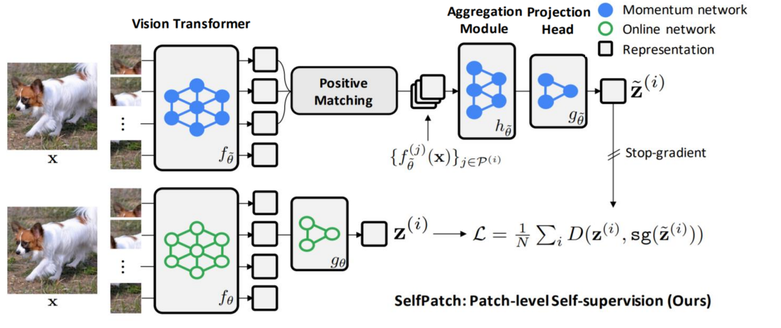
Patch-level Representation Learning for Self-supervised Vision Transformers

Sukmin Yun, Hankook Lee, Jaehyung Kim, Jinwoo Shin
Korea Advanced Institute of Science and Technology (KAIST)

CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA

TR; DR: We propose patch-level self-supervision for learning better patch-level representations

Summary



Motivation. Can we improve the quality of patch-level representations of ViTs (architectural characteristic) without human-annotated supervision?

- It can be beneficial to various downstream tasks of a dense prediction type

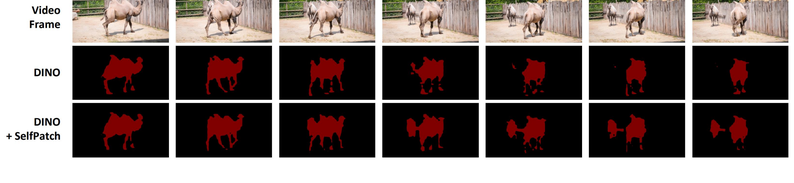
Contribution. We propose patch-level self-supervision and highlight its importance during pre-training ViTs in a self-supervised manner

- Our method can be incorporated into any image-level self-supervised ViT for learning both global and local information simultaneously

Better Patch-level Representations

Visualization of video object segmentation on the DAVIS 2017 benchmark

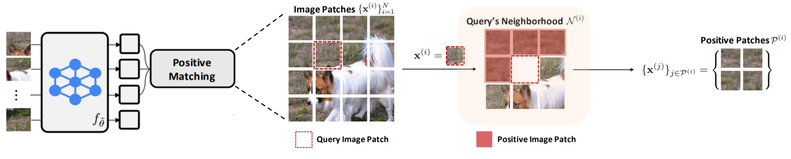
- Our method encourages the patch-level representations to learn semantic information of each object. (Incorporated with DINO)



Patch-level Self-supervision (SelfPatch)

Key Idea. Adjacent patches often share a common semantic context

- But, we do not know exactly which patches are positive
- We propose positive matching process and aggregation module

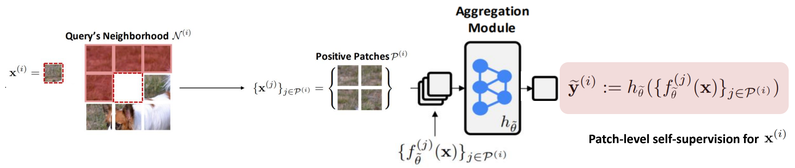


Positive Matching Process. Selecting candidates of semantically similar patches in the neighborhood

- We take *top-k* positive patches $\{\mathbf{x}^{(j)}\}_{j\in\mathcal{P}^{(i)}}$ based on the cosine similarity scores $s(i,j)$ as follow

$$s(i,j) = f_\theta^{(i)}(\mathbf{x})^\top f_\theta^{(j)}(\mathbf{x}) / ||f_\theta^{(i)}(\mathbf{x})||_2 ||f_\theta^{(j)}(\mathbf{x})||_2$$

  where $\mathcal{P}^{(i)}$ is a set of patch indices of *top-k* patches in *i*-th neighborhood $\mathcal{N}^{(i)}$
- Image patches $\{\mathbf{x}^{(i)}\}_{i=1}^N$ of an image $\mathbf{x}$
- $f_\theta^{(i)}(\mathbf{x})$ is the final representation of the *i*-th patch



