

# DLCV HW1 Report

R11921008 羅恩至

## Problem1

(2%) Draw the network architecture of method A or B.

Model A:

For model A, I use a resnet34 model and trained it from scratch. The model architecture is shown in the figure below:

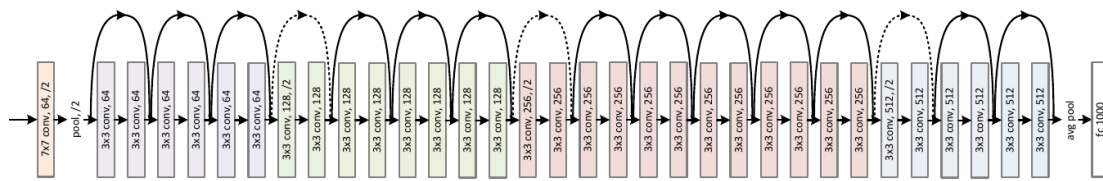


Fig. 1. The architecture of Model A(Resnet34)

Source: <https://www.796t.com/content/1546962667.html>

Model B:

For model B, I tried a pretrained Efficientnet\_b4 model and finetuned the last classifier layer to make the output channel 50, which fits with our task. The model architecture is shown in the figure below:

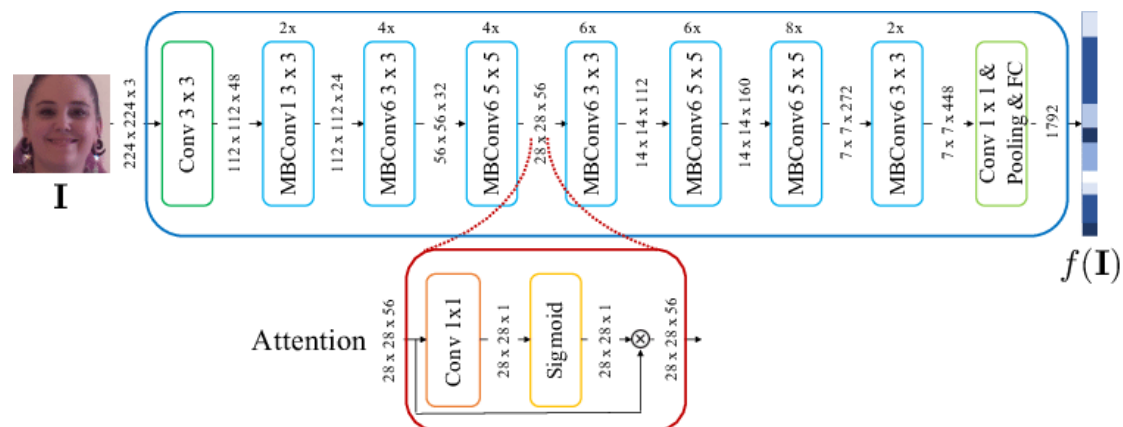


Fig. 2. The architecture of Model B(EfficientNetB4)

Source: [https://www.researchgate.net/figure/Blue-block-EfficientNetB4-model-If-the-red-block-is-embedded-into-the-network-an\\_fig2\\_340683627](https://www.researchgate.net/figure/Blue-block-EfficientNetB4-model-If-the-red-block-is-embedded-into-the-network-an_fig2_340683627)

**(1%) Report accuracy of your models (both A, B) on the validation set.**

Model	Training accuracy	Validation accuracy
Model A	0.95546	0.76523
Model B	0.93778	0.89883

**(4%) Report your implementation details of model A.**

**(1) Data Augmentation**

For this homework, I tried ColorJitter, RandomHorizontalFlip on training data. Besides, I also used 3 kinds of AutoAugment policy (IMAGENET, CIFAR10 and SVHN) then concatenated the 3 training set from different augmentation methods to get the final training set, containing  $22500 \times 3 = 67500$  images in total.

**(2) Model Architecture**

In resnet34 model, it contains 5 convolution blocks in total. The first convolution block is a  $7 \times 7$  kernel, and the rest 4 convolution blocks contain 3, 4, 6, 3 layers respectively. For each block, each layer includes two convolution layers with  $3 \times 3$  kernel, and ReLU activation function and batch normalization. Then finally the weights are fed into the adaptive average pooling and a fully connected layer to extract the output.

