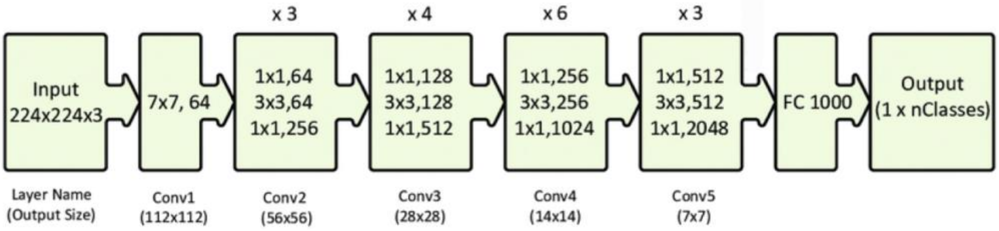
DLCV HW1

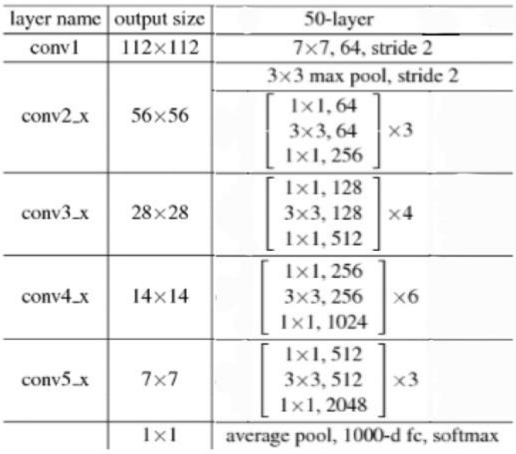
電機三 B10901041 林新晨

Problem 1

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

   Model B:





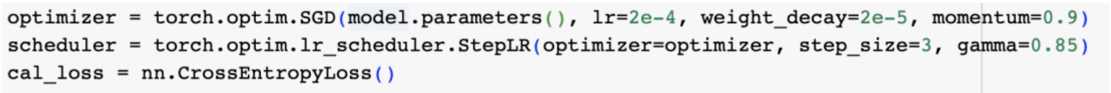
layer name	output size	50-layer
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 max pool, stride 2 [1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[1×1, 128 3×3, 128 1×1, 512] ×4
conv4_x	14×14	[1×1, 256 3×3, 256 1×1, 1024] ×6
conv5_x	7×7	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax

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

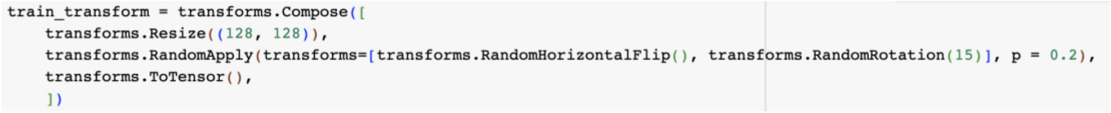
   Accuracy of A: 0.622

   Accuracy of B: 0.86367

3. **Report your implementation details of model A**



```
optimizer = torch.optim.SGD(model.parameters(), lr=2e-4, weight_decay=2e-5, momentum=0.9)
scheduler = torch.optim.lr_scheduler.StepLR(optimizer=optimizer, step_size=3, gamma=0.85)
cal_loss = nn.CrossEntropyLoss()
```

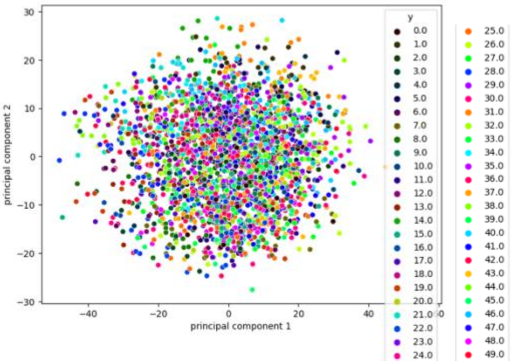


```
train_transform = transforms.Compose([
    transforms.Resize((128, 128)),
    transforms.RandomApply(transforms=[transforms.RandomHorizontalFlip(), transforms.RandomRotation(15)], p = 0.2),
    transforms.ToTensor(),
    ])
```

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

   Model B is Resnet50. The key of ResNet is the use of residual blocks, which contain skip connections (shortcut connections) allowing gradients to propagate through the network easily during training. This helps mitigate the vanishing gradient problem and allows training for deeper networks. Thus, model B has better performance than model A.