Aggregation Module. Summarizing selected positive patches via an attention-based module to construct patch-level self-supervision $\mathbf{y}^{(i)}$

- Some of them might still be noisy (*e.g.*, not positive) → Denoising effect!
- [CLS] token in aggregation module attends the selected positive patches

Experimental Results

Method	Backbone	COCO Detection AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	COCO Segmentation AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	ADE20K Segmentation mIoU	aAcc	mAcc	DAVIS Segmentation $(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
MoCo-v2	ResNet50	38.9	59.2	42.4	35.5	56.2	37.8	35.8	77.6	45.1	55.5	56.0	55.0
SwAV	ResNet50	38.5	60.4	41.4	35.4	57.0	37.7	35.4	77.5	44.9	57.4	57.6	57.3
DenseCL	ResNet50	40.3	59.9	44.3	36.4	57.0	39.2	37.2	78.5	47.1	50.7	52.6	48.9
ReSim	ResNet50	40.3	60.6	44.2	36.4	57.5	38.9	36.6	78.4	46.4	49.3	51.2	47.3
DetCo	ResNet50	40.1	61.0	43.9	36.4	58.0	38.9	37.3	78.4	46.7	56.7	57.0	56.4
MoCo-v3	ViT-S/16	39.8	62.6	43.1	37.1	59.6	39.2	35.3	78.9	45.9	53.5	51.2	55.9
MoBY	ViT-S/16	41.1	63.7	44.8	37.6	60.3	39.8	39.5	79.9	50.5	54.7	52.0	57.3
DINO	ViT-S/16	40.8	63.4	44.2	37.3	59.9	39.5	38.3	79.0	49.4	60.7	59.1	62.4
+ SelfPatch (ours)	ViT-S/16	**42.1**	**64.9**	**46.1**	**38.5**	**61.3**	**40.8**	**41.2**	**80.7**	**52.1**	**62.7**	**60.7**	**64.7**

Our method significantly improves DINO in various detection (COCO object detection) and segmentation (COCO instance, ADE20K semantic, DAVIS 2017 video object segmentation) tasks on ImageNet-1k pre-training

- It also consistently outperforms the SOTA CNN & ViT-based baselines

Ablation Study

All models are pre-trained on COCO and evaluated on DAVIS 2017 benchmark

	Neighbors $\mathcal{N}^{(i)}$	Matching	Agg	$(\mathcal{J}\&\mathcal{F})_m$
(a)	-	-	-	55.1
(b)	3×3	$k=4$	✓	**57.0**
	5×5	$k=4$	✓	56.5
	All patches	$k=4$	✓	47.3
(c)	3×3	$k=1$	✓	56.3
	3×3	$k=2$	✓	56.4
	3×3	$k=4$	✓	**57.0**
	3×3	$k=8$	✓	56.5
(d)	3×3	$k=4$	-	51.4
	3×3	$k=4$	✓	**57.0**

Method	Backbone	Negative	$(\mathcal{J}\&\mathcal{F})_m$
MoBY	ViT-Ti/16	-	54.1
+ SelfPatch (ours)	ViT-Ti/16	-	58.4
+ SelfPatch (ours)	ViT-Ti/16	✓	**58.9**
MoBY	Swin-T	-	50.8
+ SelfPatch (ours)	Swin-T	✓	**56.4**
DINO	ViT-Ti/16	-	55.1
+ SelfPatch (ours)	ViT-Ti/16	-	**57.0**
DINO	ViT-Ti/8	-	61.6
+ SelfPatch (ours)	ViT-Ti/8	-	**65.8**

Component Analysis (Left). We validate the individual effects of components: (b) Neighboring Patches (3×3, 5×5, all patches), (c) Positive Matching ($k\in\{1,2,4,8\}$) and (d) Aggregation Module (avg. pooling) on ViT-Ti/16

Compatibility Analysis (Right). We validate the compatibility of our method with (a) Image-level Self-supervision (MoBY), (b) Transformer-based Architecture (Swin Transformer), and (c) Patch-size (8×8)