# SCNet: Learning Semantic Correspondence

Kai K Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia S Schmid, Jean S Ponce

# SCNet: Learning Semantic Correspondence

Kai Han[1] Rafael S. Rezende[4,5] Bumsub Ham[2] Kwan-Yee K. Wong[1]
Minsu Cho[3] Cordelia Schmid[4,*] Jean Ponce[5,4]
[1]The University of Hong Kong [2]Yonsei Univ. [3]POSTECH [4]Inria
[5]Department of Computer Science, ENS / CNRS / PSL Research University

## Abstract

*This paper addresses the problem of establishing* semantic correspondences *between images depicting different instances of the same object or scene category. Previous approaches focus on either combining a spatial regularizer with hand-crafted features, or learning a correspondence model for appearance only. We propose instead a convolutional neural network architecture, called* SCNet, *for learning a* geometrically *plausible model for semantic correspondence. SCNet uses region proposals as matching primitives, and explicitly incorporates geometric consistency in its loss function. It is trained on image pairs obtained from the PASCAL VOC 2007 keypoint dataset, and a comparative evaluation on several standard benchmarks demonstrates that the proposed approach substantially outperforms both recent deep learning architectures and previous methods based on hand-crafted features.*

## 1. Introduction

Our goal in this paper is to establish *semantic correspondences* across images that contain different instances of the same object or scene category, and thus feature much larger changes in appearance and spatial layout than the pictures of the *same* scene used in *stereo vision*, which we take here to include broadly not only classical (narrow-baseline) stereo fusion (e.g., [31, 34]), but also optical flow computation (e.g., [15, 33, 42]) and wide-baseline matching (e.g., [30, 43]). Due to such a large degree of variations, the problem of semantic correspondence remains very challenging. Most previous approaches to semantic correspondence [2, 17, 20, 26, 37, 43] focus on combining an effective spatial regularizer with hand-crafted features such as SIFT [28], DAISY [39] or HOG [6]. With the remarkable success of deep learning approaches in visual recognition, several learning-based methods have also been proposed for both stereo vision [9, 12, 47, 48] and semantic correspondence [5, 21, 50]. Yet, none of these methods exploits the geometric consistency constraints that have proven to be a key factor to the success of their hand-crafted counterparts. Geometric regularization, if any, occurs during post-processing but not during learning (e.g., [47, 48]).

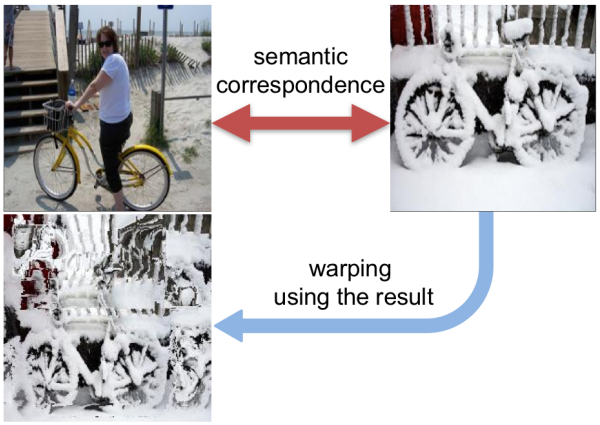


Figure 1: Learning semantic correspondence. We propose a convolutional neural network, SCNet, to learn semantic correspondence using both appearance and geometry. This allows us to handle a large degree of intra-class and scene variations. This figure shows a pair of input images (top) and a warped image (bottom) using its semantic correspondence by our method. (Best viewed in color.)

In this paper we propose a convolutional neural network (CNN) architecture, called *SCNet*, for learning geometrically plausible semantic correspondence (Figure 1). Following the *proposal flow* approach to semantic correspondence of Ham *et al*. [11], we use object proposals [29, 40, 51] as matching primitives, and explicitly incorporate the geometric consistency of these proposals in our loss function. Unlike [11] with its hand-crafted features, however, we train our system in an end-to-end manner using image pairs extracted from the PASCAL VOC 2007 keypoint dataset [7]. A comparative evaluation on several

*Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.

standard benchmarks demonstrates that the proposed approach substantially outperforms both recent deep architectures and previous methods based on hand-crafted features.

Our main contributions can be summarized as follows:

- We introduce a simple and efficient model for learning to match regions using both appearance and geometry.
- We propose a convolutional neural network, SCNet, to learn semantic correspondence with region proposals.
- We achieve state-of-the-art results on several benchmarks, clearly demonstrating the advantage of learning both appearance and geometric terms.

## 2. Related work

Here we briefly describe representative approaches related to semantic correspondence.

**Semantic correspondence.** SIFT Flow [26] extends classical optical flow to establish correspondences across similar but different scenes. It uses dense SIFT descriptors to capture semantic information beyond naive color values, and leverages a hierarchical optimization technique in a coarse-to-fine pipeline for efficiency. Kim *et al*. [20] and Hur *et al*. [17] propose more efficient generalizations of SIFT Flow. Instead of using SIFT features, Yang *et al*. [43] use DAISY [39] for an efficient descriptor extraction. Inspired by an exemplar-LDA approach [13], Bristow *et al*. [2] use whitened SIFT descriptors, making semantic correspondence robust to background clutter. Recently, Ham *et al*. [11] introduces proposal flow that uses object proposals as matching elements for semantic correspondence robust to scale and clutter. This work shows that the HOG descriptor gives better matching performance than deep learning features [23, 35]. Taniai *et al*. [37] also use HOG descriptors, and show that jointly performing cosegmentation and establishing dense correspondence are helpful in both tasks. Despite differences in feature descriptors and optimization schemes, these semantic correspondence approaches use a spatial regularizer to ensure flow smoothness on top of hand-crafted or pre-trained features.

