Learning Co-segmentation by Segment Swapping for Retrieval and Discovery

Xi Shen[1], Alexei A. Efros[2], Armand Joulin[3], and Mathieu Aubry[4]

[1,4]LIGM (UMR 8049), École des Ponts ParisTech
[2]University of California, Berkeley
[3]Facebook AI Research

Abstract

The goal of this work is to efficiently identify visually similar patterns from a pair of images, e.g. identifying an artwork detail copied between an engraving and an oil painting, or matching a night-time photograph with its day-time counterpart. Lack of training data is a key challenge for this co-segmentation task. We present a simple yet surprisingly effective approach to overcome this difficulty: we generate synthetic training pairs by selecting object segments in an image and copy-pasting them into another image. We then learn to predict the repeated object masks. We find that it is crucial to predict the correspondences as an auxiliary task and to use Poisson blending and style transfer on the training pairs to generalize on real data. We analyse results with two deep architectures relevant to our joint image analysis task: a transformer-based [64] architecture and Sparse Nc-Net [45], a recent network designed to predict coarse correspondences using 4D convolutions. We show our approach provides clear improvements for artwork details retrieval on the Brueghel dataset [1, 53] and achieves competitive performance on two place recognition benchmarks, Tokyo247 [58] and Pitts30K [59]. We then demonstrate the potential of our approach by performing object discovery on the Internet object discovery dataset [48] and the Brueghel dataset [1, 53]. Our code and data are available at http://imagine.enpc.fr/~shenx/SegSwap/.

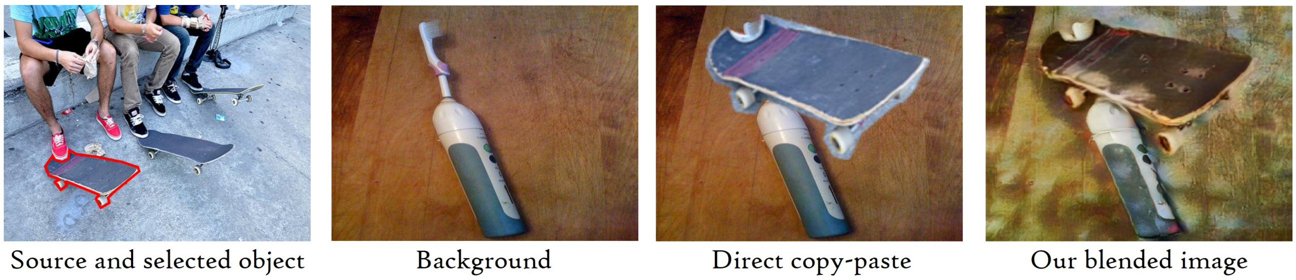
1. Introduction

Identifying repeated patterns lies at the very heart of the computer vision problem, and is a key component of Intelligence itself. Yet, in practice, our best methods for performing such a fundamental task often leave a lot to be desired. E.g. while we now have good methods for discovering *exact* pattern matches (used extensively to find copyright infringements), as well as approximate matches of *salient* objects (see object discovery and co-segmentation approaches in Section 2), detecting visually similar details within a larger visual context remains surprisingly difficult.

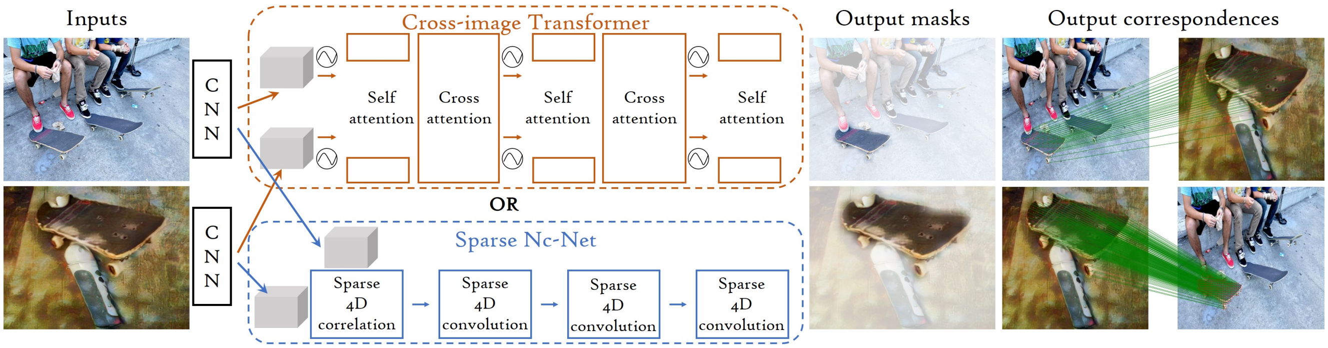
Spotting the repetition of visual detail has several applications. Identifying copied details in artworks allows art historians to discover influences, find provenance, and establish authorship [53]. Matching repeated details can boost performances in visual localisation for place recognition [19]. Reliable pair-wise image co-segmentation and correspondence identification could also enable object discovery in image collections [9]. However, identifying repeated content in image pairs remains challenging, especially in the cases where images appear very different from each other. Moreover, there is no available generic training dataset for this task.

