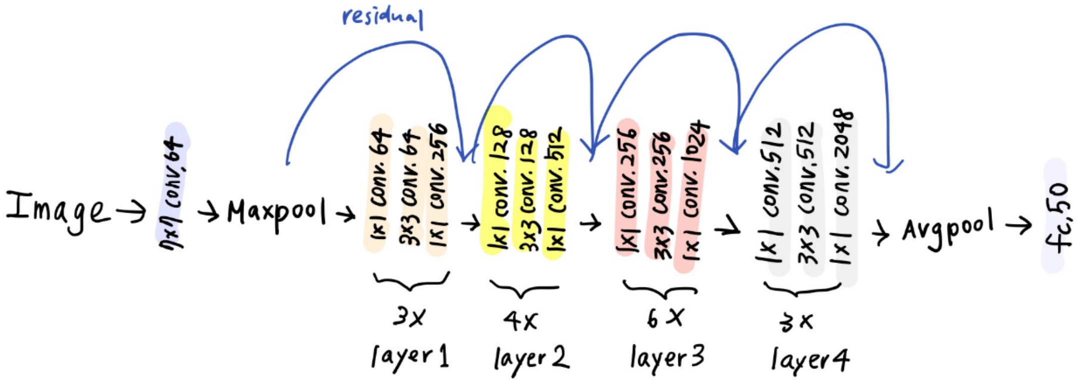
DLCV HW1 R10921A36 石子仙

Part1

1. **(2%)** Draw the network architecture of method A or B.

   My Model B is resnet50



2. **(1%)** Report accuracy of your models (both A, B) on the validation set.

   Model A: 0.456

   Model B: 0.8652

3. **(4%)** Report your implementation details of model A. Including but not limited to optimizer, loss function, cross validation method

- The model's architecture is composed of 5 layer CNN with kernel size of 3*3.
- The loss function is Cross Entropy.
- The optimizer is Adam with learning rate 0.001 and weight decay 0.001.
- I also use a learning rate scheduler called ReduceLROnPlateau, it will turn down the learning rate when it doesn't improve for n epochs. I choose n to 5.
- I've used early stop skill with maximum epochs to be 300, early stop with 30 epochs.

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

   I choose resnet50 with pretrained weights for model B. It contains more convolutional layers than model A.

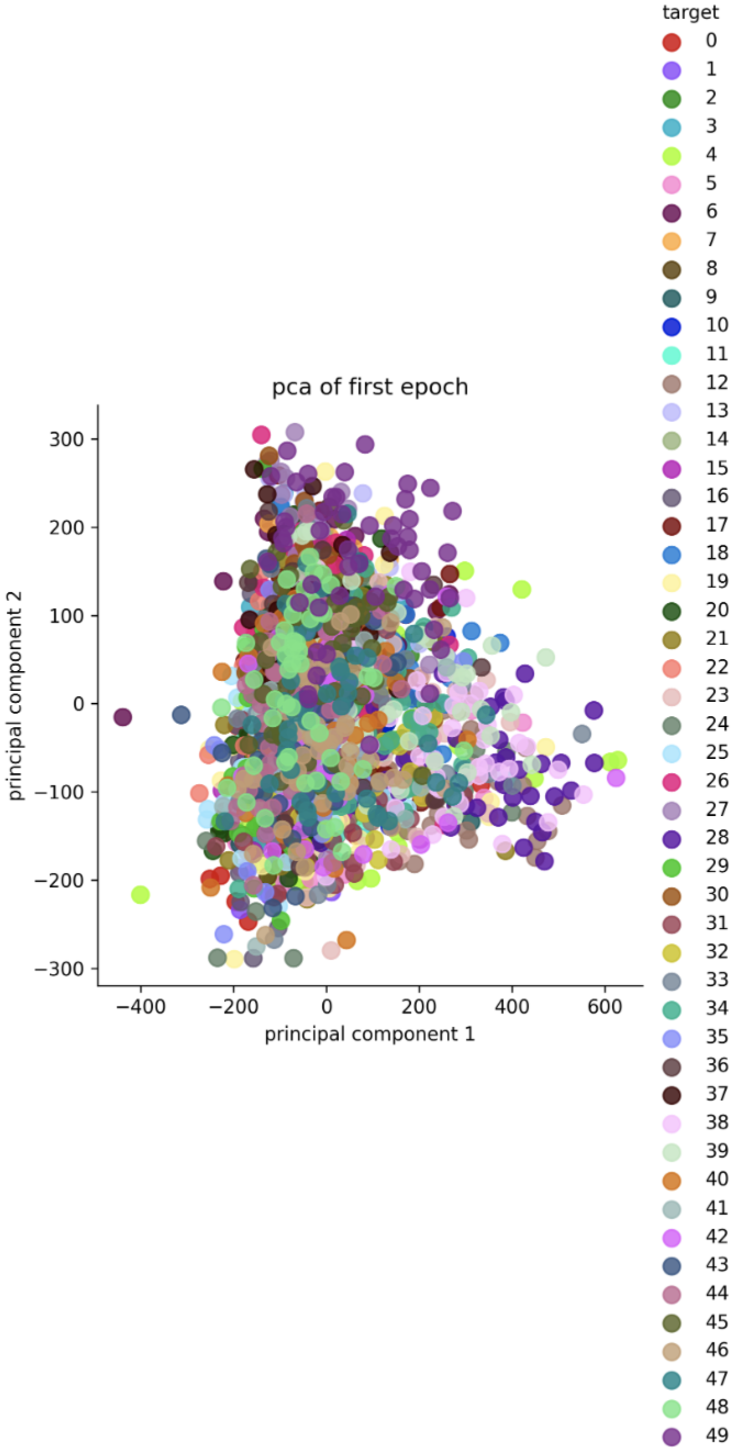
   It adds residual blocks to improve training. It also has pretrained weight, model A doesn't have.

