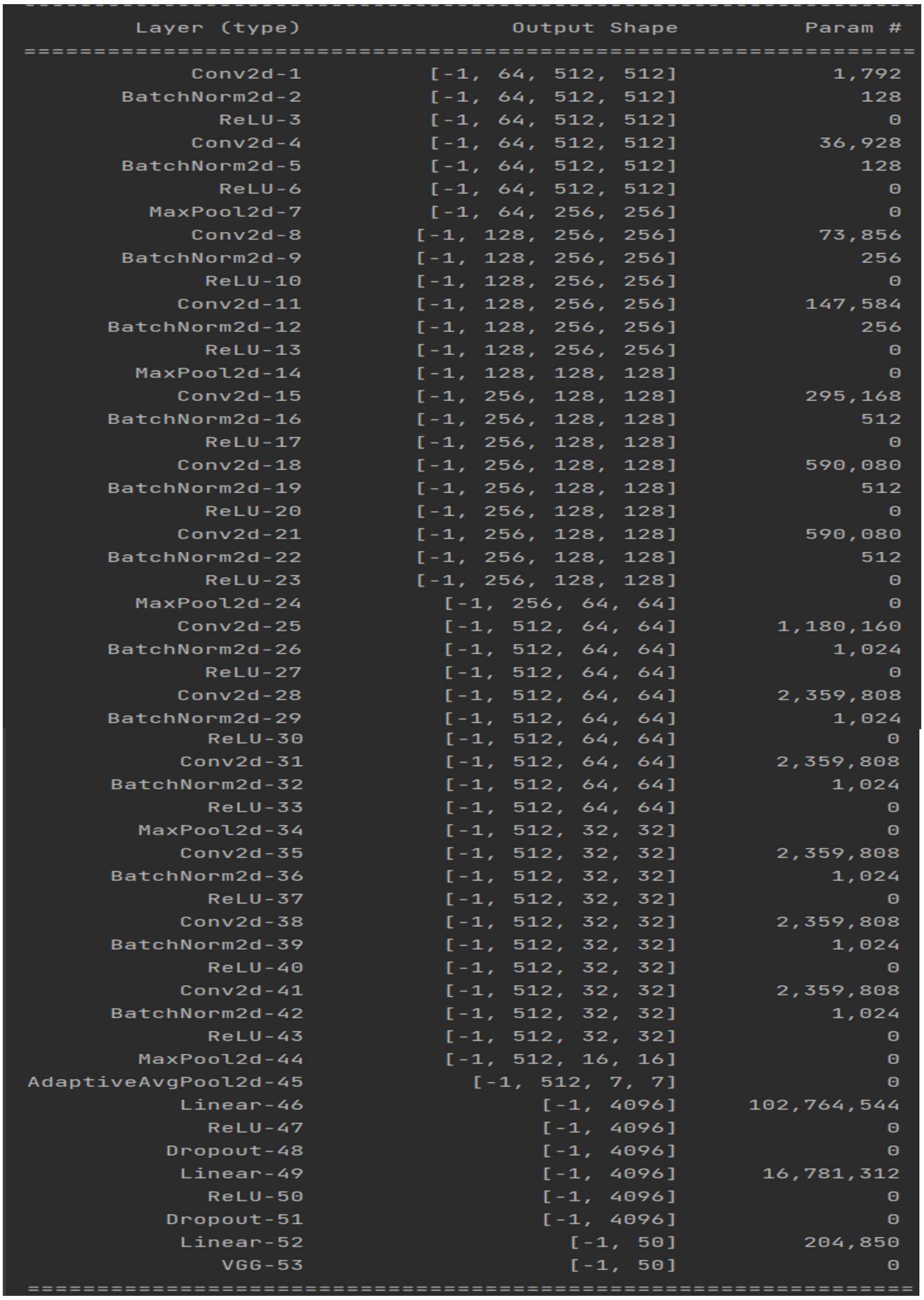
Deep Learning for Computer Vision Homework 1

R10522606 曾柏翔

Problem 1

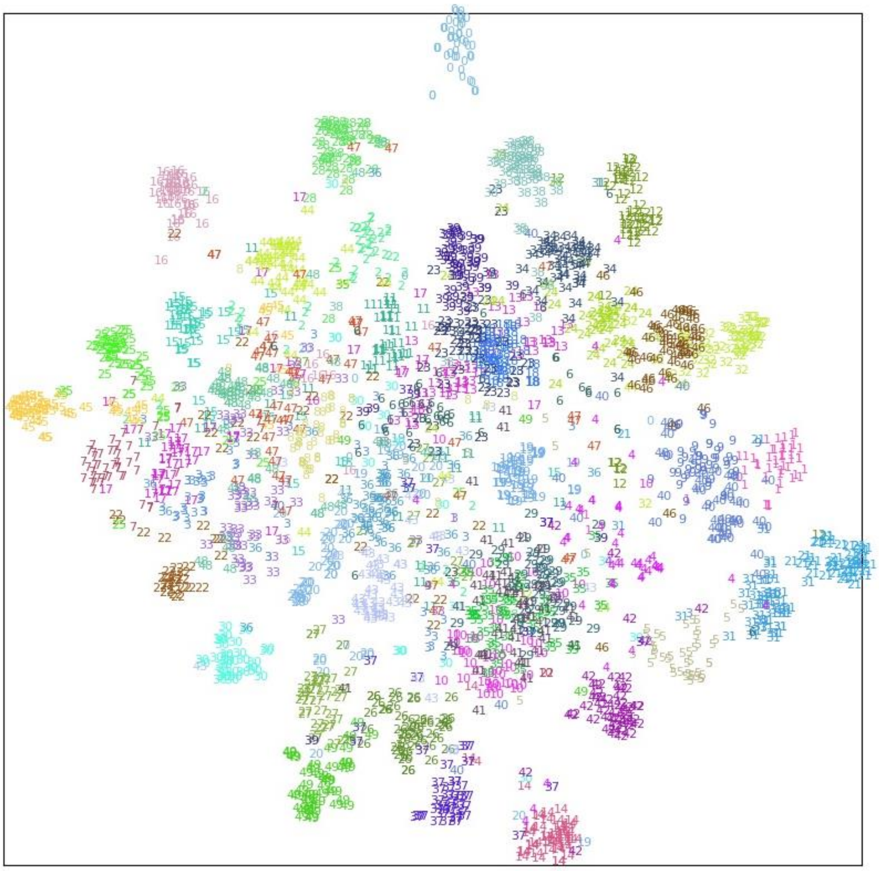
1. Print the network architecture of your model.



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1        [-1, 64, 512, 512]           1,792
       BatchNorm2d-2        [-1, 64, 512, 512]             128
              ReLU-3        [-1, 64, 512, 512]               0
            Conv2d-4        [-1, 64, 512, 512]          36,928
       BatchNorm2d-5        [-1, 64, 512, 512]             128
              ReLU-6        [-1, 64, 512, 512]               0
         MaxPool2d-7        [-1, 64, 256, 256]               0
            Conv2d-8       [-1, 128, 256, 256]          73,856
       BatchNorm2d-9       [-1, 128, 256, 256]             256
             ReLU-10       [-1, 128, 256, 256]               0
           Conv2d-11       [-1, 128, 256, 256]         147,584
      BatchNorm2d-12       [-1, 128, 256, 256]             256
             ReLU-13       [-1, 128, 256, 256]               0
        MaxPool2d-14       [-1, 128, 128, 128]               0
           Conv2d-15       [-1, 256, 128, 128]         295,168
      BatchNorm2d-16       [-1, 256, 128, 128]             512
             ReLU-17       [-1, 256, 128, 128]               0
           Conv2d-18       [-1, 256, 128, 128]         590,080
      BatchNorm2d-19       [-1, 256, 128, 128]             512
             ReLU-20       [-1, 256, 128, 128]               0
           Conv2d-21       [-1, 256, 128, 128]         590,080
      BatchNorm2d-22       [-1, 256, 128, 128]             512
             ReLU-23       [-1, 256, 128, 128]               0
        MaxPool2d-24         [-1, 256, 64, 64]               0
           Conv2d-25         [-1, 512, 64, 64]       1,180,160
