

# DLCV HW1 Report

B11901040 項達均

September 27, 2024

## 1 Problem 1

### 1.1

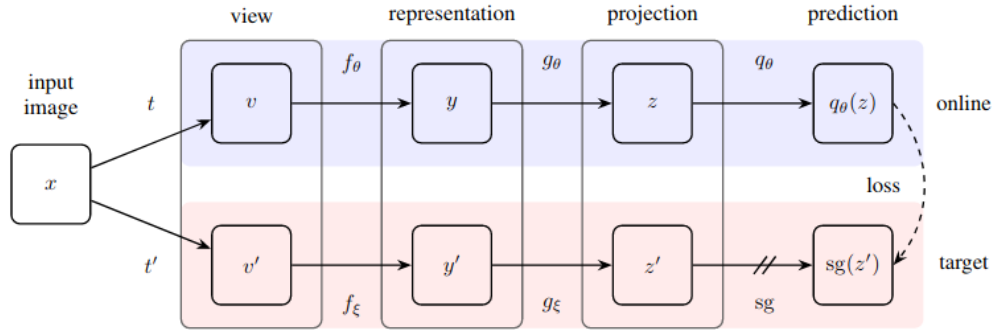


Figure 1: BYOL architecture

I use BYOL (Bootstrap Your Own Latent) as my self-supervised method. The data is augmented with Color Jitter, Random Crop, and Normalization. I chose a batch size of 32 and trained for 100 epochs. I used Adam as the optimizer and used Cosine Annealing learning rate,

$$\eta_t = 2.5 \times 10^{-5} \left( 1 + \cos \left( \frac{\pi t}{50} \right) \right).$$

I also split the training data to training and validation sets with a 9:1 ratio.

## 1.2

Setting	Pre-training (Mini-ImageNet)	Fine-tuning (Office-Home dataset)	Validation accuracy (Office-Home dataset)
A	-	Train full model (backbone + classifier)	0.5271
B	w/ label (TAs have provided this backbone)	Train full model (backbone + classifier)	0.5493
C	w/o label (Your SSL pre-trained backbone)	Train full model (backbone + classifier)	0.5025
D	w/ label (TAs have provided this backbone)	Fix the backbone. Train classifier only	0.2365
E	w/o label (Your SSL pre-trained backbone)	Fix the backbone. Train classifier only	0.1527

Table 1: Fine-tune results under the 5 settings

With setting B, using the provided supervised backbone, the validation accuracy improved from 0.5271 to 0.5493. In setting C, the validation accuracy decreased to 0.5025, my guess is this is because the augmentation applied to the data is not identical in the pretraining and fine-tuning phase, so the model is not able to adapt well to the office-home dataset. With the backbone frozen in settings D and E, The model performs poorly. This is probably because the weights of the backbone are not being updated according to the training data, causing it to learn poorly.

## 1.3

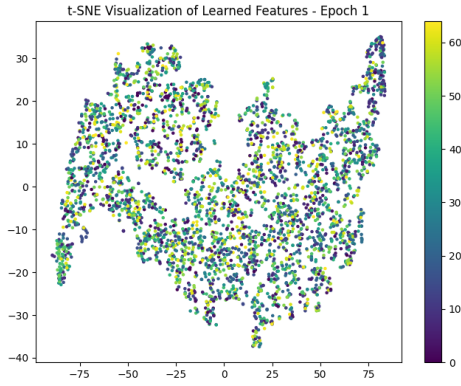


Figure 2: First Epoch

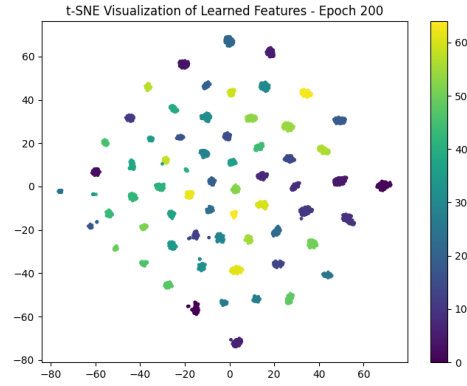


Figure 3: Last Epoch

- **First Epoch:** Before training, the data is random and scattered, indicating that the model has not yet learned to distinguish the classes well.
- **Last Epoch:** By the final epoch, the data is grouped into clusters corresponding to different classes, indicating the model has learned to differentiate between the classes

