# Learning Semantic Correspondence Exploiting an Object-level Prior

Junghyup Lee, Dohyung Kim, Wonkyung Lee, Jean Ponce, Bumsub Ham

# Learning Semantic Correspondence Exploiting an Object-level Prior

Junghyup Lee, Dohyung Kim, Wonkyung Lee,
Jean Ponce, *Fellow, IEEE,* and Bumsub Ham, *Member, IEEE*

**Abstract**—We address the problem of semantic correspondence, that is, establishing a dense flow field between images depicting different instances of the same object or scene category. We propose to use images annotated with binary foreground masks and subjected to synthetic geometric deformations to train a convolutional neural network (CNN) for this task. Using these masks as part of the supervisory signal provides an object-level prior for the semantic correspondence task and offers a good compromise between semantic flow methods, where the amount of training data is limited by the cost of manually selecting point correspondences, and semantic alignment ones, where the regression of a single global geometric transformation between images may be sensitive to image-specific details such as background clutter. We propose a new CNN architecture, dubbed SFNet, which implements this idea. It leverages a new and differentiable version of the argmax function for end-to-end training, with a loss that combines mask and flow consistency with smoothness terms. Experimental results demonstrate the effectiveness of our approach, which significantly outperforms the state of the art on standard benchmarks.

**Index Terms**—Semantic correspondence, object-level prior, differentiable argmax function.

## 1 Introduction

Establishing dense correspondences across images is one of the fundamental tasks in computer vision [1], [2], [3]. Early works have focused on handling different views of the same scene (stereo matching [1], [4]) or successive frames (optical flow [2], [5]) in a video sequence. Semantic correspondence algorithms (e.g., SIFT Flow [3]) go one step further, finding a dense flow field between images depicting different instances of the same object or scene category, which has proven useful in various computer vision tasks including object recognition [3], [6], semantic segmentation [7], co-segmentation [8], image editing [9], and scene parsing [7], [10]. Establishing dense semantic correspondences is very challenging especially in the presence of large changes in appearance or scene layout and background clutter. Classical approaches to semantic correspondence [3], [7], [11], [12], [13] typically use an objective function involving fidelity and regularization terms. The fidelity term encourages hand-crafted features (e.g., SIFT [14], HOG [15], DAISY [16]) to be matched along a dense flow field between images, and the regularization term makes it smooth while aligning discontinuities to object boundaries. Hand-crafted features, however, do not capture high-level semantics (e.g., appearance and shape variations), and they are not robust to image-specific details (e.g., texture, background clutter, occlusion).

Convolutional neural networks (CNNs) have allowed remarkable advances in semantic correspondence in the past few years. Recent methods using CNNs [17], [18], [19], [20], [21], [22], [23], [24], [25], [26] benefit from rich semantic features invariant to intra-class variations, achieving state-of-the-art results. Semantic flow approaches [17], [18], [19], [21], [22] attempt to find correspondences for individual pixels or patches. They are not seriously affected by non-rigid deformations, but are easily distracted by background clutter. They also require a large amount of data with ground-truth correspondences for training. Although pixel-level semantic correspondences impose very strong constraints, manually annotating them is extremely labor-intensive and somewhat subjective, which limits the amount of training data available [27]. An alternative is to learn feature descriptors only [18], [19], [21] or to exploit 3D CAD models together with rendering engines [22]. Semantic alignment methods [20], [23], [24], [25], [26], on the other hand, formulate semantic correspondence as a geometric alignment problem and directly regress parameters of a global transformation model (e.g., affine deformation or thin plate spline) between images. They leverage self-supervised learning where ground-truth parameters are generated synthetically using random transformations with, however, a higher sensitivity to non-rigid deformations. Moreover, background clutter prevents focusing on individual objects and interferes with the estimation of the transformation parameters. To overcome this problem, recent methods reduce the influence of distractors by inlier counting [24] or an attention process [25].

