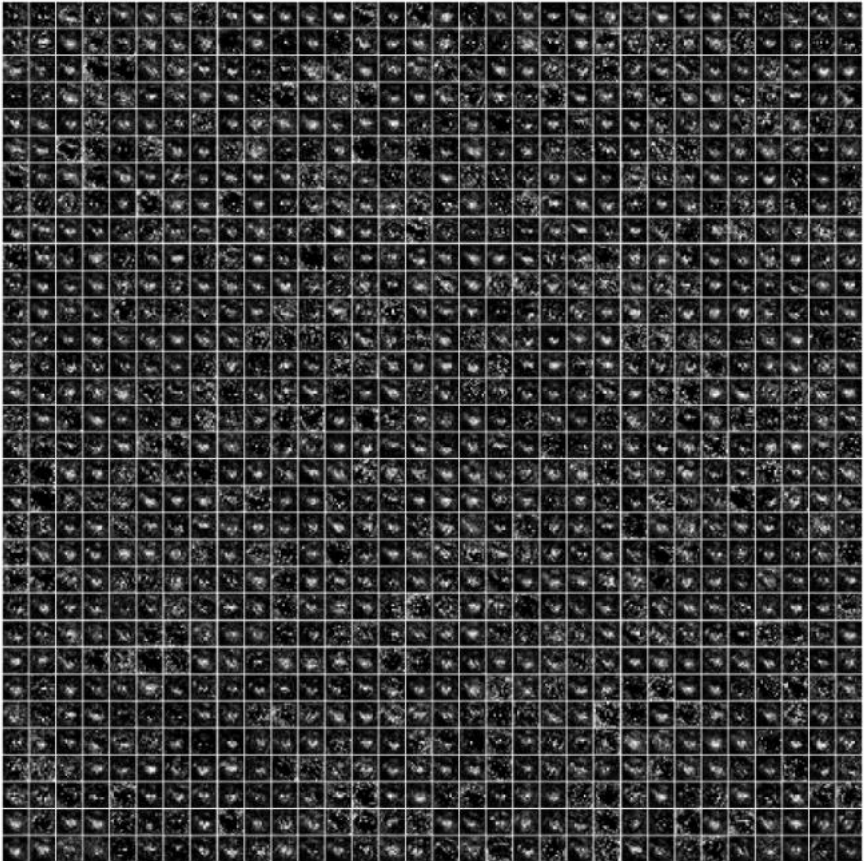
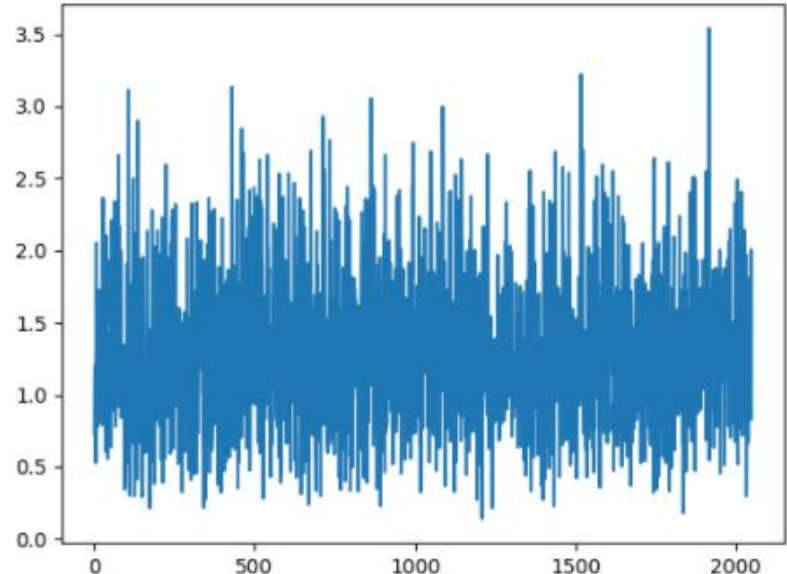
F7_layer3	
F9_avgpool	

Table 1 task A visualization

The name of layers corresponds to those in ResNet50 model in torchvision.models. More images can be found folder .resnet_vis. Note that for F9_avgpool layer, after average pooling each feature map degrades into a single value, which is the y-axis in plot, and x-axis is the serial number of that feature dimension.