2

2.1

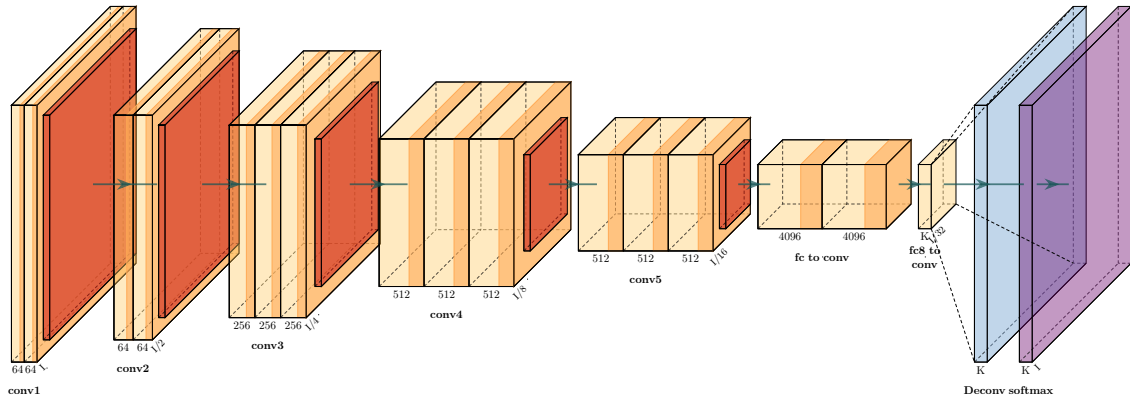


Figure 4: FCN32s architecture

2.2

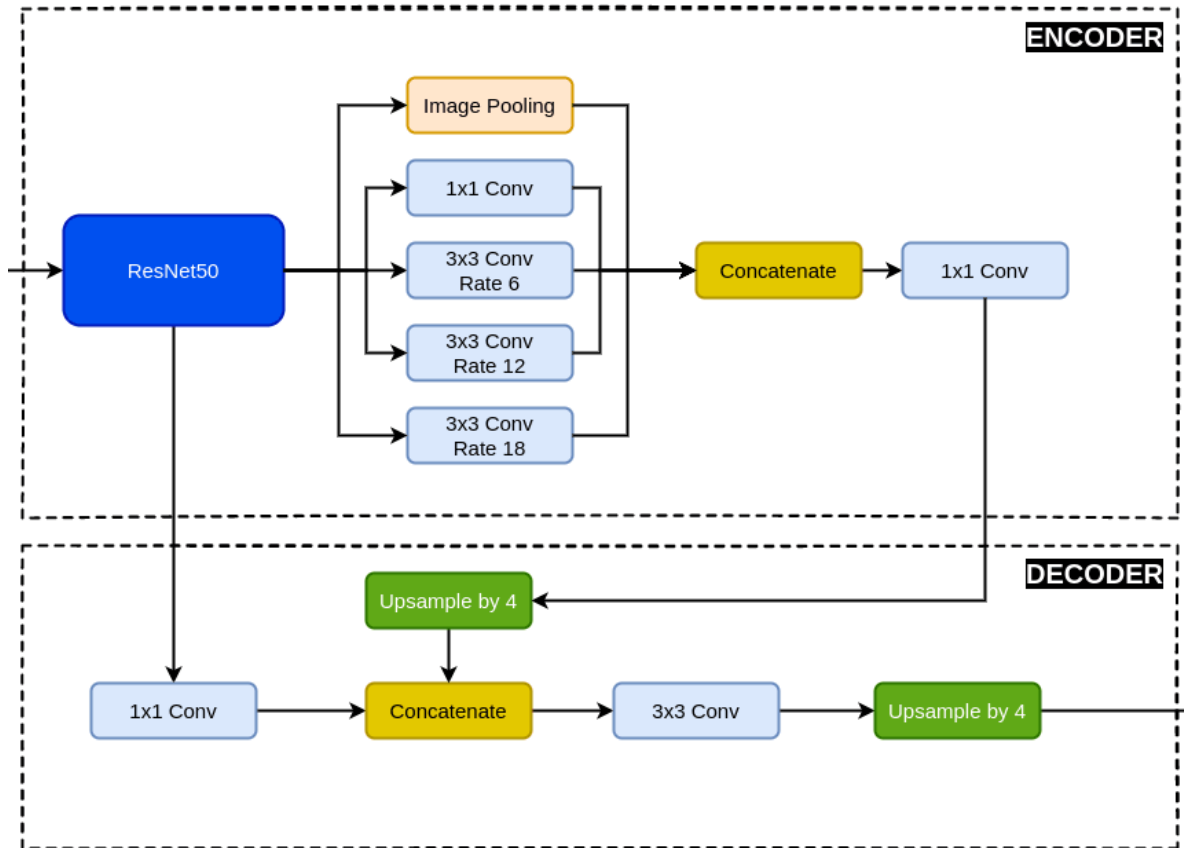


Figure 5: DeepLabv3-Resnet50 architecture

DeepLabv3 uses a ResNet50 backbone as well as a ASPP (Atrous Spatial Pyramid Pooling), which

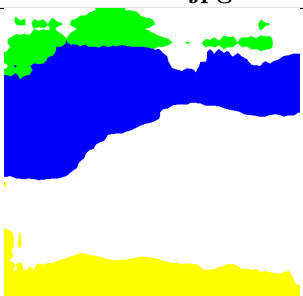
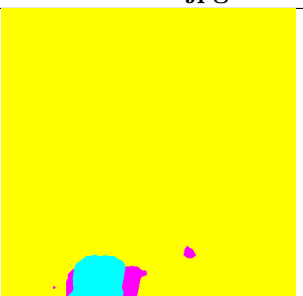



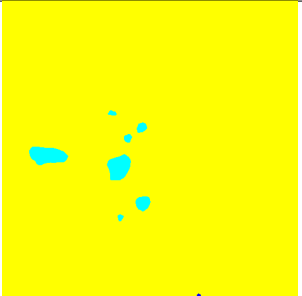


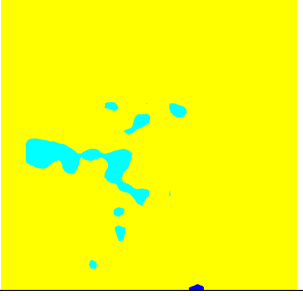
applies multiple dilated convolutions with different rates and pools together the outputs, allowing for more detailed and precise segmentation.

## 2.3

Model	FCN32s	DeepLabv3
MIoU	0.599	0.760

Table 2: MIoUs of the two models

## 2.4

image	0013_sat.jpg	0062_sat.jpg	0104_sat.jpg
Epoch 1			
Epoch 29			
Epoch 75			

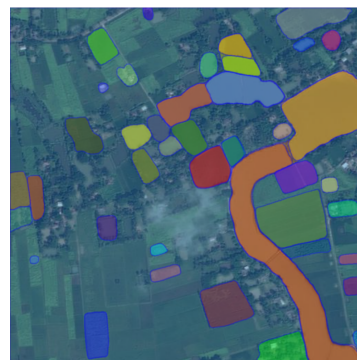
## 2.5



0006\_sat.jpg



0019\_sat.jpg



0060\_sat.jpg

I segmented 0006\_sat.jpg, 0019\_sat.jpg, 0060\_sat.jpg on Meta's Segment Anything website (<https://segment-anything.com/demo>), the model is prompted to segment everything in the image.