**Deep learning for correspondence.** Recently, CNNs have been applied to classical dense correspondence problems such as optical flow and stereo matching to learn feature descriptors [46, 47, 48] or similarity functions [12, 46, 47]. FlowNet [9] uses an end-to-end scheme to learn optical flow with a synthetic dataset, and several recent approaches also use supervision from reconstructed 3D scenes and stereo pairs [12, 46, 47, 48]. MC-CNN [47] and its efficient extension [48] train CNN models to predict how well two image patches match and use this information to compute the stereo matching cost. DeepCompare [46] learns a similarity function for patches directly from images of a 3D scene, which allows for various types of geometric and photometric transformations (e.g., rotation and illumination changes). These approaches are inherently limited to matching images of the same physical object/scene. In contrast, Long *et al*. [27] use CNN features pre-trained for ImageNet classification tasks (due to a lack of available datasets for learning semantic correspondence) with performance comparable to SIFT flow. To overcome the difficulty in obtaining ground truth for semantic correspondence, Zhou *et al*. [50] leverage 3D models, and uses flow consistency between 3D models and 2D images as a supervisory signal to train a CNN. Another approach to generating ground truth is to directly augment the data by densifying sparse keypoint annotations using warping [11, 18]. The universal correspondence network (UCN) of Choy *et al.* [5] learns semantic correspondence using an architecture similar to [48], but adds a convolutional spatial transformer networks for improved robustness to rotation and scale changes. Kim *et al*. [21] introduce a convolutional descriptor using self-similarity, called fully convolutional self-similarity (FCSS), and combine the learned semantic descriptors with the proposal flow [11] framework. These approaches to learning semantic correspondence [5, 50] or semantic descriptors [21] typically perform better than traditional hand-crafted ones. Unlike our method, however, they do not incorporate geometric consistency between regions or object parts in the learning process.

## 3. Our approach

We consider the problem of learning to match regions with arbitrary positions and sizes in pairs of images. This setting is general enough to cover all cases of region sampling used in semantic correspondence: sampling a dense set of regular local regions as in typical dense correspondence [2, 20, 26, 38] as well as employing multi-scale object proposals [1, 16, 29, 40, 51]. In this work, following proposal flow [11], we focus on establishing correspondences between object proposal boxes.

### 3.1. Model

Our basic model for matching starts from the probabilistic Hough matching (PHM) approach of [4, 11]. In a nutshell, given some potential match $m$ between two regions, and the supporting data $D$ (a set of potential matches), the PHM model can be written as

$$\begin{aligned} P(m|D) &= \sum_x P(m|x,D)P(x|D) \\ &= P_a(m)\sum_x P_g(m|x)P(x|D), \end{aligned} \tag{1}$$

where $x$ is the offset (*e.g.*, position and scale change) between all potential matches $m = [r, s]$ of two regions $r$ and $s$. $P_a(m)$ is the probability that the match between two

regions is correct based on appearance only, and $P_g(m|x)$ is the probability based on geometry only, computed using the offset $x$[1]. PHM computes a matching score by replacing geometry prior $P(x|D)$ with the Hough voting $h(x|D)$ [4]:

$$h(x|D) = \sum_{m' \in D} P_a(m') P_g(m'|x). \tag{2}$$

This turns out to be an effective spatial matching model that combines appearance similarity with global geometric consistency measured by letting all matches vote on the potential offset $x$ [4, 11].

In our learning framework, we consider similarities rather than probabilities, and rewrite the PHM score for the match $m$ as

$$\begin{aligned} z(m,w) &= f(m,w) \sum_x g(m,x) \sum_{m' \in D} f(m',w) g(m',x) \\ &= f(m,w) \sum_{m' \in D} [\sum_x g(m,x) g(m',x)] f(m',w), \end{aligned} \tag{3}$$

where $f(m,w)$ is a parameterized appearance similarity function between the two regions in the potential match $m$, $x$ is as before an offset variable (position plus scale), and $g(m,x)$ measures the geometric compatibility between the match $m$ and the offset $x$.

Now assuming that we have a total number of $n$ potential matches, and identifying matches with their indices, we can rewrite this score as

$$\begin{aligned} z(m,w) &= f(m,w) \sum_{m'} K_{mm'} f(m',w), \\ \text{where } K_{mm'} &= \sum_x g(m,x) g(m',x), \end{aligned} \tag{4}$$

and the $n \times n$ matrix $K$ is the *kernel matrix* associated with the feature vector $\varphi(m) = [g(m,x_1), \ldots, g(m,x_s)]^T$, where $x_1$ to $x_s$ form the finite set of values that the offset variable $x$ runs over: indeed $K_{mm'} = \varphi(m) \cdot \varphi(m')$.[2]

Given training pairs of images with associated true and false matches, we can learn our similarity function by minimizing with respect to $w$

$$E(w) = \sum_{m=1}^{n} l[y_m, z(m,w)] + \lambda \Omega(w), \tag{5}$$

where $l$ is a loss function, $y_m$ is the the ground-truth label (either 1 [true] or 0 [false]) for the match $m$, and $\Omega$ is a regularizer (*e.g.*, $\Omega(w) = ||w||^2$). We use the hinge loss and $L_2$ regularizer in this work. Finally, at test time, we associate any region $r$ with the region $s$ maximizing $z([r,s], w^*)$, where $w^*$ is the set of learned parameters.

[1] We suppose that appearance matching is independent of geometry matching and the offset.

[2] Putting it all together in an $n$-vector of scores, this can also be rewritten as $z(w) = f(w) \odot K f(w)$, where $z(w) = (z(1,w), \ldots, z(n,w))^T$, "$\odot$" stands for the elementwise product between vectors, and $f(w) = (f(1,w), \ldots, f(n,w))^T$.

### 3.2. Similarity function and geometry kernel

There are many possible choices for the function $f$ that computes the appearance similarity of the two regions $r$ and $s$ making up match number $m$. Here we assume a trainable embedding function $c$ (as will be shown later, $c$ will be the output of a CNN in our case) that outputs a $L_2$ normalized feature vector. For the appearance similarity between two regions $r$ and $s$, we then use a rectified cosine similarity:

$$f(m,w) = \max(0, c(r,w) \cdot c(s,w)), \tag{6}$$

that sets all negative similarity values to zero, thus making the similarity function sparser as well as insensitive to negative matches during training, with the additional benefit of giving nonnegative weights in Eq. (3).

