

Deep Learning for Computer Vision – HW #1

Problem 1 : Self-supervised pre-training for image classification

I. Implementation details of SSL method for pre-training ResNet50 backbone.

We pre-train a ResNet-50 backbone with the DINO self-supervised method. Input views follow DINO-style multi-crop: two “global” crops (scale 0.14–1.0) and multiple “local” crops (0.05–0.14), each composed with standard strong augmentations (RandomResizedCrop, horizontal flip, color jitter, random grayscale, Gaussian blur/solarization) and ImageNet mean/std normalization. Training runs for 300 epochs on unlabeled Mini-ImageNet with a batch size of 32 per GPU and 8 data-loader workers. Optimization uses SGD with learning rate 0.04 and weight decay 1e-4; the learning rate follows a 10-epoch linear warm-up then cosine decay (weight decay kept constant here). DINO’s teacher network is an exponential-moving-average of the student (teacher momentum initialized at 0.996 and scheduled upwards). The loss is the standard DINO cross-entropy between teacher and student outputs across views, with temperature sharpening and centering. BatchNorm is disabled in the DINO projection head.

II. Image classification on Office-Home dataset as the downstream task.

Setting	Pre-training (Mini-ImageNet)	Fine-tuning (Office-Home dataset)	Validation accuracy (Office-Home dataset)
A		Train full model (backbone+classifier)	18.72%
B	DINO on ImageNet-1k (TAs provided)	Train full model (backbone+classifier)	80.79%
C	w/o label (Your SSL pre-trained backbone)	Train full model (backbone+classifier)	53.45%
D	DINO on ImageNet-1k (TAs provided)	Fix the backbone. Train classifier only	80.79%
E	w/o label (Your SSL pre-trained backbone)	Fix the backbone. Train classifier only	34.48%

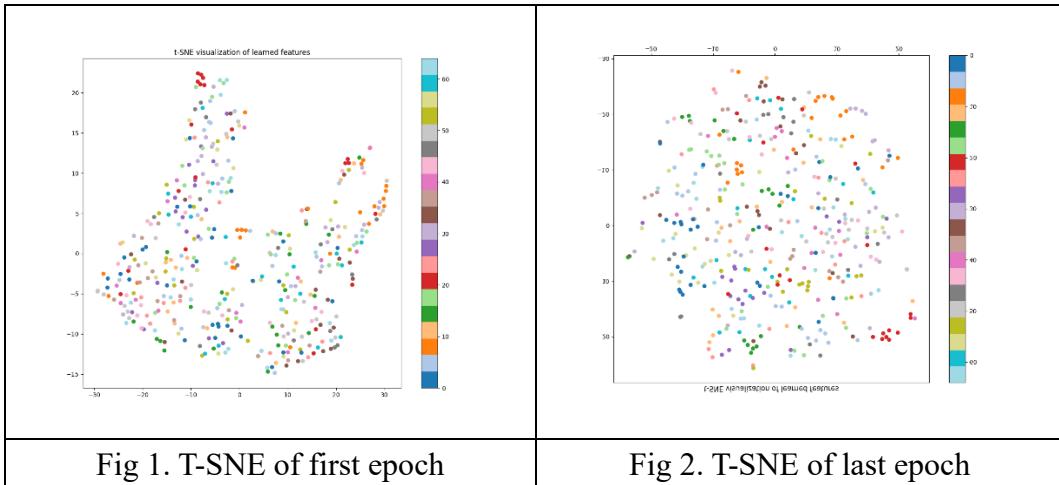
Comparing A (no pre-training) and C (our DINO-pretrained backbone), validation accuracy rises from 18.72% to 53.45% (+34.73 pp). This large gain indicates that DINO successfully learns transferable image features from unlabeled Mini-ImageNet, substantially improving downstream performance on Office-Home.

Contrasting B and D (both using the TA-provided DINO on ImageNet-1k), fine-tuning the full model (B) does not outperform freezing the backbone (D): both achieve 80.79%. This parity suggests the ImageNet-1k DINO backbone is already strong and linearly separable for this task; additional backbone updates offer little marginal benefit.

Finally, C vs E (our DINO backbone, full fine-tune vs frozen) shows 53.45% vs 34.48%. The drop when freezing (E) implies our self-supervised backbone has not fully captured features that are linearly separable for Office-Home; end-to-end fine-tuning is still necessary to adapt to the target domain.

These results (i) confirm the benefit of SSL pre-training (A→C), (ii) show that a strong, well-pretrained backbone can be frozen without hurting accuracy (B≈D), and (iii) indicate our own DINO backbone still needs end-to-end fine-tuning to adapt to Office-Home (C>E).