In this paper, we show it is possible to learn to detect repeated visual patterns – jointly predicting co-segmentation and correspondences – without any human-labelled correspondences. Instead, we generate synthetic correspondence pairs via automatic data augmentation. More precisely, we use a “segment swapping” approach, where we blend object segments in a random background using Poisson blending and apply style transfer to the resulting image to obtain challenging training image pairs (Figure 1a). We compare using as image segments either COCO [36] instance segmentation or unsupervised segments, which produced slightly lower but comparable results. On the generated image pairs, we have access to the ground-truth matchability masks as well as the correspondences which we use as supervisions for training a network (Figure 1b). Surprisingly, we find that models trained on such a dataset generalize well to real data.

We experimented with two network architectures which we adapt to predict co-segmentation and correspondences in image pairs: the recent Sparse Nc-Net [45] architecture, designed for predicting image coarse correspondences, and an architecture based on Transformers [64] which we refer to as cross-image transformer.



(a) Data generation by "segment swapping". Instead of directly pasting an object from a source image on a background (3rd column), we use Poisson blending [41] and add style transfer [22] to the result (4th column).



(b) We train our cross-image transformer or Sparse Nc-Net [45] on the generated pairs. Both networks jointly predict masks and correspondences.

Figure 1. Learning co-segmentation by "segment swapping". We generate training data with "segment swapping" (Figure 1a) and learn co-segmentation either with our cross-image transformer or Sparse Nc-Net [45] (Figure 1b).

We analyze the effectiveness of our data generation process, architectures and training strategy on two types of tasks. First, we perform retrieval tasks using the predicted pair-wise co-segementation masks and correspondences. We show clear performance improvement for artwork details retrieval on the Brueghel [53] dataset and results comparable to state of the art for visual localization on two challenging place recognition benchmarks, Tokyo247 [58] and Pitts30K [59]. This last result is especially impressive, since these benchmarks are very competitive, and many dedicated methods leveraging geo-referenced images or real correspondence for supervision have been proposed. On the contrary, our approach is generic and relies solely on our synthetic "segment swapping" training.

We further make use of the predicted masks and correspondences to build a candidate correspondence graph and perform discovery with spectral clustering [33, 40]. We demonstrate results on par with state-of-the-art on the standard co-segmentation Internet [48] dataset and show qualitative results on the challenging Brueghel [53] dataset.

Our code and data are available at http://imagine.enpc.fr/~shenx/SegSwap/.

2. Related work

Learning correspondences between different images. SIFT-Flow [37] was an early method that aligns visually distinct scenes by incorporating visual features, such as SIFT, into optical flow-style approaches. More recently, many deep learning based approaches have been developed to predict correspondences from correlations of input features [39,44,52,60–62]. Of particular interest, architectures based on attention mechanisms and Transformers [64] have been introduced to predict image correspondences. SuperGlue [49] is an attention-based graph neural network for key-point matching. Closer to this work, COTR [26] is an sequence-to-sequence transformer architecture that takes an image and 2D coordinates of a query points as inputs to predict correspondences. Finally, LoFTR [55] adopts a coarse-to-fine approach to matching with a transformer encoder. As opposed to our work, these transformer-based methods are trained on a large dataset with ground-truth poses and depth while we only train on a synthetic dataset. Additionally, our model is only composed of an encoder and outputs a mask of the common regions along with the correspondences.

Learning correspondences without annotated data. There is a large body of work that use synthetic images [13] or images with synthetic deformations [39, 44, 51, 60] to learn correspondences without real annotated training data. However, these approaches do not try to identify the matchable regions, which is essential to discover visual details. Some other approaches train directly on real images using proxy signals for correspondences, such as photometric or cycle consistency [23, 52, 61, 62, 68, 70]. Again, they focus on the quality of the correspondences and are not designed to predict matchable regions in vastly different images. On the contrary, the core of our approach is to discover these similar regions. This makes our approach particularly suited for retrieval tasks.

Our approach is also inspired by related data augmentation techniques, specifically, the CopyPaste augmentation used by Ghiasi *et al*. [16] for instance segmentation and the stylised-ImageNet augmentation used in Geirhos *et al*. [15] to increase shape bias in neural networks.

Object discovery and co-segmentation. There is a wide variety of approaches aiming at discovering objects and their location from unlabelled images. Many methods [10, 56, 66, 67] use bounding box proposals and formulate the object discovery as an optimization problem. This relies on the quality of proposals which are typically not adapted for non-photorealistic data, such as artworks. Other approaches [9, 21, 34, 35, 47, 48, 57, 65, 69] focus on predicting masks of salient objects directly. Some [34, 35, 69] require foreground masks for training, while others [9, 21, 27, 28, 34, 65] are designed to segment common repeated objects in a image collection. These approaches make strong assumptions about the frequency of appearance of an object, while, in many practical scenarios, repeated objects are rare and discovering them is about seeking a needle in a haystack [1, 53].

Our approach is related to [48, 57], as we both leverage dense correspondences to discover objects. As opposed to our work, Taniai *et al*. [57] focuses on a single pair of images while we also show results over an entire collection of images. Rubinstein *et al*. [48] makes the assumption that the common object is also the most salient in the image. This works well with images form internet queries but does not apply to artworks where the common object can be a detail in a richer scene.

3. Co-segmentation by segment swapping

We show an overview of our approach in Figure 1. In Section 3.1, we introduce our "segment swapping" data generation process (Figure 1a). We then present in Section 3.2 the two architectures we use (Figure 1b). We discuss our loss and training strategy in Section 3.3.

