FCSS: Fully Convolutional Self-Similarity for Dense Semantic Correspondence

Seungryong Kim, *Student Member, IEEE,* Dongbo Min, *Senior Member, IEEE,* Bumsub Ham, *Member, IEEE,* Stephen Lin, *Member, IEEE,* and Kwanghoon Sohn, *Senior Member, IEEE*

Abstract—We present a descriptor, called fully convolutional self-similarity (FCSS), for dense semantic correspondence. Unlike traditional dense correspondence approaches for estimating depth or optical flow, semantic correspondence estimation poses additional challenges due to intra-class appearance and shape variations among different instances within the same object or scene category. To robustly match points across semantically similar images, we formulate FCSS using local self-similarity (LSS), which is inherently insensitive to intra-class appearance variations. LSS is incorporated through a proposed convolutional self-similarity (CSS) layer, where the sampling patterns and the self-similarity measure are jointly learned in an end-to-end and multi-scale manner. Furthermore, to address shape variations among different object instances, we propose a convolutional affine transformer (CAT) layer that estimates explicit affine transformation fields at each pixel to transform the sampling patterns and corresponding receptive fields. As training data for semantic correspondence is rather limited, we propose to leverage object candidate priors provided in most existing datasets and also correspondence consistency between object pairs to enable weakly-supervised learning. Experiments demonstrate that FCSS significantly outperforms conventional handcrafted descriptors and CNN-based descriptors on various benchmarks.

Index Terms—Dense semantic correspondence, convolutional neural networks, self-similarity, weakly-supervised learning

1 Introduction

Numerous computer vision and computational photography applications require the points on an object in one image to be matched with their corresponding object points in another image, such as a motorbike wheel matched to a different model of motorbike's wheel, as exemplified in Fig. 1. Dealing with such appearance variations over object instances is essential for numerous tasks such as scene recognition, image registration, semantic segmentation, and image editing [1], [2], [3], [4], [5]. Unlike traditional dense correspondence approaches for estimating depth [6], [7] or optical flow [8], [9], in which *visually* similar images of the same scene are used as inputs, establishing dense correspondences across *semantically* similar images poses additional challenges due to intra-class appearance variations among object instances.

Often, basic visual properties such as colors and gradients are not shared among different instances within the same object or scene category. Moreover, geometric variations appear frequently among them. Those variations lead to significant differences in appearance and shape that can degrade matching by handcrafted feature descriptors [10], [11]. Although powerful optimization techniques can help by enforcing smoothness constraints over a correspondence map [2], [3], [5], [12], [13], they are limited in effectiveness without a proper feature descriptor for semantic correspondence estimation.

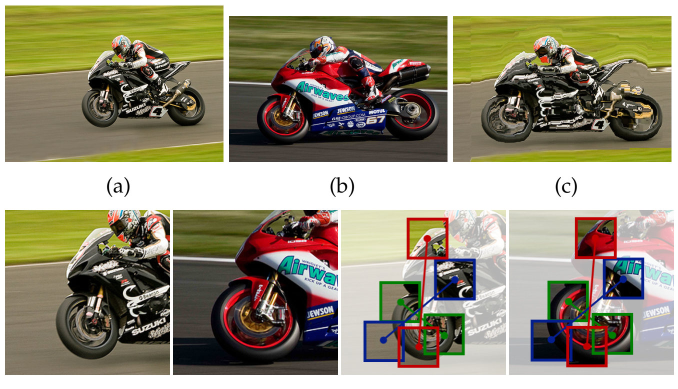


Fig. 1. Visualization of our FCSS results: (a) source image, (b) target image, (c) warped source image using dense correspondences, (d), (e) enlarged windows for source and target images, (f), (g) local self-similarities computed by our FCSS descriptor between source and target images. Even though there are significant differences in appearance among different instances within the same object category in (a) and (b), their local self-similarities computed by our FCSS descriptor are preserved as shown in (f) and (g), providing robustness to intra-class appearance and shape variations.

Over the past few years, convolutional neural network (CNN) based features have become increasingly popular for correspondence estimation thanks to their matching precision and their invariance to minor photometric and geometric deformations [14], [15], [16], [17]. However, for computing semantic correspondences within this framework, greater invariance is needed to deal with the more substantial appearance differences. This could potentially be achieved with a deeper convolutional network [18], but would come at the cost of significantly reduced spatial

- S. Kim, B. Ham, and K. Sohn are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea. E-mail: {srkim89, mimo, khsohn}@yonsei.ac.kr
- D. Min is with the Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea. E-mail: dbmin99@gmail.com
- S. Lin is with Microsoft Research, Beijing 100080, China. E-mail: stevelin@microsoft.com

Manuscript received Jan. 22, 2018

localization precision in matching (see [19], [20] for examples). Furthermore, moderate geometric variations among different object instances cannot be overcome within this framework without considering explicit geometric transformation fields. An additional challenge lies in the lack of training data with ground truth for semantic correspondence, making the use of supervised training difficult.

To address these issues, we introduce a novel CNN-based descriptor that is inherently insensitive to both intra-class appearance and shape variations while maintaining precise spatial localization ability. The key insight, illustrated in Fig. 1, is that among different object instances in the same class, their local structural layouts remain roughly the same. Even with dissimilar colors, gradients, and moderate differences in feature positions, the local self-similarity (LSS) between sampled patch pairs is basically preserved. This property has been utilized for non-rigid object detection [21], sketch retrieval [22], and cross-modal correspondence estimation [23], [24]. However, existing LSS-based techniques are mainly handcrafted and need further robustness to capture reliable matching evidence from semantically similar images.

Our proposed descriptor, called fully convolutional self-similarity (FCSS), formulates the LSS within a fully convolutional network in a manner where the patch sampling patterns and self-similarity measure are both learned. We propose a convolutional self-similarity (CSS) layer that encodes the LSS structure and possesses differentiability, allowing for end-to-end training of the proposed network. The convolutional self-similarities are measured at multiple scales, using skip layers [19] to forward intermediate convolutional activations. To address geometric variations such as affine transformations among different object instances, we propose a convolutional affine transformer (CAT) layer that estimates explicit affine transformation fields to transform the sampling patterns and corresponding receptive fields. Furthermore, since limited training data is available for semantic correspondence, we propose a weakly-supervised feature learning scheme that leverages correspondence consistency within object candidate priors provided in existing datasets. With this learning scheme, we examine two kinds of loss functions for training the proposed network: a correspondence contrastive loss which aims to minimize/maximize convolutional activation differences between matching/non-matching pixel pairs, and a correspondence classification loss which treats correspondence as a classification problem among candidate pixels.

Experimental results show that the FCSS descriptor outperforms conventional handcrafted descriptors and CNN-based descriptors on various benchmarks, including that of Taniai *et al.* [12], Proposal Flow-WILLOW [13], Proposal Flow-PASCAL [25], and the CUB-200-2011 dataset [26], and on different applications, including non-parametric part segmentation on the PASCAL-VOC part dataset [27], foreground mask detection on Caltech-101 [28], non-parametric object segmentation on PASCAL-VOC 2012 [29], and non-parametric object detection on Proposal Flow-PASCAL [25].

This manuscript extends the conference version of this work [30]. It newly adds (1) an affine invariant extension of the FCSS, called CAT-FCSS; (2) an examination of two kinds of loss functions for training the proposed network; and (3) an extensive comparative study with existing semantic correspondence methods using various datasets. The source code of our work is available online at our project webpage: `http://diml.yonsei.ac.kr/∼srkim/FCSS/`.

2 Related Work

2.1 Feature Descriptors

Conventional gradient-based and intensity comparison-based descriptors, such as SIFT [10], HOG [31], DAISY [11], and BRIEF [32], have shown limited performance in dense correspondence estimation across semantically similar but different object instances. Besides these handcrafted features, several attempts have recently been made using deep CNNs to learn discriminative descriptors for local patches from large-scale datasets. Some of these techniques have extracted intermediate convolutional activations as the descriptor [33], [34], [35], [36], which have shown to be effective for patch-level matching. Other methods have directly learned similarity measures for comparing patches using a convolutional similarity network [14], [15], [16], [17]. Even though these CNN-based descriptors encode a discriminative structure with a deep architecture, they have inherent limitations in handling large intra-class variations [16], [37]. Furthermore, some of those methods are tailored to estimate sparse correspondences [14], [17], and cannot in practice provide dense descriptors due to their high computational complexity. Of particular importance, current research on semantic correspondence lacks an appropriate benchmark with dense ground-truth correspondences, making supervised learning of CNNs less feasible for this task.

Use of the LSS descriptor, proposed in [21], has led to impressive results in object detection, image retrieval by sketching [21], deformable shape class retrieval [22], and cross-modal correspondence estimation [23], [24]. Among the more recent cross-modal descriptors inspired by LSS is the dense adaptive self-correlation (DASC) descriptor [23], which provides relatively good performance but is unable to handle non-rigid deformations due to its fixed patch pooling scheme. The deep self-correlation (DSC) descriptor [24] reformulates LSS in a deep non-CNN architecture. As all of these techniques use handcrafted descriptors, they lack the robustness to deformations that is possible with CNNs.

2.2 Dense Semantic Correspondence

Conventionally, many techniques for dense semantic correspondence have employed handcrafted features such as SIFT [10] or HOG [31]. To improve matching quality, they focus on optimization. Graph-based matching algorithms [38], [39] attempt to find category-level feature matches by leveraging a flexible graph representation of images, but they are designed to handle sparsely sampled or detected features. Among these methods are some based on SIFT Flow [2], [3], which uses hierarchical dual-layer belief propagation (BP). Inspired by this, Kim *et al.* [3] proposed the deformable spatial pyramid (DSP) which performs multi-scale regularization with a hierarchical graph. Other instances include methods that perform matching with an exemplar-LDA approach [40], through joint image set alignment [5], or together with cosegmentation [12].

More recently, CNN-based descriptors have been used for establishing dense semantic correspondences. Pretrained ConvNet features [41] were employed with the SIFT Flow algorithm [36] and with semantic flow using object proposals [13]. Zhou *et al.* [42] proposed a deep network consisting of a feature encoder and a flow decoder to predict cross-instance correspondences, which exploits cycle-consistency with a 3-D CAD model [43] as a supervisory signal. However, the need to have 3-D CAD model for each object class limits its applicability. Furthermore, none of those methods are able to handle non-rigid geometric variations. Choy *et al.* [44] proposed a deep convolutional descriptor based on fully convolutional feature learning. As those methods formulate the networks only by combining successive convolutions, they face a tradeoff between appearance invariance and localization precision, which limits their effectiveness for semantic correspondence.

Several methods aim to alleviate geometric variations through extensions of SIFT Flow, including scale-less SIFT Flow (SLS) [45], scale-space SIFT Flow (SSF) [46], and generalized DSP (GDSP) [47]. However, they have a critical and practical limitation that their computation linearly increases with the search space size. Tau *et al.* [48] proposed a dense correspondence algorithm that propagates scales estimated from sparse interest points and uses them to optimize correspondence fields. However, limited performance has been achieved due to propagation of erroneous scales. A generalized PatchMatch algorithm [49] was proposed for efficient matching that leverages a randomized search scheme. It was utilized by HaCohen *et al.* [1] in a non-rigid dense correspondence (NRDC) algorithm, but employs weak matching evidence that cannot guarantee reliable performance. Geometric invariance to scale and rotation is provided by Daisy Filter Flow (DFF) [4], but its implicit smoothing model which relies on randomized sampling and propagation of good estimates in the direct neighborhood often induces mismatches. While those aforementioned techniques provide some amount of geometric invariance, none of them can deal with affine transformations across images, which are a frequent occurrence in semantic correspondence. More recently, Kim *et al.* [50] proposed a discrete-continuous transformation matching (DCTM) framework where dense affine transformation fields are inferred using a handcrafted energy function and optimization.

