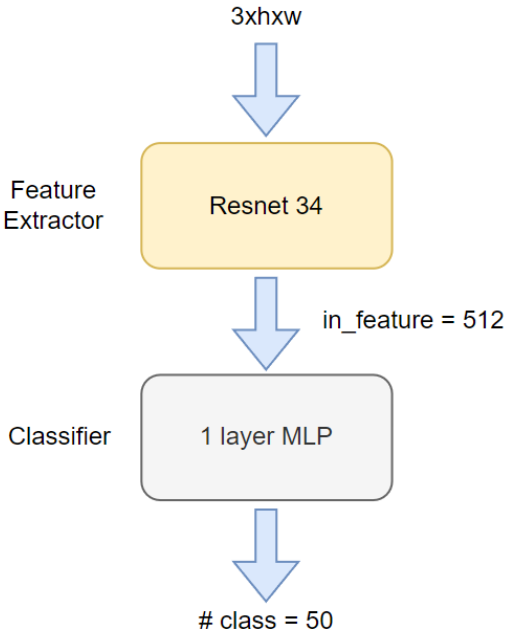
DLCV hw1_report

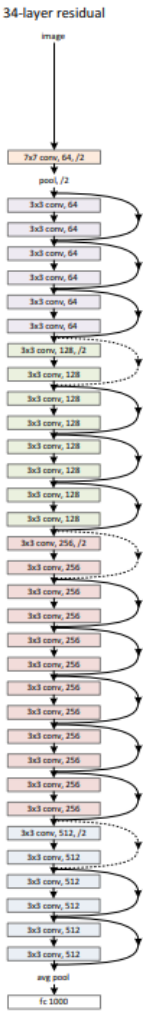
R11922166 陳敬和

Problem 1 - Image Classification

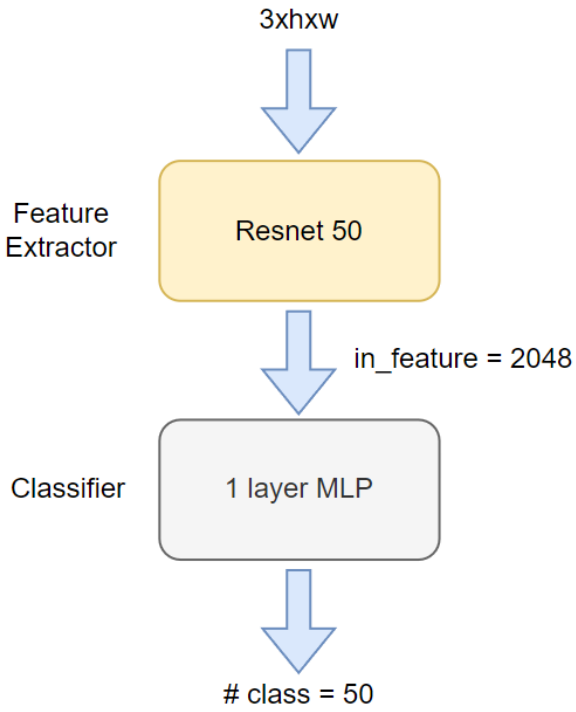
1. Draw the network architecture of method A or B.

- Model A





- Model B

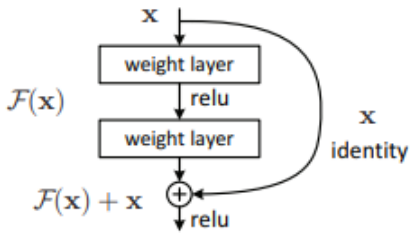


2. Report accuracy of your models (both A, B) on the validation set.

	Accuracy (%)	**# of correct**
Model A	72.48	1812
Model B	86.84	2171

3. Report your implementation details of model A

To implement model A, I choose ResNet34 to write from scratch. At the beginning, I constructed the Residual Block as below, which can reuse through the network. In the architecture, there are four blocks, and each containing different number of layers. Adding the layers one by one along with the Residual Block constructed before. At the end of network, connecting 1-layer MLP to implement classification.



To pre-processing the data, implementing data augmentation when loading dataset, like resize, rotate, horizontal flip, crop, and etc. The performance improve a lot after implementing data augmentation.

When starting to training, I use Adam as my optimizer, select MultiStepLR as learning rate scheduler, and use Cross Entropy Loss as loss function. After recording the training loss, the ‘milestone’ parameter in MultiStepLR would be adjust, in order to update learning rate in specific step according to the information from the loss curve.