In this paper, we present a new approach to establishing an object-aware semantic flow and propose to exploit binary foreground masks as a supervisory signal during training (Fig. 1). Our approach builds upon the insight that correspondences of high quality between images allow to segment common objects from background. To implement this idea, we introduce a new CNN architecture, dubbed SFNet, that outputs a semantic flow field at a sub-pixel level. We leverage a new and differentiable version of the argmax function, the kernel soft argmax, together with mask/flow consistency and smoothness terms to train SFNet end to end, establishing object-aware correspondences while filtering out distracting details. Our approach has the following advantages: First, it is a good compromise between current semantic flow

*The first two authors contributed equally.*

- *J. Lee, D. Kim, W. Lee, and B. Ham are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Korea.*
  *E-mail: {junghyup.lee, do.hyung, leewk92, bumsub.ham}@yonsei.ac.kr*
- *J. Ponce is with Inria and DI-ENS, Département d'Informatique de l'ENS, CNRS, PSL University, Paris 75005, France.*
  *E-mail: jean.ponce@inria.fr*

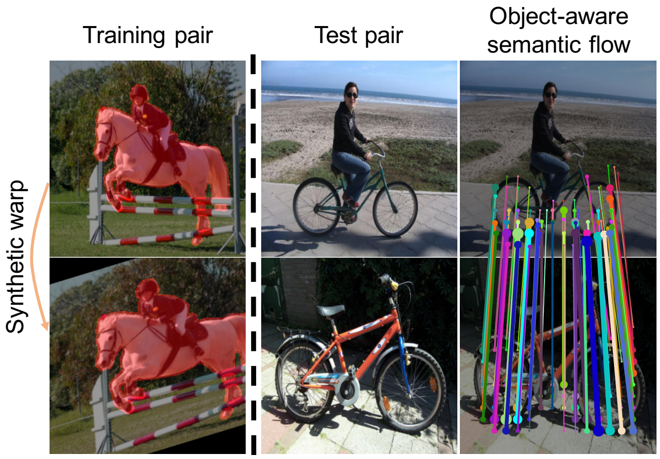


Fig. 1: We use pairs of warped foreground masks obtained from a single image (left) as a supervisory signal to train our model. This allows us to establish object-aware semantic correspondences across images depicting different instances of the same object or scene category (right). No masks are required at test time. (Best viewed in color.)

and alignment methods, since foreground masks are available for large datasets, and they provide an object-level prior for the semantic correspondence task. Exploiting these masks during training makes it possible to focus on learning correspondences between prominent objects and scene elements (masks are of course not used at test time). Second, our method establishes a dense non-parametric flow field (i.e., semantic flow), which is more robust to non-rigid deformations than parametric regression (i.e., semantic alignment). Finally, using the kernel soft argmax allows us to train the whole network end to end, and hence our approach further benefits from high-level semantics specific to the task of semantic correspondence. The main contributions of this paper can be summarized as follows:

- We propose to exploit binary foreground masks, that are widely available and can be annotated more easily than individual point correspondences, to learn semantic flow by incorporating an object-level prior in the learning task.
- We introduce a kernel soft argmax function, making our model quite robust to multi-modal distributions while providing a differentiable flow field at a sub-pixel level.
- We set a new state of the art on standard benchmarks for semantic correspondence, mask transfer, and pose keypoint propagation, clearly demonstrating the effectiveness of our approach. We also provide an extensive experimental analysis with ablation studies.

A preliminary version of this work appeared in [28]. This version adds (1) a detailed description of related works exploiting object priors for semantic correspondence; (2) an in-depth presentation of SFNet including the kernel soft argmax and loss terms; (3) more comparisons with the state of the art on different benchmarks including the TSS [8] and recent SPair-71k [29] datasets; (4) an evaluation on the task of pose keypoint propagation with the JHMDB dataset [30]; and (5) an extensive experimental evaluation including a runtime comparison and a performance analysis on SFNet trained using noisy labels (i.e., bounding boxes) or with different datasets. To encourage comparison and future work, our code and model are available online: https://cvlab-yonsei.github.io/projects/SFNet.

## 2 Related Work

Correspondence problems cover a broad range of topics in computer vision including stereo, motion analysis, object recognition and shape matching. Giving a comprehensive review on these topics is beyond the scope of this paper. We thus focus on representative works related to ours.