Task B

1. Insights and analysis

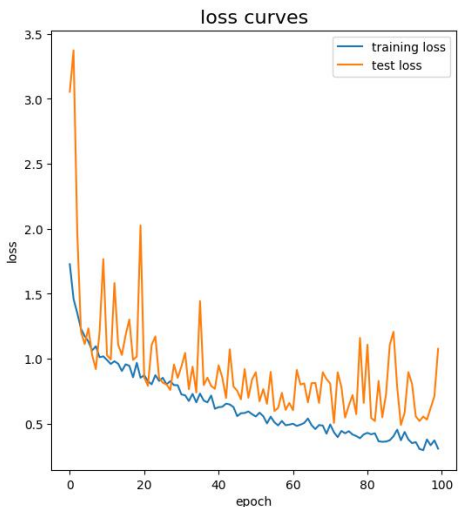
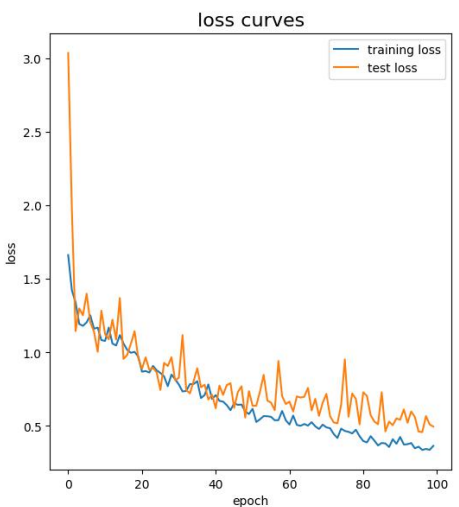
In this task we try to modify a DenseNet121 model to be fitted on our small dataset. We start from DenseNet because it has fewer parameters than ResNet counterpart, but deeper layers to obtain more abundant information from limited inputs. In small dataset it is easy for deep models to get overfitted, while we want to take advantages of deep models to capture features at different levels. **To further deal with the conflict between small dataset and large quantity of parameters, our work is also done following these two principles:**

A) To reduce number of parameters. Inspired by operation used in InceptionV3, we use kernel factorization to replace standard 3x3 convolution filter with a 1x3 plus a 3x1. If we are going to use deeper DenseNet as backbone, group convolution can be added to release explosion in parameters.

B) To enrich information from limited data. Data augmentation has been widely used to enhance robustness of model, but we try to introduce this method into . Disturbance should not be added at very deep layer because that could has a over-strong influence on final output and mislead the model. We choose to add a gaussian noise at first six convolutional layers point-wisely, whose elements are zero-centered and have a standard deviation of 0.005. Here to keep randomness of gaussian noise, the global random state (by setup_seed function) can not be used.

2. experiment results (without data augmentation or learning rate scheduling)