Our geometry kernel $K_{mm'}$ records the fact that two matches (roughly) correspond to the same offset: Concretely, we discretize the set of all possible offsets into bins. Let us denote by $h$ the function mapping a match $m$ onto the corresponding bin $x$, we now define $g$ by

$$g(m,x) = \begin{cases} 1, & \text{if } h(m) = x \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Thus, the kernel $K_{mm'}$ simply measures whether two matches share the same offset bin or not:

$$K_{mm'} = \begin{cases} 1, & \text{if } h(m) = h(m') \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

In practice, $x$ runs over a grid of predefined offset values, and $h(m)$ assigns match $m$ to the nearest offset point. Our kernel is sparse, which greatly simplifies the computation of the score function in Eq. (4): Indeed, let $B_x$ denote the set of matches associated with the bin $x$, the score function $z$ reduces to

$$z(m,w) = f(m,w) \sum_{m' \in B_{h(m)}} f(m',w). \tag{9}$$

This trainable form of the PHM model from [4, 11] can be used within Eq. (5).

Note that since our simple geometry kernel is only dependent on matches' offsets, we obtain the same geometry term value of $\sum_{m' \in B_{h(m)}} f(m',w)$ for any match $m$ that falls into the same bin $h(m)$. This allows us to compute this geometry term value only once for each non-empty bin $x$ and then share it for multiple matches in the same bin. This sharing makes computing $z$ several times faster in practice.[3]

### 3.3. Gradient-based learning

The feature embedding function $c(m,w)$ in the model above can be implemented by any differentiable architecture, for example a CNN-based one, and the score function

[3] If the geometry kernel is dependent on something other than offsets, e.g., matches' absolute position or their neighborhood structure, this sharing is not possible.

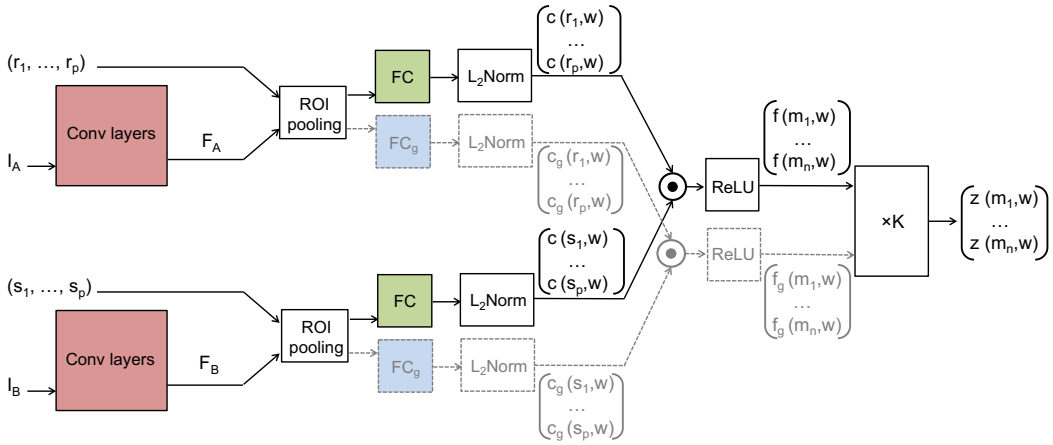


Figure 2: The SCNet architectures. Three variants are proposed: SCNet-AG, SCNet-A, and SCNet-AG+. The basic architecture, SCNet-AG, is drawn in solid lines. Colored boxes represent layers with learning parameters and the boxes with the same color share the same parameters. "$\times K$" denotes the voting layer for geometric scoring. A simplified variant, SCNet-A, learns appearance information only by making the voting layer an identity function. An extended variant, SCNet-AG+, contains an additional stream drawn in dashed lines. SCNet-AG learns a single embedding $c$ for both appearance and geometry, whereas SCNet-AG+ learns an additional and separate embedding $c_g$ for geometry. See text for details. (Best viewed in color.)

$z$ can be learned using stochastic gradient descent. Let us now consider the problem of minimizing the objective function $E(w)$ defined by Eq. (5).[4] This requires computing the gradient with respect to $w$ of the function $z$:

$$\nabla z(m,w) = [\sum_{m' \in D} K_{mm'} f(m',w)] \nabla f(m,w) + f(m,w) \sum_{m' \in D} K_{mm'} \nabla f(m',w). \quad (10)$$

Denoting by $n$ the size of $D$, this involves $n$ evaluations of both $f$ and $\nabla f$. Computing the full gradient of $E$ thus requires at most $n^2$ evaluations of both $f$ and $\nabla f$, which becomes computationally intractable when $n$ is large enough. The score function of Eq. (9) with the sparse kernel of Eq. (8), however, greatly reduces the gradient computation:

$$\nabla z(m,w) = [\sum_{m' \in B_{h(m)}} f(m',w)] \nabla f(m,w) + f(m,w) \sum_{m' \in B_{h(m)}} \nabla f(m',w). \quad (11)$$

Note that computing the gradient for match $m$ involves only a small set of matches falling into the same offset bin $h(m)$.

## 4. SCNet architecture

Among many possible architectures implementing the proposed model, we propose using a convolutional neural network (CNN), dubbed *SCNet*, that efficiently processes regions and learns our matching model. Three variants, SCNet-AG, SCNet-A, SCNet-AG+, are illustrated in Fig. 2.

In each case, SCNet takes as input two images $I_A$ and $I_B$, and maps them onto feature maps $F_A$ and $F_B$ by CNN layers. Given region proposals $(r_1, \ldots, r_p)$ and $(s_1, \ldots, s_p)$ for the two images, parallel ROI pooling layers [10, 14] extract feature maps of the same size for each proposal. This is an efficient architecture that shares convolutional layers over all region proposals.