**(3) Hyperparameters**

The hyperparameters are shown below:

Optimizer	Adam (with $1e-05$ weight decay)
Loss function	CrossEntropyLoss
Learning rate	0.0001
Batch size	64
Epoch (with best performance)	74

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

**(1) Data augmentation**

Same as model A.

**(2) Model Architecture**

In Efficientnet\_b4 model, as shown in Fig. 2. and Fig. 4 below, the whole architecture contains 7 main blocks, includes 2, 4, 4, 6, 6, 8, 2 subblocks respectively. In Fig. 3., it contains three different kinds of subblock. The subblocks include 2D convolution, batch normalization, and also some methods not utilized in Resnet34 like depthwise convolution, global average pooling, rescaling, etc. In addition, by using SiLU activation function instead of ReLU, Efficientnet\_b4 model has a better performance than Resnet34.

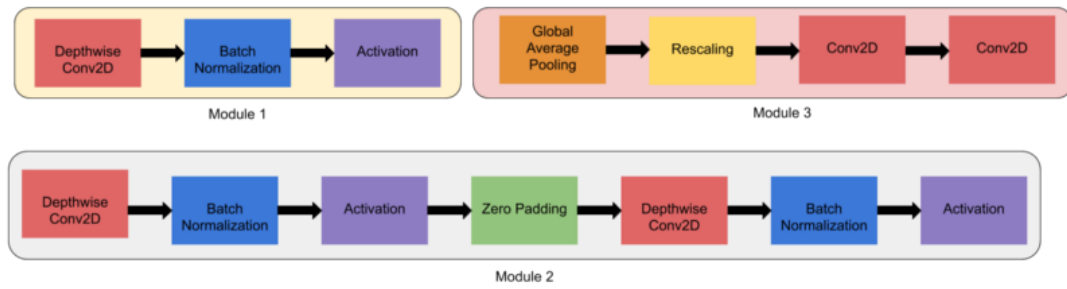


Fig. 3. Three different kinds of subblock in Efficientnet\_b4 model

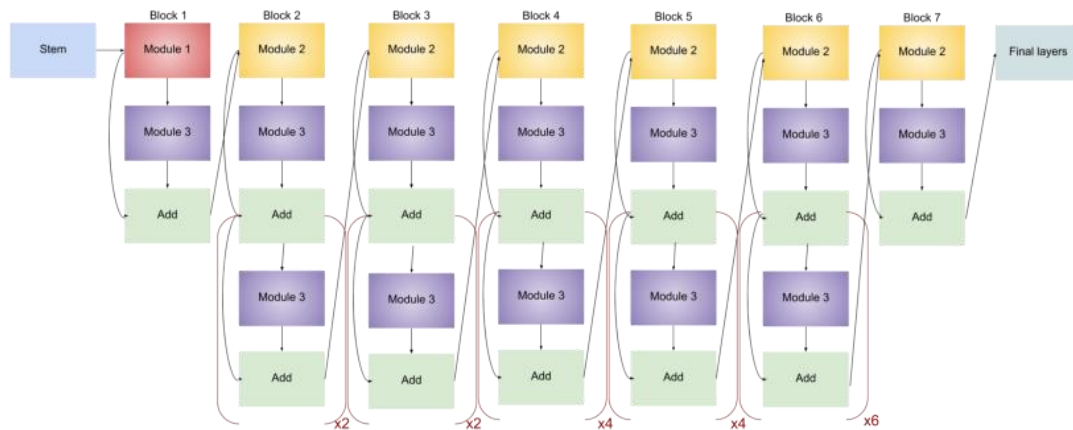


Fig. 4. Whole model architecture of Efficientnet\_b4 model

Source: <https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142>

### (3) Hyperparameters

Optimizer	Adam (with 1e-05 weight decay)
Loss function	CrossEntropyLoss
Learning rate	0.0001
Batch size	64
Epoch (with best performance)	4

With the usage of pretrained weights and finetuned the output classifier layer, I trained only a few epochs and get a good performance.

**(7%) 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.**

Below is the PCA result of model A at epoch 54(with highest validation accuracy):



Fig. 5. Model A PCA result at epoch 54

Fig. 5. is the visualization of the classification result on 2500 validation sets by implementing PCA on the output of the second last layer of model A(Resnet34). From the figure, we can not see the clearly clusters between the labels, instead, all labels fuse together in the left and center of the figure.