   I used the same learning rate scheduler but I changed the optimizer to SGD to get better accuracy.

5. **(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.

   The PCA is not very good but we can see the class 28(dolphin) and 49(sunflower) have a little bit of being a cluster.

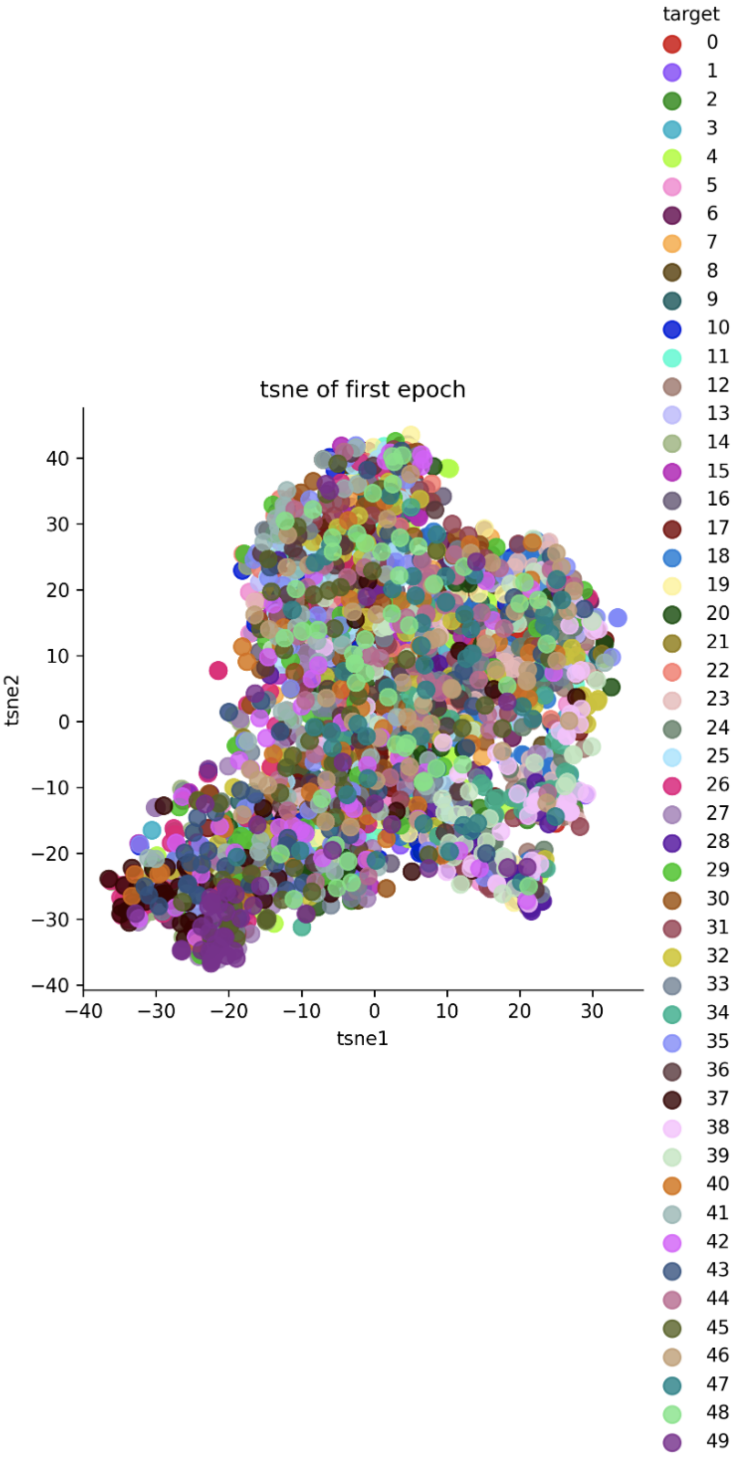
   I think if the model has better performance, it can have a clearly separated PCA visualization.