### 2.1 Semantic Flow

Classical approaches focus on finding sparse correspondences, e.g., for instance matching [14], or establishing dense matches between nearby views of the same scene/object, e.g., for stereo matching [1], [4] and optical flow estimation [2], [5]. Unlike these, semantic correspondence methods estimate dense matches across pictures containing different instances of the same object or scene category. Early works on semantic correspondence focus on matching local features from hand-crafted descriptors, such as SIFT [3], [7], [11], [12], DAISY [13] and HOG [8], [27], [31], together with spatial regularization using graphical models [3], [7], [8], [11] or random sampling [13], [32]. However, hand-crafting features capturing high-level semantics is extremely hard, and similarities between them are easily distracted, e.g., by clutter, texture, occlusion and appearance variations. There have been many attempts to estimate correspondences robust to background clutter or scale changes between objects/object parts. These use object proposals as candidate regions for matching [27], [31] or perform matching in scale space [33].

Recently, image features from CNNs have demonstrated a capacity to both representing high-level semantics and being robust to appearance and shape variations [34], [35], [36]. Long et al. [37] apply CNNs to establish semantic correspondences between images. They follow the same procedure as the SIFT Flow [3] method, but exploit off-the-shelf CNN features trained for the ImageNet classification task [38] due to a lack of training datasets with pixel-level annotations. This problem can be alleviated by synthesizing ground-truth correspondences from 3D models [22] or augmenting the number of match pairs in a sparse keypoint dataset using interpolation [8]. More recently, new benchmarks for semantic correspondence have been released. PF-PASCAL [39] provides 1300+ image pairs of 20 image categories with ground-truth annotations from the PASCAL 2011 keypoint dataset [40]. SPair-71k [29] consists of over 70k of image pairs from PASCAL 3D+ [41] and PASCAL VOC 2012 [42] with rich annotations including keypoints, segmentation masks and bounding boxes. These enable learning local features [17], [21], [29], [43] specific to the task of semantic correspondence. FCSS [21] introduces a learnable local self-similarity descriptor robust to intra-class variations. SCNet [17] and HPF [29] present region descriptors exploiting geometric consistency among object parts. NCN [43] analyzes neighborhood consensus patterns in the 4D space of all possible correspondences in order to find spatially consistent matches, disambiguating feature matches on repetitive patterns. Although these approaches using CNN features outperform early methods by large margins, the loss functions they use for training typically do not involve a spatial regularizer mainly due to a lack of differentiability of the flow field. In contrast, our flow field is differentiable, allowing us to train the whole network end to end with a spatial regularizer.

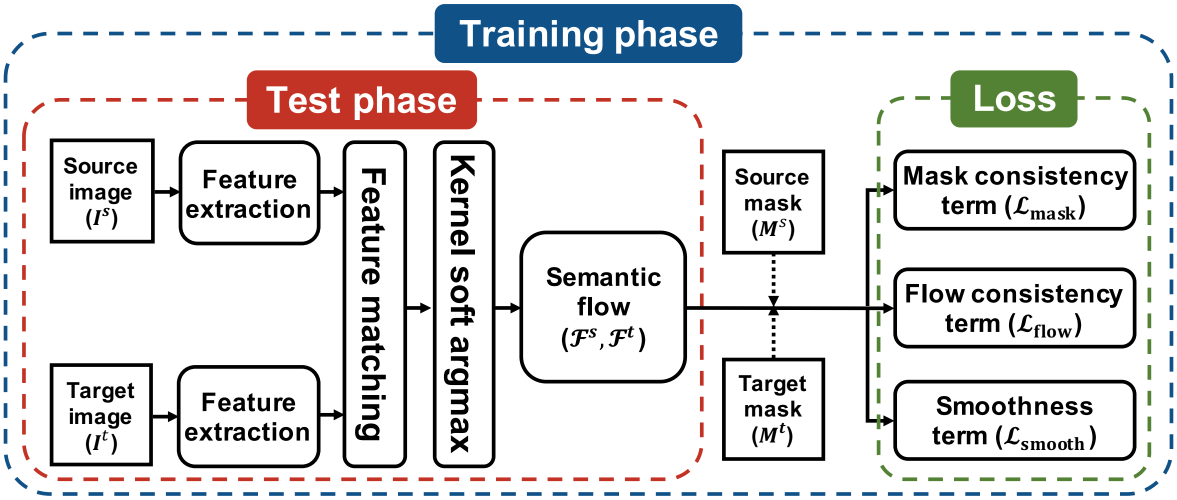

Training phase
Test phase
Loss
Source image ($I^s$)
Target image ($I^t$)
Feature extraction
Feature extraction
Feature matching
Kernel soft argmax
Semantic flow ($\mathcal{F}^s, \mathcal{F}^t$)
Source mask ($M^s$)
Target mask ($M^t$)
Mask consistency term ($\mathcal{L}_{\text{mask}}$)
Flow consistency term ($\mathcal{L}_{\text{flow}}$)
Smoothness term ($\mathcal{L}_{\text{smooth}}$)

Fig. 2: Overview of SFNet. SFNet takes an input pair of source and target images, and extracts local features using a siamese network. It then computes pairwise matching scores between features and establishes semantic flow for source and target images using the kernel soft argmax. At training time, corresponding foreground masks for the two images are used to compute mask consistency, flow consistency, and smoothness terms. See text for details.