2.3 Transformation Invariance in CNNs

Most CNN-based approaches tolerate just minor geometric variations by simply employing spatial pooling layers or data augmentation techniques [18]. Recently, Laptev *et al.* [51] proposed a transformation-invariant pooling operator (TI-pooling), but it only considers a set of pre-defined geometric transformations. Spatial transformer networks (STNs) [52] offer a way to deal with geometric variations within CNNs by warping feature maps through a global parametric transformation. More recently, Lin *et al.* [53] proposed inverse compositional spatial transformer networks (IC-STNs) that replaces the feature warping with transformation parameter propagation. However, these methods consider a global image transformation only, and thus they cannot provide tolerance to spatially-varying geometric variations, which frequently appear in dense semantic correspondence. To overcome these limitations, Choy *et al.* [44] developed a descriptor, called universal correspondence network (UCN), based on convolutional STNs that enables spatially-varying feature manipulation. Dai *et al.* [54] introduced deformable convolutional networks (DCN) to encode spatially-varying geometric variations in CNNs.

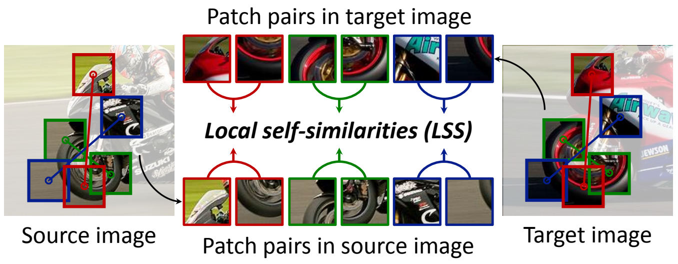

Patch pairs in target image
Local self-similarities (LSS)
Source image
Patch pairs in source image
Target image

Fig. 2. Visualization of LSS descriptor. This descriptor represents local self-similarity between certain patch pairs within a local support window.

3 The FCSS Descriptor

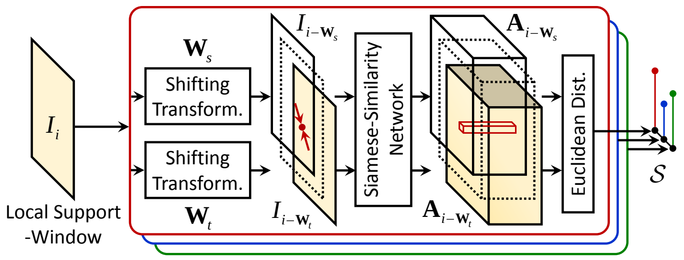
3.1 Problem Formulation and Overview

Given an image I and image point I_i for pixel $i=[i_\mathbf{x},i_\mathbf{y}]^T$, a dense descriptor D_i is designed to extract a robust representation on a local support window. For LSS, this descriptor represents locally self-similar structures around a given pixel by recording the similarity between certain patch pairs within a local support window, as shown in Fig. 2. Formally, LSS can be described as a vector of feature values such that $D_i=\{D_i(l)\}$ for $l=\{1,...,L\}$ with the number of sampling patterns L, where the feature values $D_i(l)$ are computed as

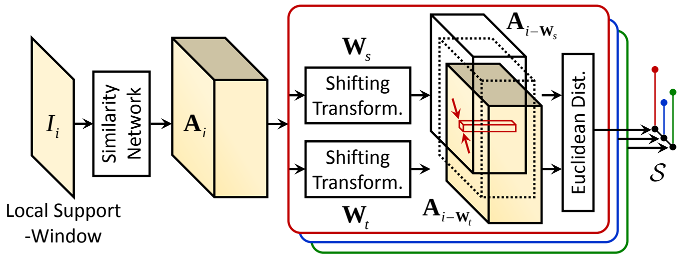
$$D_i(l)=\max\nolimits_{j\in N_i(l)}\exp(-\mathcal{S}(j;s_l,t_l)/\lambda). \tag{1}$$

$\mathcal{S}(i;s_l,t_l)$ is a self-similarity distance between two patches P_{i-s_l} and P_{i-t_l} sampled on s_l and t_l, the l^{th} selected sampling pattern, around center pixel i. To alleviate the effects of outliers, the self-similarity responses are encoded by non-linear mapping with an exponential function of bandwidth λ [55]. For spatial invariance to the position of the sampling pattern, the maximum self-similarity within a spatial window $N_i(l)$ is computed. Based on this framework, LSS has been formulated in various ways, using different self-similarity distances and different sampling strategies for the patch pairs [21], [23], [24].

By leveraging CNNs, our objective is to design a dense descriptor that formulates LSS in a fully convolutional and end-to-end manner for robust estimation of dense semantic correspondences. We design the dense descriptor by first considering translational transformations (Section 3.2, Section 3.3). Our network is built as a multi-scale series of convolutional self-similarity (CSS) layers where each includes a two-stream shifting transformer for applying a learned sampling pattern. This is then extended to provide invariance to geometric distortions such as affine transformations (Section 3.4). Specifically, convolutional affine transformer (CAT) layers are proposed, which transform the sampling patterns and corresponding receptive fields. To learn the network in a weakly-supervised manner, we utilize correspondence consistency between pairs of input images within object bounding boxes provided in most existing datasets [26], [27], [28], [29] (Section 3.5).



(a) Straightforward implementation of CSS layers



(b) Efficient implementation of CSS layers

Fig. 3. Convolutional self-similarity (CSS) layers, which measure convolutional self-similarity $\mathcal{S}(i;\mathbf{W}_s,\mathbf{W}_t)$ between two patches $P_{i-\mathbf{W}_s}$ and $P_{i-\mathbf{W}_t}$. (a) Straightforward version. (b) Efficient version, which equivalently solves for convolutional self-similarity while avoiding repeated computations for convolutions.

3.2 CSS: Convolutional Self-Similarity Layer

Previous LSS-based techniques [21], [23], [24] evaluate Eq. (1) by sampling patch pairs and then computing their similarity using handcrafted metrics, which often fails to yield effective matching evidence for estimating semantic correspondences. To overcome this limitation, we propose the convolutional self-similarity (CSS) layer that learns the sampling patterns and computes the similarity of sampled patch pairs through CNNs.

3.2.1 Two-Stream Shifting Transformer

With l omitted for simplicity, convolutional self-similarity between a patch pair P_{i-s} and P_{i-t} is formulated through a Siamese network, followed by a simple Euclidean distance as shown in Fig. 3(a). The sampling patterns (s,t) of patch pairs are a critical element of local self-similarity. In our CSS layer, a sampling pattern for a pixel i can be generated by shifting the image I_i by s and t to form two different images from which self-similarity is measured. To learn this spatial manipulation of data within the network, we formulate a novel learnable module, called a two-stream shifting transformer layer, in which the shift transformations with s and t are defined as network parameters that can be learned. In this way, the optimized sampling patterns can be learned in the CNN.

Concretely, the sampling patterns are defined as network parameters $\mathbf{W}_s=[W_{s_\mathbf{x}},W_{s_\mathbf{y}}]^T$ and $\mathbf{W}_t=[W_{t_\mathbf{x}},W_{t_\mathbf{y}}]^T$ for all (s,t). Since the shifted sampling is repeated in the image domain, the image I_i is shifted without interpolation according to the fixed sampling patterns as

$$I_{i-\mathbf{W}_s}=\mathcal{F}(I_i;\mathbf{W}_s),\quad I_{i-\mathbf{W}_t}=\mathcal{F}(I_i;\mathbf{W}_t). \tag{2}$$

3.2.2 Convolutional Similarity Network

To compute the convolutional self-similarity, we extract convolutional activations from $I_{i-\mathbf{W}_s}$ and $I_{i-\mathbf{W}_t}$ through feed-forward processes $\mathcal{F}(I_{i-\mathbf{W}_s};\mathbf{W}_c)$ and $\mathcal{F}(I_{i-\mathbf{W}_t};\mathbf{W}_c)$ with similarity network parameters $\mathbf{W}_c$ and measure self-similarity based on the Euclidean distance, such that

$$\mathcal{S}(i;\mathbf{W}_s,\mathbf{W}_t)=\|\mathcal{F}(I_{i-\mathbf{W}_s};\mathbf{W}_c)-\mathcal{F}(I_{i-\mathbf{W}_t};\mathbf{W}_c)\|^2. \tag{3}$$

Note that a convolutional self-similarity $\mathcal{S}(i;\mathbf{W}_s,\mathbf{W}_t)$ is a vector defined for all $(\mathbf{W}_s,\mathbf{W}_t)$.

Our approach employs the Siamese network to measure self-similarity within a single image, in contrast to recent CNN-based descriptors [16], [17] that directly measure the similarity between patches from two different images.

3.2.3 Efficient Computation

Computing $\mathcal{S}(i;\mathbf{W}_s,\mathbf{W}_t)$ for all $(\mathbf{W}_s,\mathbf{W}_t)$ in this network is time-consuming, since the number of iterations through the similarity network is linearly proportional to the number of sampling patterns. To expedite this computation, we instead generate the convolutional activations of an entire image I by passing it through the CNN such that $\mathbf{A}=\mathcal{F}(I;\mathbf{W}_c)$, similar to [56], and then measure the self-similarity for the sampling patterns directly on the convolutional activations, as shown in Fig. 3(b). Formally, we first define the sampled activations through a two-stream shifting transformer

$$\mathbf{A}_{i-\mathbf{W}_s}=\mathcal{F}(\mathbf{A}_i;\mathbf{W}_s),\quad \mathbf{A}_{i-\mathbf{W}_t}=\mathcal{F}(\mathbf{A}_i;\mathbf{W}_t). \tag{4}$$

From this, convolutional self-similarity is then defined as

$$\mathcal{S}(i;\mathbf{W}_s,\mathbf{W}_t)=\|\mathbf{A}_{i-\mathbf{W}_s}-\mathbf{A}_{i-\mathbf{W}_t}\|^2. \tag{5}$$

With this scheme, the self-similarity can be measured by running the similarity network only once, regardless of the number of sampling patterns. Interestingly, a similar computational scheme was also used to reduce computational redundancy when locally extracting convolutional features as in [56], [57], [58].

For end-to-end learning of the proposed descriptor, the derivatives for the CSS layer must be computable, so that the gradients of the final loss can be back-propagated to the convolutional similarity and shifting transformer layers. The differentiability of convolutional self-similarity is derived in the supplementary materials.

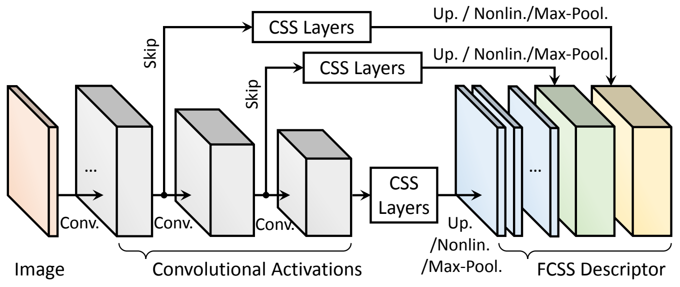


Fig. 4. Network configuration of the FCSS descriptor, consisting of convolutional self-similarity layers at multiple scales.