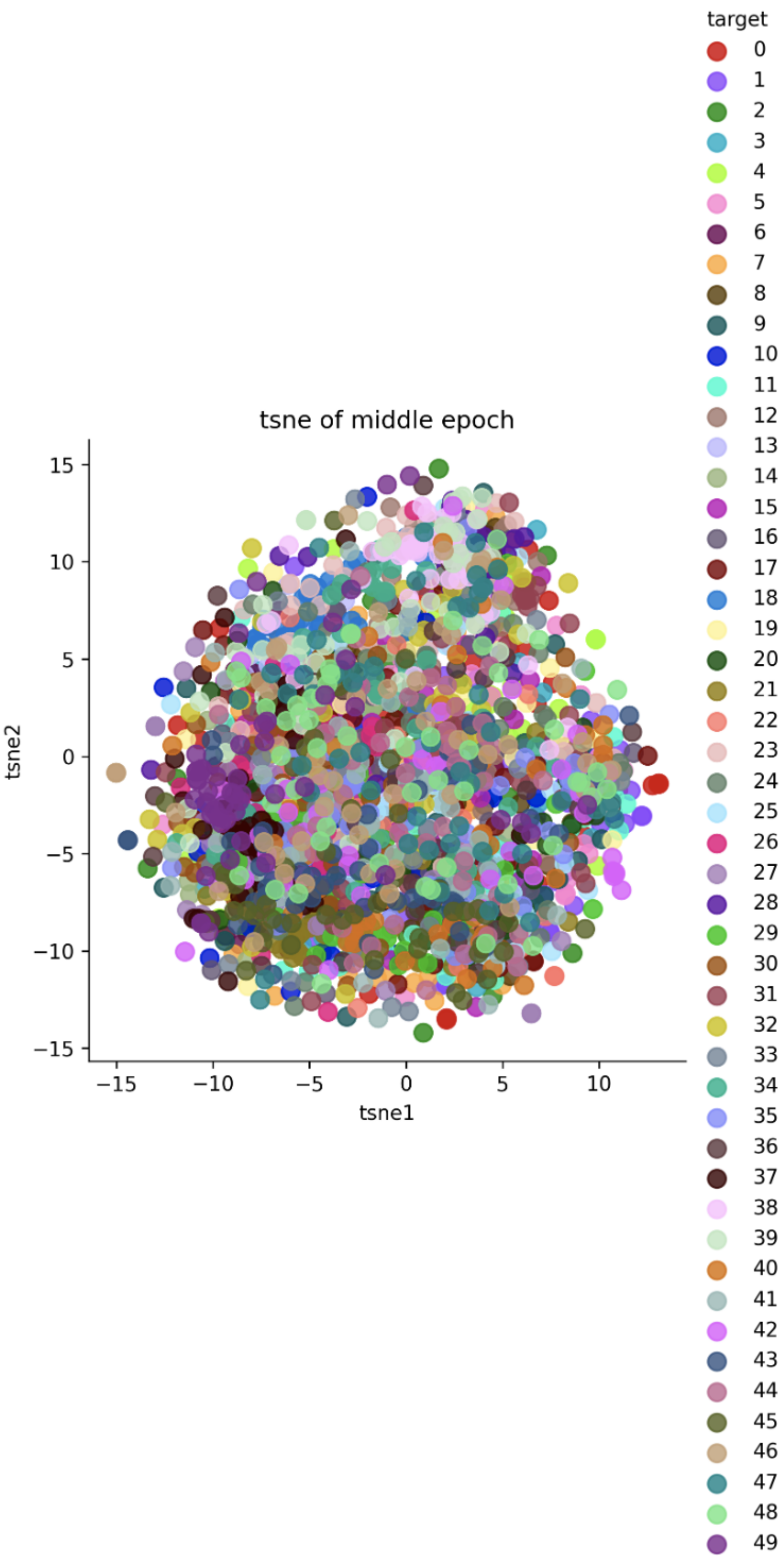
6. **(7%)** Visualize the learned visual representation of **model A**, again on the output of the second last layer, but using **t-SNE** (t-distributed Stochastic Neighbor Embedding) instead. Depict your visualization from **three different epochs** including the first one and the last one. Briefly explain the above results.