Model	Best accuracy
DenseNet121(baseline)	0.84
Model_B(ours)	0.86

Curve	DenseNet121 (baseline)	Model_B(ours)
Loss accuracy		

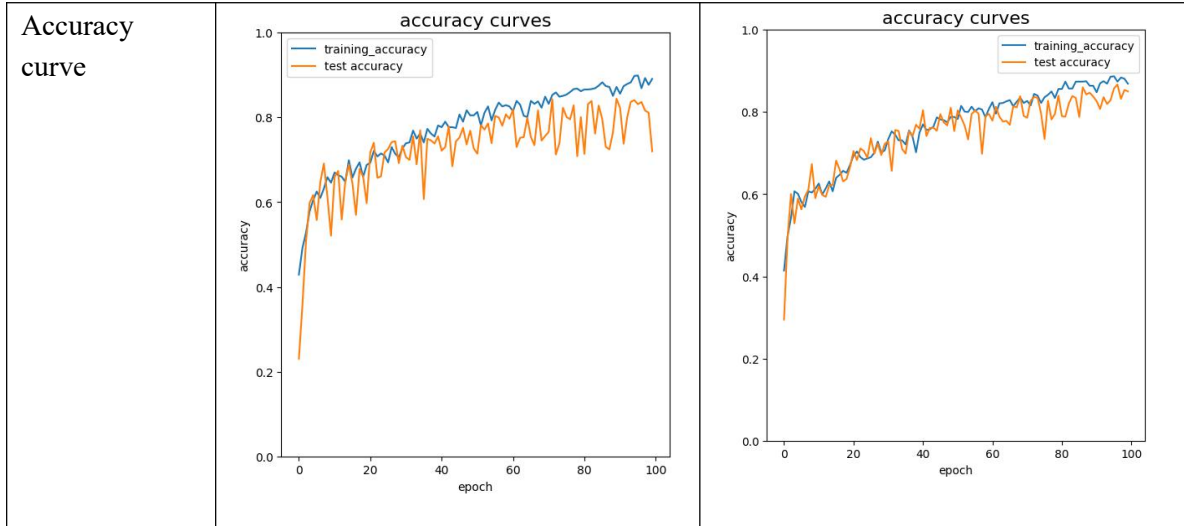


Table 2 training and test curves in baseline and proposed model

We could compare training graphs to see that our model has partially mitigated vibration of loss curves throughout training process and achieved a better performance than baseline. Note that in our model in last few epochs the test loss still decreases, while in baseline model there is no improvement in last 40 epochs.

3. Data augmentation and learning rate scheduling

In the section above, though relieved, the problem of loss vibration of our model still exists. We could further improve with input data augmentation and learning rate scheduling policy.

1) How to choose data augmenting methods?

We try to compare some samples from dataset to select reasonable methods.

Below are four representative image samples from four categories. The first one comes from 'sealake' and second one from 'forest'. Obviously, there is no apparent difference of their shapes. The main difference to distinguish them is their color. The same applies to the third (highway) and fourth (river), which are both lines or curves. It denies or at least limits the usage of color-jitter technique.

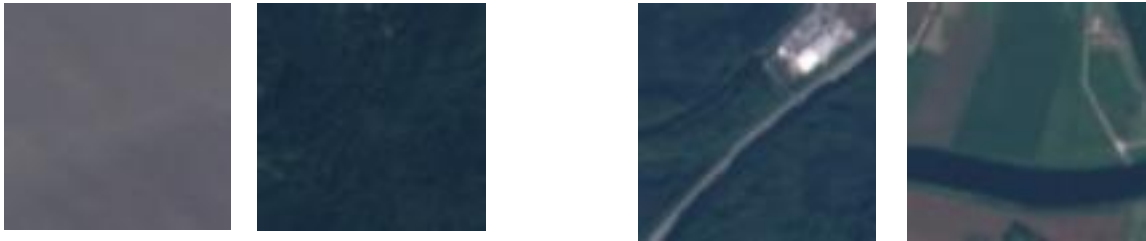


Figure 3 Sample images from different categories

Except those related to colors, we try to randomly choose from different transforms as

many as possible to increase the diversity of inputs.

2) How to choose learning rate scheduler?

Since the problem of loss vibration seems still severe, we would like learning rate to be smaller at latter stage. Thus we use ReduceOnPlateau as scheduler here.

3) More tricks

To further relieve overfitting, we choose smooth-labeled crossentropy as loss function with smoothing factor 0.1.

4) Experiment results

Model	Best accuracy
DenseNet121(baseline)	0.84
Model_B(without tricks)	0.86
Model_B(with tricks)	0.892

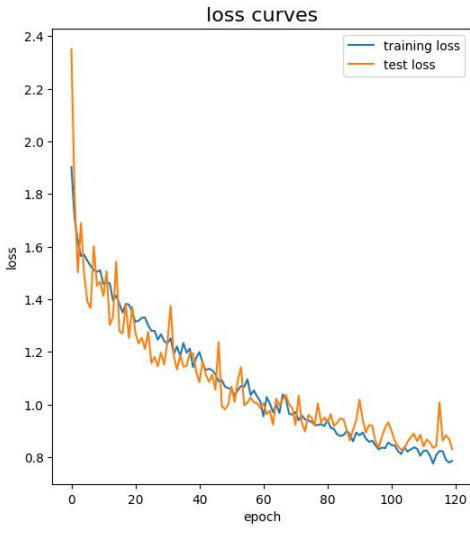
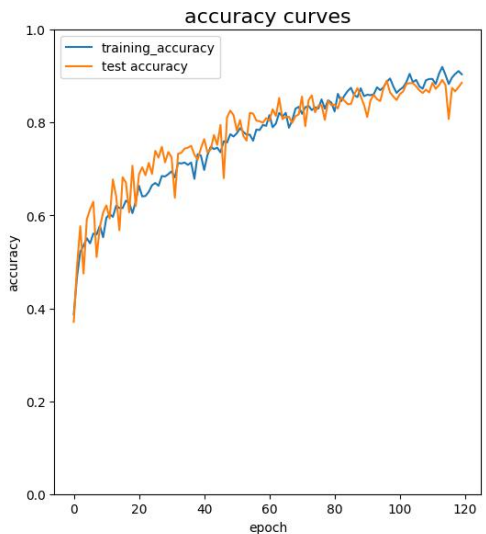
	Loss curve	Accuracy curve
Model_B (with training tricks)		

Table 3 training and test curves in proposed model with tricks

After using tricks including data augmentation, learning rate scheduling and label smoothing (which contributes to a higher loss value), we can see that best model accuracy further increased by 3.2%. At the same time, the generalization gap decreased, meaning the overfitting problem has been efficiently solved.