**SCNet-AG.** The proposal features are fed into a fully-connected layer, mapped onto feature embedding vectors, and normalized into unit feature vectors $c(r_i, w)$ and $c(s_j, w)$, associated with the regions $r_i$ and $s_j$ of $I_A$ and $I_B$, respectively. The value of $f(m,w)$ for the match $m$ associated with regions $r_i$ and $s_j$ is computed as the rectified dot product of $c(r_i)$ and $c(s_j)$ (Eq. (6)), which defines the appearance similarity $f(m,w)$ for match $m$. Geometric consistency is enforced with the kernel described in Sec. 3.2, using a voting layer, denoted as "$\times K$", that computes score $z(m,w)$ from the appearance similarity and geometric consensus of proposals. Finally, matching is performed by identifying the maximal $z(m,w)$ scores, using both appearance and geometric similarities.

**SCNet-A.** We also evaluate a similar architecture without the geometry term. This architecture drops the voting layer (denoted by $\times K$ in Fig. 2) from SCNet-AG, directly using $f(m,w)$ as a score function. This is similar to the universal correspondence network (UCN) [5]. The main differences are the use of object proposals and the use of a different loss function.

[4] We take $\Omega(w) = 0$ for simplicity in this section, but tackling a nonzero regularizer is easy.

**SCNet-AG+.** Unlike SCNet-AG, which learns a single embedding $c$ for both appearance and geometry, SCNet-AG+ learns an additional and separate embedding $c_g$ for geometry that is implemented by an additional stream in the SCNet architecture (dashed lines in Fig. 2). This corresponds to a variant of Eq. (9), as follows:

$$z^+(m,w) = f(m,w) \sum_{m' \in B_{h(m)}} f_g(m',w), \qquad (12)$$

where $f_g$ is the rectified cosine similarity computed by $c_g$. Compared to the original score function, this variant allows the geometry term to learn a separate embedding function for geometric scoring. This may be beneficial particularly when a match's contribution to geometric scoring needs to be different from the appearance score. For example, a match of rigid object parts (wheel of cars) may contribute more to geometric scoring than that of deformable object parts (leg of horses). The separate similarity function $f_g$ allows more flexibility in learning the geometric term.

**Implementation details.** We use the VGG16 [36] model that consists of a set of convolutional layers with $3 \times 3$ filters, a ReLU layer and a pooling layer. We find that taking the first 4 convolutional layers is a good trade-off for our semantic feature extraction purpose without loosing localization accuracy. These layers output features with 512 channels. For example, if the net takes input of $224 \times 224 \times 3$ images, the convolutional layers produce features with the size of $14 \times 14 \times 512$. For the ROI pooling layer, we choose a $7 \times 7$ filter following the fast R-CNN architecture [10], which produces a feature map with size of $7 \times 7 \times 512$ for each proposal. To transform the feature map for each proposal into a feature vector, we use the $FC$ layer with a size of $7 \times 7 \times 512 \times 2048$. The 2048 dimensional feature vector associated with each proposal are then fed into the $L_2$ normalization layer, followed by the dot product layer, ReLU, our geometric voting layer, and loss layer. The convolutional layers are initialized by the pretrained weights of VGG16 and the fully connected layers have random initialization. We train our SCNet by mini-batch SGD, with learning rate $0.001$, and weight decay $0.0005$. During training, each mini-batch arises from a pair of images associated with a number of proposals. In our implementation, we generated 500 proposals for each image, which leads to $500 \times 500$ potential matches.

For each mini-batch, we sample matches for training as follows. (1) Positive sampling: For a proposal $r_i$ in $I_A$, we are given its ground truth match $r_i'$ in $I_B$. We pick all the proposals $s_j$ in $I_B$ with $\text{IoU}(s_j, r_i') > T_{pos}$ to be positive matches for $r_i$. (2) Negative sampling: Assume we obtain $k$ positive pairs w.r.t $r_i$. We also need to have $k$ negative pairs w.r.t $r_i$. To achieve this, we first find the proposals $s_t$ in $I_B$ with $IoU(s_t, r_i') < T_{neg}$. Assuming $p$ proposals satisfying the IoU constraint, we find the proposals with top $k$ appearance similarity with $r_i$ among those $p$ proposals. In our experiment, we set $T_{pos} = 0.6$, and $T_{neg} = 0.4$.

## 5. Experimental evaluation

In this section we present experimental results and analysis. Our code and models will be made available online: http://www.di.ens.fr/willow/research/scnet/.

### 5.1. Experimental details

**Dataset.** We use the PF-PASCAL dataset that consists of 1300 image pairs selected from PASCAL-Berkeley keypoint annotations[5] of 20 object classes. Each pair of images in PF-PASCAL share the same set of non-occluded keypoints. We divide the dataset into 700 training pairs, 300 validation pairs, and 300 testing pairs. The image pairs for training/validation/testing are distributed proportionally to the number of image pairs of each object class. In training, we augment the data into a total of 1400 pairs by horizontal mirroring. We also test our trained models with the PF-WILLOW dataset [11], Caltech-101 [8] and PASCAL Parts [49] to further validate a generalization of the models.

**Region proposal.** Unless stated otherwise, we choose to use the method of Manen *et al.* (RP) [29]. The use of RP proposals is motivated by the superior result reported in [11], which is verified once more by our evaluation. In testing we use 1000 proposals for each image as in [11], while in training we use 500 proposals for efficiency.

**Evaluation metric.** We use three metrics to compare the results of SCNet to other methods. First, we use the probability of correct keypoint (PCK) [44], which measures the precision of dense flow at sparse keypoints of semantic relevance. It is calculated on the Euclidean distance $d(\phi(p), p^*)$ between a warped keypoint $\phi(p)$ and ground-truth one $p^*$[6]. Second, we use the probability of correct regions (PCR) introduced in [11] as an equivalent of the the PCK for region based correspondence. PCR measures the precision of a region matching between region $r$ and its correspondent $r^*$ on the intersection over union (IoU) score $1 - \text{IoU}(\phi(r), r^*)$. Both metrics are computed against a threshold $\tau$ in $[0, 1]$ and we measure PCK@$\tau$ and PCR@$\tau$ as the percentage correct below $\tau$. Third, we capture the quality of matching proposals by the mean IoU of the top $k$ matches (mIoU@$k$). Note that these metrics are used to evaluate two different types of correspondence. Indeed, PCK is an evaluation metric for dense flow field, whereas PCR and mIoU@$k$ are used to evaluate region-based correspondences [11].