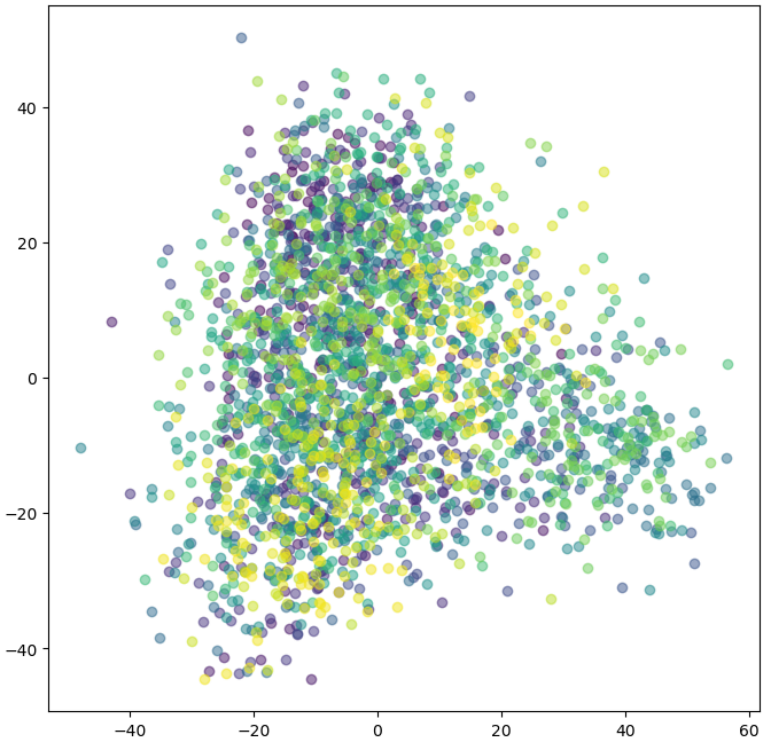
4. Report your alternative model or method in B, and describe its difference from model A.

The model B use ResNet50 as model backbone, which is more deeper than model A which use ResNet34. From the paper, 50-layer ResNet replace each 2-layer block in the 34-layer net with 3-layer bottelneck block, resulting in 50-layer ResNet (see below table). Furthermore, ResNet50 use projections for increasing dimensions. As the result, 50-layer ResNet is more accurate than 34-layer ResNet. After conquering the degradation problem, it seems that the more deeper depth on the network, the more accurate the model is.

layer name	output size	18-layer	34-layer	50-layer	101-layer
conv1	112×112	7×7, 64, stride 2			
conv2_x	56×56	3×3 max pool, stride 2			
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax			
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9

5. Visualize the learned visual representations of model A on the validation set by implementing PCA (Principal Component Analysis) on the output of the second last layer. Briefly explain your result of the PCA visualization.

After implementing PCA on the dataset, the original features will turn into Principal Components, which is the linear combination of original features. Principal Components are not as readable and interpretable as original features. Furthermore, the most significant drawback of PCA is Information Lost. After dimension reduction, information lost is inevitable. So I think this is the main reason why all points are grouped together.

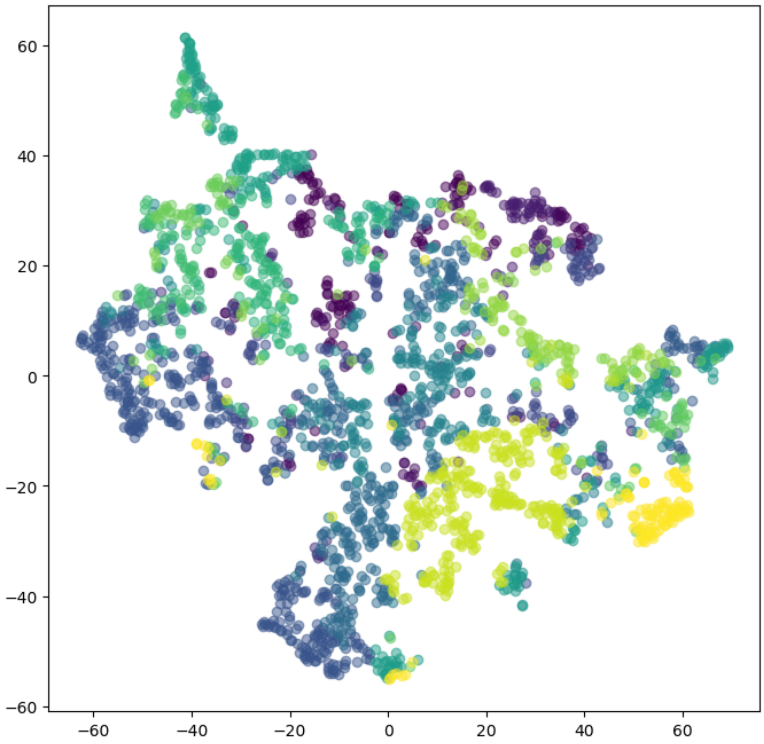


6. Visualize the learned visual representation of model A, again on the output of the second last layer, but using t-SNE (t-distributed Stochastic Neighbor Embedding) instead. Depict your visualization from three different epochs including the first one and the last one. Briefly explain the above results.

