Deep Learning for Computer Vision HW1 Report

Note. Only Assignment 1 provides a template. Please make sure you complete all the requirements for each question and respond in the order of the question numbers

P1.

(5%) Describe the implementation details of your SSL method for pre-training the ResNet50 backbone. (**including but not limited to** the name of the SSL method & data augmentation techniques you used, learning rate schedule, optimizer, and batch size setting for this pre-training phase. 150-300 words are enough)

Answers (Your implementation details):

I adopted the designated SSL method DINO (self-distillation with no labels) for pre-training the ResNet-50 backbone. I modified the training code heavily from the official DINO repo. DINO trains a student network to match the output distribution of a teacher network. Both networks share the same ResNet50 architecture. A crucial component in DINO is its multicrop augmentation strategy. During each iteration, two large global crop (224*224) and several small local crop (96*96) are generated from each image. The student network process all crops, while the teacher network receives only the global ones.

As for data augmentation, I used several methods from torchvision.transforms, including random horizontal flip, color jitter, and gray scale. Besides the hyperparameters list in the below table, weight_decay = 1e-4, warmup_epochs = 10, local_crop = 6, local_crop_scale (0.05, 0.14), global_crop_scale (0.14, 1).

The model is trained on Mini-ImageNet dataset, which consists of 38,400 84x84 RGB images of 64 classes (each class has 600 images).

Parameter:

Learning Rate	Batch Size	Optimizer	Scheduler	Epochs	Loss Function
3e-2	128	sgd	cosine	500	cross entropy

(20%) Please conduct the Image classification on Office-Home dataset as the downstream task. Also, please complete the following Table, which contains different image classification settings, and **discuss/analyze the results**.

Set	Accuracy
A	33.50 %
B	80.05 %
C	69.46 %
D	66.75 %
E	51.72 %

Answers:

For all five settings, the models were finetuned on Office-Home dataset using the same hyperparameters:
epoch: 25
batch_size = 32
learning_rate = 1e-4
weight_decay = 1e-4
scheduler = Reduce LROnPlateau

From the accuracy results, two observations can be made.

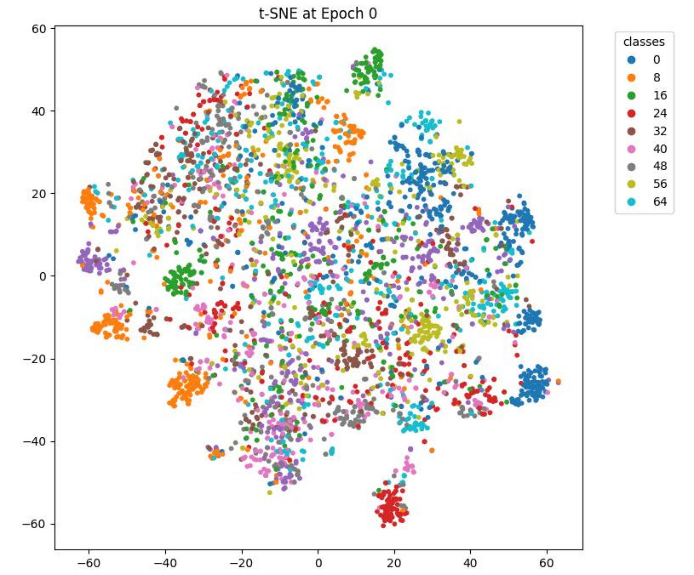
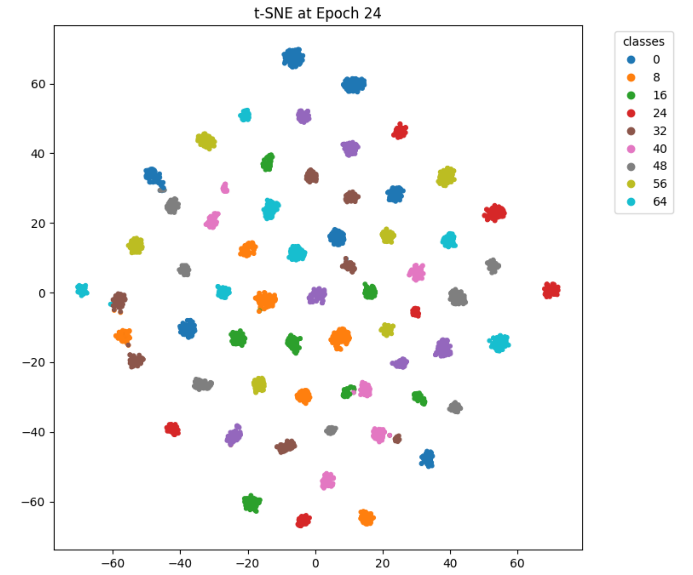
First, the models use official DINO pre-trained weights on ImageNet-1k as the backbone outperform those pre-trained on Mini-ImageNet (B: 80.05% vs. C: 69.46%, D: 66.75% vs. E: 51.72%). This suggest that pretraining on larger and more diverse data generally provides stronger ability.

