Deep ViT Features as Dense Visual Descriptors: Supplementary Material

CVPR 2022 Submission ID: 1653

1 Implementation Details

In all our applications (unless specified otherwise) we use `dino_vits8` model from the official DINO Github repository [1, 2], with `stride=4` (see Sec. 2).

Co-segmentation parameters (§5.1). We extracted the keys from the last layer (11^{th} starting from 0), concatenated all the heads to receive a descriptor for each patch. We used the FAISS library [3, 4] for computing k-means. In the co-segmentation experiments our elbow coefficient is 0.975, saliency threshold is 0.065, majority percentage is 75%. We resize the input images to have the shorter edge of size 320[pix].

Global Outlier Filtering One of the challenges in the Internet300 [6] dataset is handling images that do not contain the common object at all. We term these *global outlier images*, and filter them automatically before applying the co-segmentation pipeline using the descriptor of the [CLS] token. We compute the average of all the [CLS] descriptors on the entire set of images, and reject images that have cosine similarity lower than 0.7 from the average descriptor.

Co-segmentation and Part Co-segmentation Ablations. Supervised ViT weights are from `timm` repository [7]. We used keys from the 9^{th} layer because they exhibited better part separation than the 11^{th} layer, giving supervised ViT a fair chance. We used `vit_small_patch16_224` with `stride=4`. In saliency baseline we used a saliency threshold of 0.04. DINO and supervised `ResNet-50` weights are from DINO and `timm` repositories respectively. In PASCAL-Co ablations for `ResNet-50` we replace the last three strides with dilation to receive high resolution feature maps, as if features were computed at `stride=4` of the input resolution. All models are trained on ImageNet data.

Part Co-segmentation parameters (§5.2). We use the same parameters as co-segmentation application. For CelebA [5], we choose the salient segments based if there average distance from the center of the image was under 0.2, and if their compactness was higher than 0.5.

Part Co-Segmentation of Image Pairs. (Fig. 9) We present our part co-segmentation results in an extreme setting – operating on two images under significant variations of quantity, background clutter, pose, scale and appearance. We use flip and random-crop (95% of the original images) augmentations to compensate for the low number of images. We also introduce three clustering stages instead of two – one for fg/bg separation, one for removing uncommon foreground objects and one for part segmentation. This extreme setting is sensitive to hyper-parameters, but we found using 40 random-crop augmentations, and elbow coefficient of 0.94 works well for most cases.

Correspondence parameters (§5.3). For compatibility with NBB we resized the images to 224×224. We use saliency threshold of 0.05. We use log-binning with 2 hierarchies (17 bins, like shown in Fig. 7).

t-SNE. (Fig. 4) We used the same configurations as mentioned previously, besides these modifications: we used Layer 11 in supervised ViT and `stride=8` in both models.

Architecture	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
ViT-S/16	61.8	60.2	63.4
ViT-B/16	62.3	60.7	63.9
ViT-S/8	69.9	66.6	73.1
ViT-B/8	71.4	67.9	74.9
Ours	**72.2**	**67.9**	**76.5**

Table 1: DAVIS 2017 Video Object Segmentation.

PCA. (Fig. 3) We used `dino_vits16` and `vit_small_patch16_224` models with `stride=8`. We resized the input images to size 224×224.

2 Resolution Increase (§4.1)

We use `timm` repository [7] for ViT architecture and supervised weights, and [1] for DINO-ViT weights. We increase the resolution of ViT features maps by altering the phase of patch preparation. Instead of taking non-overlapping patches we take *overlapping patches*. In practice, the separation to patches and linear embedding is done by passing the image through a single convolution layer, with stride that equals the patch size and number of out channels as the embedding dimension. We alter the *stride* of this convolution layer to achieve overlapping patches. For example, using `stride=8` for a ViT trained with patch size 16 will increase the ViT feature's resolution times two. We assume the input size $\{H_{in}, W_{in}\}$ is divided by the patch size without remainder. If that is not the case, we remove the remainder pixels from the image. The output size is given by:

$$H_{\text{out}} = \frac{H_{\text{in}} - \text{patch\_size}}{\text{stride}} + 1$$
$$W_{\text{out}} = \frac{W_{\text{in}} - \text{patch\_size}}{\text{stride}} + 1$$

DAVIS Label Propagation. We empirically show the usefulness of test-time resolution increase by applying it to one of the applications shown in [1] - using pre-trained DINO features for DAVIS label propagation. We used a `dino_vits8` model with `stride=4`. Table 1 exhibits a significant improvement in results when using our alteration, and exhibit results even better than `dino_vitb8`.

Spair71k Keypoint Matching. In Tab. 2 we ablate our keypoint matching method with and without resolution increase. Evidently, increasing resolution enables higher spatial granularity which improves the performance of the method.

category	NBB	CATs	Stride 4	Stride 8
aeroplane	0.44	0.57	**0.69**	0.64
bicycle	0.28	0.48	**0.50**	0.49
bird	0.67	**0.89**	0.82	0.78
boat	0.12	0.39	**0.47**	0.43
bottle	0.17	**0.44**	0.37	0.33
bus	0.20	**0.63**	0.42	0.36
car	0.28	**0.60**	0.53	0.52
cat	0.30	0.65	**0.66**	0.62
chair	0.20	0.34	**0.45**	0.39
cow	0.29	0.73	**0.75**	0.63
dog	0.37	**0.65**	**0.65**	0.63
horse	0.13	**0.60**	0.46	0.38
motorbike	0.51	**0.80**	0.69	0.68
person	0.14	**0.66**	0.48	0.38
pottedplant	0.15	**0.48**	0.44	0.44
sheep	0.11	**0.70**	0.65	0.62
train	0.23	**0.83**	0.54	0.45
tvmonitor	0.26	**0.62**	0.59	0.55
all	0.27	**0.61**	0.56	0.52

Table 2: Spair71k keypoint matching with different strides

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021. 1, 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers: Github repository. `https://github.com/facebookresearch/dino`, 2021. 1

[3] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 1

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus: Github repository. `https://github.com/facebookresearch/faiss`, 2017. 1

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *ICCV*, 2015. 1

[6] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. *CVPR*, 2013. 1

[7] Ross Wightman. Pytorch image models. *GitHub repository*, 2019. 1, 2