      BatchNorm2d-26         [-1, 512, 64, 64]           1,024
             ReLU-27         [-1, 512, 64, 64]               0
           Conv2d-28         [-1, 512, 64, 64]       2,359,808
      BatchNorm2d-29         [-1, 512, 64, 64]           1,024
             ReLU-30         [-1, 512, 64, 64]               0
           Conv2d-31         [-1, 512, 64, 64]       2,359,808
      BatchNorm2d-32         [-1, 512, 64, 64]           1,024
             ReLU-33         [-1, 512, 64, 64]               0
        MaxPool2d-34         [-1, 512, 32, 32]               0
           Conv2d-35         [-1, 512, 32, 32]       2,359,808
      BatchNorm2d-36         [-1, 512, 32, 32]           1,024
             ReLU-37         [-1, 512, 32, 32]               0
           Conv2d-38         [-1, 512, 32, 32]       2,359,808
      BatchNorm2d-39         [-1, 512, 32, 32]           1,024
             ReLU-40         [-1, 512, 32, 32]               0
           Conv2d-41         [-1, 512, 32, 32]       2,359,808
      BatchNorm2d-42         [-1, 512, 32, 32]           1,024
             ReLU-43         [-1, 512, 32, 32]               0
        MaxPool2d-44         [-1, 512, 16, 16]               0
AdaptiveAvgPool2d-45           [-1, 512, 7, 7]               0
           Linear-46                [-1, 4096]     102,764,544
             ReLU-47                [-1, 4096]               0
          Dropout-48                [-1, 4096]               0
           Linear-49                [-1, 4096]      16,781,312
             ReLU-50                [-1, 4096]               0
          Dropout-51                [-1, 4096]               0
           Linear-52                  [-1, 50]         204,850
              VGG-53                  [-1, 50]               0
================================================================
```