3.3 Network Configuration for Dense Descriptor

3.3.1 Multi-Scale Convolutional Self-Similarity Layers

In building the descriptor through a CNN architecture, there is a trade-off between robustness to semantic variations and fine-grained localization precision [19], [20].

Inspired by the skip layer scheme in [19], we formulate the CSS layers in a multi-scale manner to encode multi-scale

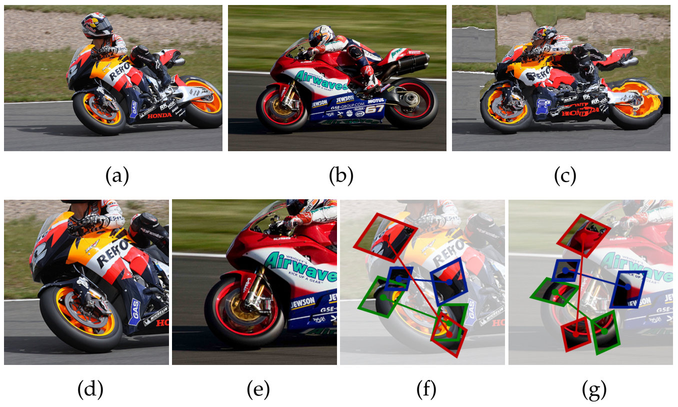


Fig. 5. Visualization of our CAT-FCSS results: (a) source image, (b) target image, (c) warped source image using dense correspondences, (d), (e) enlarged windows for source and target images, (f), (g) local self-similarities computed by our CAT-FCSS descriptor between source and target images. Our CAT-FCSS descriptor provides robustness to geometric variations such as affine transformations.

self-similarities as shown in Fig. 4. Even though the CSS layer itself provides robustness to semantic variations and fine-grained localization precision, this scheme enables the descriptor to boost both robustness and localization precision. Specifically, the CSS layers are located after multi-scale intermediate activations, and their outputs are concatenated to construct the proposed descriptor. In this way, the descriptor naturally encodes self-similarity at multiple scales of receptive fields, and further learns optimized sampling patterns on each scale. It should be noted that many existing descriptors [15], [20] also employ a multi-scale description to improve matching quality.

For intermediate activations $\mathbf{A}^k = \mathcal{F}(\mathbf{A}^{k-1}; \mathbf{W}^k_c)$, where $k \in \{1, ..., K\}$ is the level of convolutional activations and $\mathbf{W}^k_c$ represents convolutional similarity network parameters at the k^{th} level, the self-similarity at the the k^{th} level is measured according to sampling patterns $\mathbf{W}^k_s$ and $\mathbf{W}^k_t$ as

$$\mathcal{S}^k(i; \mathbf{W}^k_s, \mathbf{W}^k_t) = \|\mathbf{A}^k_{i-\mathbf{W}^k_s} - \mathbf{A}^k_{i-\mathbf{W}^k_t}\|^2. \quad (6)$$

Since the intermediate activations are of smaller spatial resolution than the original image resolution due to the stride and/or max-pooling operation, we apply a bilinear upsampling layer [19] after each CSS layer.

3.3.2 Non-linear Mapping and Max-Pooling Layer

Since the pre-learned sampling patterns used in the CSS layers are fixed over an entire image, they may be sensitive to non-rigid deformation as described in [24]. To address this, we perform the max-pooling operation within a spatial window centered at a pixel i after the non-linear mapping:

$$D^k_i(l) = \max_{j \in N^k_i(l)} \exp(-\mathcal{S}^k(j; \mathbf{W}^k_{s,l}, \mathbf{W}^k_{t,l})/\mathbf{W}^k_\lambda), \quad (7)$$

where $\mathbf{W}^k_\lambda$ is a learnable Gaussian kernel parameter for scale k. Note that this non-linear mapping is different from rectified linear units (ReLUs) [59] commonly used after each convolution in that it is designed to reduce the effects of outliers when computing self-similarity as in [21], [23], [24]. The max-pooling layer provides an effect similar to using pixel-varying sampling patterns, providing robustness to minor non-rigid deformations. The descriptor for each pixel then undergoes L_2 normalization. Finally, the proposed

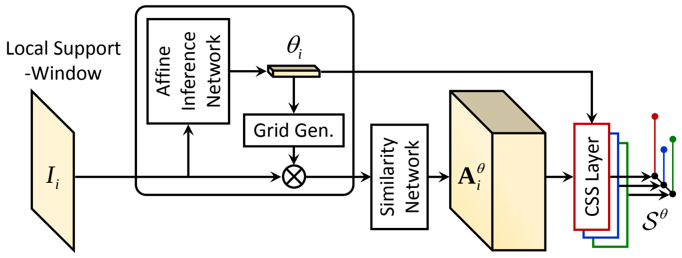


Fig. 6. For affine invariance in FCSS descriptor, a convolutional affine transformer (CAT) layer estimates an affine transformation field $\mathbf{T}(\theta_i)$ to transform the sampling patterns and corresponding receptive fields.

descriptor $D_i = \{D^k_i(l)\}$ for all k and l is built by concatenating the feature responses across all scales. Fig. 4 displays an overview of FCSS descriptor construction.

3.4 Affine-Invariant Dense Descriptor

It is known that CNN-based descriptors provide geometric invariance to some extent thanks to spatial pooling layers [16], [44]. However, it is rather limited and does not obviate the need for explicit consideration of geometric variations. Even our descriptor as described to this point cannot deal with geometric variations due to the lack of a mechanism for explicitly considering geometric transformation fields.

To overcome this issue, we adopt the idea of the spatial transformer layer [52], [53] to explicitly estimate geometric variation fields in the CNN architecture. However, instead of estimating a global image transformation as in [52], [53], we allow each pixel to undergo an independent transformation to deal with locally-varying geometric variations. Specifically, spatially-varying affine transformation fields are first extracted through an additional layer, called the convolutional affine transformer (CAT). With the estimated transformation fields, we then transform the sampling patterns and corresponding receptive fields for measuring self-similarities in the similarity network, as illustrated in Fig. 5. Note that while some methods such as UCN [44] and DCN [54] also transform receptive fields in a similar manner, we extend them to transform the sampling patterns as well as receptive fields tailored to the FCSS descriptor.

3.4.1 CAT: Convolutional Affine Transformer

The CAT layer first infers the affine transformation fields through successive convolutions such that $\theta_i = \mathcal{F}(I_i; \mathbf{W}_a)$ with affine inference network parameters $\mathbf{W}_a$ of the CAT layer, and then uses them to transform the sampling patterns and receptive fields, as shown in Fig. 6.

With an affine transformation field θ_i, sampling patterns $\mathbf{W}_s$ are transformed into affine-varying sampling patterns $\mathbf{W}^\theta_{i,s} = [W^\theta_{i,s_\mathbf{x}}, W^\theta_{i,s_\mathbf{y}}]^T$ as follows:

$$\begin{aligned} \begin{bmatrix} W^\theta_{i,s_\mathbf{x}} \\ W^\theta_{i,s_\mathbf{y}} \end{bmatrix} &= \mathbf{T}(\theta_i) \begin{bmatrix} W_{s_\mathbf{x}} \\ W_{s_\mathbf{y}} \end{bmatrix} \\ &= \begin{bmatrix} \theta^{11}_i & \theta^{12}_i \\ \theta^{21}_i & \theta^{22}_i \end{bmatrix} \begin{bmatrix} W_{s_\mathbf{x}} \\ W_{s_\mathbf{y}} \end{bmatrix}, \end{aligned} \quad (8)$$

where $\mathbf{T}(\theta_i)$ is a matrix form of $\theta_i = [\theta^{11}_i, \theta^{12}_i, \theta^{21}_i, \theta^{22}_i]^T$. Similarly, $\mathbf{W}^\theta_{i,t}$ can be computed from $\mathbf{W}_t$. Compared to sampling patterns $\mathbf{W}_s$ and $\mathbf{W}_t$, affine-varying sampling

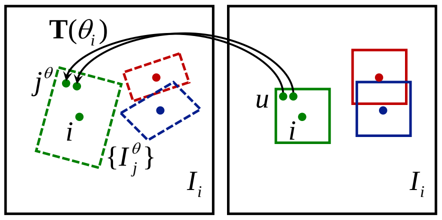


Fig. 7. Visualization of applying an affine-varying receptive field with parameterized sampling grids j^θ according to affine transformation $\mathbf{T}(\theta_i)$.

patterns $\mathbf{W}_{i,s}^\theta$ and $\mathbf{W}_{i,t}^\theta$ are spatially-varying on each pixel i according to its corresponding affine field $\mathbf{T}(\theta_i)$.

For full affine invariance, the receptive fields for measuring self-similarity need to be transformed as well. This warping process can be implemented through image sampling on a parameterized grid, similar to spatial transformer layers [52]. However, a spatially-varying parameterized sampling grid cannot be directly realized with the existing spatial transform layer [52]. To address this issue, we first define a spatially-varying parameterized sampling grid independently for each sample within a receptive field and then apply them sequentially for image sampling. Specifically, for a desired output grid sample $u = [u_\mathbf{x}, u_\mathbf{y}]^T$ within an affine-varying local receptive field, the input grid sample $j^\theta = [j_\mathbf{x}^\theta, j_\mathbf{y}^\theta]^T$ is defined as transforming j with an affine transformation field $\mathbf{T}(\theta_i)$:

$$\begin{bmatrix} j_\mathbf{x}^\theta \\ j_\mathbf{y}^\theta \end{bmatrix} = \begin{bmatrix} \theta_i^{11} & \theta_i^{12} \\ \theta_i^{21} & \theta_i^{22} \end{bmatrix} \begin{bmatrix} u_\mathbf{x} \\ u_\mathbf{y} \end{bmatrix} \tag{9}$$

for all pixels i and all samples u within receptive fields on the regular grid. For each input grid sample j^θ, receptive fields for the convolutional similarity layer are warped through the bilinear sampler [52] independently such that

$$I_j^\theta = \sum\nolimits_i I_i \max(0, 1 - |j_\mathbf{x}^\theta - i_\mathbf{x}|)\max(0, 1 - |j_\mathbf{y}^\theta - i_\mathbf{y}|). \tag{10}$$

This affine-varying local receptive field can be represented in a vector form $\{I_j^\theta\}$ for all samples u, as shown in Fig. 7. With this, affine-varying convolutional activations are computed such that $\mathbf{A}_i^\theta = \mathcal{F}(\{I_j^\theta\}; \mathbf{W}_c)$. Note that since the length of vector $\{I_j^\theta\}$ is $z \times |P_i|$ for a vector I_i of length z and the number of samples within P_i is $|P_i|$, the convolutional parameters for the first convolutional layers with size $h \times w \times z$ should be resized to $1 \times 1 \times hwz$. Then, affine-varying convolutional self-similarity is defined such that

$$\mathcal{S}^\theta(i; \mathbf{W}_{i,s}^\theta, \mathbf{W}_{i,t}^\theta) = \|\mathbf{A}_{i-\mathbf{W}_{i,s}^\theta}^\theta - \mathbf{A}_{i-\mathbf{W}_{i,t}^\theta}^\theta\|^2. \tag{11}$$

For reliable estimates of affine transformation fields θ_i, the network parameters $\mathbf{W}_a$ must be effectively learned. In our approach, affine transformation fields $\mathbf{T}(\theta_i)$ are used to transform both the sampling patterns and the inputs of the similarity network, so $\mathbf{W}_a$ is trained from both the CSS layer and similarity network. The differentiability of convolutional self-similarity is derived in the supplementary materials.