### 2.2 Semantic Alignment

Several recent methods [20], [23], [24], [25], [26] formulate semantic correspondence as a geometric alignment problem using parametric models. In particular, these methods first compute feature correlations between images. The feature correlations are then fed into a regression layer to estimate parameters of a global transformation model (e.g., affine, homography, and thin plate spline) to align images. This makes it possible to leverage self-supervised learning [20], [23], [24], [25] using synthetically-generated data, and to train the entire CNNs end to end. These approaches apply the same transformation to all pixels, which has the effect of an implicit spatial regularization, providing smooth matches and often outperforming semantic flow methods [17], [18], [21], [22], [27]. However, they are easily distracted by background clutter and occlusion [20], [23], since correlations between pairs of features are noisy and include outliers (e.g., between different backgrounds). Although this can be alleviated by using attention models [25] or suppressing outlier matches [24], global transformation models are highly sensitive to non-rigid deformations or local geometric variations. Alternative approaches include estimating local transformation models in a coarse-to-fine scheme [26] or applying the geometric transformation recursively [44], but they are computationally expensive. In contrast, our method avoids the problem efficiently by establishing semantic correspondences directly from feature correlations.

### 2.3 An Object-level Prior for Semantic Correspondence

Several methods [10], [17], [21], [22], [26], [27], [31] leverage object priors (e.g., object proposals, bounding boxes or foreground masks) to learn semantic correspondence. Proposal flow [27] and its CNN version [17] use object proposals as matching primitives, and consider appearance and geometric consistency constraints to establish region correspondences. OADSC [31] also exploits object proposals, but leverages hierarchical graphs built on the proposals in a coarse-to-fine manner, allowing pixel-level correspondences. Similar to ours, other methods leverage bounding boxes or foreground masks for semantic correspondence. They, however, do not incorporate the object location prior explicitly into loss functions, and use the prior for pre-processing training samples instead. For example, PARN [26] and FCSS [21] use bounding boxes or foreground masks to generate positive/negative matches within object regions at training time. In [10], [22], bounding boxes are used to limit the candidate regions for matching at both training and test time. Contrary to these methods, we incorporate this prior (e.g., bounding boxes or foreground masks) directly into the loss functions to train the network, and outperform the state of the art by a significant margin.

## 3 Approach

In this section, we describe our approach to establishing object-aware semantic correspondences including the network architecture (Section 3.1) and loss functions (Section 3.2). An overview of our method is shown in Fig. 2.

### 3.1 Network Architecture

Our model consists of three main parts (Fig. 2): We first extract features from source and target images, $I^s$ and $I^t$, respectively, using a fully convolutional siamese network, where each of the two branches has the same structure with shared parameters. We then compute matching scores between all pairs of local features in the two images, and assign the best match for each feature using the kernel soft argmax function defined in Sec 3.1.3. All components are differentiable, allowing us to train the whole network end to end. In the following, we describe the network architecture for source to target matching in detail. A target to source match is computed in the same manner.

#### *3.1.1 Feature Extraction*

For feature extraction, we exploit a ResNet-101 [36] pretrained for the ImageNet classification task [38]. Although such CNN features give rich semantics, they typically fire on highly discriminative parts for classification [45]. This may be less adequate for feature matching that requires capturing a spatial deformation for fine-grained localization. We thus use additional adaptation layers to