3.1. Training data generation by segment swapping

Training pairs generation We generate training pairs using images from the COCO dataset [36]. We first sample a source image, from which we extract one or two segments (as explained below). We then build the target image by applying geometric transformations to the objects and blending them into a random background image using Poisson blending [41]. The geometric transformations include rotation, translation, scaling, and thin-plate spline (TPS). A style augmentation is then performed on both the source and target images using an AdaIN [22] model trained on the Brueghel dataset [1, 53]. An example of training pair can be seen in Figure 1a and we provide more examples of training samples in the supplementary material [2].

Segments definition The simplest way to define segments for our data generation process is to use the instance annotations from COCO [36]. We compare this approach to a completely unsupervised segment extraction, which we defined using the following strategy: (1) given an image, we compute bounding box proposals via the Selective Search [63]; (2) we compute a simple saliency for each feature in the bounding box, i.e. for each Moco [7] *Conv4* feature inside the box we use as saliency its average cosine distance with its neighbours and only keep boxes which average saliency is high enough; (3) the final mask for each box is obtained using GrabCut [46] initialized by the saliency map.

More details about the unsupervised procedure and example of unsupervised segments and the associated image pairs are available in the supplementary material [2].

3.2. Architectures

Our networks take as input a source image $\mathbf{I}^s$ and a target image $\mathbf{I}^t$, from which features maps $\mathbf{F}^s$ and $\mathbf{F}^t$ of spatial dimension $W \times H$ are extracted by a feature extraction backbone.

These feature maps are then processed either by our cross-image transformer or our sparse Nc-Net [45] architecture to predict both the masks of the repeated objects in the source and target images, $\mathbf{M}^s \in [0,1]^{W\times H}$ and $\mathbf{M}^t \in [0,1]^{W\times H}$ respectively, and the correspondences both from source to target $\mathbf{C}^{s\to t}$ and target to source $\mathbf{C}^{t\to s}$. Both $\mathbf{C}^{s\to t}$ and $\mathbf{C}^{t\to s}$ are represented as matrices of size $W \times H \times 2$. To simplify notation, we sometime use the masks as continuous 2D functions, which in practice is done by performing bilinear interpolation.

Cross-image transformer We built an architecture based on the classic transformer encoder [64] which alternates multi-headed attention and fully connected feed-forward networks (FFN) blocks.

Feat. + Methods	mAP	
	Retrieval	Det.(IoU > 0.3)
Brueghel [53] + cos [53]	75.5	75.3
Brueghel [53] + discovery [53]	76.6	76.4
MocoV2 [7] + cos [53]	79.0	78.7
MocoV2 [7] + discovery [53]	80.8	79.6
Ours + Unsupervised segments		
Transformer	83.3	79.8
Sparse-Ncnet	82.4	73.5
Ours + COCO segments [36]		
Transformer	**84.4**	**81.8**
Sparse-Ncnet	83.3	73.7

Table 1. Art detail retrieval and detection on Brueghel [1, 53]. For detection, we employ ArtMiner (Brueghel [53] + cos [53]) as a post-processing and reports results with IoU > 0.3 [53]

Method	Supervision	Tokyo 24/7 [58]			Pitts30k-test [59]		
		R@1	R@5	R@10	R@1	R@5	R@10
AP-GEM [19, 43]	Image location	40.3	55.6	65.4	75.3	89.3	92.5
DenseVLAD [19, 58]	Image location	59.4	67.3	72.1	77.7	88.3	91.6
NetVLAD [3, 19]	Image location	73.3	82.9	86.0	86.0	93.2	95.1
CRN [14, 30]	Image location	75.2	83.8	87.3	-	-	-
SARE [14, 38]	Image location	79.7	86.7	90.5	-	-	-
IBL [14]	Image location	**85.4**	**91.1**	**93.3**	-	-	-
Re-ranking Top-100 from NetVLAD [3, 19]							
Patch-NetVLAD [19]	Image location	81.9	85.7	87.9	88.6	94.5	95.8
Patch-NetVLAD [19] + RANSAC	Image location	86.0	88.6	**90.5**	**88.7**	94.5	95.9
SuperGlue [19, 49]*	Pose+Depth	**88.2**	**90.2**	90.2	**88.7**	**95.1**	**96.4**
Ours + Unsupervised segments							
Transformer	Segment swapping	76.5	82.9	85.4	83.5	92.9	95.3
Nc-Net	Segment swapping	83.2	87.0	87.6	85.6	94.1	95.5
Ours + COCO segments [36]							
Transformer	Segment swapping	80.0	86.0	87.9	84.7	93.5	95.6
Nc-Net	Segment swapping	85.4	88.3	89.2	86.8	94.4	95.8

* uses learnt keypoint detector Superpoint [12]

Table 2. Image-based localization on Tokyo 24/7 [58] and Pitts30k [59]. We follow Patch-NetVLAD [19] and re-rank the top-100 images ranked by NetVLAD [3] features.

The FFN blocks contain two layers with a ReLu non-linearity. Similar to [49], we use two types of attention layers: one is the standard self-attention (SA) layer, the other one is a cross attention (CA) layer where the attention is only computed between features from different images. We include the same 2D positional encoding as DeTR [5] on top of the feature map before SA. Our transformer alternates these two types of attention layers as shown in Figure 1b, with a total of five attention and FFN blocks. Each attention layer has 2 heads and the dimension of the features is 256. Our last layer is followed by a sigmoid and has three outputs, that we interpret as masks and correspondences for each image. We provide an ablation study of this architecture in the supplementary material [2].