My model A’s second last layer is 32*214*214. I flatten them then do the tsne. I think my model A didn’t have good performance so the tsne is awful. Or it can’t have good separation due to being reduced to 2 dimensions. Maybe in 3d spaces which can contain more dimensions can have better results.

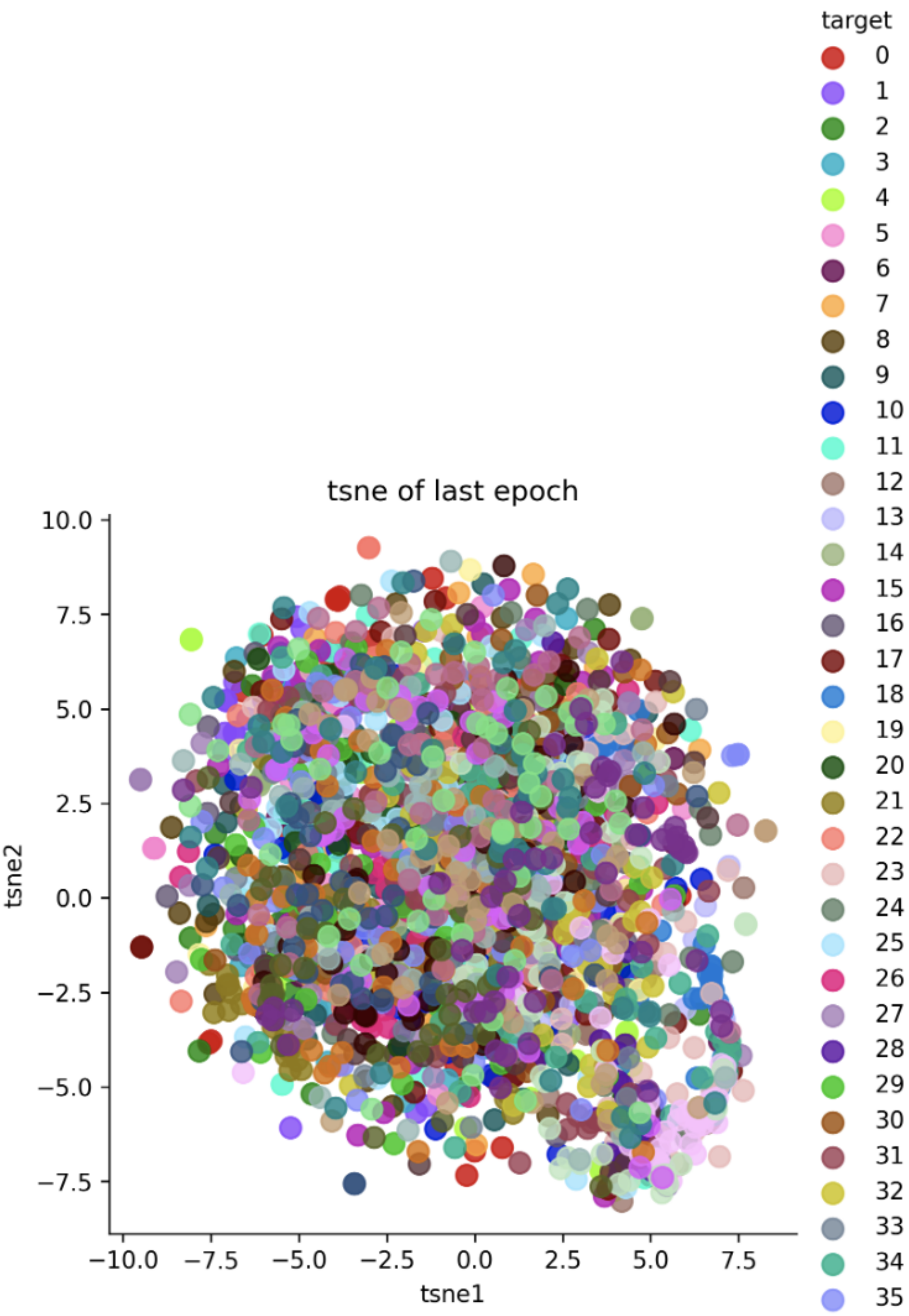
First Epoch:



Middle Epoch:

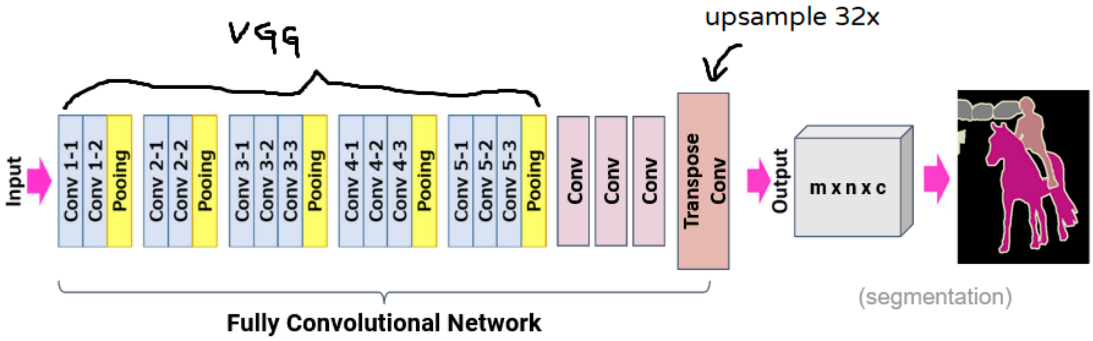

tsne of middle epoch
tsne2
15
10
5
0
−5
−10
−15
tsne1
−15
−10
−5
0
5
10
target
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

Last Epoch:


tsne of last epoch
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
tsne2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
tsne1
target
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