2. Report accuracy of model on the validation set.
   → Accuracy : 0.8268

3. Visualize the classification result on validation set by implementing t-SNE on output features of the second last layer. Briefly explain your result of the t-SNE visualization.

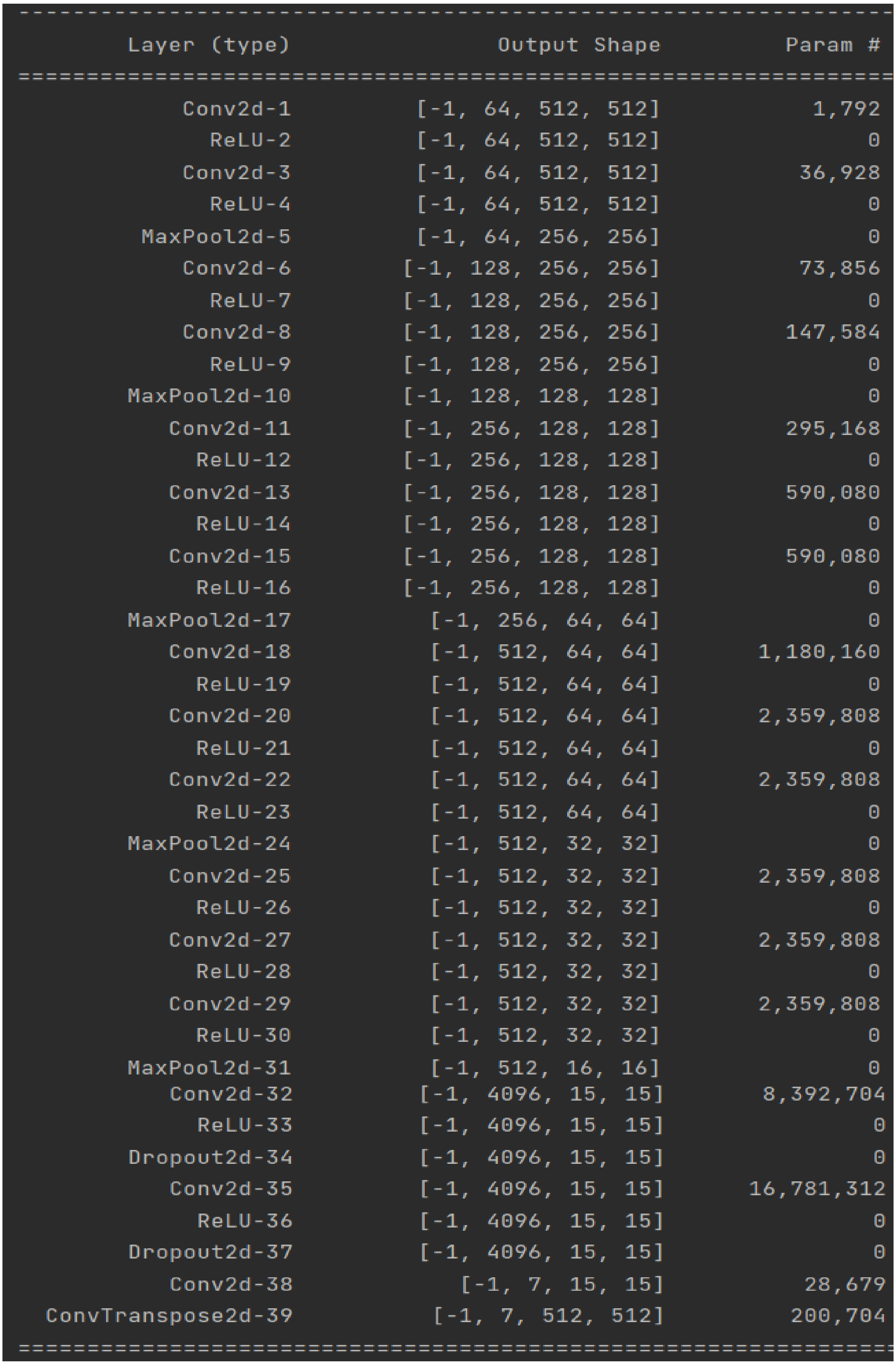


從 t-SNE 可以看出各個 class 的特性，可以注意到有些 claas 是自成一區，而有些則會與其他 calss 重疊。這可以簡單想成 model 對於各個 class 的辨識能力，若是自成一區，代表 class 對於有著鮮明的特徵，使得 model 能夠輕易地辨識出來；而若是與其他的重疊，代表重疊的 class 之間可能存在著某些相似的特徵，造成 model 混淆。

舉例來說，可以很明顯的注意到 t-SNE 圖最上方，class 0 就自成一區，其對應到的圖片-腳踏車，在訓練中，並沒有與其相似的 data，因此 model 能夠輕易地辨識腳踏車的特徵；而可以看到 class 32 與 class 46 幾乎都混在一起，兩者所對應到的皆為車輛相關的圖片，因此對於 model 來說就很容易辨識錯誤。(32 與 46 甚至我用肉眼都沒辦法分辨)

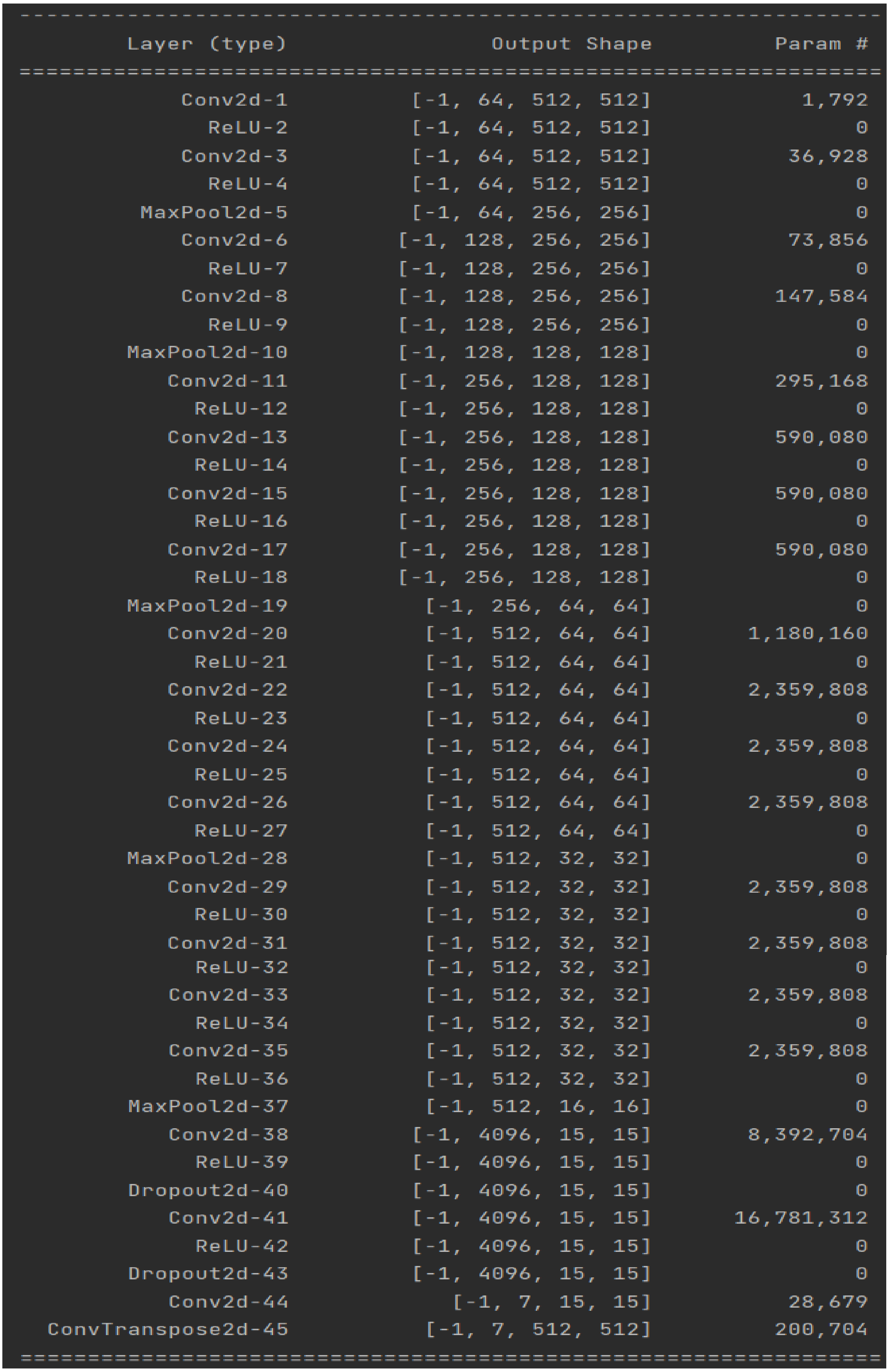
Problem 2

1. Print the network architecture of your VGG16-FCN32s model.



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 512, 512]           1,792
              ReLU-2         [-1, 64, 512, 512]               0
            Conv2d-3         [-1, 64, 512, 512]          36,928