Sparse Nc-Net Nc-Net [45] is designed to learn coarse correspondences under weak supervision. It takes as input the correlations between $\mathbf{F}^s$ and $\mathbf{F}^t$, seen as a 4D volume of affinities $\mathcal{A}_{input} \in \mathbb{R}^{W\times H\times W\times H}$, and processes them with 4D convolutions. The final 4D convolution predicts affinities $\mathcal{A}_{pred} \in \mathbb{R}^{W\times H\times W\times H}$, on which softmax functions are applied in dimensions corresponding to source and target giving $\mathcal{A}^s_{pred}(i,j,k,l) = \frac{\exp(\mathcal{A}_{pred}(i,j,k,l))}{\sum_{k,l}\exp(\mathcal{A}_{pred}(i,j,k,l))}$ and $\mathcal{A}^t_{pred}(i,j,k,l) = \frac{\exp(\mathcal{A}_{pred}(i,j,k,l))}{\sum_{i,j}\exp(\mathcal{A}_{pred}(i,j,k,l))}$. We use the maxima of these affinities as source and target masks, i.e., $\mathbf{M}^s(i,j) = \max_{k,l}\mathcal{A}^s_{pred}(i,j,k,l)$ and $\mathbf{M}^t(k,l) = \max_{i,j}\mathcal{A}^t_{pred}(i,j,k,l)$. Correspondences are obtained with soft-argmax:

$$\mathbf{C}^s(i,j) = \left(\sum_{k,l}\frac{k}{W}\mathcal{A}^s_{pred}(i,j,k,l), \sum_{k,l}\frac{l}{H}\mathcal{A}^s_{pred}(i,j,k,l)\right),$$
$$\mathbf{C}^t(k,l) = \left(\sum_{i,j}\frac{i}{W}\mathcal{A}^t_{pred}(i,j,k,l), \sum_{i,j}\frac{j}{H}\mathcal{A}^t_{pred}(i,j,k,l)\right) \tag{1}$$

Since 4D convolutions are computational heavy, we instead use sparse 4D convolutions with the same architecture as Sparse Nc-Net [45].

3.3. Loss and training

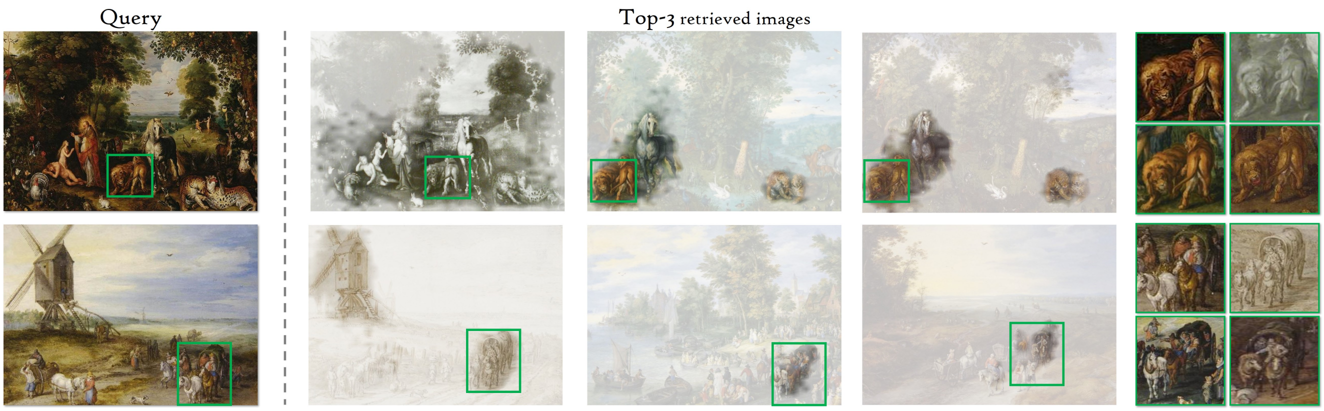
On our synthetic training data we have access to the ground truth masks $\mathbf{M}^s_{gt}$ and $\mathbf{M}^t_{gt}$ and ground truth correspondences $\mathbf{C}^{s\to t}_{gt}$ and $\mathbf{C}^{t\to s}_{gt}$ on the source and target images. Our loss is the sum of two symmetric terms for source and target, for simplicity we write only the source loss $\mathcal{L}^s$. It includes a cross-entropy (CE) loss on the predicted mask $\mathcal{L}_m$ and the transported mask $\mathcal{L}_{tm}$, as well as a regression loss $\mathcal{L}_{corr}$ on the correspondences:

$$\begin{aligned}\mathcal{L}^s =& \underbrace{CE(\mathbf{M}^s_{gt}, \mathbf{M}^s)}_{\mathcal{L}_m} + \underbrace{CE(\mathbf{M}^s_{gt}, \mathbf{M}^t(\mathbf{C}^{s\to t}))}_{\mathcal{L}_{tm}} \\ &+ \underbrace{\eta\frac{1}{\sum_{i,j}\mathbf{M}^s_{gt}(i,j)}\sum_{i,j}\mathbf{M}^s_{gt}(i,j)\|\mathbf{C}^{s\to t}(i,j) - \mathbf{C}^{s\to t}_{gt}(i,j)\|}_{\mathcal{L}_{corr}}\end{aligned} \tag{2}$$

where i and j correspond to the feature coordinates, η is a scalar hyper-parameter, and $CE(\mathbf{M}_{gt}, \mathbf{M}) = -\frac{1}{W\times H}\sum_{i,j}\mathbf{M}_{gt}(i,j)\log(\mathbf{M}(i,j)) + (1 - \mathbf{M}_{gt}(i,j))\log(1 - \mathbf{M}(i,j))$. Note that this loss is computed both for positive pairs (source and target pairs generated by segment swapping) and negative pairs (sampled from two different pairs, without repeated objects) for which $\mathbf{M}^s_{gt} = \mathbf{M}^t_{gt} = 0$ and by convention $\mathcal{L}_{corr} = 0$.

