Splicing ViT Features for Semantic Appearance Transfer
Supplementary Material (SM)

CVPR 2022 Submission ID:2561

We provide implementation details for our architecture and training regime.

1 Generator Network Architecture

We base our generator G_θ network on a `U-Net` architecture [4], with a 5-layer encoder and a symmetrical decoder. All layers comprise 3×3 Convolutions, followed by `BatchNorm`, and `LeakyReLU` activation. The encoder's channels dimensions are $[3 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 128]$ (the decoder follows a reversed order). In each level of the encoder, we add an additional 1×1 Convolution layer and concatenate the output features to the corresponding level of the decoder. Lastly, we add a 1×1 Convolution layer followed by `Sigmoid` activation to get the final RGB output.

2 ViT Feature Extractor Architecture

As described in Sec. 3, we leverage a pre-trained ViT model (DINO-ViT [1]) trained in a self-supervised manner as a feature extractor. We use the 12 layer pretrained model in the 8×8 patches configuration (`ViT-B/8`), downloaded from the official implementation at GitHub.

3 Training Details

We implement our framework in PyTorch [3] (code will be made available). We optimize our full objective (Eq. 4, Sec. 3.3), with relative weights: $\alpha = 0.1$, $\beta = 0.1$. We use the Adam optimizer [2] with a constant learning rate of $\lambda = 2 \cdot 10^{-3}$. Each batch contains $\{\tilde{I}_s, \tilde{I}_t\}$, the augmented views of the source structure image and the target appearance image respectively. Every 75 iterations, we add $\{I_s, I_t\}$ to the batch (i.e., do not apply augmentations). The resulting images $\{G(\tilde{I}_s), G(\tilde{I}_t)\}$ and $\tilde{I}_t$ are then resized down to 224[pix] (maintaining aspect ratio) using bicubic interpolation, before extracting DINO-ViT features for estimating the losses. Training on an input image pair of size 512×512 takes $\sim$20 minutes to train on a single GPU (Nvidia RTX 6000) for a total of 2000 iterations.

4 Data Augmentations (§3.3)

We apply data augmentations to the input image pair $\{I_s, I_t\}$ to create multiple *internal examples* $\{I_s^i, I_t^i\}_{i=1}^N$. Specifically, at each training step, we apply the following augmentations:

Augmentations to the source structure image I_s:

- random cropping: we uniformly sample a NxN crop such that N is between 95% - 100% of the height of I_s.
- random horizontal-flipping, applied in probability p=0.5.
- random color jittering: in probability p=0.5 we jitter the brightness, contrast, saturation and hue of the image.
- random Gaussian blurring: in probability p=0.5 we apply a Gaussian blurring 3x3 filter (σ is uniformly sampled between 0.1-2.0).

Augmentations to the target appearance image I_t:

- random cropping: we uniformly sample a NxN crop such that N is between 95% - 100% of the height of I_t.
- random horizontal-flipping, applied in probability p=0.5.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021. 1

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015. 1