              ReLU-4         [-1, 64, 512, 512]               0
         MaxPool2d-5         [-1, 64, 256, 256]               0
            Conv2d-6        [-1, 128, 256, 256]          73,856
              ReLU-7        [-1, 128, 256, 256]               0
            Conv2d-8        [-1, 128, 256, 256]         147,584
              ReLU-9        [-1, 128, 256, 256]               0
        MaxPool2d-10        [-1, 128, 128, 128]               0
           Conv2d-11        [-1, 256, 128, 128]         295,168
             ReLU-12        [-1, 256, 128, 128]               0
           Conv2d-13        [-1, 256, 128, 128]         590,080
             ReLU-14        [-1, 256, 128, 128]               0
           Conv2d-15        [-1, 256, 128, 128]         590,080
             ReLU-16        [-1, 256, 128, 128]               0
        MaxPool2d-17          [-1, 256, 64, 64]               0
           Conv2d-18          [-1, 512, 64, 64]       1,180,160
             ReLU-19          [-1, 512, 64, 64]               0
           Conv2d-20          [-1, 512, 64, 64]       2,359,808
             ReLU-21          [-1, 512, 64, 64]               0
           Conv2d-22          [-1, 512, 64, 64]       2,359,808
             ReLU-23          [-1, 512, 64, 64]               0
        MaxPool2d-24          [-1, 512, 32, 32]               0
           Conv2d-25          [-1, 512, 32, 32]       2,359,808
             ReLU-26          [-1, 512, 32, 32]               0
           Conv2d-27          [-1, 512, 32, 32]       2,359,808
             ReLU-28          [-1, 512, 32, 32]               0
           Conv2d-29          [-1, 512, 32, 32]       2,359,808
             ReLU-30          [-1, 512, 32, 32]               0
        MaxPool2d-31          [-1, 512, 16, 16]               0
           Conv2d-32         [-1, 4096, 15, 15]       8,392,704
             ReLU-33         [-1, 4096, 15, 15]               0
        Dropout2d-34         [-1, 4096, 15, 15]               0