Implementation details We implement our approach using the Pytorch library. We use as backbone features the *conv4* features of a ResNet-50 [20] trained on ImageNet [11] with MOCO-v2 [7].

We freeze the backbone during the training, as learning backbone features leads to overfiting on the synthetic train-



(a) Retrieval results on Brueghel [1, 53]. green bounding-boxes are one-shot detection results.

(b) Retrieval results on Tokyo24/7 [58].

(c) Retrieval results on Pitts30k [59].

Figure 2. Visual results for retrieval on different datasets. For each query image (1st column), we show its 3 most similar images with the predicted masks as transparency. For Brueghel [1, 53], we also show the detection results.

ing set. For all the experiments, we optimise the loss defined in Equation 2 with $\eta = 8$ and use the Adam optimiser [31] with momentum terms $\beta_1 = 0.5$ and $\beta_2 = 0.999$. At each iteration, we sample 5 positive and 15 negative pairs. For the transformer architecture, after training 200k iterations with learning rate 2e-4, we train with hard negative pairs and learning rate 1e-5 for 5k iterations. Hard negatives are obtained by sampling a pool of $N_{pool} = 500$ images from different synthetic pairs, computing predicted masks for all the pairs of images in the pool, and keeping those with mask prediction higher than a threshold $\tau = 0.04$ in a hard negative pair pool for $K_{hard} = 1000$ iterations of training. For Sparse Nc-Net [45] training 200k iterations with learning rate 2e-4 without hard negative mining leads to the best performance. The entire trainings of the transformer and Sparse Nc-Net [45] take approximately 30 hours and 15 hours respectively on a single GPU Tesla-V100-16GB. An ablation study of the architectures and more training details are provided in the supplementary material [2].

4. Application to Image Retrieval

In this section, we show how our model can be used for retrieval tasks. We first explain how we use it to compute an image similarity score in Section 4.1. We then present experimental results in Section 4.2, including art detail retrieval on the Brueghel dataset [1, 53] and place recognition on Pitts30k [59] and Tokyo 24/7 [58]. More visual results and ablation studies are provided in the supplementary material [2].

4.1. Score between a pair of images

We propose the following score $\mathcal{S}$ to measure the similarity between a pair of images based on predicted correspondences and masks. $\mathcal{S}$ is the sum of weighted local features similarities, where our predicted correspondences are used to associate features and the weight $\mathbf{M}^s_{joint}$

Dataset		Losses			Cross-image Transformer	
Posson blending [41]	Style transfer [22]	$\mathcal{L}_m$	$\mathcal{L}_{tm}$	$\mathcal{L}_{corr}$	Brueghel [1,53] mAP	Tokyo 24/7 [58] R@1
✓	✓	✓	✓	✓	**84.4**	**80.0**
✗	✓	✓	✓	✓	75.1	60.0
✓	✗	✓	✓	✓	75.6	57.8
✓	✓	✗	✓	✓	80.9	67.8
✓	✓	✓	✗	✓	79.8	61.3
✓	✓	✓	✓	✗	8.5	13.3

Table 3. Ablation study. We report retrieval mAP on Brueghel [1,53] and R@1 on Tokyo 24/7 [58] with our cross-image transformer using COCO segments [36].

is the product of the source and transported target mask $\mathbf{M}^s_{joint}(i,j) = \mathbf{M}^t(\mathbf{C}^{s\to t}(i,j))\mathbf{M}^s(i,j)$:

$$\mathcal{S}(\mathbf{I}^s,\mathbf{I}^t) = \sum_{i,j} \underbrace{\mathbf{M}^s_{joint}(i,j)}_{\text{Mask}} \underbrace{cos(\mathbf{F}^s(i,j), \mathbf{F}^t(\mathbf{C}^{s\to t}(i,j)))}_{\text{Feat. similarity}} \tag{3}$$

Ablations in supplementary material [2] show that the mask term is the key part of this score.

4.2. Experiments

Qualitative results on our different datasets can be seen in Figure 2. The predicted masks, shown with transparency, are able to capture repeated regions even in challenging cases, such as large difference of scale, viewpoints, lightening conditions and depiction styles. More visual results are provided in the supplementary material [2].

Art detail retrieval We evaluate our approach on the Brueghel dataset [1,53] in Table 1. Our score allows us to directly retrieve images from a selected query detail. To further compare with the detection performance in ArtMiner [53], we crop a 320 × 320 patch around the predicted regions and use ArtMiner [53] as a post-processing to obtain the bounding box prediction. The correspondences are more accurate for the cross-image transformer which achieves much better results for detection. We also observe that, in this benchmark, the performances with unsupervised segments are close to the ones using COCO [36] instance annotations, which suggests that our approach does not depend on human annotations. Note that the best performance of ArtMiner is obtained with a discovery score which is expensive to compute and involves multi-scale feature matching and RANSAC. Our approach is thus simpler, faster and more effective.

Place recognition In Table 2 we compare our approach to state of the art for place recognition on the Pitts30k [59] and Tokyo 24/7 [58] datasets. The descriptions of the datasets are in the supplementary material [2].