3.4.2 *Multi-Scale Convolutional Affine Transformer Layers*

Since the CSS layers in our descriptor are formulated in a multi-scale manner to encode multi-scale self-similarities, the CAT layers are also built in a multi-scale manner, providing multi-scale geometric invariance. Since optimal affine transformation fields may differ among scales, the CAT layers, $\theta_i^k = \mathcal{F}(\mathbf{A}_i^{k-1}; \mathbf{W}_a^k)$ with network parameters $\mathbf{W}_a^k$ at the k^{th} scale level, are placed before each convolutional activation and the CSS layers. This finally yields a fully affine-invariant FCSS descriptor, which we refer to as CAT-FCSS. Fig. 8 displays an overview of CAT-FCSS descriptor construction.

3.5 Weakly-Supervised Feature Learning

A major challenge of semantic correspondence estimation with CNNs is the lack of ground-truth correspondence maps for training data. To deal with this problem, we propose a weakly-supervised feature learning scheme that obtains putative training samples during training based on correspondence consistency between image pairs. Unlike existing CNN-based descriptor learning methods which use a set of *patch pairs* [14], [15], [16] for training, we use a set of *image pairs*. Such an image-wise learning scheme also expedites feature learning by reducing the computational redundancy that occurs when computing convolutional activations for two adjacent pixels in the image. Our approach is conceptually similar to [44], but we learn the descriptor in a weakly-supervised manner that leverages correspondence consistency between image pairs.

3.5.1 *Correspondence Consistency*

Intuitively, the correspondence relation from a source image to a target image should be consistent with that from the target image to the source image. After forward-propagation with the training image pairs $\mathcal{F}(I; \mathbf{W})$ and $\mathcal{F}(I'; \mathbf{W})$, the best match i^* for each pixel i is computed by comparing descriptors from the two images through nearest neighbor (NN) search [60]:

$$i^* = \text{argmin}_{i'} \|\mathcal{F}(I_i; \mathbf{W}) - \mathcal{F}(I'_{i'}; \mathbf{W})\|^2, \tag{12}$$

where $\mathbf{W} = \{\mathbf{W}_c^k, \mathbf{W}_a^k, \mathbf{W}_s^k, \mathbf{W}_t^k, \mathbf{W}_\lambda^k \mid k = 1, ..., K\}$ represents all network parameters. After running NN twice for the source and target images respectively, we identify the pixel pairs that correspond to each other as putative positive samples. With these putative positive samples, network parameters are learned with loss functions defined in the image domain, as described in the following section. These putative samples are updated at each iteration during training. The feature learning begins by initializing the shifting transform with randomly selected sampling patterns. We found that even initial descriptors generated from random patterns provide enough putative samples to be used for weakly-supervised feature learning, which will be described in detail in the experiments section. A similar observation was also reported in [23].

To boost the computation and convergence of this feature learning, we limit the correspondence candidate regions according to object location priors such as a bounding box or a mask containing the target object to be matched, which are provided in most benchmarks [27], [28], [29]. Similar to [5], [13], [42], it is assumed that true matches exist only within the object region. Utilizing this prior mitigates the side effects that may occur due to background clutter when directly running the NN search, and also expedites the feature learning process.

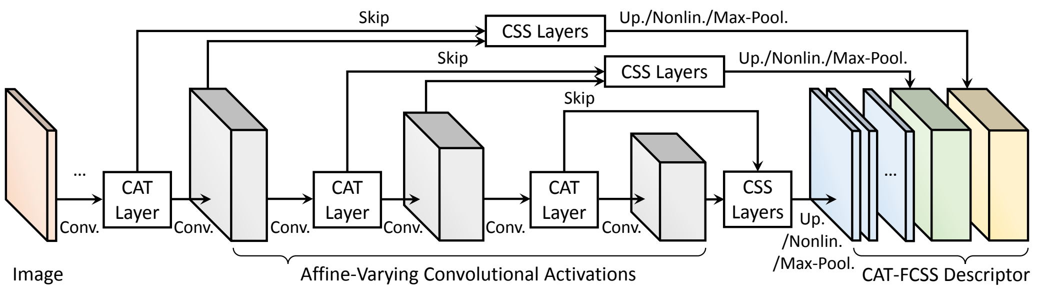


Fig. 8. Network configuration of the CAT-FCSS descriptor, consisting of convolutional affine transformation layers and convolutional self-similarity layers at multiple scales.

3.5.2 *Correspondence Contrastive Loss*

Since our method adopts image domain learning, the loss function is also defined in the image domain. For training the network with image pairs with putative positive samples, the correspondence contrastive loss can be used similarly to [16], [17], [44] such that

$$\begin{aligned}\mathcal{L}_{co}(\mathbf{W}) = \frac{1}{2N}\textstyle\sum_{i\in\Omega_{co}} l_i\|\mathcal{F}(I_i;\mathbf{W}) - \mathcal{F}(I'_{i'};\mathbf{W})\|^2 \\ + (1-l_i)\mathrm{max}(0, C - \|\mathcal{F}(I_i;\mathbf{W}) - \mathcal{F}(I'_{i'};\mathbf{W})\|^2),\end{aligned} \tag{13}$$

where i and i' are either a matching or non-matching pixel pair, and l_i denotes a class label that is 1 for a positive pair and 0 otherwise. Ω_{co} represents the set of training samples, and N is the number of training samples. C is the maximal cost. It should be noted that in many supervised feature learning methods [16], [17], [44], the class label l_i is given from ground truth correspondence maps. Contrarily, in our approach the class label l_i is actively determined via the correspondence consistency. Among a set of correspondence candidates computed from NN search, the pixel pairs with consistent matches are used as positive samples (i.e., $l_i = 1$), and they are taken as negative samples otherwise (i.e., $l_i = 0$). We randomly select the training samples among the positive and negative samples at each iteration during training. Since the negative samples ensue from erroneous local minima in the energy cost, they provide the effects of hard negative mining during training [16].

Although this contrastive loss function yields satisfactory performance [16], [17], [44], it has two inherent limitations. First, obtaining proper negative samples is often problematic, which is essential for learning a network well [16], [17], [44]. Second, it is difficult to consider all possible matching candidates for each pixel, often leading to solutions trapped in erroneous local minima [44].

3.5.3 *Correspondence Classification Loss*

The correspondence estimation task is generally formulated as a pixel-labeling problem. Thus, to establish reliable correspondences, a descriptor should have high discriminability to distinguish the true correspondence from other candidates. To this end, we propose a correspondence classification loss that computes a softmax loss for each pixel across all possible correspondence candidates. Specifically, for each pixel i and its possible match candidates, the correspondence classification loss is defined as

$$\mathcal{L}_{cl}(\mathbf{W}) = -\frac{1}{2N}\sum_{i\in\Omega_{cl}}\sum_{i'} p_T(I'_{i'})\log(p(I'_{i'};I_i,\mathbf{W})), \tag{14}$$

where i' is an index over all possible matching candidates. $p_T(I'_{i'})$ is a class label defined as 1 if $i' = i^*$, and 0 otherwise. Ω_{cl} represents the set of training samples. The function $p(I'_{i'};I_i,\mathbf{W})$ is a softmax probability defined as

$$p(I'_{i'};I_i,\mathbf{W}) = \frac{\exp(1 - \|\mathcal{F}(I_i;\mathbf{W}) - \mathcal{F}(I'_{i'};\mathbf{W})\|^2)}{\sum_l \exp(1 - \|\mathcal{F}(I_i;\mathbf{W}) - \mathcal{F}(I'_l;\mathbf{W})\|^2)}, \tag{15}$$

where l is an index over all possible matching candidates.

In contrast to the contrastive loss $\mathcal{L}_{co}$, the classification loss $\mathcal{L}_{cl}$ can consider all matching candidates through aggregating all of their derivatives, thus providing boosted learning performance. Furthermore, it does not need to define hard negative samples. However, since the classification loss $\mathcal{L}_{cl}$ needs the aggregation of all derivatives across possible candidates, its computational time at each iteration during training is larger than that of the contrastive loss $\mathcal{L}_{co}$. Nevertheless, the descriptor learned with the classification loss $\mathcal{L}_{cl}$ can provide highly boosted matching accuracy.

4 Experimental Results and Discussion

4.1 Experimental Settings

For our experiments, we implemented the FCSS and CAT-FCSS descriptor using the `VLFeat MatConvNet toolbox` [61] on an Intel Core i7-3770 CPU with an NVIDIA GeForce GTX TITAN X GPU. For convolutional similarity networks, we used the ImageNet pretrained VGG-Net [18] from the bottom conv1 to the conv3-4 layer, with their network parameters as initial values. CSS layers are located after conv2-2, conv3-2, and conv3-4, thus $K = 3$. For the CAT-FCSS descriptor, CAT layers are located after conv2-1, conv3-1, and conv3-3, followed by three CSS layers. Considering the tradeoff between efficiency and robustness, the number of sampling patterns is set to 64, thus the total dimension of the descriptor is $L = 192$. Before each CSS layer, convolutional activations undergo L_2 normalization to reduce the effect of outliers [62]. To learn the network, we employed the Caltech-101 dataset [28] excluding testing image pairs used in the experiments. The number of training samples N is 1024. C is set to 0.2. The learned parameters were used for all the experiments.

In the following, we comprehensively evaluated our descriptor through comparisons to state-of-the-art handcrafted descriptors, including SIFT [10], DAISY [11], HOG [31], LSS [21], and DASC [23], as well as recent CNN-based feature descriptors, including MatchNet (MatchN.) [14], Deep Compare (DeepC.) [15], Deep Descriptor (DeepD.) [16],

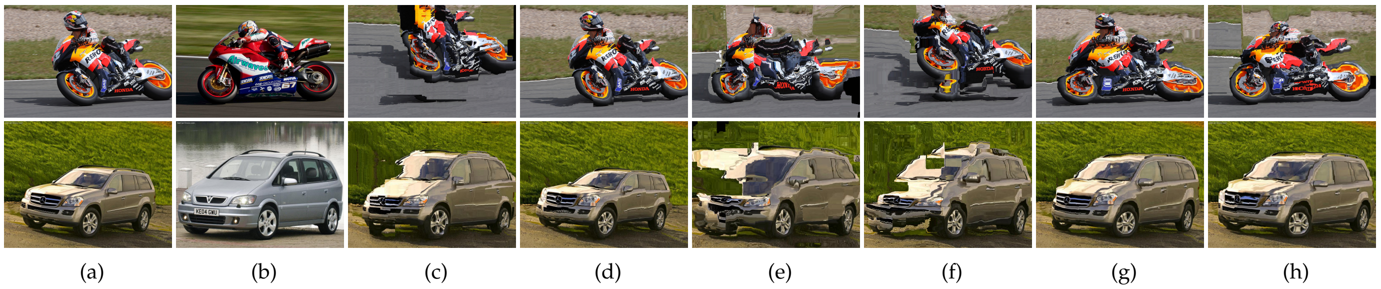


Fig. 10. Qualitative results for various feature descriptors with fixed SF optimization on the Taniai benchmark [12]: (a) source image, (b) target image, (c) SIFT [10], (d) DASC [23], (e) DeepD. [16], (f) MatchN. [14], (g) FCSS, and (h) CAT-FCSS w/$\mathcal{L}_{cl}$.