Second, full model finetuning outperforms finetuning only the classifier. (B: 80.45% vs. D: 66.75%, C: 69.46% vs. E: 51.72%). Adapting the entire backbone to the target domain yields better performance. Nevertheless, finetuning only the classifier still offers some improvement of the downstream task.

(5%) Visualize the learned visual representation of **setting C** on the **train set** by implementing **t-SNE** (t-distributed Stochastic Neighbor Embedding) on the output of **the second last layer**.

a. Depict your visualization from both **the first and the last epochs**.(3%)

b. Briefly explain the results.(2%)

t-SNE the first epochs	t-SNE the last epochs
	

Answers(Briefly explain the results) :

For first epoch, the classes are not well separated in the t-SNE visualization. Although the network uses weights from SSL pretrained backbone on Mini-ImageNet, network's weights are still unfinetuned on Office-Home dataset. As a result, The features extracted from the second-last layer do not yet encode meaningful distinctions between most classes. {pints from different classes appear largely mixed in the embedding space.

By the last epoch, after the model has been finetuned for the Office-Home dataset for 25 epochs, the second-last layer has learned to produce highly discriminative representations. Points corresponding to the same class cluster together, indicating that the network has successfully captured class-specific semantic information. However, some classes clusters remain closer to each other, suggesting that certain classes are more visually similar or inherently harder to differentiate. Overall, the t-SNE visualization provides a qualitative illustration of how the network learns and organized semantic representations overtraining.

P 2.

1. **(4%)** Do an ablation study. You need to modify the U-Net model. During training, randomly select one skip connection between encoder and decoder and drop it, so the decoder at that layer will not receive the encoder features. Report and discuss/analyze the performance difference with the standard U-Net (model A).

Ans:

a. Explain you skip which layer:

I choose to drop the deepest skip connection in the U-Net, which corresponds to the encoder feature map at index 4. This connection carrying the highest-level semantic information. Such features are crucial for classifying large regions in satellite imagery. By removing this connection, I expected the decoder to lose some important semantic context, resulting in weaker segmentation accuracy compared to the full U-Net.

b. Result for standard U-Net:

Mean IoU: 0.697

c. Result for your skip U-Net:

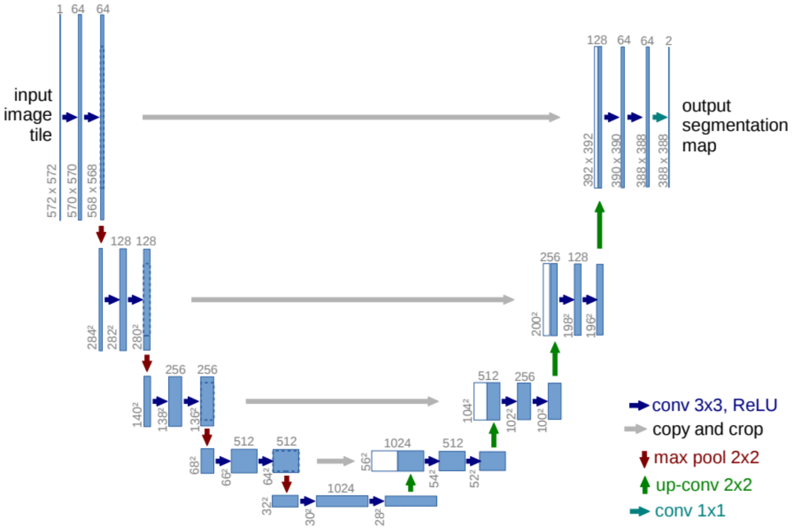
Mean IoU: 0.680

d. Explain the result:

Surprisingly, in my experiments, removing the deepest skip connection in U-Net did not cause a huge performance drop, though it consistently underperformed compared to the standard one. By removing the deepest skip connection, I initially expected the loss of high-level semantic features will make class distinctions much harder, leading to a strong negative impact on mIoU. However, the performance drop was modest. This may suggests that for satellite image segmentation, the certain level of semantic features many not be as critical as anticipated, since the network could still leverage other levels features for reasonable segmentation. Still, dropping the skip connection in U-Net leads to some degree of performance drop.