We follow the standard evaluation protocol [4,14,17,50,58,59]. The query image is correctly localized if one of the top N retrieved database images is within d = 25 meters from the ground truth TUM coordinate of the query. The recall is then reported for N = 1, 5, 10. For Tokyo 24/7 we follow [14,58] and perform spatial non-maximal suppression on ranked database images before evaluation. To enable fast evaluation, we follow PatchVlad [19] and evaluate our score on the top-100 images given by NetVLAD [3]. Although our approach is not specifically designed for place recognition, it achieves performances comparable to Patch-NetVLAD [19] without RANSAC. Note that the competing approaches either employ specific supervisions or more complicated process such as RANSAC, while our approach is trained only with our synthetic segment swapping data. Note that on this task where retrieving discriminative repeated regions is sufficient and correspondence accuracy is not critical, the Nc-Net architecture preforms better. Similar to the Brueghel results, leveraging COCO [36] annotated segments leads to superior performance. Training with unsupervised segments still leads to competitive results using the NC-Net. However, it gives clearly worst results using the transformer architecture on Tokyo 24/7. We think this performance gap could be bridged using more advanced unsupervised segments.

Ablation study An ablation study of our approach using the cross-image transformer architecture and COCO [36] annotated segments is shown in Table 3 on the Brueghel [1,53] and Tokyo24/7 [58] datasets. We notice that: *(i)* Poisson blending [41] and style transfer [22] are both critical; *(ii)* the three terms of the loss are necessary for good performance. More analysis on the importance of learning correspondences, the similarity score and the architectures are provided in the supplementary material [2].

Method	Airplane		Car		Horse		Avg	
	$\mathcal{P}$	$\mathcal{J}$	$\mathcal{P}$	$\mathcal{J}$	$\mathcal{P}$	$\mathcal{J}$	$\mathcal{P}$	$\mathcal{J}$
DOCS [35]*	0.946	0.64	0.940	0.83	0.914	0.65	0.933	0.70
Sun et al. [54]	0.886	0.36	0.870	0.73	0.876	0.55	0.877	0.55
Joulin et al. [27]	0.493	0.15	0.587	0.37	0.638	0.30	0.572	0.27
Kim et al. [29]	0.802	0.08	0.689	0.00	0.751	0.06	0.754	0.05
Rubinstein et al. [48]	0.880	0.56	0.854	0.64	0.828	0.52	0.827	0.43
Chen et al. [8]	0.902	0.40	0.876	0.65	0.893	0.58	0.890	0.54
Quan et al. [42]	0.910	0.56	0.885	0.67	0.893	0.58	0.896	0.60
Hati et al. [18]	0.777	0.33	0.621	0.43	0.738	0.20	0.712	0.32
Chang et al. [6]	0.726	0.27	0.759	0.36	0.797	0.36	0.761	0.33
Lee et al. [32]	0.528	0.36	0.647	0.42	0.701	0.39	0.625	0.39
Jerripothula et al. [25]	0.905	0.61	0.880	0.71	0.883	0.61	0.889	0.64
Jerripothula et al. [24]	0.818	0.48	0.847	0.69	0.813	0.50	0.826	0.56
Hsu et al. [21]	0.936	0.66	0.914	0.79	0.876	0.59	0.909	0.68
Chen et al. [9]	**0.941**	0.65	**0.940**	**0.82**	**0.922**	**0.63**	**0.935**	**0.70**
Ours + Unsupervised segments								
transformer	**0.941**	0.66	0.919	0.79	0.887	0.57	0.916	0.67
Nc-Net	0.682	0.19	0.791	0.56	0.774	0.27	0.749	0.34
Ours + COCO segments [36]								
transformer	**0.941**	**0.67**	0.928	**0.82**	0.916	0.60	0.928	**0.70**
Nc-Net	0.655	0.23	0.857	0.61	0.873	0.43	0.795	0.42

* learned with strong supervision (i.e., manually annotated object masks)

Table 4. Co-segmentation on Internet [48]. We report pixel level precision $\mathcal{P}$ and Jaccard index $\mathcal{J}$

5. Application to Object Discovery and co-segmentation

In this section, we show how to use our predicted masks and correspondences for object discovery. We first introduce the correspondences graph on which we perform spectral analysis in Section 5.1. We then present experimental results on the Internet [48] dataset for co-segmentation and Brueghel [1, 53] for discovery in Section 5.2.

5.1. Correspondences graph and clustering

In the spirit of [33], we see object discovery as a graph clustering problem, where the vertices $\mathcal{V}$ of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ are correspondences between images and the weights of the edges encodes consistency between the correspondences. Let us consider a set of N images $(I_1, \ ..., \ I_N)$. For every pair of images our network predicts correspondences that we add to the set of vertices $\mathcal{V}$ if the associated mask value is higher than a threshold. Each vertex $v_i = (s_i, t_i, x_i^s, x_i^t, m_i)$ in the graph is thus associated to a predicted correspondence and defined by the indices s_i and t_i of the images it connects, the associated coordinates x_i^s and x_i^t and the predicted mask value m_i. We use the masks values and cycle consistency between the correspondences to define the weights of the edges between the different vertices. More precisely, we only connect correspondences which have exactly one image in common. For example, let's assume that we have two vertices v_i and v_j such that $s_i = s_j = s$ and $t_i \neq t_j$. We use our network to predict correspondence fields $\mathbf{C}^{t_i \rightarrow t_j}$ and $\mathbf{C}^{t_j \rightarrow t_i}$ and we define the weight $\epsilon_{i,j}$ of the edge between v_i and v_j as:

$$\epsilon_{i,j} = \frac{1}{2} m_i m_j exp(-\frac{\|x_i^s - x_j^s\|}{\sigma})(exp(-\frac{\|x_i^t - \mathbf{C}^{t_j \rightarrow t_i}(x_j^t)\|}{\sigma}) + exp(-\frac{\|x_j^t - \mathbf{C}^{t_i \rightarrow t_j}(x_i^t)\|}{\sigma})) \tag{4}$$