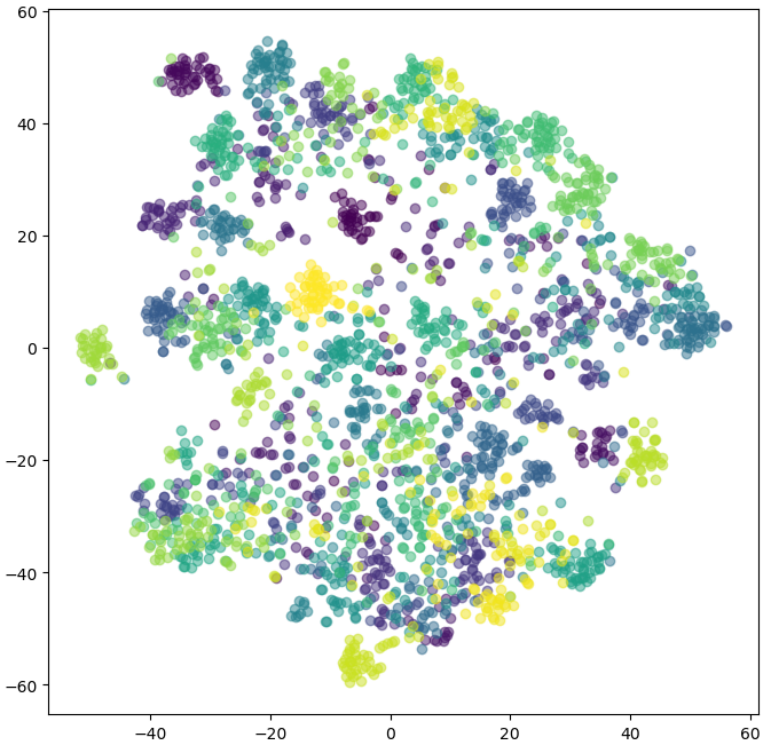
In the first epoch, almost all groups are blended with each other. The accuracy in the first epoch is also the lowest. After 90 epochs and at the last epoch, the intra class distance of the features is become small, however the inter class distance is still large.

I think the reason is because the accuracy of this model is only about 72%. Although the intra class difference is smaller, the inter class among different features need to improve. I think using a more deeper model like ResNet101 or more complex loss function would have better result.