[5] http://www.di.ens.fr/willow/research/proposalflow/

[6] To better take into account the different sizes of images, we normalize the distance by dividing by the diagonal of the warped image, as in [5]

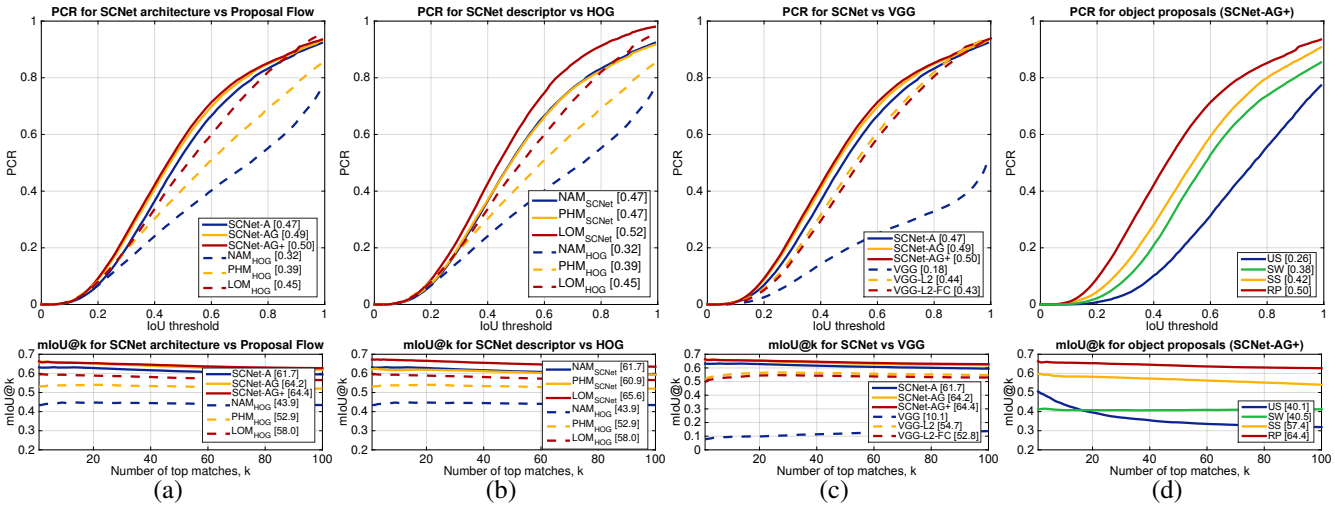


Figure 3: (a) Performance of SCNet on PF-PASCAL, compared to Proposal Flow methods [11]. (b) Performance of SCNet and HOG descriptors on PF-PASCAL, evaluated using Proposal Flow methods [11]. (c) Comparison to ImageNet-trained baselines. (d) Comparison of different proposals. PCR and mIoU@$k$ plots are shown at the top and bottom, respectively. AuC is shown in the legend. (Best viewed in pdf.)

### 5.2. Proposal flow components

We use the PF-PASCAL dataset to evaluate region matching performance. This setting allows our method to be tested against three other methods in [11]: NAM, PHM and LOM. NAM finds correspondences using handcrafted features only. PHM and LOM additionally consider global and local geometric consistency, respectively, between region matchings. We also compare our SCNet-learned feature against whitened HOG [6], the best performing handcraft feature of [11]. Experiments on the PF-WILLOW dataset [11] showed similar results with the ones on the PF-PASCAL dataset. For details, refer to our project webpage.

**Quantitative comparison.** Figure 3(a) compares SCNet methods with the proposal flow methods [11] on the PF-PASCAL dataset. Our SCNet models outperform the other methods that use the HOG feature. Our geometric models (SCNet-AG, SCNet-AG+) substantially outperform the appearance-only model (SCNet-A), and SCNet-AG+ slightly outperform SCNet-AG. This can also be seen from the area under curve (AuC) presented in the legend. This clearly show the effectiveness of deep learned features as well as geometric matching. In this comparison, we fix the VGG16 layer and only learn the FC layers. In our experiment, we also learned all layers including VGG 16 and the FC layers in our model (fully finetuned), but the improvement over the partially learned model was marginal. Figure 3(b) shows the performance of NAM, PHM, LOM matching when replacing HOG feature with our learned feature in SCNet-A. We see that SCNet features greatly improves all the matching methods. Interestingly, LOM using SCNet feature outperforms our best performing SCNet model, SCNet-AG+. However, the LOM method is more than 10 times slower than SCNet-AG+: on average the method takes $0.21s$ for SCNet-A feature extraction and $3.15s$ for the actual matching process, whereas our SCNet-AG+ only takes $0.33s$ in total. Most of the time in LOM is spent in computing its geometric consistency term. We further evaluated three additional baselines using ImageNet-trained VGG (see Figure 3(c)). Namely (i) VGG: We directly use the features from ImageNet-trained VGG, followed by ROI-pooling to make the features for each proposal of the same size ($7 \times 7 \times 512$). We then flatten the features into vectors of dimension 175616. (ii) VGG-L2: We l2-normalize the flattened feature of (i). (iii) VGG-L2-FC: We perform a random projection from (ii) to a feature of dimension 2048 (the same dimension with SCNet, 12.25 times smaller than (i) and (ii)) by adding a randomly initialized FC layer on top of (ii). Note that this is exactly equivalent to SCNet-A without training on the target dataset.