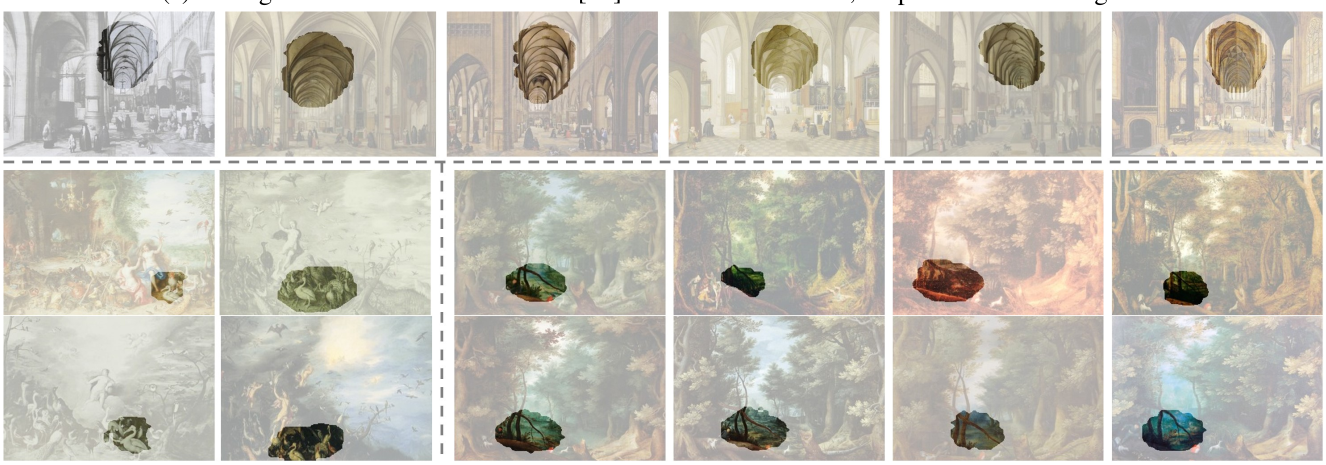
where σ is a scalar hyper-parameter. The edges are defined similarly in the cases $s_i = t_j$, $t_i = s_j$ and $t_i = t_j$. More details about the way we define the graph and in particular strategies to limit the number of vertices are given in the supplementary material [2].

Given the correspondence graph, we use the spectral decomposition of its adjacency matrix [33, 40] either to obtain clusters of correspondences for object discovery, or a foreground potential for co-segmentation. For discovery we first compute N_{eig} principal eigenvectors then performing K-means with $K_{cluster}$ clusters. For co-segmentation, we directly use the first eigenvector to define a foreground potential. Note that because we only consider in the graph correspondences with mask values higher than a threshold, the full graph is extremely sparse that the eigen-decomposition can be efficiently computed.

5.2. Experiments

Object co-segmentation on Internet dataset [48] We build the correspondences graph using for each image only the correspondences in the five most similar images according to the retrieval score of Equation 3. We then use the principal eigen-vector of the correspondence graph to define a seed for GrabCut [46]. More precisely, for every image we associate to each position the sum of the eigen-vector values for the correspondences at this position. Note that GrabCut [46] is crucial to achieve good performance on this dataset, and is widely used by competing approaches such as [9, 21, 25, 42, 48]. More details about the GrabCut [46] can be found in the supplementary material [2]. We follow the standard evaluation protocol [9, 48] and report pixel-level precision $\mathcal{P}$ and the Jaccard index $\mathcal{J}$ on three subsets: Airplane, Car, Horse. The precision $\mathcal{P}$ measures pixel accuracy. The Jaccard index $\mathcal{J}$ is the IoU between the segmented object and ground truth object. Quantitative results are presented in Table 4 and qualitative results in Figure 3a. Our cross-image transformer obtains performance comparable to the state of the art unsupervised approaches. Again, the performances using annotated COCO [36] segments and unsupervised segments are close, which demonstrates that the success of our approach does not come from implicitly leveraging annotated object segmentations. Sparse Nc-Net performances are clearly worse for this task. This can be understood by looking at qualitative results: the segmentation masks predicted by Nc-Net tend to be more localized in discriminative regions.

(a) Co-segmentation results in Internet [48] dataset for the Horse, Airplane and Car categories.



(b) Examples of discovered clustered on Brueghel [1, 53].

Figure 3. Visual results for object discovery. We show co-segmentation results on the Internet [48] in Figure 3a and some discovered clusters in Brueghel [1, 53] in Figure 3b.

Discovery on Brueghel dataset [1, 53] Images are resized to 640 × 640, as many repeated details in Brueghel [1, 53] are small. We also remove duplicate images and images with similar borders to focus on more interesting repeated details. Again, we only include in the graph the correspondences from the five most similar images according to the retrieval score to limit the size of the graph and we perform K-means for $K_{cluster} = 500$ clusters with $N_{eig} = 100$ principal eigen vectors. The graph has ∼900K nodes and it took 10 hours to compute predictions of all the pairs and 2 hours to perform the eigen-decomposition and clustering. Figure 3b presents some interesting clusters that are not covered by ArtMiner [53][1]. More results and details are in the supplementary material [2].