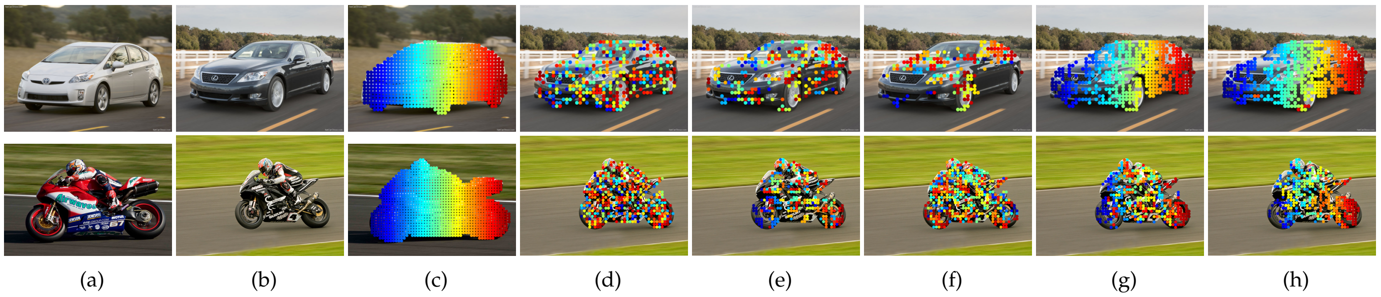


Fig. 11. Qualitative results for various feature descriptors with fixed NN optimization on the Taniai benchmark [12]: (a) source image, (b) target image, (c) source key-points within object regions, (d) LSS [21], (e) LIFT [17], (f) DeepC. [15], (g) VGG [18], and (h) FCSS. For the visualization, color-coded source key-points were warped to the target images using correspondences.

UCN [44], learned invariant feature transform (LIFT) [17][1], and Zhou *et al.* [42]. Note that while all of these CNN-based feature descriptors were learned in a fully supervised manner, our FCSS and CAT-FCSS were learned in a weakly-supervised manner. Furthermore, geometry-robust methods including SLS [45], SSF [46], SegSIFT [63], Lin *et al.* [64], DFF [4], GDSP [47], Proposal Flow (PF) [13], and WarpNet [65] were evaluated. The performance was measured on the Taniai *et al.* benchmark [12], Proposal Flow-WILLOW dataset [13], Proposal Flow-PASCAL dataset [25], CUB-200-2011 dataset [26], PASCAL-VOC part dataset [27], and Caltech-101 benchmark [28]. For ablation experiments to validate the components of the FCSS descriptor, we evaluated the initial VGG-Net (conv3-4) [18] (VGG), the VGG-Net with learned single-scale CSS layer (VGG w/S-CSS) and learned multi-scale CSS layers (VGG w/M-CSS)[2]. Furthermore, we evaluated the CAT-FCSS descriptor in comparison to the FCSS descriptor, and the performance gain of the proposed descriptor with the correspondence classification loss (FCSS w/$\mathcal{L}_{cl}$, CAT-FCSS w/$\mathcal{L}_{cl}$) in place of the correspondence contrastive loss. As an optimizer for estimating dense correspondence maps, we used the hierarchical dual-layer belief propagation (BP) of the SIFT Flow (SF) optimization [2], whose code is publicly available. The performance of our descriptor when combined with other powerful optimizers was also examined using PF [13]. Furthermore, to evaluate the performance of the descriptor itself, we used simple nearest neighbor (NN) optimization[3].

1. Since MatchN. [14], DeepC. [15], DeepD. [16], UCN [44], and LIFT [17] were developed for sparse correspondence, sparse descriptors were first built by forward-propagating images through networks and then were upsampled.

2. In 'VGG w/S-CSS' and 'VGG w/M-CSS', the sampling patterns were only learned with VGG-Net layers fixed.

3. Due to complexity issues, we uniformly sampled points on the foreground with a stride of 8 as keypoints for matching, similar to [65].

TABLE 1
Matching accuracy for various feature descriptors with fixed SF optimization on the Taniai benchmark [12]. VGG w/S-CSS† denotes results with randomly selected sampling patterns.

Methods	FD3D.	JODS	PASC.	Avg.
SIFT [2]	0.632	0.509	0.360	0.500
DAISY [11]	0.636	0.373	0.338	0.449
LSS [21]	0.644	0.349	0.359	0.451
DASC [23]	0.668	0.454	0.261	0.461
DeepD. [16]	0.684	0.315	0.278	0.426
DeepC. [15]	0.753	0.405	0.335	0.498
MatchN. [14]	0.561	0.380	0.270	0.404
LIFT [17]	0.730	0.318	0.306	0.451
UCN [44]	0.741	0.321	0.311	0.458
VGG [18]	0.756	0.490	0.360	0.535
VGG w/S-CSS†	0.762	0.521	0.371	0.551
VGG w/S-CSS	0.775	0.552	0.391	0.573
VGG w/M-CSS	0.806	0.573	0.451	0.610
FCSS	0.830	0.656	0.494	0.660
FCSS w/$\mathcal{L}_{cl}$	0.832	0.662	0.512	0.668
CAT-FCSS	0.798	0.625	0.500	0.641
CAT-FCSS w/$\mathcal{L}_{cl}$	**0.858**	**0.680**	**0.522**	**0.687**

4.2 Matching Results

4.2.1 Results on Taniai Benchmark

We first evaluated the FCSS and CAT-FCSS descriptors on the Taniai benchmark [12], which consists of 400 image pairs divided into three groups: FG3DCar [66], JODS [67], and PASCAL [68]. As in [12], matching accuracy was measured by computing the proportion of foreground pixels with an absolute flow endpoint error that is smaller than a certain threshold T, after resizing images so that its larger dimension is 100 pixels. Table 1 summarizes the matching accuracy for various feature descriptors with the SF optimization fixed ($T = 5$ pixels). Interestingly, while both the CNN-based descriptors [14], [15], [16], [17], [44] and the handcrafted descriptors [10], [11], [21], [23] tend to show similar performance, our method outperforms both of these

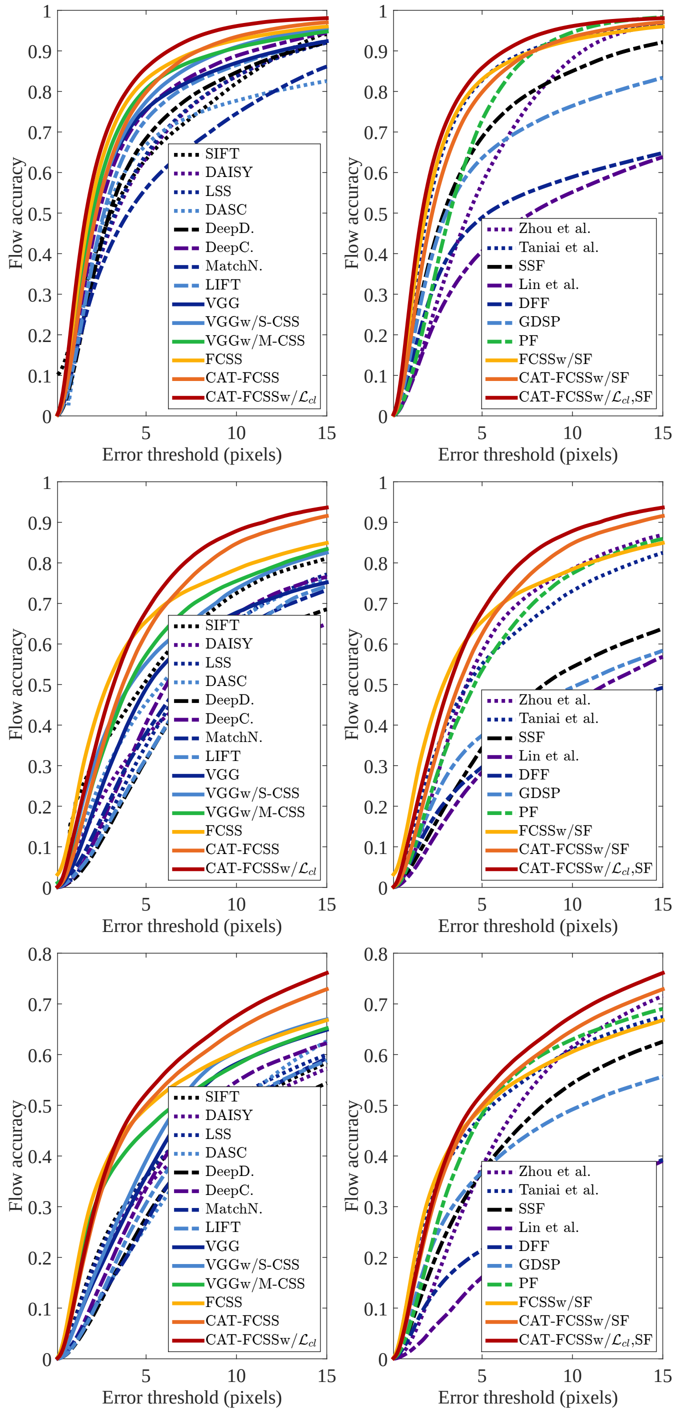


Fig. 9. Average matching accuracy with respect to endpoint error threshold on the Taniai benchmark [12]: (a) various feature descriptors with SF optimization, (b) state-of-the-art correspondence techniques on image pairs within (from top to bottom) FG3DCar, JODS, and PASCAL on the Taniai benchmark [12].

approaches. Fig. 9 shows the flow accuracy with varying error thresholds. Fig. 10 and Fig. 11 show qualitative results. Table 2 compares the matching accuracy ($T = 5$ pixels) with other correspondence techniques. Taniai *et al.* [12] and Proposal Flow [13] provide plausible flow fields, but their methods have limitations due to their usage of handcrafted features. Thanks to its invariance to intra-class variations and precise localization ability, our FCSS achieves the best

TABLE 2
Matching accuracy compared to state-of-the-art correspondence techniques on the Taniai benchmark [12].

Methods	FG3D.	JODS	PASC.	Avg.
SIFT Flow [2]	0.632	0.509	0.360	0.500
DSP [3]	0.487	0.465	0.382	0.445
Zhou *et al.* [42]	0.721	0.514	0.436	0.556
Taniai *et al.* [12]	0.830	0.595	0.483	0.636
SLS [45]	0.525	0.519	0.320	0.457
SSF [46]	0.687	0.344	0.370	0.467
SegSIFT [63]	0.612	0.421	0.331	0.457
Lin *et al.* [64]	0.406	0.283	0.161	0.283
DFF [4]	0.489	0.296	0.214	0.333
GDSP [47]	0.639	0.374	0.368	0.459
Proposal Flow [13]	0.786	0.653	0.531	0.657
FCSS w/PF [13]	0.839	0.635	0.582	0.685
CAT-FCSS w/PF [13]	0.842	0.641	0.586	0.690
CAT-FCSS w/$\mathcal{L}_{cl}$,PF [13]	**0.846**	**0.667**	**0.591**	**0.701**

TABLE 3
Matching accuracy for various feature descriptors with SF optimization on the Proposal Flow-WILLOW benchmark [13]. LIFT† denotes results of LIFT [17] with densely sampled windows.