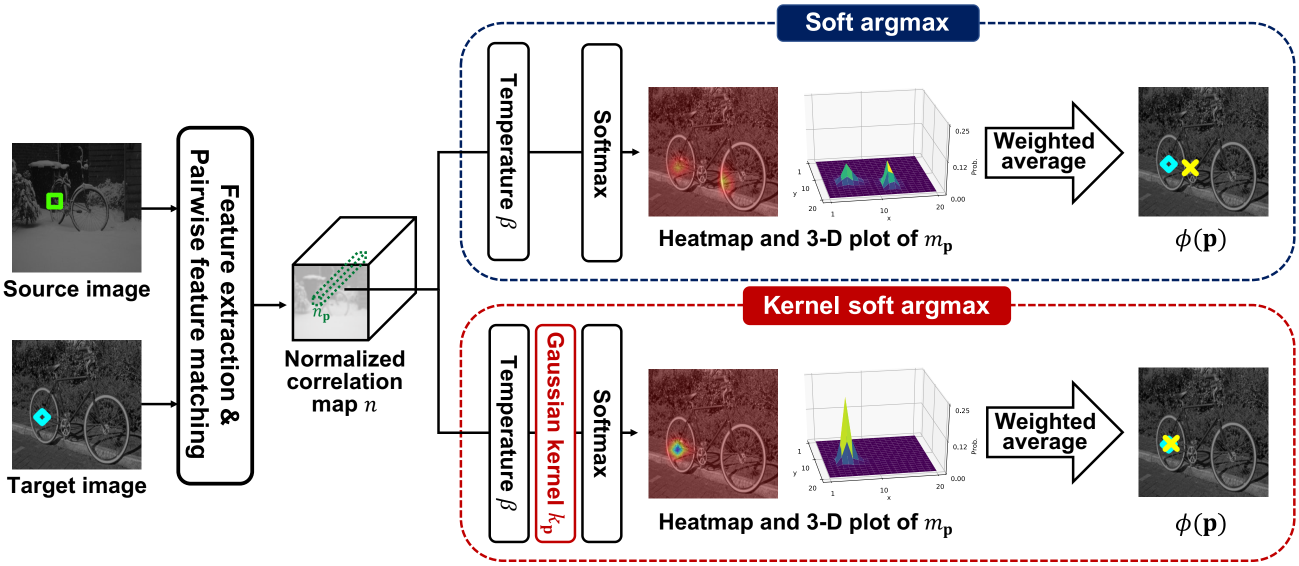


Fig. 3: Visualization of soft and kernel soft argmax operations. A point in the source image and its ground-truth correspondence in the target image are shown as the square and the diamond, respectively. A matching point computed by either the soft or kernel soft argmax operators is shown as the cross. When multiple features are highly correlated, the soft argmax often gives incorrect matches. The kernel soft argmax avoids this problem while maintaining differentiability. (Best viewed in color.)

extract features specific to the task of semantic correspondence, making them highly discriminative w.r.t both appearance and spatial context. This gives a feature map of size $h \times w \times d$ for each image that corresponds to $h \times w$ grids of $d$-dimensional local features. We then apply L2 normalization to the individual $d$-dimensional features. As will be seen in our experiments, the adaptation layers boost the matching performance drastically.

### 3.1.2 Feature Matching

Matching scores are computed as the dot product between local features, resulting in a 4-dimensional correlation map of size $h \times w \times h \times w$ as follows:

$$c(\mathbf{p}, \mathbf{q}) = f^s(\mathbf{p})^\top f^t(\mathbf{q}), \tag{1}$$

where we denote by $f^s(\mathbf{p})$ and $f^t(\mathbf{q})$ $d$-dimensional features at positions $\mathbf{p} = (p_x, p_y)$ and $\mathbf{q} = (q_x, q_y)$ in the source and target images, respectively.

### 3.1.3 Kernel Soft Argmax Layer

We could assign the best matches by applying the argmax function over a 2-dimensional correlation map $c_{\mathbf{p}}(\mathbf{q}) = c(\mathbf{p}, \mathbf{q})$, w.r.t all features $f^t(\mathbf{q})$ at each spatial location $\mathbf{p}$. However, argmax is not differentiable. The soft argmax function [46], [47] computes an output by a weighted average of all spatial positions with corresponding matching probabilities (i.e., an expected value of all spatial coordinates weighted by corresponding probabilities). Although it is differentiable and enables fine-grained localization at a sub-pixel level, its output is influenced by all spatial positions, which is problematic especially in the case of multi-modal distributions (Fig. 3). In other words, the soft argmax best approximates the discrete argmax when the matching probability is uni-modal having one clear peak.