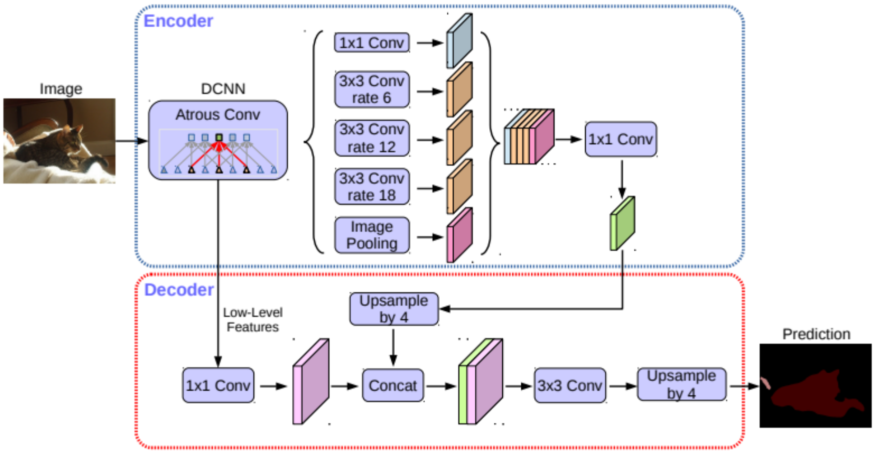
2. **(4%)** Draw the network architecture of the improved model (model B) and explain it differs from your UNet model (model A).

U-Net is a fully convolutional encoder-decoder architecture designed for semantic segmentation. The encoder progressively down samples the input image to extract hierarchical features. On the other hand, the decoder upsamples the feature maps to reconstruct the output segmentation mask. U-Net uses multiple skip connections between corresponding encoder and decoder stages, which allow U-Net to preserve fine-grained spatial details and small structures.



U-net architecture (Ronneberger et al., 2015)

The model B I used is DeepLabV3+. The decoder utilizes Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale context through atrous convolutions with varying dilation rates. This enables the model to effectively handle objects of different scales. The encoder extracts features from the input, while the decoder refines the segmentation result, especially along object boundaries. Compared to U-Net, DeepLabV3+ relies less on multiple skip connections and more on multi-scale feature aggregation.

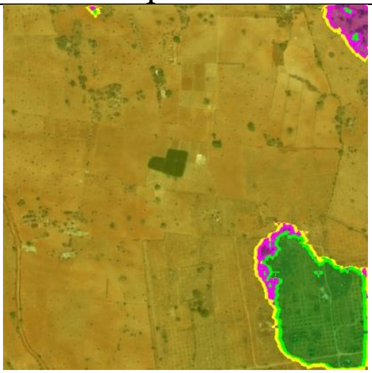
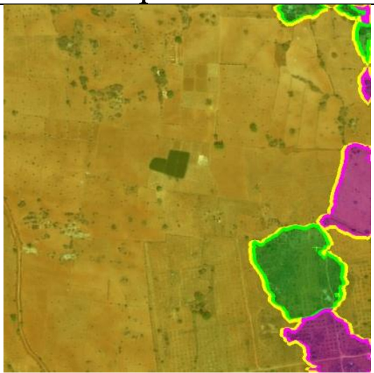
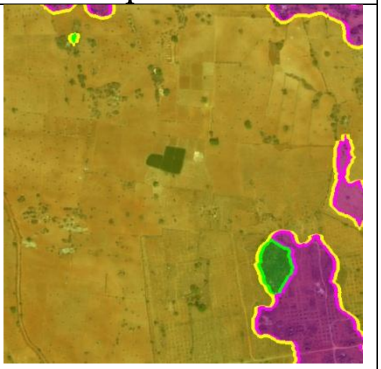
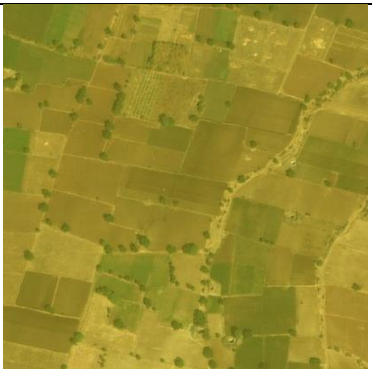
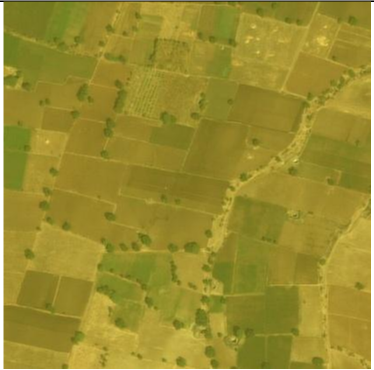
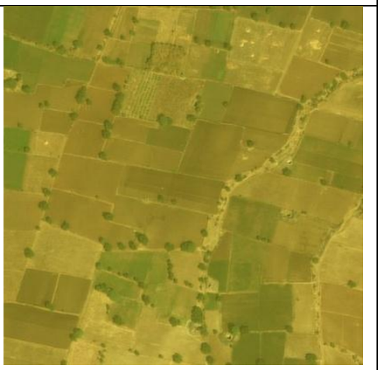
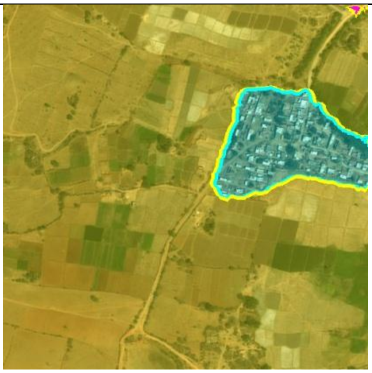
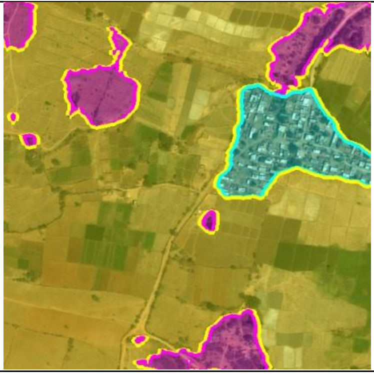
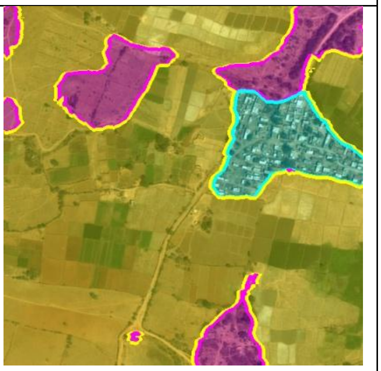