Methods	PCK		
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
SIFT [2]	0.247	0.380	0.504
DAISY [11]	0.324	0.456	0.555
LSS [21]	0.347	0.504	0.626
DASC [23]	0.255	0.411	0.564
DeepD. [16]	0.187	0.308	0.430
DeepC. [15]	0.212	0.364	0.518
MatchN. [14]	0.205	0.338	0.476
LIFT [17]	0.197	0.322	0.449
LIFT† [17]	0.224	0.346	0.489
UCN [44]	0.221	0.354	0.492
VGG [18]	0.224	0.388	0.555
VGG w/S-CSS	0.239	0.422	0.595
VGG w/M-CSS	0.344	0.514	0.676
FCSS	0.354	0.532	0.681
FCSS w/$\mathcal{L}_{cl}$	0.356	0.534	0.684
CAT-FCSS	0.361	0.541	0.686
CAT-FCSS w/$\mathcal{L}_{cl}$	**0.362**	**0.546**	**0.692**

results both quantitatively and qualitatively. Furthermore, since almost all of the image pairs on the Taniai benchmark [12] have nearly identical poses and do not have severe geometric variations, our CAT-FCSS provides rather degraded performance compared to FCSS. However, on other following benchmarks with severe geometric variations, CAT-FCSS exhibits clearly boosted matching accuracy. In the results of 'VGG w/S-CSS†', we found that even initial random patterns of the FCSS network can provide relatively reliable performance, which demonstrates that FCSS can be learned with enough initial putative training samples. Furthermore, when using the classification loss $\mathcal{L}_{cl}$ to learn the FCSS and CAT-FCSS, the matching accuracy was highly improved compared to the cases of using the contrastive loss $\mathcal{L}_{co}$, which means that with a more powerful loss fuction, the CAT layer can provide robustness even for image pairs having similar geometric configurations.

4.2.2 *Results on Proposal Flow-WILLOW Benchmark*

We also evaluated our descriptor on the Proposal Flow-WILLOW benchmark [13], which includes 10 object subclasses with 10 keypoint annotations for each image. For the evaluation metric, we used the probability of correct keypoint (PCK) between flow-warped keypoints and the ground truth [13], [36]. The warped keypoints are deemed

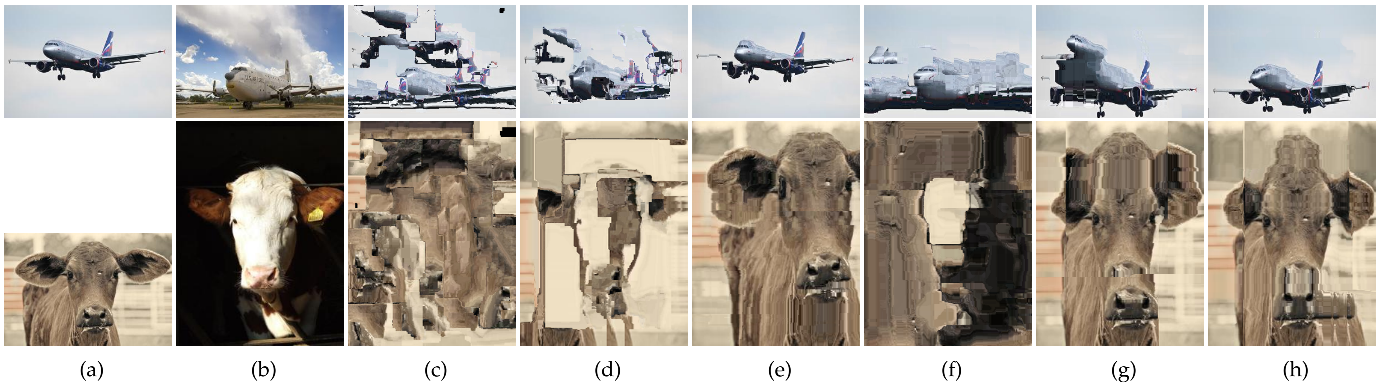


Fig. 12. Qualitative results for various feature descriptors with fixed SF optimization on the Proposal Flow-PASCAL benchmark [25]: (a) source image, (b) target image, (c) DAISY [11], (d) LSS [21], (e) DASC [23], (f) VGG [18], (g) FCSS w/$\mathcal{L}_{cl}$, and (h) CAT-FCSS w/$\mathcal{L}_{cl}$.

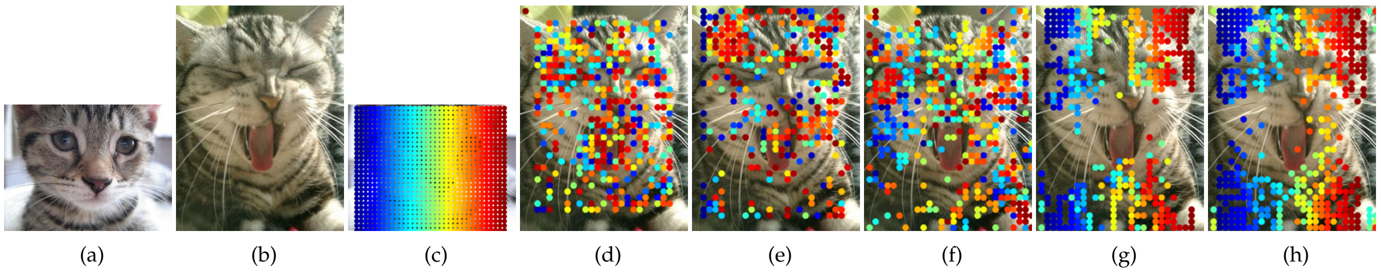


Fig. 13. Qualitative results for various feature descriptors with fixed NN optimization on the Proposal Flow-PASCAL benchmark [25]: (a) source image, (b) target image, (c) source key-points, (d) LSS [21], (e) DASC [23], (f) VGG [18], (g) FCSS w/$\mathcal{L}_{cl}$, and (h) CAT-FCSS w/$\mathcal{L}_{cl}$. For the visualization, color-coded source key-points were warped to the target images using correspondences.

TABLE 4
Matching accuracy compared to state-of-the-art correspondence techniques on the Proposal Flow-WILLOW benchmark [13].

Methods	PCK		
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
SIFT Flow [2]	0.247	0.380	0.504
DSP [3]	0.239	0.364	0.493
Zhou *et al.* [42]	0.197	0.524	0.664
SSF [46]	0.292	0.401	0.531
Lin *et al.* [64]	0.192	0.354	0.487
DFF [4]	0.241	0.362	0.510
GDSP [47]	0.242	0.487	0.512
Proposal Flow [13]	0.284	0.568	0.682
FCSS w/PF [13]	0.295	0.584	0.715
CAT-FCSS w/PF [13]	0.301	**0.587**	0.721
CAT-FCSS w/$\mathcal{L}_{cl}$,PF [13]	**0.311**	0.579	**0.725**

TABLE 5
Matching accuracy for various feature descriptors with SF optimization on the Proposal Flow-PASCAL benchmark [25].

Methods	PCK		
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
SIFT [2]	0.192	0.334	0.492
DAISY [11]	0.189	0.324	0.489
LSS [21]	0.201	0.331	0.481
DASC [23]	0.204	0.364	0.601
DeepD. [16]	0.189	0.324	0.412
DeepC. [15]	0.207	0.367	0.473
MatchN. [14]	0.212	0.343	0.397
LIFT [17]	0.227	0.359	0.421
UCN [44]	0.234	0.367	0.431
FCSS w/$\mathcal{L}_{cl}$	0.269	0.462	0.636
CAT-FCSS w/$\mathcal{L}_{cl}$	**0.274**	**0.468**	**0.647**

to be correctly predicted if they lie within $\alpha \cdot \max(h_b, w_b)$ pixels of the ground-truth keypoints for $\alpha \in [0, 1]$, where h_b and w_b are the height and width of the object bounding box, respectively. The PCK values were measured for various feature descriptors with SF optimization in Table 3, and for different correspondence techniques in Table 4. Our FCSS descriptor with SF optimization shows competitive performance compared to recent state-of-the-art correspondence methods. When combined with PF optimization instead, our method significantly outperforms the existing state-of-the-art descriptors and correspondence techniques.

4.2.3 Results on Proposal Flow-PASCAL Benchmark

We evaluated our descriptor on the Proposal Flow-PASCAL benchmark [25], which samples 1,351 image pairs for 20 object categories from PASCAL keypoint annotations [69]. For the evaluation metric, we used the PCK between flow-warped keypoints and the ground truth [13] as in the experiments on the Proposal Flow-WILLOW benchmark [13].

The PCK values were measured for various feature descriptors with SF optimization in Table 5, and for different correspondence techniques in Table 6. Fig. 12 and Fig. 13 show qualitative results for dense flow estimation. Our FCSS descriptor exhibits performance competitive to state-of-the-art handcrafted and CNN-based descriptors. Our CAT-FCSS descriptor was found to be especially reliable for severe geometric deformations.

4.2.4 Results on CUB-200-2011 Benchmark

Lastly, we evaluated our descriptor on the CUB-200-2011 dataset [26], which contains 11,788 images of 200 bird categories, with 15 parts annotated. We followed the experimental configuration in [65], which utilizes the training set to extract training pairs and 5,000 other image pairs from the validation subset as testing pairs. For the evaluation metric, we used the PCK between flow-warped keypoints and the ground truth [65], where a match is considered correct if the predicted point is within $\alpha \cdot L_d$ of the mean diagonal length

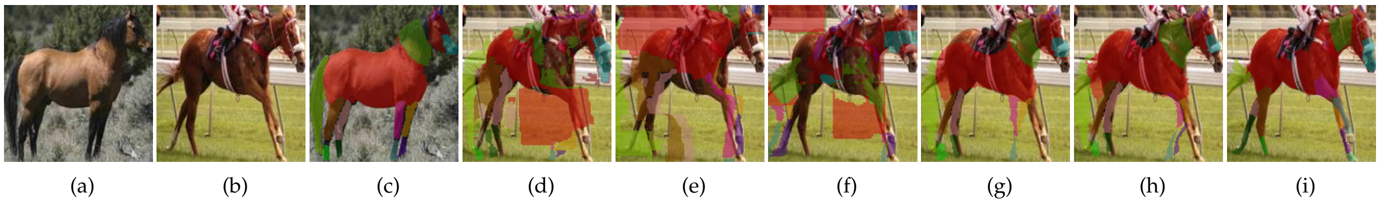


Fig. 15. Visualizations of non-parametric part segmentation on the PASCAL-VOC part dataset [27]: (a) source image, (b) target image, (c) source mask, (d) LSS [70], (e) DeepD. [16], (f) LIFT [17], (g) FCSS, (h) CAT-FCSS w/$\mathcal{L}_{cl}$, and (i) target mask.