**Results with different object proposals.** SCNet can be combined with any region proposal methods. In this experiment, we train and evaluate SCNet-AG+ on PF-PASCAL with four region proposal methods: randomized prim (RP) [29], selective search (SS) [41], random uniform sampling (US), and sliding window (SW). US and SW are extracted using the work of [16], and SW is similar to regular grid sampling used in other popular methods [20, 26, 33]. Figure 3(d) compares matching performance in PCR and mIoU@$k$ when using the different proposals. RP performs best, and US performs worst with a large margin. This

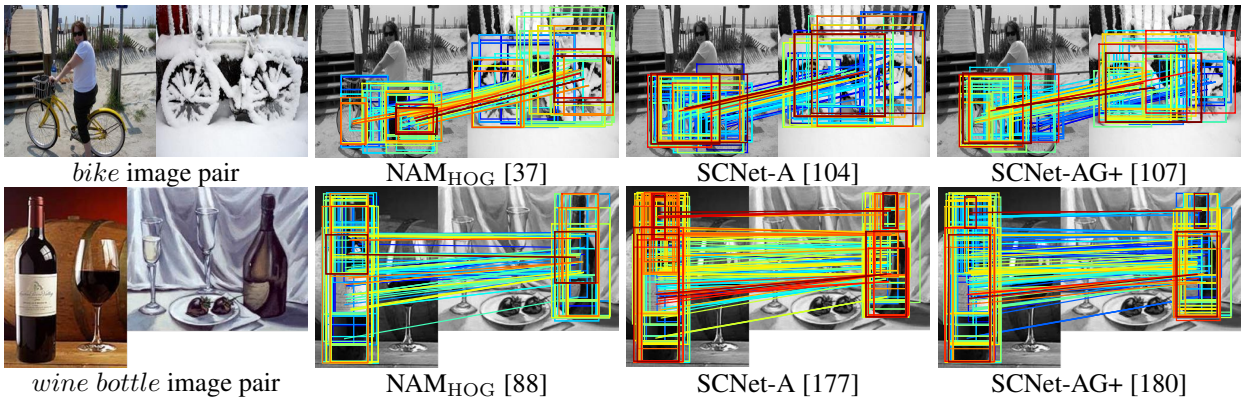


Figure 4: Region matching examples. Numbers beside methods stand for numbers of correct matches.

Table 1: Per-class PCK on PF-PASCAL at $\tau = 0.1$. For all methods using object proposals, we use 1000 RP proposals [29].

| Method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | d.table | dog | horse | moto | person | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NAM$_{\text{HOG}}$ [11] | 72.9 | 73.6 | 31.5 | 52.2 | 37.9 | 71.7 | 71.6 | 34.7 | 26.7 | 48.7 | 28.3 | 34.0 | 50.5 | 61.9 | 26.7 | 51.7 | 66.9 | 48.2 | 47.8 | 59.0 | 52.5 |
| PHM$_{\text{HOG}}$ [11] | 78.3 | 76.8 | 48.5 | 46.7 | 45.9 | 72.5 | 72.1 | 47.9 | 49.0 | 84.0 | 37.2 | 46.5 | 51.3 | 72.7 | 38.4 | 53.6 | 67.2 | 50.9 | 60.0 | 63.4 | 60.3 |
| LOM$_{\text{HOG}}$ [11] | 73.3 | 74.4 | 54.4 | 50.9 | 49.6 | 73.8 | 72.9 | 63.6 | 46.1 | 79.8 | 42.5 | 48.0 | 68.3 | 66.3 | 42.1 | 62.1 | 65.2 | 57.1 | 64.4 | 58.0 | 62.5 |
| UCN [5] | 64.8 | 58.7 | 42.8 | 59.6 | 47.0 | 42.2 | 61.0 | 45.6 | 49.9 | 52.0 | 48.5 | 49.5 | 53.2 | 72.7 | 53.0 | 41.4 | **83.3** | 49.0 | **73.0** | 66.0 | 55.6 |
| SCNet-A | 67.6 | 72.9 | 69.3 | 59.7 | **74.5** | 72.7 | 73.2 | 59.5 | 51.4 | 78.2 | 39.4 | 50.1 | 67.0 | 62.1 | **69.3** | **68.5** | 78.2 | 63.3 | 57.7 | 59.8 | 66.3 |
| SCNet-AG | 83.9 | 81.4 | **70.6** | 62.5 | 60.6 | 81.3 | 81.2 | 59.5 | 53.1 | 81.2 | **62.0** | 58.7 | 65.5 | 73.3 | 51.2 | 58.3 | 60.0 | 69.3 | 61.5 | **80.0** | 69.7 |
| SCNet-AG+ | **85.5** | **84.4** | 66.3 | **70.8** | 57.4 | **82.7** | **82.3** | **71.6** | **54.3** | **95.8** | 55.2 | **59.5** | **68.6** | **75.0** | 56.3 | 60.4 | 60.0 | **73.7** | 66.5 | 76.7 | **72.2** |

shows that the region proposal process is an important factor for matching performance.

**Qualitative comparison.** Region matching results for NAM, SCNet-A, and SCNet-AG+ are shown in Figure 4. In this example, at the IoU threshold 0.5, the numbers of correct matches are shown for all methods. We can see that SCNet models perform significantly better than NAM with HOG feature, and SCNet-A is outperformed by SCNet-AG+ that learns a geometric consistency term.

### 5.3. Flow field

Given a sparse region matching result and its corresponding scores, we generate dense semantic flow using a densifying technique in [11]. In brief, we select out a region match with the highest score, and assign dense correspondences to the pixels within the matched regions by linear interpolation. This process is repeated without replacement of the region match until we assign dense correspondences to all pixels in the source image. The results are evaluated on PF-PASCAL dataset. To evaluate transferability performance of the models, we also test them on other datasets such as PF-WILLOW [11], Caltech-101 [8] and PASCAL Parts [49] datasets, and compare with state-of-the-art results on these datasets. In these cases direct comparison between learning-based methods may not be fair in the sense that they are trained on different datasets.