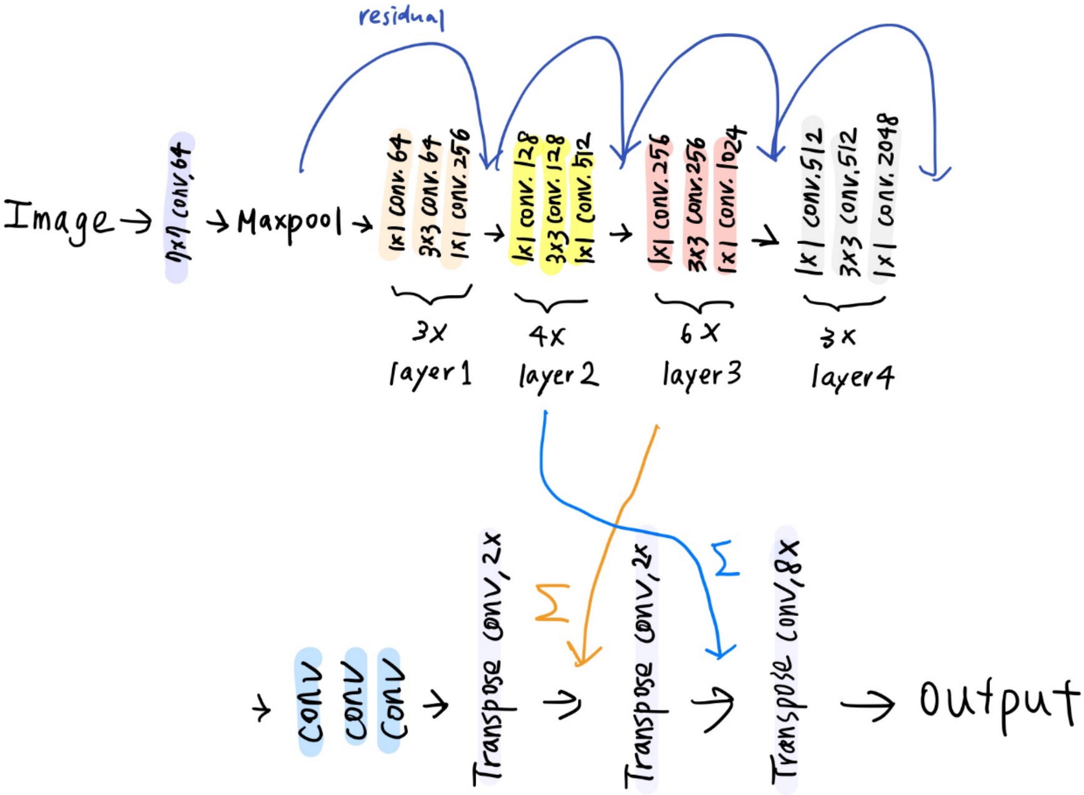
Part 2

1. **(5%)** Draw the network architecture of your VGG16-FCN32s model (model A).



2. **(5%)** Draw the network architecture of the improved model (model B)
and explain it differs from your VGG16-FCN32s model.

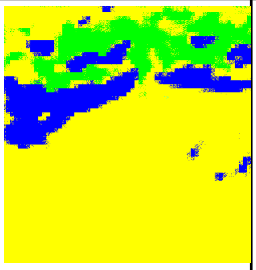
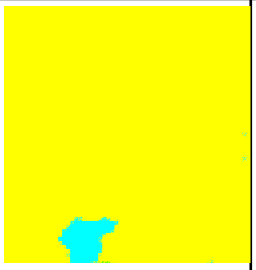
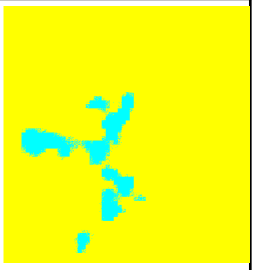
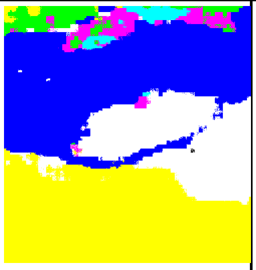
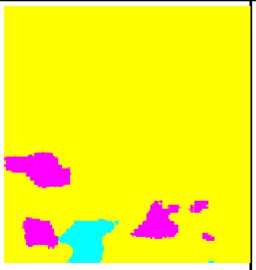
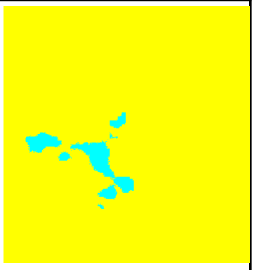
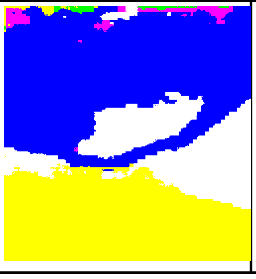
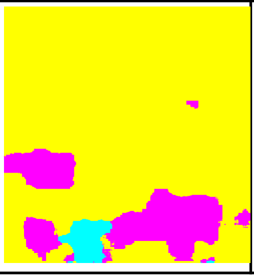
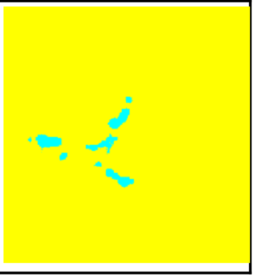
I use the ResNet50-FCN8s model. The backbone is changed to resnet50, and the FCN8 part considers more layers output, which can get clearer maps.



3. **(3%)** Report mIoUs of two models on the validation set.
Model A: 0.610185
Model B: 0.704725

4. **(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.

I use early stop for finding best model, the best model is at 16 epoch

Epoch\Image	0013	0062	0104
Epoch=0			
Epoch=8			
Epoch=16			

- Tips: Given n epochs training, you could save the 1st, (n/2)-th, n-th epoch model, and draw the predicted mask by loading these saved models.