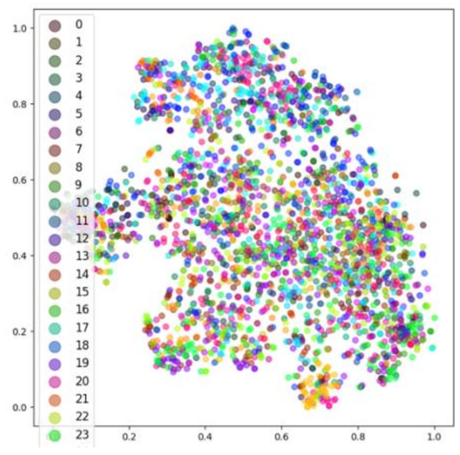
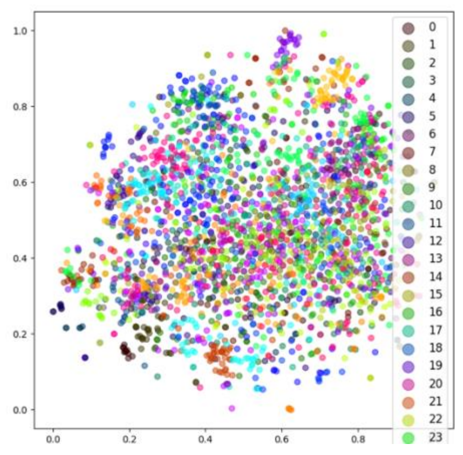
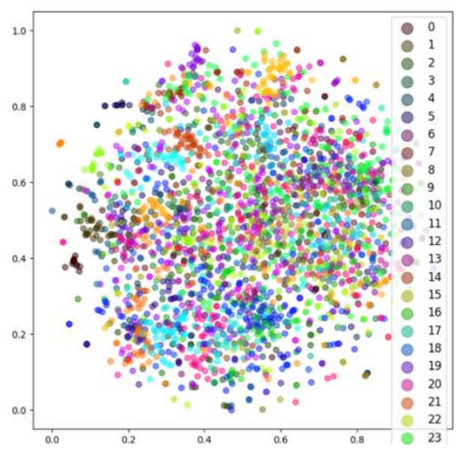
5. **Visualize the learned visual representations of model A on the validation set by implementing PCA (Principal Component Analysis) on the output of the second last layer. Briefly explain your result of the PCA visualization.**



Clustering is not significant in the PCA visualization, and this may result from the poor performance of the model.

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.**

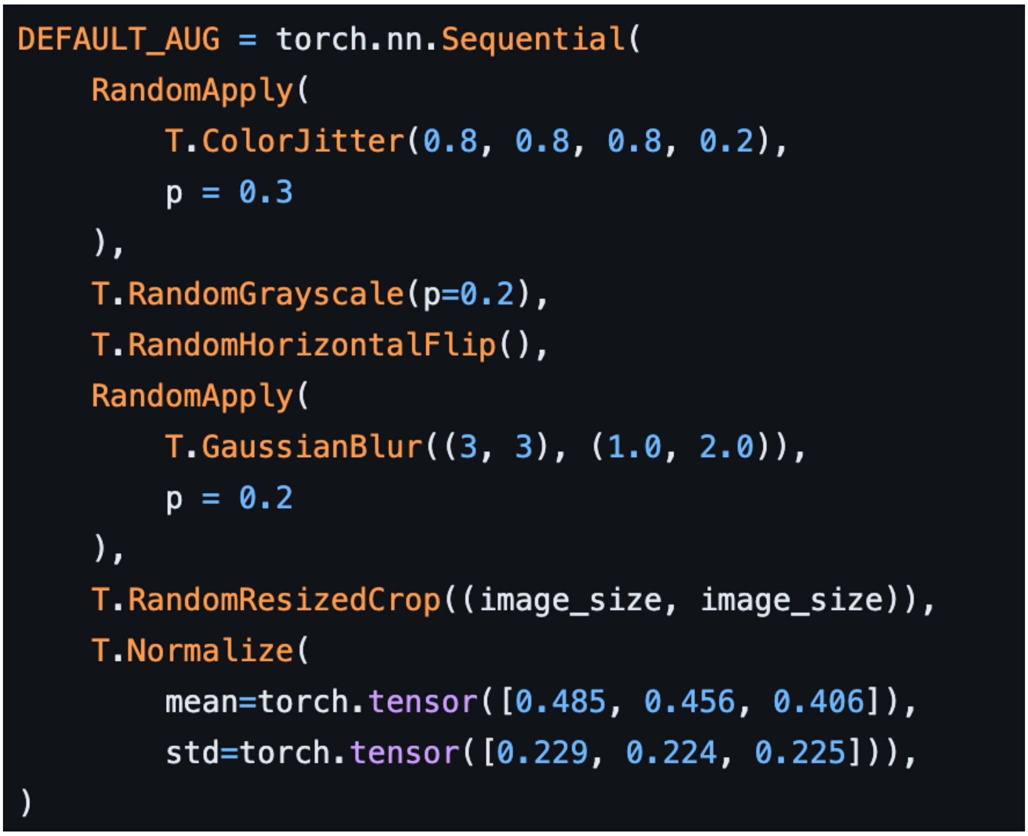
Cluster starts to form as epoch increases from 0 to 99, which means that similar features gather together from early training stage to the late stage.