III. Learned visual representation of setting C model on train set.



At initialization, classes are intermixed, indicating limited transfer of separable structure from our DINO-pretrained backbone to Office-Home. After end-to-end fine-tuning, clusters become more compact and margins widen, showing effective

adaptation. However, persistent overlap aligns with the modest validation accuracy ($\sim 53\%$), implying features are not sufficiently linearly separable. This is consistent with a domain gap between Mini-ImageNet pre-training and Office-Home and suggests stronger pre-training or additional tuning/regularization may further improve separability.

Problem 2 : Semantic segmentation

I. Ablation study of U-Net (baseline model).

We modified U-Net to drop one encoder–decoder skip. For the reported run, I removed the shallowest skip (idx 0) while keeping the training protocol, data, and hyper-parameters identical to model A. Baseline U-Net (A) reached **10.22%** pixel accuracy; the ablated model achieved **9.86%** (-0.36 pp). In principle, shallow skips carry high-resolution edges and textures that help the decoder localize boundaries; removing them should hurt segmentation slightly. The small drop we observe is consistent with that intuition. However, both models plateau near $\sim 10\%$ accuracy and exhibit poor convergence, so the -0.36 pp gap is unlikely to be statistically meaningful—training noise likely masks the true effect of the skip. In other words, this ablation suggests the skip is beneficial, but the experiment is under-informative because the base model itself has not learned a useful representation.

II. Network architecture of the improved model.

Model B is a DeepLabV3-ResNet50: an encoder plus a lightweight context head. A ResNet-50 processes the image; in its later stages we use dilated convolutions so the final feature map stays at about 1/16 input resolution while the receptive field grows. On this low-resolution map, an ASPP head runs five parallel paths—a 1×1 conv, three 3×3 atrous convs at increasing rates, and an image-level pooling branch—then concatenates and projects them to class logits, which are finally upsampled once to the input size. In contrast, U-Net is an encoder–decoder. It downsamples through the encoder, records high-resolution “skip” features at each scale, then upsamples step by step; at every stage the decoder concatenates the corresponding skip to rebuild spatial detail before a final 1×1 classifier. Practically, DeepLabV3 emphasizes multi-scale context at low resolution and forgoes a heavy decoder (often with an auxiliary mid-level loss to stabilize training), yielding cheaper memory and computation but potentially softer boundaries. U-Net, by explicitly reinjecting early high-resolution features during a multi-stage upsampling path, tends to produce sharper edges and better tiny-object recovery at the cost of a larger decoder. In short, DeepLabV3 “sees” more scene-level context; U-Net “recovers” more fine detail.

```

Image
↓
ResNet-50 encoder (conv1 → layer1 → layer2 → layer3 → layer4)
└ uses dilation in later layers → large receptive field at ~1/16 size
↓
ASPP head (multi-scale context)
├─ 1x1 conv
├─ 3x3 atrous r=small
├─ 3x3 atrous r=medium
└─ 3x3 atrous r=large
└ image-level pooling
  → concatenate → 1x1 projection → logits (C classes)
↓
Upsample logits back to input size

```

Fig 3. Network architecture of DeepLabV3_ResNet101

```

Image
↓
Encoder (downsample 4x: convs + pooling)
  ↘ skip1  ↘ skip2  ↘ skip3  ↘ skip4
Decoder (upsample 4x)
  ↑ concat with skip4 → convs
  ↑ concat with skip3 → convs
  ↑ concat with skip2 → convs
  ↑ concat with skip1 → convs
  ↓
1x1 conv → logits (C classes)

```

Fig 4. Network architecture of UNet

III. mIoUs of two models on the validation set.

mIoU of U-Net	mIoU of ResNet101+DeepLabV3
10.22%	73.36%

IV. Predicted segmentation mask of images in validation set generated by three stages of training process of improved model.

	0018_sat.jpg	0065_sat.jpg	0109_sat.jpg
model at 1 st epoch			
model at 50 th epoch			
model at 100 th epoch			

V. Predicted segmentation mask of images in validation set generated by segment anything model (SAM).

0018_sat.jpg	0065_sat.jpg	0109_sat.jpg

We pick the same satellite images in the previous question for easily comparing between the predicted segmentation masks of improved model and those of SAM. Besides, we use the SAM demo on the website (<https://segment-anything.com/demo>) and press the “Everything” bottom. Note that SAM uses an MAE pre-trained ViT-H image encoder and was trained on SA-1B.