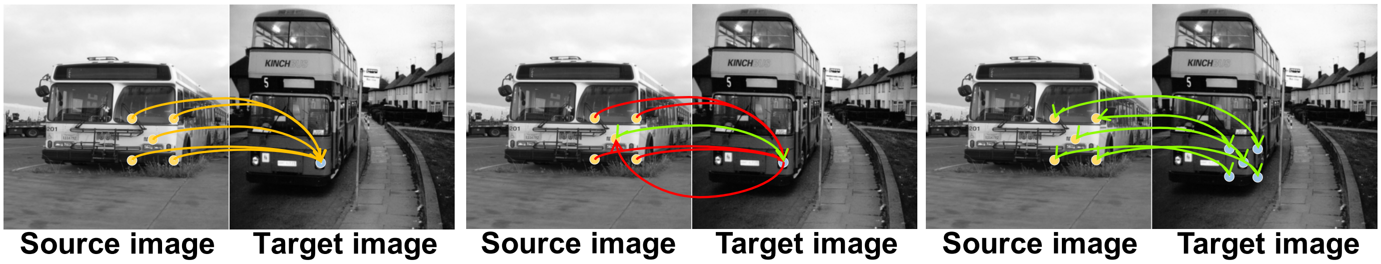
We introduce a hybrid version, the *kernel soft argmax*, that takes advantage of both the soft and discrete argmax. Concretely, it computes correspondences $\phi(\mathbf{p})$ for individual locations $\mathbf{p}$ as an average of all coordinate pairs $\mathbf{q} = (q_x, q_y)$ weighted by matching probabilities $m_{\mathbf{p}}(\mathbf{q})$ as follows.

$$\phi(\mathbf{p}) = \sum_{\mathbf{q}} m_{\mathbf{p}}(\mathbf{q})\mathbf{q}. \tag{2}$$

The matching probability $m_{\mathbf{p}}$ is computed by applying a spatial softmax function to a L2-normalized version $n_{\mathbf{p}}$ of the correlation map $c_{\mathbf{p}}$:

$$m_{\mathbf{p}}(\mathbf{q}) = \frac{\exp(\beta k_{\mathbf{p}}(\mathbf{q}) n_{\mathbf{p}}(\mathbf{q}))}{\sum_{\mathbf{q}' \in n_{\mathbf{p}}} \exp(\beta k_{\mathbf{p}}(\mathbf{q}') n_{\mathbf{p}}(\mathbf{q}'))}. \tag{3}$$

We perform L2 normalization on the 2-dimensional correlation map $c_{\mathbf{p}}$, adjusting the matching scores $f^s(\mathbf{p})^\top f^t(\mathbf{q})$ to a common scale before applying the softmax function. $\beta$ is a "temperature" parameter adjusting the distribution of the softmax output. As the temperature parameter $\beta$ increases, the softmax function approaches the discrete one with one clear peak, but this may cause an unstable gradient flow at training time. $k_{\mathbf{p}}$ is a 2-dimensional Gaussian kernel centered on the position obtained by applying the discrete argmax to the correlation map, i.e., $\text{argmax}_{\mathbf{q}}\, n_{\mathbf{p}}(\mathbf{q})$. The Gaussian kernel allows us to retain the scores $n_{\mathbf{p}}$ near the output of the discrete argmax while suppressing others. That is, the kernel $k_{\mathbf{p}}$ has the effect of restricting the range of averaging in (2), and makes the kernel soft argmax less susceptible to multi-modal distributions (e.g., from ambiguous matches in clutter and repetitive patterns) while maintaining differentiability. Note that the center position of the Gaussian kernel is changed at every iteration during training, and the matching probability $m_{\mathbf{p}}$ is differentiable, since we do not train the Gaussian kernel itself and no gradients are propagated through the discrete argmax. Note also that the normalization of the correlation map is particularly important for semantic alignment methods [23], [24], [25], [26] (see, for example, Table 2 in [23]) but its purpose is different from ours. They use the normalization to penalize features having multiple highly-correlated matches, boosting the scores of discriminative matches.



(a) Many-to-one matching. (b) Consistent and inconsistent flows. (c) One-to-one matching.

Fig. 5: Using the mask consistency term alone may cause a many-to-one matching problem: (a) multiple yellow points in the source image can be matched to the single blue one in the target image. The flow consistency term (b) penalizes inconsistent correspondences and (c) favors a one-to-one matching. We denote by green and red arrows consistent and inconsistent matches, respectively. (Best viewed in color.)