**Results on PF-PASCAL.** We compare SCNet with Proposal Flow [11] and UCN [5] on the PF-PASCAL dataset, and summarize the result in Table 1. The UCN is retrained using the code provided by the authors on the PF-PASCAL dataset for fair comparison. Using the raw network of [5] trained on a different subset of PASCAL yields as expected lower performance, with a mean PCK of 36.0 as opposed to the 55.6 obtained for the retrained network. The three variants of SCNet do consistently better than UCN as well as all methods in [11], with a PCK of 66.3 or above. Among all the methods, SCNet-AG+ performs best with a PCK of 72.2. Figure 5 presents two examples of dense matching for PF-PASCAL. Ground truth are presented as circles and predicted keypoints are presented as crosses. We observe a better performance of SCNet-AG and SCNet-AG+.

**Results on PF-WILLOW.** For evaluating transferability, we test (PF-PASCAL trained) SCNet and UCN on the PF-WILLOW dataset [11] and compare the results with recent methods in Table 2 where PCK is averaged over all classes. The postfix 'w/SF' and 'w/PF' represent that matching is performed by SIFT Flow [26] and Proposal Flow [11], respectively. On this dataset where the data has a different distribution, SCNet-AG slightly outperforms the A and AG+ variants (PCK@0.05). We observe that all SCNet models significantly outperform UCN, which is trained on the same dataset with the SCNet models, as well as other methods

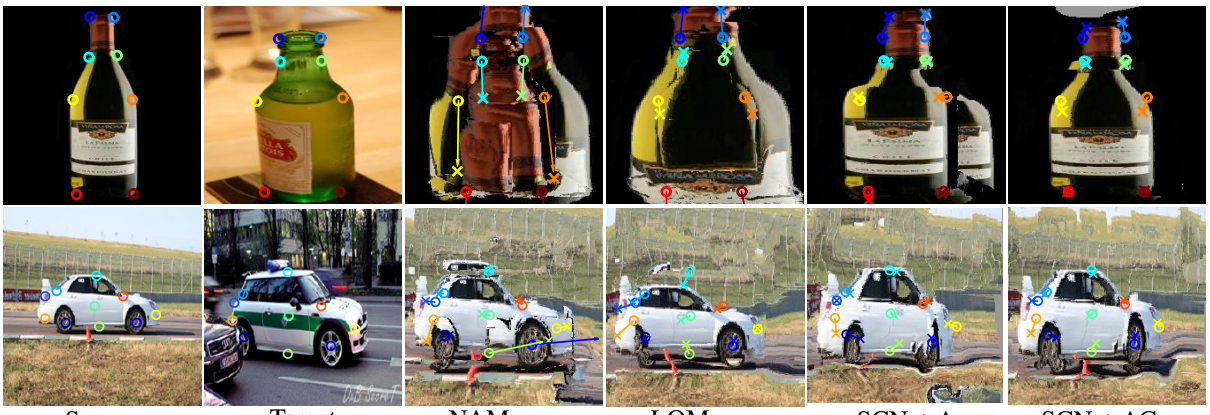


Figure 5: Quantitative comparison of dense correspondence. We show the keypoints of the target image in circles and the predicted keypoints of the source in crosses, with a vector that depicts the matching error. (Best viewed in pdf.)

Table 2: Fixed-threshold PCK on PF-WILLOW.

| Method | PCK@0.05 | PCK@0.1 | PCK@0.15 |
|---|---|---|---|
| SIFT Flow [26] | 0.247 | 0.380 | 0.504 |
| DAISY w/SF [43] | 0.324 | 0.456 | 0.555 |
| DeepC w/SF [46] | 0.212 | 0.364 | 0.518 |
| LIFT w/SF [45] | 0.224 | 0.346 | 0.489 |
| VGG w/SF [36] | 0.224 | 0.388 | 0.555 |
| FCSS w/SF [21] | 0.354 | 0.532 | 0.681 |
| FCSS w/PF [21] | 0.295 | 0.584 | 0.715 |
| LOM$_{\text{HOG}}$ [11] | 0.284 | 0.568 | 0.682 |
| UCN[5] | 0.291 | 0.417 | 0.513 |
| SCNet-A | 0.390 | **0.725** | **0.873** |
| SCNet-AG | **0.394** | 0.721 | 0.871 |
| SCNet-AG+ | 0.386 | 0.704 | 0.853 |

Table 3: Results on Caltech-101.

| Methods | LT-ACC | IoU | LOC-ERR |
|---|---|---|---|
| NAM$_{\text{HOG}}$ [11] | 0.70 | 0.44 | 0.39 |
| PHM$_{\text{HOG}}$ [11] | 0.75 | 0.48 | 0.31 |
| LOM$_{\text{HOG}}$ [11] | 0.78 | 0.50 | 0.26 |
| DeepFlow [33] | 0.74 | 0.40 | 0.34 |
| SIFT Flow [26] | 0.75 | 0.48 | 0.32 |
| DSP [20] | 0.77 | 0.47 | 0.35 |
| FCSS w/SF [21] | 0.80 | 0.50 | **0.21** |
| FCSS w/PF [21] | **0.83** | **0.52** | 0.22 |
| SCNet-A | 0.78 | 0.50 | 0.28 |
| SCNet-AG | 0.78 | 0.50 | 0.27 |
| SCNet-AG+ | 0.79 | 0.51 | 0.25 |

Table 4: Results on PASCAL Parts.

| Methods | IoU | PCK |
|---|---|---|
| NAM$_{\text{HOG}}$ [11] | 0.35 | 0.13 |
| PHM$_{\text{HOG}}$ [11] | 0.39 | 0.17 |
| LOM$_{\text{HOG}}$ [11] | 0.41 | 0.17 |
| Congealing [24] | 0.38 | 0.11 |
| RASL [32] | 0.39 | 0.16 |
| CollectionFlow [19] | 0.38 | 0.12 |
| DSP [20] | 0.39 | 0.17 |
| FCSS w/SF [21] | 0.44 | 0.28 |
| FCSS w/PF [21] | 0.46 | **0.29** |
| SCNet-A | 0.47 | 0.17 |
| SCNet-AG | 0.47 | 0.17 |
| SCNet-AG+ | **0.48** | 0.18 |

using hand-crafted features [26, 43, 22] and learned features [35, 46, 12, 45, 36, 21].