epoch 0	epoch 39	epoch 99
		

Problem 2

1. **Describe the implementation details of your SSL method for pre-training the ResNet50 backbone.(Include but not limited to the name of the SSL method you used, data augmentation for SSL, learning rate schedule, optimizer, and batch size setting for this pre-training phase)**

   Use BYOL with the recommend github link.
   Data augmentation: same as default augmentation in the BYOL repo.
   optimizer = Adam(lr = 3e-4)
   Batch size: 32



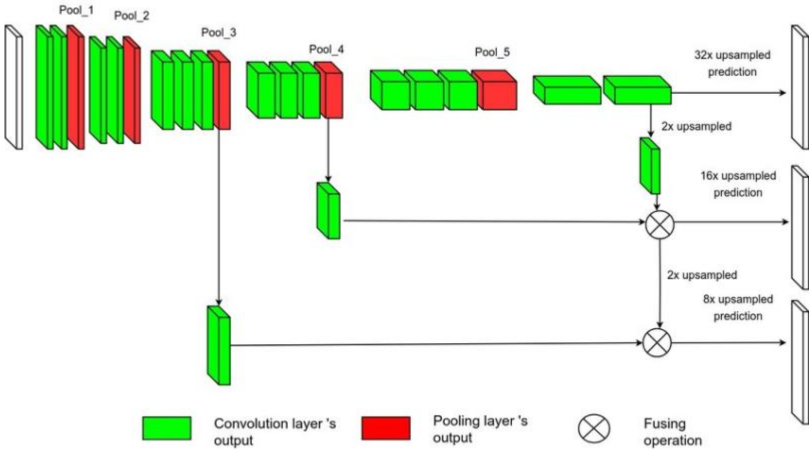
```
DEFAULT_AUG = torch.nn.Sequential(
    RandomApply(
        T.ColorJitter(0.8, 0.8, 0.8, 0.2),
        p = 0.3
    ),
    T.RandomGrayscale(p=0.2),
    T.RandomHorizontalFlip(),
    RandomApply(
        T.GaussianBlur((3, 3), (1.0, 2.0)),
        p = 0.2
    ),
    T.RandomResizedCrop((image_size, image_size)),
    T.Normalize(
        mean=torch.tensor([0.485, 0.456, 0.406]),
        std=torch.tensor([0.229, 0.224, 0.225])),
)
```

2. **Please conduct the Image classification on Office-Home dataset as the downstream task. Also, please complete the following Table, which contains different image classification setting, and discuss/analyze the results.**