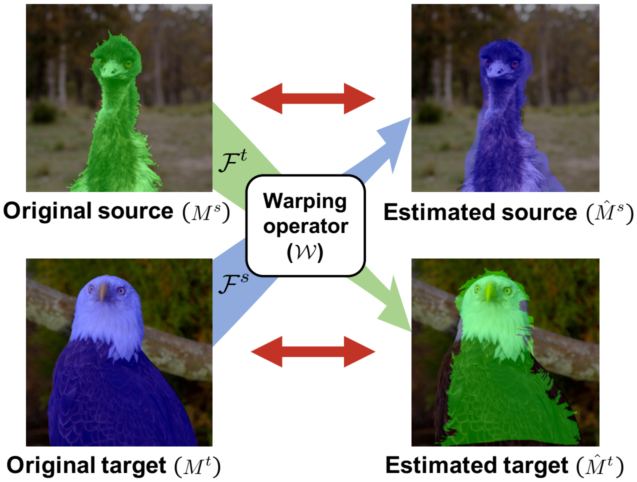

Original source ($M^s$)
Estimated source ($\hat{M}^s$)
$\mathcal{F}^t$
Warping operator ($\mathcal{W}$)
$\mathcal{F}^s$
Original target ($M^t$)
Estimated target ($\hat{M}^t$)

Fig. 4: Illustration of the mask consistency loss. We estimate the binary source mask $\hat{M}^s$ by warping the target one $M^t$ using the flow field $\mathcal{F}^s$. The target mask $\hat{M}^t$ is similarly estimated. We then compute the average error between original and estimated masks to compute the mask consistency loss. This penalizes the correspondences between foreground and background regions, and vice versa. We show foreground parts only for the purpose of visualization. (Best viewed in color.)

We visualize soft and kernel soft argmax operators in Fig. 3, which shows that the soft argmax yields an incorrect correspondence in the presence of multiple highly correlated features, since a weighted average of matching probabilities $m_{\mathbf{p}}$ having multi-modal distributions accumulates positional errors. The kernel soft argmax instead suppresses matching probabilities $m_{\mathbf{p}}$ for the highest value, making them have an (approximately) uni-modal distributions and favoring correct correspondences.

### 3.2 Loss

We exploit binary foreground masks as a supervisory signal to train the network, which gives a strong object prior. To this end, we define three losses that guide the network to learn object-aware correspondences without pixel-level ground truth as

$$\mathcal{L} = \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{flow}}\mathcal{L}_{\text{flow}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}, \tag{4}$$

which consists of mask consistency $\mathcal{L}_{\text{mask}}$, flow consistency $\mathcal{L}_{\text{flow}}$ and smoothness $\mathcal{L}_{\text{smooth}}$ terms, balanced by the weight parameters ($\lambda_{\text{mask}}$, $\lambda_{\text{flow}}$, $\lambda_{\text{smooth}}$). In the following, we describe each term in detail.

#### 3.2.1 Mask Consistency Term

We define a flow field $\mathcal{F}^s$ from source to target images as

$$\mathcal{F}^s(\mathbf{p}) = \phi(\mathbf{p}) - \mathbf{p}. \tag{5}$$

Similarly, a flow field $\mathcal{F}^t(\mathbf{q})$ from target to source images is defined as $\phi(\mathbf{q}) - \mathbf{q}$. We denote by $M^s$ and $M^t$ the binary masks of the source and target images, respectively. Values of 0 and 1 in the masks respectively indicate background and foreground regions. We assume that the binary mask in the source images can be reconstructed by warping the mask in the target image and vice versa, if we have discriminative features and correct dense correspondences. To implement this idea, we transfer the target mask $M^t$ by warping [48] using the flow field $\mathcal{F}^s$ and obtain an estimate of the source mask $\hat{M}^s$ as follows.