**Results on Caltech-101.** We also evaluate our approach on the Caltech-101 dataset [8]. Following the experimental protocol in [20], we randomly select 15 pairs of images for each object class, and evaluate matching accuracy with three metrics: Label transfer accuracy (LT-ACC) [25], the IoU metric, and the localization error (LOC-ERR) of corresponding pixel positions. Table 3 shows that SCNet achieves comparable results with the state of the art. The best performer, FCSS [21], is trained on images from the same Caltech-101 dataset, while SCNet models are not.

**Results on PASCAL Parts.** Following [11], we use the dataset provided by [49] where the images are sampled from the PASCAL part dataset [3]. For this experiment, we measure the weighted IoU score between transferred segments and the ground truth, with weights determined by the pixel area of each part. To evaluate alignment accuracy, we measure the PCK metric ($\alpha = 0.05$) using keypoint annotations for the PASCAL classes. Following [11] once again, we use selective search (SS) to generate proposals for SCNet in this experiment. The results are summarized in Table 4. SCNet models outperform all other results on the dataset in IoU, and SCNet-AG+ performs best among them. FCSS w/PF [21] performs better in PCK on this dataset.

These results verify that SCNet models have successfully learned semantic correspondence.

## 6. Conclusion

We have introduced a novel model for learning semantic correspondence, and proposed the corresponding CNN architecture that uses object proposals as matching primitives and learns matching in terms of appearance and geometry. The proposed method substantially outperforms both recent deep learning architectures and previous methods based on hand-crafted features. The result clearly demonstrates the effectiveness of learning geometric matching for semantic correspondence. In future work, we will explore better models and architectures to leverage geometric information.

**Acknowledgments.** This work was supported by the ERC grants VideoWorld and Allegro, the Institut Universitaire de France, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2017R1C1B2005584) as well as the MSIT (Ministry of Science and ICT), Korea, under the ICT Consilience Creative program (IITP-2017-R0346-16-1007). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X Pascal GPU used for this research. We also thank JunYoung Gwak and Christopher B. Choy for their help in comparing with UCN.

## References

[1] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2014. 2

[2] H. Bristow, J. Valmadre, and S. Lucey. Dense semantic correspondence where every pixel is a classifier. In *Proc. Int. Conf. Comp. Vision*, 2015. 1, 2

[3] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, et al. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2014. 8

[4] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 2, 3

[5] C. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *Proc. Neural Info. Proc. Systems*, 2016. 1, 2, 4, 5, 7, 8

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2005. 1, 6

[7] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 (voc2007) results. 1

[8] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Patt. Anal. Mach. Intell.*, 28(4):594–611, 2006. 5, 7, 8

[9] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 1, 2

[10] R. Girshick. Fast r-cnn. In *Proc. Int. Conf. Comp. Vision*, 2015. 4, 5

[11] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2016. 1, 2, 3, 5, 6, 7, 8

[12] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 1, 2, 8

[13] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Proc. European Conf. Comp. Vision*, pages 459–472. Springer, 2012. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. European Conf. Comp. Vision*, 2014. 4

[15] B. K. Horn and B. G. Schunck. Determining optical flow: A retrospective. *Artificial Intelligence*, 59(1):81–87, 1993. 1

[16] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *IEEE Trans. Patt. Anal. Mach. Intell.*, 2015. 2, 6

[17] J. Hur, H. Lim, C. Park, and S. C. Ahn. Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 1, 2

[18] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2016. 2

[19] I. Kemelmacher-Shlizerman and S. M. Seitz. Collection flow. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2012. 8

[20] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2013. 1, 2, 6, 8

[21] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2017. 1, 2, 8

[22] S. W. Kim, D. Min, B. Ham, and K. Sohn. Dasc: Dense adaptative self-correlation descriptor for multi-modal and multi-spectral correspondence. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 8

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Neural Info. Proc. Systems*, 2012. 2

[24] E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Trans. Patt. Anal. Mach. Intell.*, 28(2):236–250, 2006. 8

[25] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. Patt. Anal. Mach. Intell.*, 33(12):2368–2382, 2011. 8

[26] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. Patt. Anal. Mach. Intell.*, 33(5):978–994, 2011. 1, 2, 6, 7, 8

[27] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Proc. Neural Info. Proc. Systems*, 2014. 2

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Comp. Vision*, 60(2):91–110, 2004. 1

[29] S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized Prim's algorithm. In *Proc. Int. Conf. Comp. Vision*, 2013. 1, 2, 5, 6, 7

[30] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 1

[31] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(4):353–363, 1993. 1

[32] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for

linearly correlated images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 34(11):2233–2246, 2012. 8

[33] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deepmatching: Hierarchical deformable dense matching. *ArXiv e-prints*, 2015. 1, 6, 8

[34] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2011. 1

[35] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proc. Int. Conf. Comp. Vision*, 2015. 2, 8

[36] K. Simonyan and andrew Zisserman. Very deep convolutional networks for large-scale visual recognition. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2014. 5, 8

[37] T. Taniai, S. N. Sinha, and Y. Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2016. 1, 2

[38] M. Tau and T. Hassner. Dense correspondences across scenes and scales. *IEEE Trans. Patt. Anal. Mach. Intell.*, 2015. 2

[39] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Patt. Anal. Mach. Intell.*, 32(5):815–830, 2010. 1, 2

[40] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *Int. J. of Comp. Vision*, 104(2):154–171, 2013. 1, 2

[41] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *Int. J. of Comp. Vision*, 104(2):154–171, 2013. 6

[42] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proc. Int. Conf. Comp. Vision*, 2013. 1

[43] H. Yang, W.-Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2014. 1, 2, 8

[44] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Patt. Anal. Mach. Intell.*, 35(12):2878–2890, 2013. 5

[45] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *Proc. European Conf. Comp. Vision*, 2016. 8

[46] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 2, 8

[47] J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 1, 2

[48] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. 1, 2

[49] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2015. 5, 7, 8

[50] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2016. 1, 2

[51] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Proc. European Conf. Comp. Vision*, 2014. 1, 2