**(7%) 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.**

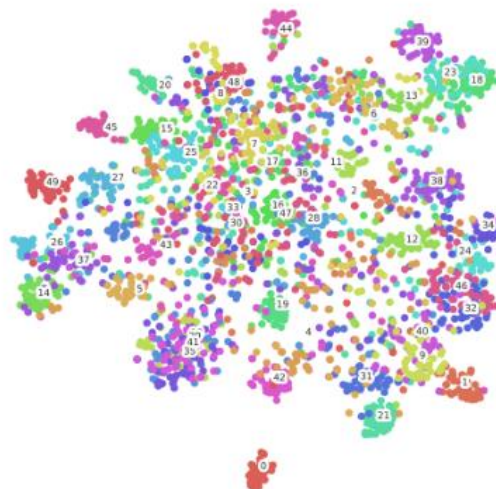


Fig. 6. Model A t-SNE result at epoch 1

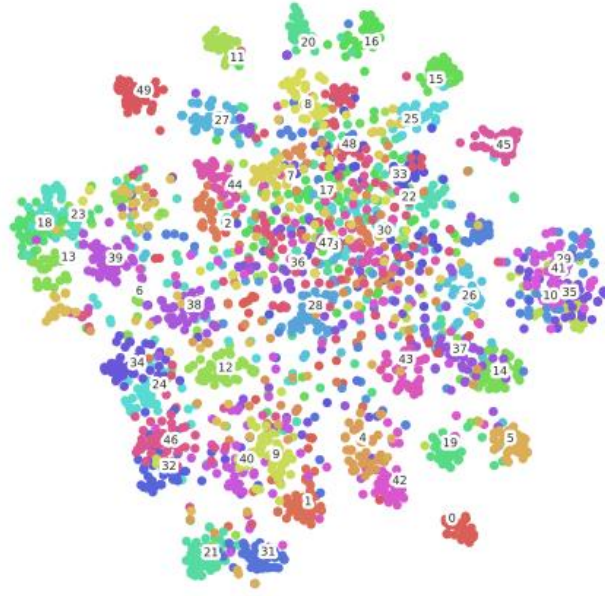


Fig. 7. Model A t-SNE result at epoch 30

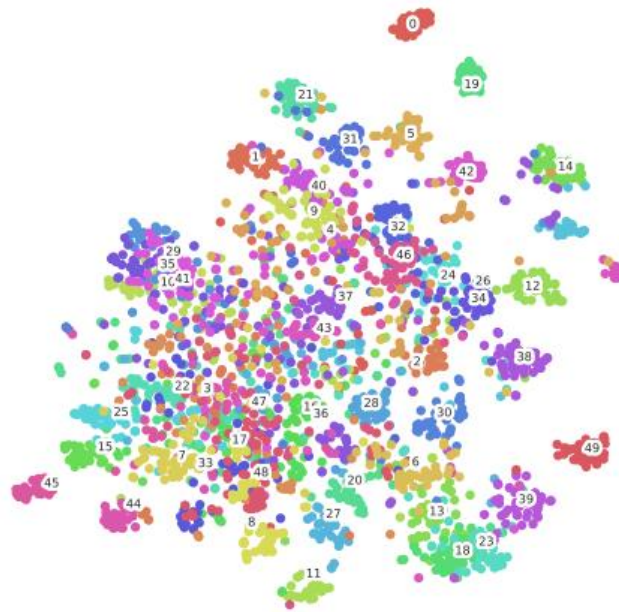


Fig. 8. Model A t-SNE result at epoch 74

Fig. 6. To 8. are the visualization of the classification result on 2500 validation sets by implementing t-SNE on the output of the second last layer of model A(Resnet34). Fig. 6. is from epoch1(accuracy = 0.29492), Fig. 7. is from epoch30(accuracy = 0.71719) and Fig. 8. is from epoch74 with the best validation accuracy 0.76523. From the 3 figures above, we can see that when the during training, we can see more clearly clusters with more epochs, and this shows that the network is

able to distinguish the labels more clearly. From the figures, we can clearly see the clusters on some labels like label 0, 5, 12, 14, 19, 45 and 49, etc., while some labels such as label 37, 43 and 47, etc. are still all in the center of the plot and very to each other. This implies that the model still can't classify correctly between these labels.