           Conv2d-35         [-1, 4096, 15, 15]      16,781,312
             ReLU-36         [-1, 4096, 15, 15]               0
        Dropout2d-37         [-1, 4096, 15, 15]               0
           Conv2d-38            [-1, 7, 15, 15]          28,679
  ConvTranspose2d-39          [-1, 7, 512, 512]         200,704
================================================================
```

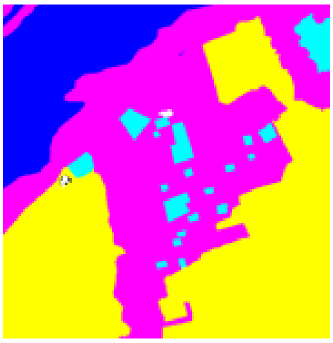
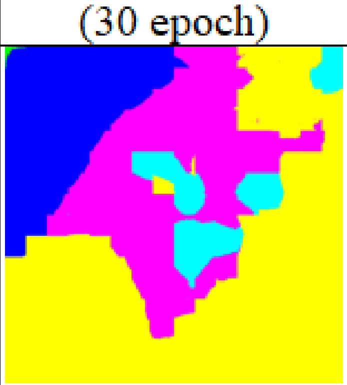
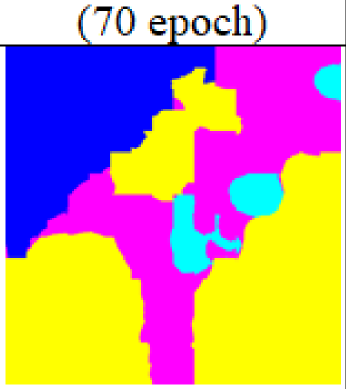
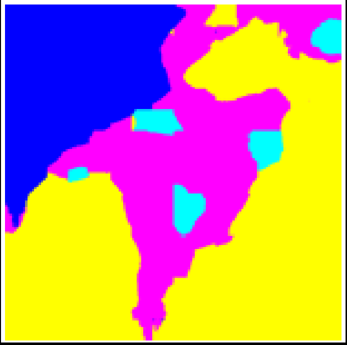
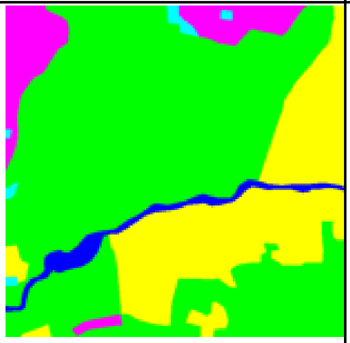
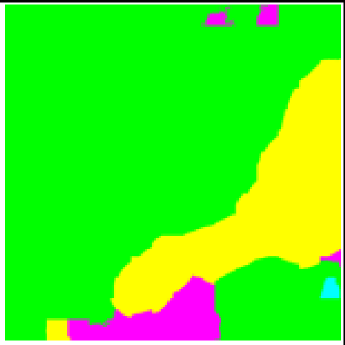
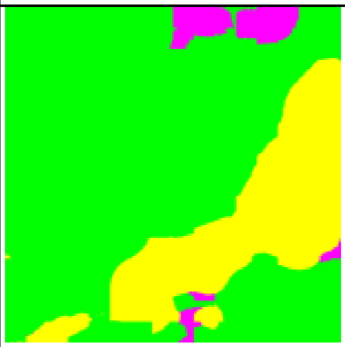
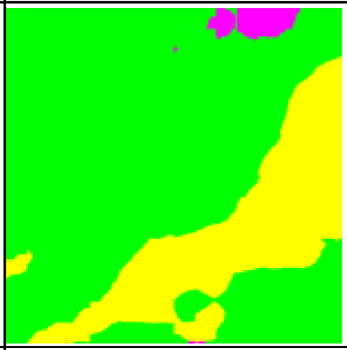
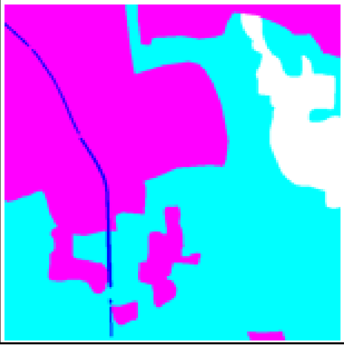
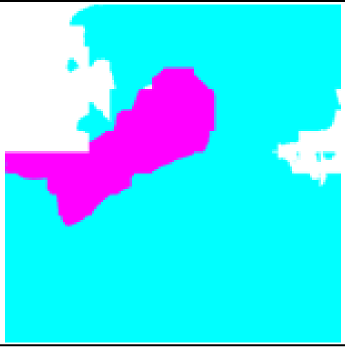
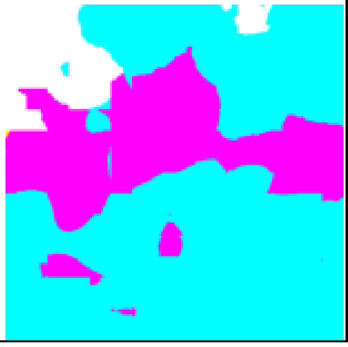
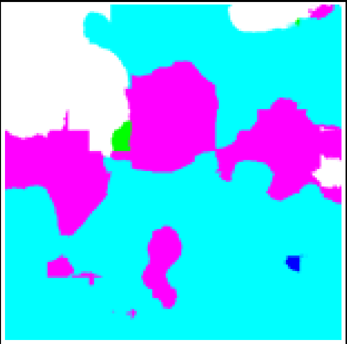
2. Implement an improved model which performs better than your baseline model. Print the network architecture of this model.