Setting	Validation Accuracy
A	0.048
B	0.31
C	0.228
D	0.272
E	0.198

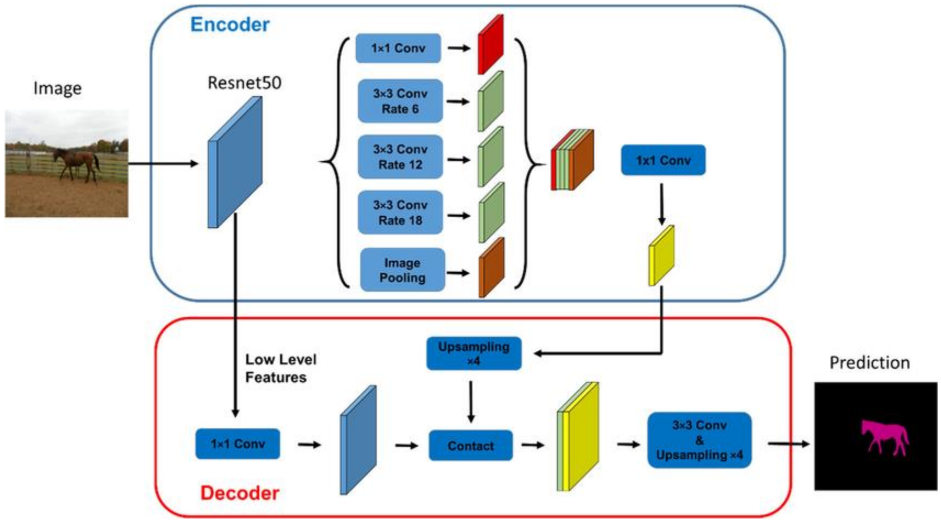
Problem 3

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.**

   Deeplabv3-ResNet50



   The difference:

   B: Deeplabv3-ResNet50 employs ResNet50 backbone, a deeper and more powerful architecture known for its feature extraction capabilities. It utilizes dilated convolutions and a spatial pyramid pooling module.

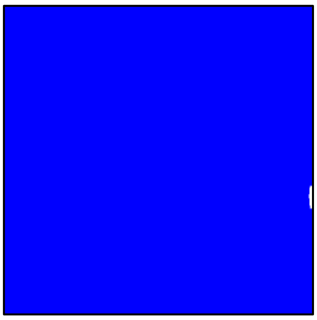
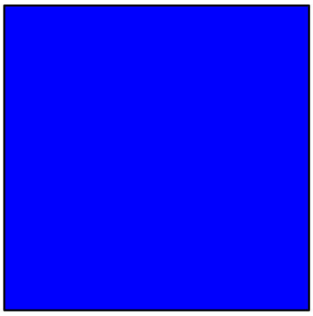
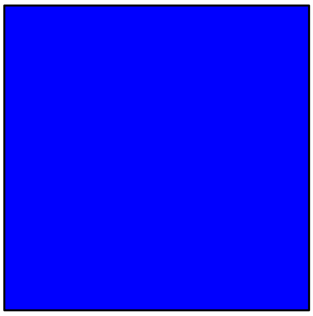
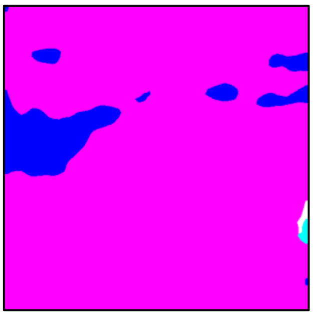
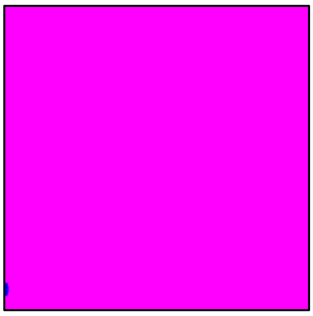
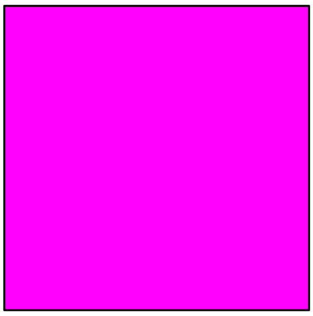
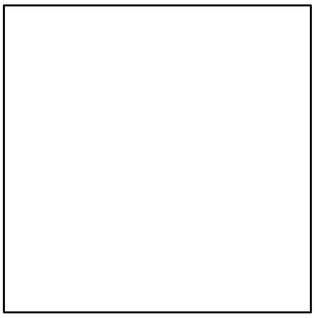
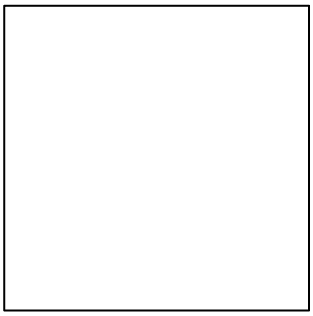
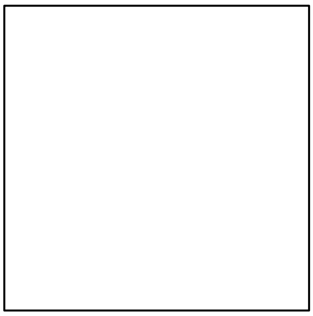
   A: FCN32 generate pixel-wise segmentation maps using skip connections and upsampling layers. It may not capture fine-grained details as effectively as DeepLabV3.

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

   Model A: 0.533

   Model B: 0.7478

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.**

	013	062	104
early epoch0			
mid epoch18			
final epoch50			
Ground truth