## **Problem 2**

**(5%) Draw the network architecture of your VGG16-FCN32s model (model A).**

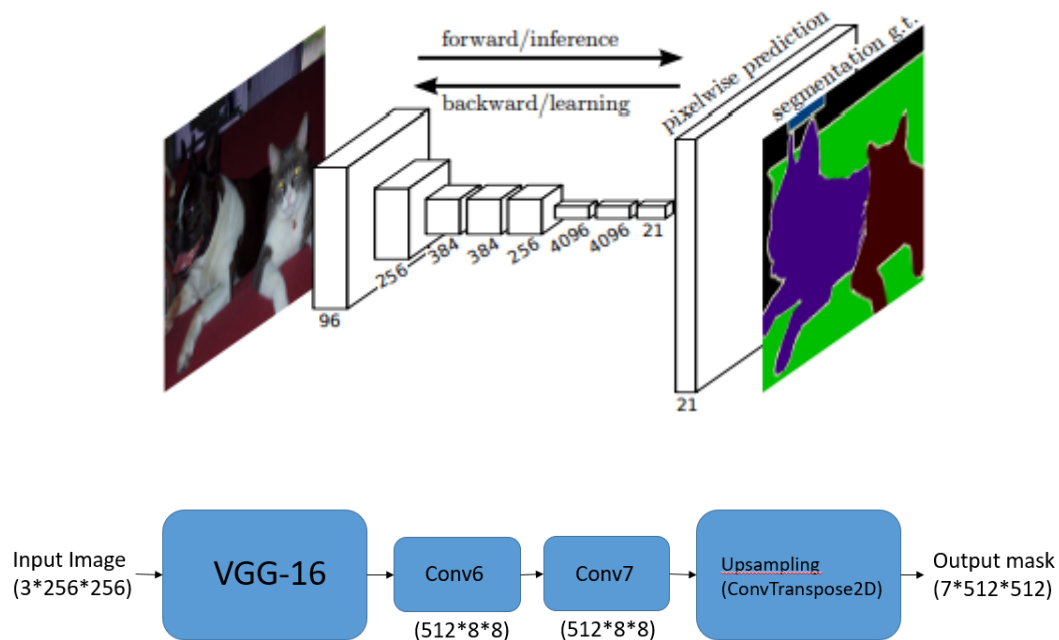


Fig. 9. Network architecture of VGG16-FCN32s model

Source: Fully Convolutional Networks for Semantic Segmentation

<https://arxiv.org/abs/1605.06211v1>

**(5%) Draw the network architecture of the improved model (model B) and explain it differs from your VGG16-FCN32s model.**

In model B, I tried VGG16-FCN4s as my improved model. The difference between VGG16-FCN4s and VGG16-FCN32s is that VGG16-FCN4s also extracts the features of pooling2 to 5 layer, then fuse all features with a total 4 times upsampling to get the output, where VGG16-FCN32s only extracts the feature from pooling5 layer and only do the upsampling once.