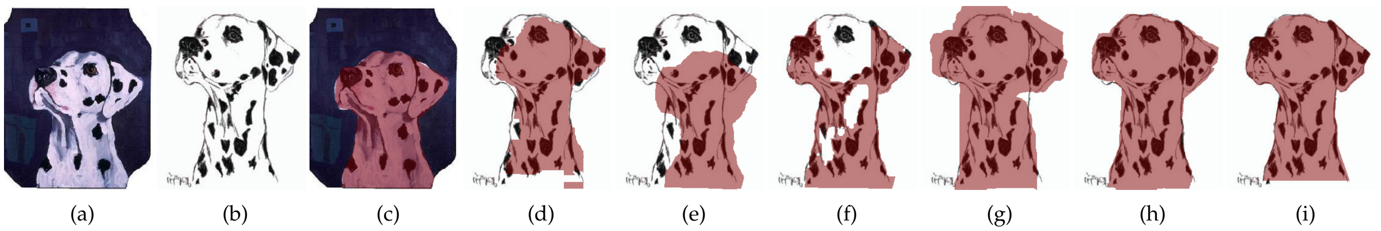


Fig. 16. Visualizations of foreground mask transfer on the Caltech-101 dataset [28]: (a) source image, (b) target image, (c) source mask, (d) SIFT [10], (e) DASC [23], (f) MatchN. [14], (g) LIFT [17], (h) FCSS, and (i) target mask.

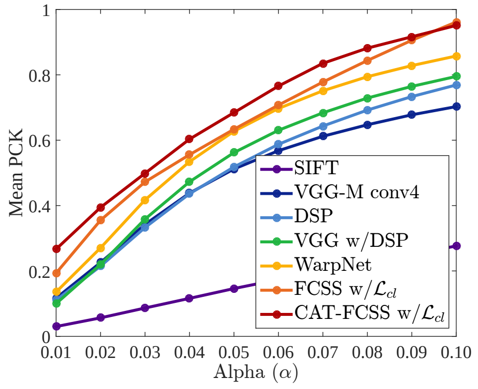


Fig. 14. Average PCK on the CUB-200-2011 benchmark [26].

TABLE 6
Matching accuracy compared to state-of-the-art correspondence techniques on the Proposal Flow-PASCAL benchmark [25].

Methods	PCK $\alpha = 0.05$	PCK $\alpha = 0.1$	PCK $\alpha = 0.15$
SIFT Flow [2]	0.192	0.334	0.492
DSP [3]	0.198	0.372	0.414
Zhou *et al.* [42]	0.181	0.410	0.624
SSF [46]	0.210	0.382	0.511
Lin *et al.* [64]	0.204	0.368	0.498
DFF [4]	0.214	0.372	0.522
GDSP [47]	0.222	0.412	0.524
Proposal Flow [13]	0.242	0.451	0.640
CAT-FCSS w/$\mathcal{L}_{cl}$,PF [13]	0.270	**0.472**	0.646

of the two images L_d. We uniformly sample points on the foreground with a stride of 8 as keypoints for matching.

The average PCK was measured for various descriptors and correspondence techniques in Fig. 17. In this experiment, we evaluated our descriptor using nearest neighbor (NN) search to evaluate the performance contribution of the descriptor, following [65]. Our FCSS and CAT-FCSS descriptors with NN search show competitive performance compared to recent state-of-the-art optimization based methods such as DSP [3] and WarpNet [65].

4.3 Applications

4.3.1 Non-parametric Part Segmentation

To verify the superiority of our descriptor, we applied our descriptor to the non-parameteric part segmentation task on the dataset provided by [5], where the images are sampled from the PASCAL-VOC part dataset [27]. In this application, part segments of the target image were estimated by warping ground truth part segments of the source image using dense correspondences, enabling non-parametric part segmentation. With human-annotated part segments, we measured part matching accuracy using the weighted intersection over union (IoU) score between transferred segments and ground truths, with weights determined by the pixel area of each part. To evaluate alignment accuracy, we measured the PCK metric using keypoint annotations for the 12 rigid PASCAL classes [71]. Table 7 summarizes the matching accuracy compared to state-of-the-art correspondence methods. Fig. 15 visualizes estimated dense flow with color-coded part segments. From the results, our FCSS descriptor is found to yield the highest matching accuracy.

4.3.2 Foreground Mask Detection

Furthermore, we applied our descriptor to the foreground mask detection task on the Caltech-101 dataset [28], where the ground truth foreground mask of the source images were transferred to the target images by using dense correspondences. Following the experimental protocol in [3], we randomly selected 15 pairs of images for each object class, and evaluated matching accuracy with three metrics: label transfer accuracy (LT-ACC) [73], the IoU metric, and the localization error (LOC-ERR) of corresponding pixel positions. Table 8 summarizes the matching accuracy compared to the state-of-the-art correspondence methods. Fig. 16 visualizes estimated dense flow fields with mask transfer. For the results, our FCSS descriptor clearly outperforms the comparison techniques.

4.3.3 Non-parametric Semantic Segmentation

We also applied our descriptor to the non-parameteric semantic segmentation task on the PASCAL-VOC 2012 benchmark [29]. Similar to the preceding experiments, we cast the non-parametric segmantic segmentation problem as a segmentation mask transfer between source and target images. In [73], similar approaches were used for single image

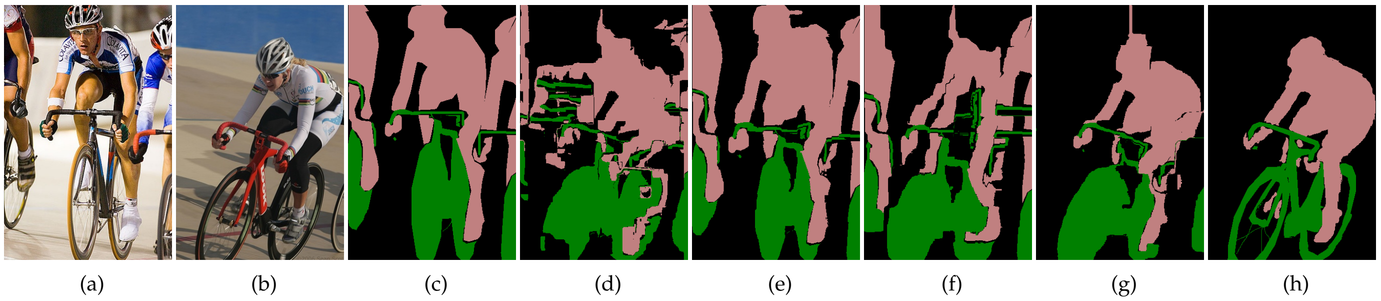


Fig. 17. Visualization of non-parametric semantic segmentation on the the PASCAL-VOC 2012 benchmark [29]: (a) source image, (b) target image, (c) source semantic segments, (d) DAISY [11], (e) VGG [18], (f) FCSS w/$\mathcal{L}_{cl}$, (g) CAT-FCSS w/$\mathcal{L}_{cl}$, and (h) target semantic segments. For visualization, color-coded source semantic segments were warped to the target images using correspondences.

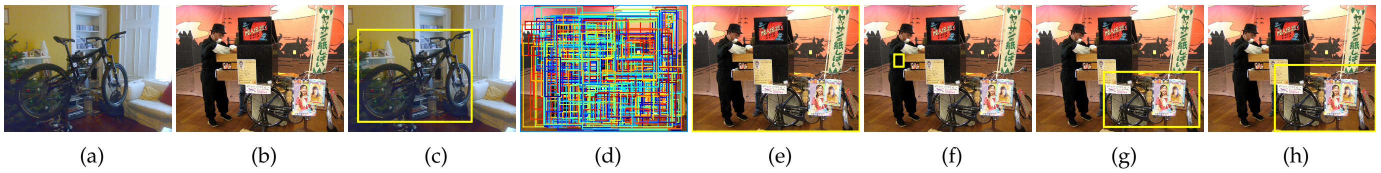


Fig. 18. Visualization of non-parametric object detection on the Proposal Flow-PASCAL benchmark [25]: (a) source image, (b) target image, (c) source ground truth object bounding box, (d) object proposals [72] in target image, object detection results using (e) VGG [18], (f) UCN [44], (g) CAT-FCSS w/$\mathcal{L}_{cl}$, and (h) target ground truth object bounding box.

TABLE 7
Quantitative results for non-parametric part segmentation on the PASCAL-VOC part dataset [27].

Methods	IoU	PCK	
		$\alpha = 0.05$	$\alpha = 0.1$
DFF [4]	0.36	0.14	0.31
GDSP [47]	0.40	0.16	0.34
FlowWeb [3]	0.43	0.26	-
Zhou *et al.* [42]	-	-	0.24
Proposal Flow [13]	0.41	0.17	0.36
UCN [44]	-	0.26	0.44
FCSS w/SF [2]	0.44	0.28	0.47
FCSS w/PF [13]	0.46	0.29	0.46
CAT-FCSS w/SF [2]	0.41	**0.31**	0.45
CAT-FCSS w/PF [13]	**0.47**	0.30	0.48
CAT-FCSS w/$\mathcal{L}_{cl}$,SF [2]	0.45	0.29	0.49
CAT-FCSS w/$\mathcal{L}_{cl}$,PF [13]	0.46	0.29	**0.51**

TABLE 8
Quantitative results for foreground mask detection on the Caltech-101 dataset [28].

Methods	LT-ACC	IoU	LOC-ERR
DSP [3]	0.77	0.47	0.35
SIFT Flow [2]	0.75	0.48	0.32
Proposal Flow [13]	0.78	0.50	0.25
VGG [18] w/SF [2]	0.78	0.51	0.25
FCSS w/SF [2]	0.80	0.50	0.21
FCSS w/PF [2]	0.83	0.52	0.22
CAT-FCSS w/$\mathcal{L}_{cl}$,SF [2]	0.81	0.53	**0.19**
CAT-FCSS w/$\mathcal{L}_{cl}$,PF [13]	**0.84**	**0.55**	0.20

scene parsing by leveraging its nearest neighbors from a large database containing fully annotated images. In this experiment, we only consider a simplified version of [73] in which we transfer ground truth segmentation masks from a single target image queried using GIST [74]. For quantitative evaluations, we adopted the mean intersection over union (mIoU) between the predicted segmentations and ground truths on the validation sets of [29]. Fig. 17 shows the predicted semantic segmentation using dense correspondences. Table 9 presents quantitative comparisons to state-of-the-art correspondence methods. Our FCSS and CAT-FCSS show state-of-the-art performance even for challenging scenarios

TABLE 9
Quantitative results for non-parametric semantic segmentation on the PASCAL-VOC 2012 benchmark [29].

Methods	mIoU
DeepD. [16]	0.39
DeepC. [15]	0.38
MatchN. [14]	0.46
LIFT [17]	0.51
VGG [18]	0.46
FCSS w/$\mathcal{L}_{cl}$	0.59
CAT-FCSS w/$\mathcal{L}_{cl}$	**0.62**

TABLE 10
Quantitative results for non-parametric object detection on the Proposal Flow-PASCAL benchmark [25].

Methods	mAP (%)
DeepD. [16]	0.46
DeepC. [15]	0.42
MatchN. [14]	0.49
LIFT [17]	0.45
VGG [18]	0.51
FCSS w/$\mathcal{L}_{cl}$	0.60
CAT-FCSS w/$\mathcal{L}_{cl}$	**0.63**

in semantic segmentation.