DeepLabV3+ Architecture (Chen et al., 2018)

3. **(2%)** Report mIoUs of two models on the validation set.

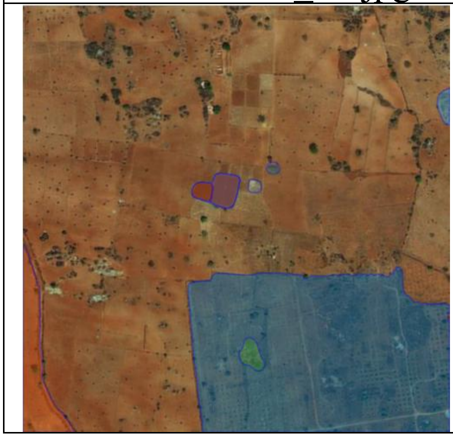
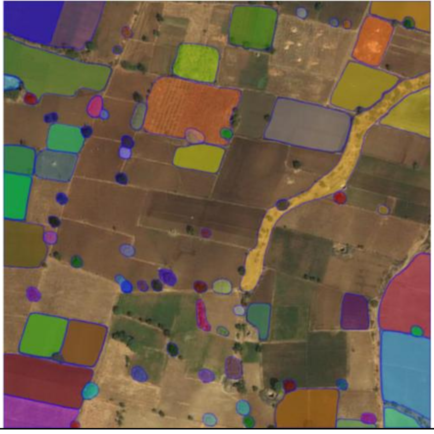
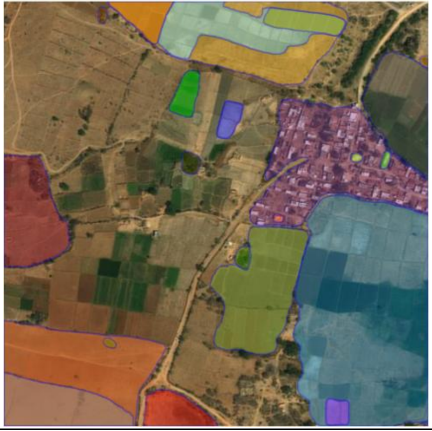
A	B
0.697	0.761

4. **(3%)** Show the predicted segmentation mask of "validation/0013_sat.jpg", "validation/0065_sat.jpg", "validation/0104_sat.jpg" during the early, middle, and the final stage during the training process of the improved model.

	Epoch 1	Epoch 47	Epoch 118
0018_sat.jpg			
0065_sat.jpg			
0109_sat.jpg			

5. (7%) Use segment anything model (SAM) to segment three of the images in the validation dataset, report the result images and the method you use.

Your Answer:

validation/0018_sat.jpg	validation/0065_sat.jpg	validation/0109_sat.jpg
		

Segment Anything Model (SAM) is introduced in the paper “Segment Anything” by Meta AI Research, FAIR in 2023. SAM is a foundation model for image segmentation that can generate object masks from prompts such as points, boxes, or texts. It is built on Vision Transformer, separates image and prompt encoding. Trained on over 1B masks on 11M images, makes SAM highly effective for zero-shot segmentation.

In this task, I use the online demo website by Meta AI (https://segment-anything.com/demo) to generate the results of segmentations of 3 images in validation set.