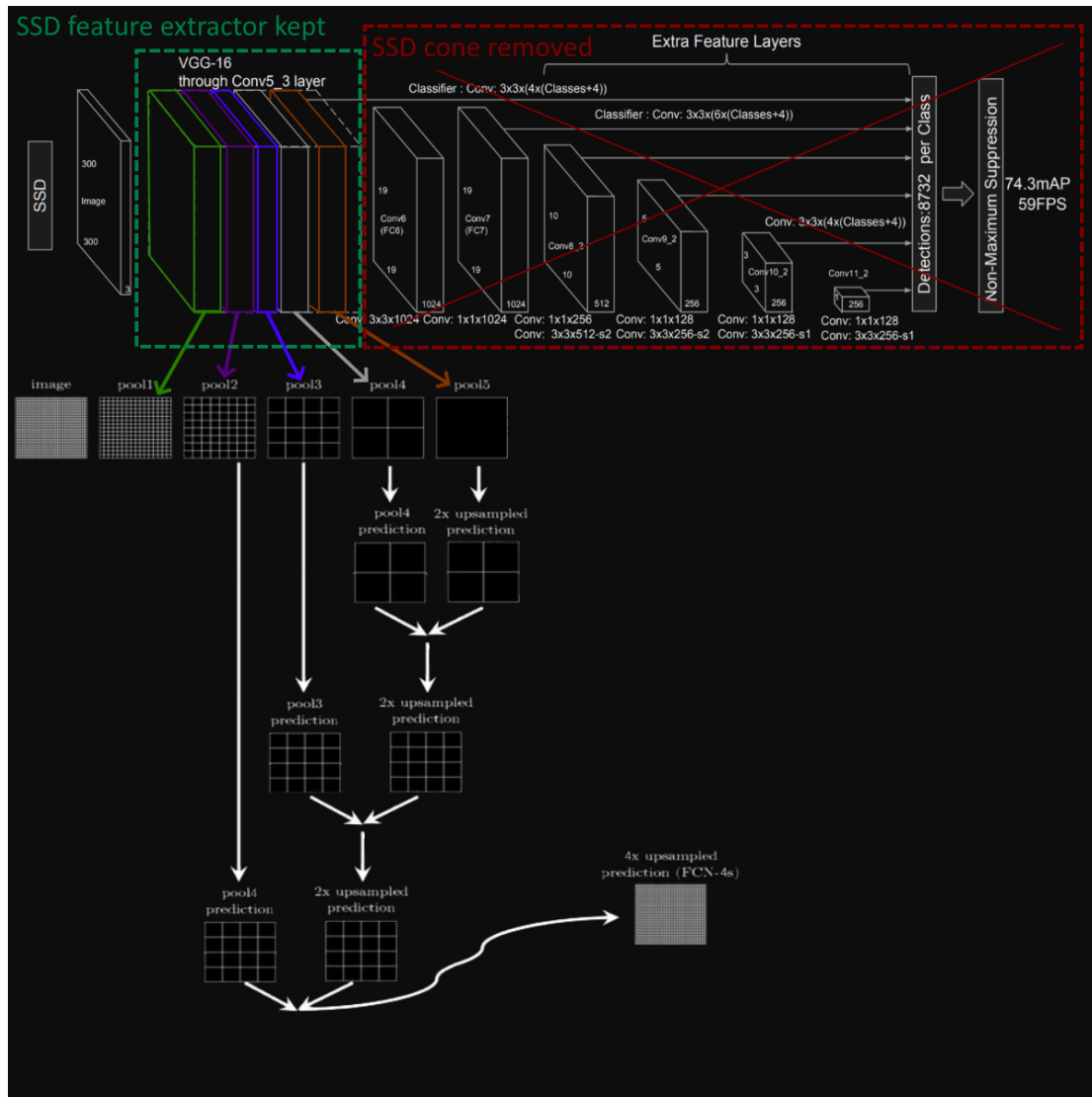


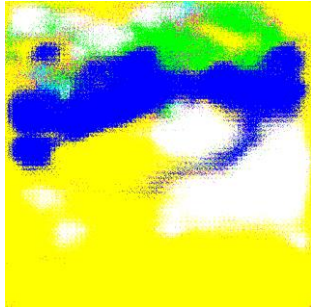
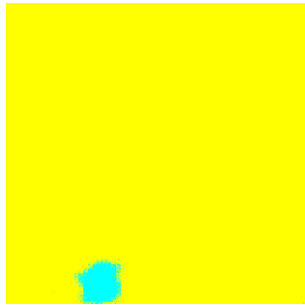
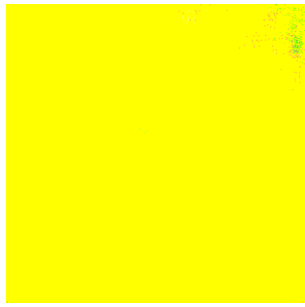
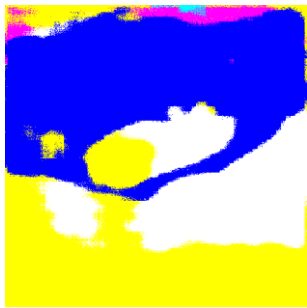





Fig. 10. Network architecture of VGG16-FCN4s model

Source: <https://foundationsofdl.com/2021/03/03/segmentation-model-implementation/>

**(3%) Report mIoUs of two models on the validation set.**

Model	Model A (VGG16-FCN32s)	Model B (VGG16-FCN4s)
mIoU class #0	0.71420	0.74612
mIoU class #1	0.85889	0.88470
mIoU class #2	0.30653	0.39438
mIoU class #3	0.80198	0.79518
mIoU class #4	0.71738	0.74725
mIoU class #5	0.65805	0.71448
Mean mIoU	0.67617	0.71369

**(7%) 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.**

	validation/0013_sat.jpg	validation/0062_sat.jpg	validation/0104_sat.jpg
Early stage (Epoch 2)			
Middle stage (Epoch 14)			
Final stage			
Ground Truth	