[1] http://imagine.enpc.fr/~shenx/ArtMiner/visualRes/brueghel/brueghel.html

6. Conclusion

In this work, we presented a "segment swapping" approach to generate pairs of images with repeated patterns from which we show it is possible to train co-segementation and correspondence prediction networks. We evaluated two architectures, a cross-image transformer and Sparse Nc-Net [45]. We also compared using annotated segments in COCO [36] and segments extracted in a completely unsupervised way, which shows that our approach is not reliant on COCO [36] object annotations. The trained models shows competitive or better performance on various datasets and different tasks, including art detail retrieval, place recognition and object discovery.

Acknowledgement This work was supported in part by ANR project EnHerit ANR-17-CE23-0008, project Rapid Tabasco, and IDRIS under the allocation AD011011160R1 made by GENCI.

References

[1] Brueghel family: Jan brueghel the elder." the brueghel family database. university of california, berkeley. http://www.janbrueghel.net/. Accessed: 2018-10-16. 1, 3, 4, 5, 6, 7, 8

[2] Supplementary material. http://imagine.enpc.fr/~shenx/SegSwap/suppMat.pdf. 3, 4, 5, 6, 7, 8

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 4, 6

[4] Relja Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*, 2014. 6

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 4

[6] Haw-Shiuan Chang and Yu-Chiang Frank Wang. Optimizing the decomposition for multiple foreground cosegmentation. *CVIU*, 2015. 7

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, 2020. 3, 4

[8] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014. 7

[9] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation. *PAMI*, 2020. 1, 3, 7

[10] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015. 3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 4

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 3

[14] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *ECCV*, 2020. 4, 6

[15] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 3

[16] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 3

[17] Petr Gronat, Guillaume Obozinski, Josef Sivic, and Tomas Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *CVPR*, 2013. 6

[18] Avik Hati, Subhasis Chaudhuri, and Rajbabu Velmurugan. Image co-segmentation using maximum common subgraph matching and region co-growing. In *ECCV*, 2016. 7

[19] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *CVPR*, 2021. 1, 4, 6

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[21] Kuang-Jui Hsu, Yen-Yu Lin, Yung-Yu Chuang, et al. Co-attention cnns for unsupervised object co-segmentation. In *IJCAI*, 2018. 3, 7

[22] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2, 3, 6

[23] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *ECCV*, 2018. 3

[24] Koteswar Rao Jerripothula, Jianfei Cai, Jiangbo Lu, and Junsong Yuan. Object co-skeletonization with co-segmentation. In *CVPR*, 2017. 7

[25] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Image co-segmentation via saliency co-fusion. *TOMM*, 2016. 7

[26] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. *arXiv*, 2021. 2

[27] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 3, 7

[28] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 3

[29] Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011. 7

[30] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *CVPR*, 2017. 4

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 5

[32] Chulwoo Lee, Won-Dong Jang, Jae-Young Sim, and Chang-Su Kim. Multiple random walkers and their application to image cosegmentation. In *CVPR*, 2015. 7

[33] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005. 2, 7

[34] Bo Li, Zhengxing Sun, Qian Li, Yunjie Wu, and Anqi Hu. Group-wise deep object co-segmentation with co-attention recurrent neural network. In *ICCV*, 2019. 3

[35] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018. 3, 7

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 3, 4, 6, 7, 8

[37] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2010. 2

[38] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic attraction-repulsion embedding for large scale image localization. In *ICCV*, 2019. 4

[39] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. In *WACV*, 2019. 2, 3

[40] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *NeurIPS*, 2001. 2, 7

[41] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *SIGGRAPH*, 2003. 2, 3, 6

[42] Rong Quan, Junwei Han, Dingwen Zhang, and Feiping Nie. Object co-segmentation via graph optimized-flexible manifold ranking. In *CVPR*, 2016. 7

[43] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 4

[44] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017. 2, 3

[45] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, 2020. 1, 2, 3, 4, 5, 8

[46] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 2004. 3, 7

[47] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, 2006. 3

[48] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013. 1, 2, 3, 7, 8

[49] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 4

[50] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 6

[51] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *ECCV*, 2018. 3

[52] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. In *ECCV*, 2020. 2, 3

[53] Xi Shen, Alexei A Efros, and Mathieu Aubry. Discovering visual patterns in art collections with spatially-consistent feature learning. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[54] Jian Sun and Jean Ponce. Learning dictionary of discriminative part detectors for image categorization and cosegmentation. *IJCV*, 2016. 7

[55] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *arXiv*, 2021. 2

[56] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014. 3

[57] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016. 3

[58] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 1, 2, 4, 5, 6

[59] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *CVPR*, 2013. 1, 2, 4, 5, 6

[60] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *CVPR*, 2020. 2, 3

[61] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. *arXiv*, 2021. 2, 3

[62] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Warp consistency for unsupervised learning of dense correspondences. *arXiv*, 2021. 2, 3

[63] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013. 3

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 3

[65] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 3

[66] Huy V Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *CVPR*, 2019. 3

[67] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *ECCV*, 2020. 3

[68] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, 2018. 3

[69] Ze-Huan Yuan, Tong Lu, Yirui Wu, et al. Deep-dense conditional random fields for object co-segmentation. In *IJCAI*, 2017. 3

[70] Tinghui Zhou, Yong Jae Lee, Stella X Yu, and Alexei A Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, 2015. 3