Task C

1. Insights and analysis

In this task we would choose long-tailed learning problem to complete. Our inspiration comes from a new paper, *Decoupling Representation and classifier for long-tailed recognition* by Facebook AI, in which they proposed to **train representation learner and a classifier separately to combat the imbalanced-dataset problem**. In our training process, at first stage **we inherit all training tricks from task B, plus oversampling policy** of imbalanced training set to train an adjusted-Densenet121 model. When this training is done, we use adjusted-Densenet121 as a feature extractor to map training images to vectors in low-dimension space, which are subsequently used to train the head.

From theory in machine learning we have known that **support vector machine (SVM) is a classifier that is very robust in imbalanced dataset**, since the hyperplane depends on marginal samples only. Moreover, we can set larger values of penalty coefficients for smaller categories to pay more attention on classifying errors occurring in those disadvantaged classes.

2. Experiment results

Since under sklearn framework plotting training curves is not supported, we turn to plot curves during training backbone below.

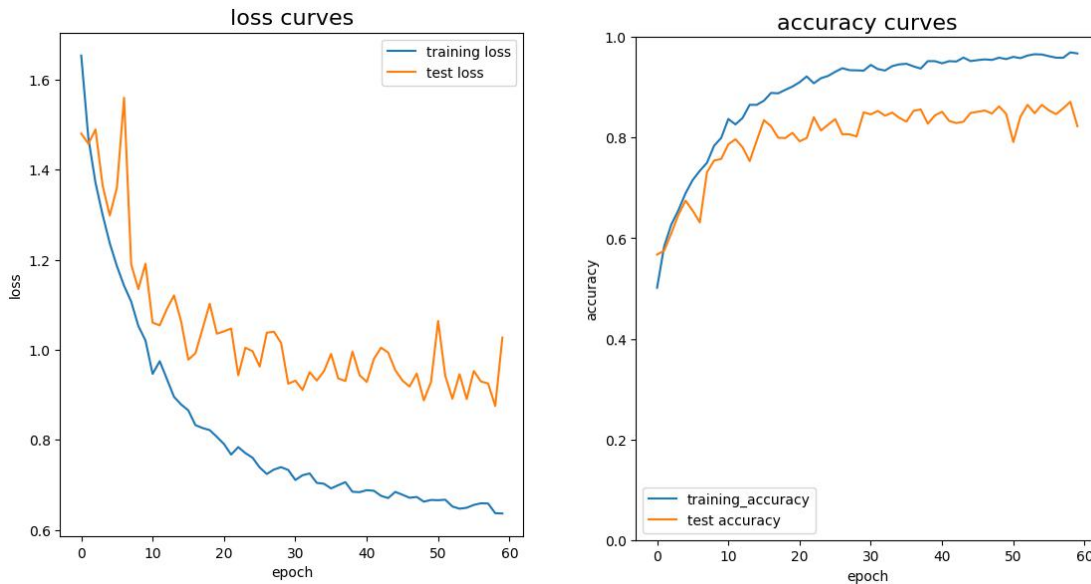


Figure 4 training and test curves in backbone

We can see from curves that our backbone overfits in the imbalanced training set. Then we try to use the SVM head to classify on the base on backbone. With hyperparameter $\text{class_weight} = \{0:0.0001, 1:0.001, 2:0.001, 3:1, 4:1, 5:1, 6:1, 7:0.01, 8:1, 9:1\}$, we get the accuracy

of 0.865 (1.5% increase than backbone only).