4.3.4 *Non-parametric Object Detection*

Finally, we applied our descriptor to the non-parametric object detection task on the Proposal Flow-PASCAL benchmark [25]. In this experiment, we demonstrate that dense feature descriptors such as our FCSS and CAT-FCSS can be used to detect objects in a non-parametric manner. Specifically, for each feature descriptor of ground truth bounding boxes in the source image, we first estimate the similarities between the features of possible object bounding boxes detected by an object proposal method [72] in the target image, and then detect the objects based on the similarities. We extracted features for the k-th object proposals D_k^{op} using spatial pyramid matching [75] from dense descriptors D_i as in [25]. We also measured the normalized feature similarity that minimizes $\exp(-|D_k^{\mathrm{op}} - D_k^{\prime,\mathrm{op}}|)$. Note that we did not apply any post-processing techniques such as

TABLE 11
Evaluation of computation time in constructing a descriptor in seconds (†Runtime measured on a GPU).

Image size	SIFT	DAISY	VGG	FCSS	CAT-FCSS
800×600	252	3.8	$7.3/1.6^\dagger$	$9.4/2.6^\dagger$	$11.2/3.2^\dagger$

non-maximum suppression. For quantitative evaluation, we measured the mean average precision (mAP) as in [57], [58], [76] for 1,351 image pairs from [25]. Fig. 18 visualizes the object detection results, and Table 10 shows quantitative evaluations. These experiments demonstrate that our FCSS and CAT-FCSS can be used for the non-parameteric object detection task.

4.4 Runtime Analysis

In Table 11, we compared the computational speed of FCSS and CAT-FCSS to state-of-the-art descriptors including handcrafted and CNN-based descriptors. For the CNN-based methods including VGG [18], FCSS, and CAT-FCSS, the computation times were also measured on a GPU. Even though our descriptors need more computation compared to the handcrafted descriptors such as DAISY [11] on a CPU, they exhibit clearly better matching performance.

5 Conclusion

We presented the FCSS descriptor, which formulates local self-similarity within a fully convolutional network. In contrast to the previous LSS-based techniques, the sampling patterns and the self-similarity measure were jointly learned within the proposed network in an end-to-end and multi-scale manner. Furthermore, to address affine deformations in dense semantic correspondence, we proposed the CAT layer that first estimates explicit affine transformation fields at each pixel and then transforms the sampling patterns and corresponding receptive fields. The network was additionally trained in a weakly-supervised manner, using correspondence consistency within object bounding boxes provided in the training image pairs.

Even though our descriptor has shown reliable performance on various benchmarks and applications, it has some limitations. Since our descriptor can be learned with sailent-object database, it might not estimate correspondences of the background well as shown in Fig. 10, which would limit the applicability of the descriptor for scene-level correspondence. Furthermore, the computation time of our descriptor does not allow it to be used in real-time applications. To overcome these, our further works will be focused on the generalizaion of our descriptor to deal with scene-level matching and more efficient computation.

Acknowledgments

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069370). The work of B. Ham was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2017R1C1B2005584).

References

[1] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *In: SIGGRAPH*, 2011.
[2] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 815–830, 2011.
[3] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," *In: CVPR*, 2013.
[4] H. Yang, W. Y. Lin, and J. Lu, "Daisy filter flow: A generalized discrete approach to dense correspondences," *In: CVPR*, 2014.
[5] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros, "Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences," *In: CVPR*, 2015.
[6] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1, pp. 7–42, 2002.
[7] J. Lu, H. Yang, D. Min, and M. N. Do, "Patchmatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," *In: CVPR*, 2013.
[8] D. Butler, J. Wulff, G. Stanley, and M. Black, "A naturalistic open source movie for optical flow evaluation," *In: ECCV*, 2012.
[9] D. Sun, S. Roth, and M. J. Black, "Secret of optical flow estimation and their principles," *In: CVPR*, 2010.
[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[11] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. PAMI*, vol. 32, no. 5, pp. 815–830, 2010.
[12] T. Taniai, S. N. Sinha, and Y. Sato, "Joint recovery of dense correspondence and cosegmentation in two images," *In: CVPR*, 2016.
[13] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," *In: CVPR*, 2016.
[14] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," *In: CVPR*, 2015.
[15] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," *In: CVPR*, 2015.
[16] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," *In: ICCV*, 2015.
[17] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," *In: ECCV*, 2016.
[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *In: ICLR*, 2015.
[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *In: CVPR*, 2015.
[20] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," *In: CVPR*, 2015.
[21] E. Schechtman and M. Irani, "Matching local self-similarities across images and videos," *In: CVPR*, 2007.
[22] K. Chatfield, J. Philbin, and A. Zisserman, "Efficient retrieval of deformable shape classes using local self-similarities," *In: ICCV Workshop*, 2009.
[23] S. Kim, D. Min, B. Ham, S. Ryu, M. N. Do, and K. Sohn, "Dasc: Dense adaptive self-correlation descriptor for multi-modal and multi-spectral correspondence," *In: CVPR*, 2015.
[24] S. Kim, D. Min, S. Lin, and K. Sohn, "Deep self-correlation descriptor for dense cross-modal correspondence," *In: ECCV*, 2016.
[25] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE Trans. PAMI*, 2017.
[26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., 2011.
[27] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasum, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," *In: CVPR*, 2014.
[28] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. PAMI*, vol. 28, no. 4, pp. 594–611, 2006.
[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisseman, "The pascal visual object classes (voc) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[30] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn, "Fcss: Fully convolutional self-similarity for dense semantic correspondence," *In: CVPR*, 2017.
[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *In: CVPR*, 2005.
[32] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief : Computing a local binary descriptor very fast," *IEEE Trans. PAMI*, vol. 34, no. 7, pp. 1281–1298, 2011.
[33] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional acitivation features," *In: ECCV*, 2014.
[34] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: A comparison to sift," *arXiv:1405.5769*, 2014.
[35] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *In: ICML*, 2014.
[36] J. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" *In: NIPS*, 2014.
[37] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: Dsp-sift," *In: CVPR*, 2015.
[38] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," *In: ICCV*, 2011.
[39] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," *In: CVPR*, 2012.
[40] H. Bristow, J. Valmadre, and S. Lucey, "Dense semantic correspondence where every pixel is a classifier," *In: ICCV*, 2015.
[41] K. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *In: NIPS*, 2012.
[42] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros, "Learning dense correspondence via 3d-guided cycle consistency," *In: CVPR*, 2016.
[43] Online., http://www.shapenet.org/.
[44] C. B. Choy, Y. Gwak, and S. Savarese, "Universal correspondence network," *In: NIPS*, 2016.
[45] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On sifts and their scales," *In: CVPR*, 2012.
[46] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu, "Scale-space sift flow," *In: WACV*, 2014.
[47] J. Hur, H. Lim, C. Park, and S. C. Ahn, "Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation," *In: CVPR*, 2015.
[48] M. Tau and T. Hassner, "Dense correspondences across scenes and scales," *IEEE Trans. PAMI*, vol. 38, no. 5, pp. 875–888, 2016.
[49] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," *In: ECCV*, 2010.
[50] S. Kim, D. Min, S. Lin, and K. Sohn, "Dctm: Discrete-continuous transformation matching for semantic flow," *In: ICCV*, 2017.
[51] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, "Ti-pooling: Transformation-invariant pooling for feature learning in convolutional neural networks," *In: CVPR*, 2016.
[52] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *In: NIPS*, 2015.
[53] C. H. Lin and S. Lucey, "Inverse compositional spatial transformer networks," *In: CVPR*, 2017.
[54] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *In: ICCV*, 2017.
[55] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. IP*, vol. 7, no. 3, pp. 421–432, 1998.
[56] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *JMLR*, vol. 17, no. 1, pp. 2287–2318, 2016.
[57] R. Girshick, "Fast r-cnn," *In: ICCV*, 2015.
[58] S. Ren, K. He, R. Girchick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. PAMI*, vol. 39, no. 6, pp. 1137–1149, 2017.
[59] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *In: ICML*, 2010.
[60] V. Garcia, E. Devreuve, F. Nielsen, and M. Barlaud, "K-nearest neighbor search: Fast gpu-based implementations and application to high-dimensional feature matching," *In: ICIP*, 2010.
[61] Online., http://www.vlfeat.org/matconvnet/.
[62] H. O. Song, Y. Xiang, S. Jegelk, and S. Savarese, "Deep metric learing via lifted structured feature embedding," *In: CVPR*, 2016.
[63] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. M. Noguer, "Dense segmentation-aware descriptors," *In: CVPR*, 2013.
[64] W. Y. Lin, L. Liu, Y. Matsushita, K. L. Low, and S. Liu, "Aligning images in the wild," *In: CVPR*, 2012.
[65] A. Kanazawa, D. W. Jacobs, and M. Chandraker, "Warpnet: Weakly supervised matching for single-view reconstruction," *In: CVPR*, 2016.
[66] Y. L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis, "Jointly optimizing 3d model fitting and fine-grained classification," *In: ECCV*, 2014.
[67] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," *In: CVPR*, 2013.
[68] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," *In: ICCV*, 2011.
[69] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations,," *In: ICCV*, 2009.
[70] S. Saleem and R. Sablatnig, "A robust sift descriptor for multispectral images," *IEEE SPL*, vol. 21, no. 4, pp. 400–403, 2014.
[71] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," *In: WACV*, 2014.
[72] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.
[73] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. PAMI*, vol. 33, no. 12, pp. 2368–2382, 2011.
[74] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, vol. 42, no. 3, pp. 145–175, 2001.
[75] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *In: CVPR*, 2006.
[76] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. PAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.

Seungryong Kim received the B.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2012 and 2018, respectively. He is currently a Post-Doctoral Researcher in Electrical and Electronic Engineering at Yonsei University. His current research interests include 2D/3D computer vision, computational photography, and machine learning, in particular, sparse/dense feature descriptor and continuous/discrete optimization.

Dongbo Min received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2003, 2005, and 2009, respectively. He was a Post-Doctoral Researcher with the Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, from 2009 to 2010. From 2010 to 2015, he was with the Advanced Digital Sciences Center, Singapore. From 2015 to 2018, he was an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon, South Korea. Since 2018, he has been an Assistant Professor with the Department of Computer Science and Engineering, Ewha Womans University, Seoul, South Korea. His current research interests include computer vision, 2D/3D video processing, computational photography, augmented reality, and continuous/discrete optimization.

Bumsub Ham is an an Assistant Professor of Electrical and Electronic Engineering at Yonsei University in Seoul, Korea. He received the B.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University in 2008 and 2013, respectively. From 2014 to 2016, he was Post-Doctoral Research Fellow with Willow Team of INRIA Rocquencourt, École Normale Supérieure de Paris, and Centre National de la Recherche Scientifique. His research interests include computer vision, computational photography, and machine learning, in particular, regularization and matching, both in theory and applications.

Stephen Lin received the B.S.E. degree in electrical engineering from Princeton University, NJ, and the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor. He is a Principal Researcher at Microsoft Research Asia. His research interests include computer vision, image processing, and computer graphics. He is on the editorial board of the International Journal of Computer Vision and served as a Program Co-Chair of the International Conference on Computer Vision 2011.

Kwanghoon Sohn received the B.E. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1983, the M.S.E.E. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1985, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 1992. He was a Senior Member of the Research engineer with the Satellite Communication Division, Electronics and Telecommunications Research Institute, Daejeon, Korea, from 1992 to 1993, and a Post-Doctoral Fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, USA, in 1994. He was a Visiting Professor with Nanyang Technological University, Singapore, from 2002 to 2003. He is currently an Underwood Distinguished Professor with the School of Electrical and Electronic Engineering, Yonsei University. His research interests include 3D image processing and computer vision.