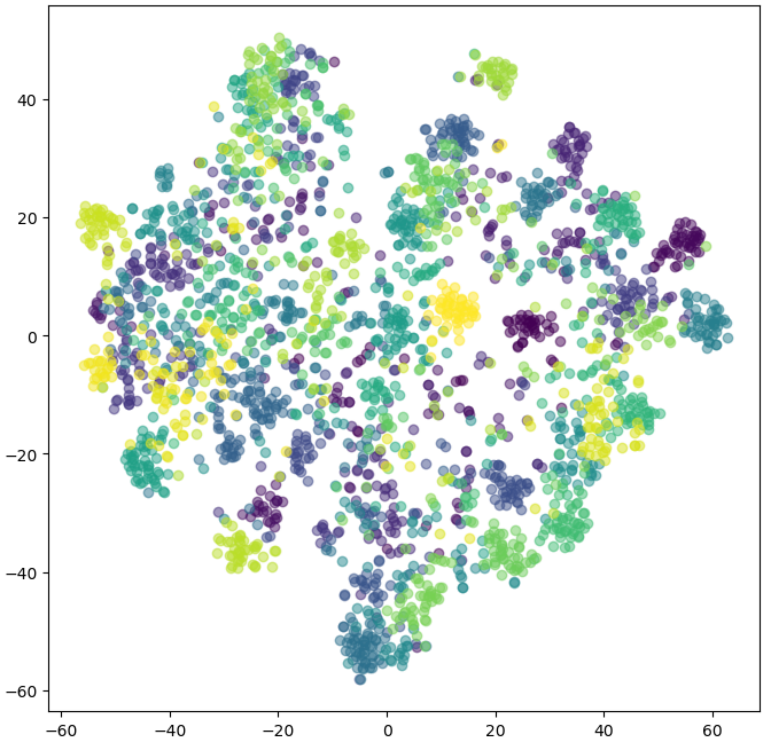
- First epoch



- Epoch 90

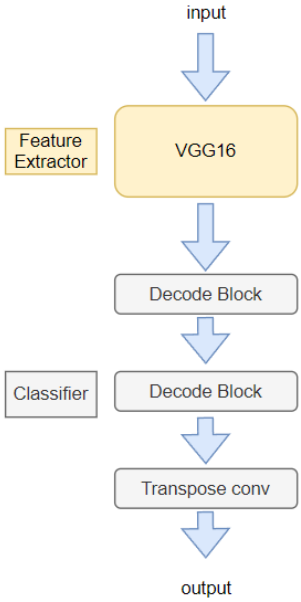


- Last epoch

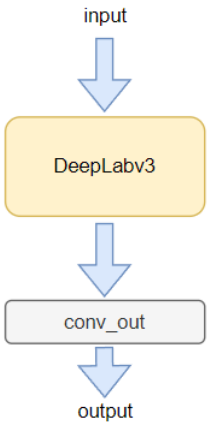


Problem 2 - Image Classification

1. Draw the network architecture of your VGG16-FCN32s model (model A)



2. Draw the network architecture of the improved model (model B) and explain it differs from your VGG16-FCN32s model



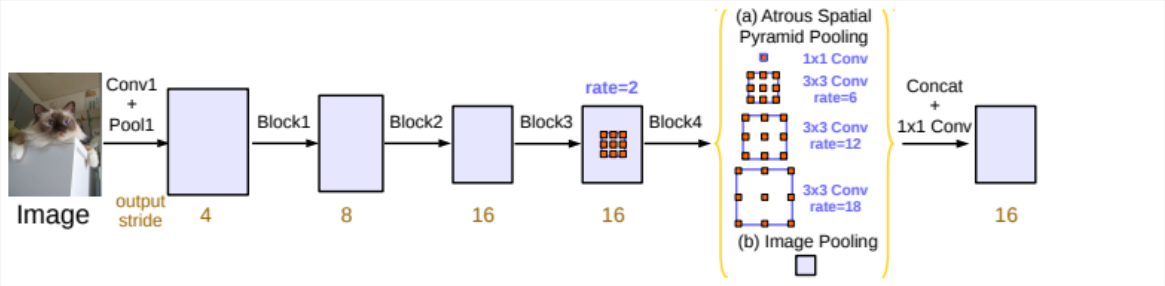


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

One of the drawbacks of VGG was vanishing gradient problem. As a neural network gets deeper, the gradients start to shrink to zero, and the weights are not updated. ResNet solved the problem by using skip connection.

And here, the improve model - DeepLabv3 use a bigger and deeper architecture - ResNet101 as model backbone instead of using fc layer as VGG. According to the paper, to encode multi-scale information, DeepLabv3 use atrous spatial pyramid pooling (ASPP) module augmented with image-level features probes the features with filters at multiple sampling rates and effective field-of-views. DeepLabv3 also contains pretrained weight on CoCo dataset, which make the model converges faster than model A and stop with lower loss.

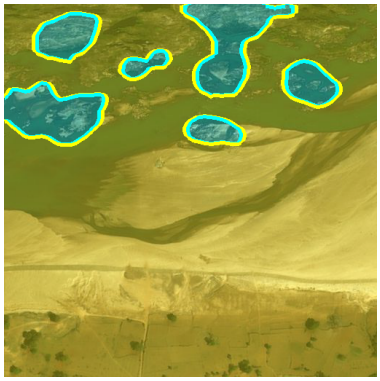
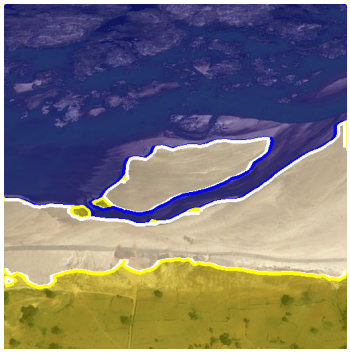
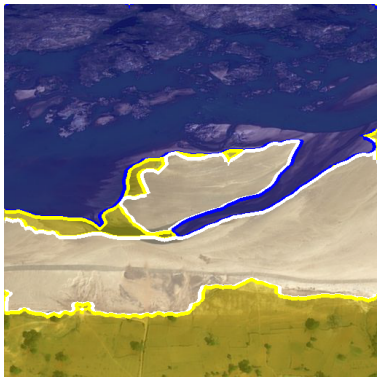
3. Report mIoUs of two models on the validation set.

	mIoU
Model A	0.6032
Model B	0.7354

4. Show the predicted segmentation mask of “validation/0013_sat.jpg”, “validation/0062_sat.jpg”, “validation/0104_sat.jpg” during the early, middle, and the final stage during the training process of the improved model.

▼ 0013_sat.jpg

Early stage | Middle stage | Final stage

▼ 0062_sat.jpg

Early stage | Middle stage | Final stage



▼ 0104_sat.jpg

Early stage | Middle stage | Final stage



Reference

https://arxiv.org/pdf/1512.03385.pdf

https://arxiv.org/pdf/1706.05587.pdf