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1        [-1, 64, 512, 512]           1,792
              ReLU-2        [-1, 64, 512, 512]               0
            Conv2d-3        [-1, 64, 512, 512]          36,928
              ReLU-4        [-1, 64, 512, 512]               0
         MaxPool2d-5        [-1, 64, 256, 256]               0
            Conv2d-6       [-1, 128, 256, 256]          73,856
              ReLU-7       [-1, 128, 256, 256]               0
            Conv2d-8       [-1, 128, 256, 256]         147,584
              ReLU-9       [-1, 128, 256, 256]               0
        MaxPool2d-10       [-1, 128, 128, 128]               0
           Conv2d-11       [-1, 256, 128, 128]         295,168
             ReLU-12       [-1, 256, 128, 128]               0
           Conv2d-13       [-1, 256, 128, 128]         590,080
             ReLU-14       [-1, 256, 128, 128]               0
           Conv2d-15       [-1, 256, 128, 128]         590,080
             ReLU-16       [-1, 256, 128, 128]               0
           Conv2d-17       [-1, 256, 128, 128]         590,080
             ReLU-18       [-1, 256, 128, 128]               0
        MaxPool2d-19         [-1, 256, 64, 64]               0
           Conv2d-20         [-1, 512, 64, 64]       1,180,160
             ReLU-21         [-1, 512, 64, 64]               0
           Conv2d-22         [-1, 512, 64, 64]       2,359,808
             ReLU-23         [-1, 512, 64, 64]               0
           Conv2d-24         [-1, 512, 64, 64]       2,359,808
             ReLU-25         [-1, 512, 64, 64]               0
           Conv2d-26         [-1, 512, 64, 64]       2,359,808
             ReLU-27         [-1, 512, 64, 64]               0
        MaxPool2d-28         [-1, 512, 32, 32]               0
           Conv2d-29         [-1, 512, 32, 32]       2,359,808
             ReLU-30         [-1, 512, 32, 32]               0
           Conv2d-31         [-1, 512, 32, 32]       2,359,808
             ReLU-32         [-1, 512, 32, 32]               0
           Conv2d-33         [-1, 512, 32, 32]       2,359,808
             ReLU-34         [-1, 512, 32, 32]               0
           Conv2d-35         [-1, 512, 32, 32]       2,359,808
             ReLU-36         [-1, 512, 32, 32]               0
        MaxPool2d-37         [-1, 512, 16, 16]               0
           Conv2d-38        [-1, 4096, 15, 15]       8,392,704
             ReLU-39        [-1, 4096, 15, 15]               0
        Dropout2d-40        [-1, 4096, 15, 15]               0
           Conv2d-41        [-1, 4096, 15, 15]      16,781,312
             ReLU-42        [-1, 4096, 15, 15]               0
        Dropout2d-43        [-1, 4096, 15, 15]               0
           Conv2d-44           [-1, 7, 15, 15]          28,679
  ConvTranspose2d-45         [-1, 7, 512, 512]         200,704
================================================================
```

3. Report mIoU of the improved model on the validation set.
   → mIoU : 0.688

4. Show the predicted segmentation mask of “0010_sat.jpg”, “0097_sat.jpg”, “0107_sat.jpg” during the early, middle, and the final stage during the training process of this improved model.

	Validation	early stage (30 epoch)	middle stage (70 epoch)	final stage (100 epoch)
0010_sat.jpg				
0097_sat.jpg				
0107_sat.jpg				

Reference

[1] Vgg16&FCN32
https://blog.csdn.net/gbz3300255/article/details/105582572

[2] t-SNE
https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

[3] Print the network architecture of model.
https://stackoverflow.com/questions/42480111/model-summary-in-pytorch

[4] How to use “register_forward_hook”.
https://zhuanlan.zhihu.com/p/87853615