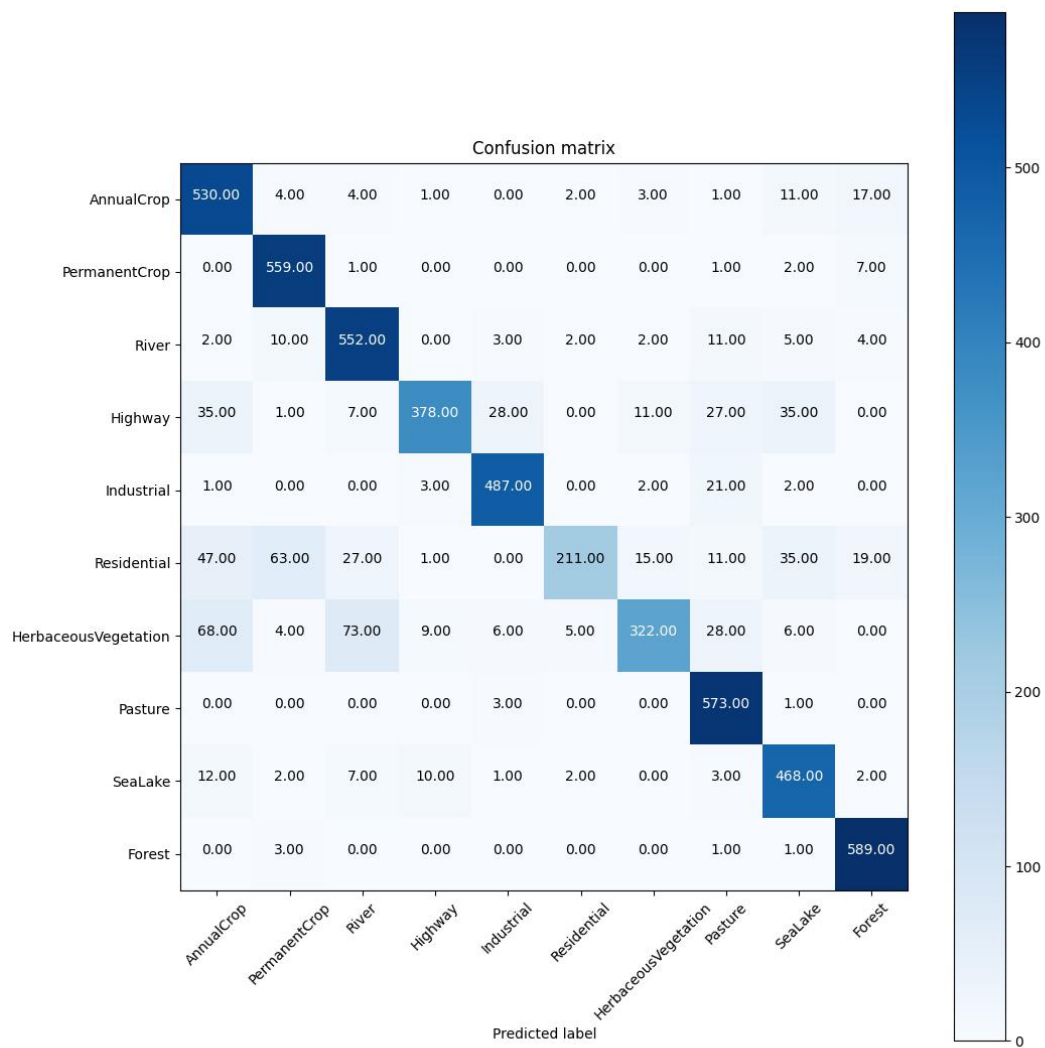


Figure 5 confusion matrix of classifier

Acknowledgement

Sincerely thanks everyone who has given me patient help during this homework including two TAs and a senior student Zhang Zheng from my lab, as well as a young giant student in our course. It can never be easy for me, a non-CS student, who decided to be committed to AI for future study but now poor with coding, to finish this homework. In the past two weeks, all the time my work is writing codes, checking information and documents online, asking questions and debugging. I have confronted different bugs whose amount was almost more than I had met all before. Everywhere are giants on this campus, some of whom really completed this homework even in three days with a high quality. Sometimes having got upset, but I really appreciate this course because it guides me to overcome one difficulty after another that may be unimaginable for me before this course. Life is not easy, just go ahead.

References

- [1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778, 2016
- [2] python 可视化 resnet50. <https://blog.csdn.net/u012435142/article/details/84711978>
- [3] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. ICML, 2019.
- [4] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, Mu Li: Bags of Tricks for Image Classification with CNN.
- [5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2261–2269, 2017.
- [6] Pytorch 使用 albumentations 实现 数据 增强 . <https://blog.csdn.net/zhangyuexiang123/article/details/107705311>
- [7] <https://github.com/albumentations-team/albumentations/issues/29>
- [8] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, Yannis Kalantidis. Decoupling Representation and Classifier for Long-Tailed Recognition. International Conference on Learning Representations (ICLR), 2020