$$\hat{M}^s = \mathcal{W}(M^t; \mathcal{F}^s), \tag{6}$$

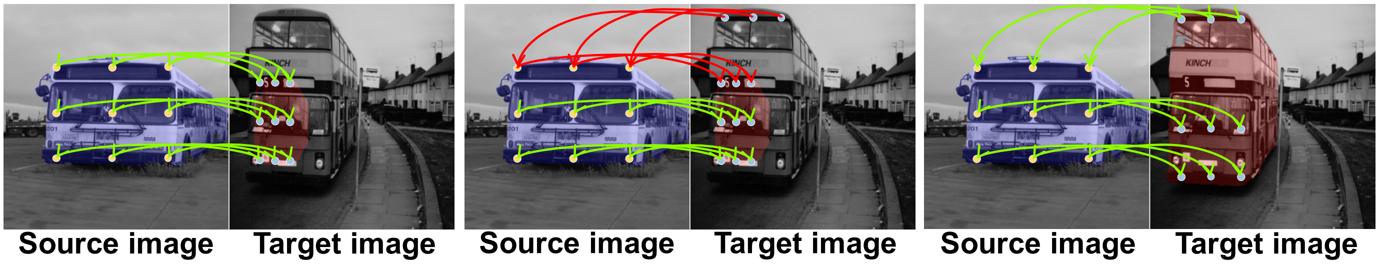
where $\mathcal{W}$ denotes a warping operator using the flow field, e.g.,

$$\mathcal{W}(M^t; \mathcal{F}^s)(\mathbf{p}) = M^t(\mathbf{p} + \mathcal{F}^s(\mathbf{p})). \tag{7}$$

We then compute the difference between the source mask $M^s$ and its estimate $\hat{M}^s$. Similarly, we reconstruct the target mask $\hat{M}^t$ from $M^s$ using the flow field $\mathcal{F}^t$ and compute the difference between $\hat{M}^t$ and $M^t$. Accordingly, we define the mask consistency loss (Fig. 4) as

$$\mathcal{L}_{\text{mask}} = \sum_{i\in\{s,t\}} \left( \frac{1}{|N^i|} \sum_{\mathbf{p}} (M^i(\mathbf{p}) - \hat{M}^i(\mathbf{p}))^2 \right), \tag{8}$$

where $|N^i|$ is the number of pixels in the mask $M^i$. Although the mask consistency loss does not constrain the background itself, it prevents matches from foreground to background regions and vice versa by penalizing them. This encourages correspondences to be established between features within foreground and background masks, guiding our model to learn object-aware correspondences. Note that the consistency loss using binary masks does not prevent a many-to-one matching (Fig. 5(a)). That is, it does not penalize a case when many foreground features in an image are matched to a single one in other image. For example, the foreground mask in the source image even can be reconstructed, when all points in the foreground region are matched to a single foreground point in the target image.



(a) Flow consistency for the source image. (b) Flow consistency for the target image. (c) Flow consistency for both images.

Fig. 6: Using a symmetric loss: (a) considering the flow consistency loss w.r.t a source image only may cause a flow shrinkage problem; (b) we can overcome this problem by computing the loss w.r.t a target image as well and penalizing inconsistent matches; (c) this symmetric loss allows us to perform object-level matching. We use green and red arrows to show consistent and inconsistent matches, respectively. (Best viewed in color.)

#### 3.2.2 *Flow Consistency Term*

To address the many-to-one matching problem, we propose to use a flow consistency loss. It measures consistency between flow fields $\mathcal{F}^s$ and $\mathcal{F}^t$ within foreground masks as

$$\mathcal{L}_{\text{flow}} = \sum_{i\in\{s,t\}} \left( \frac{1}{|N_F^i|} \sum_{\mathbf{p}} ||(\mathcal{F}^i(\mathbf{p}) + \hat{\mathcal{F}}^i(\mathbf{p})) \odot M^i(\mathbf{p})||_2^2 \right), \tag{9}$$

where $|N_F^i|$ is the number of foreground pixels in the mask $M^i$, and

$$\hat{\mathcal{F}}^s = \mathcal{W}(\mathcal{F}^t; \mathcal{F}^s), \tag{10}$$

which aligns the flow field $\mathcal{F}^t$ with respect to $\mathcal{F}^s$ by warping. $\hat{\mathcal{F}}^t$ is computed similar to (10). We denote by $\|\cdot\|_2$ and $\odot$ the L2 norm and element-wise multiplication, respectively. The multiplication is applied separately for each $x$ and $y$ component.

The flow consistency term penalizes inconsistent correspondences (Fig. 5(b)), and favors one-to-one matching (Fig. 5(c)), alleviating the many-to-one matching problem in the mask consistency loss. For example, when the flow fields are consistent with each other, $\mathcal{F}^s$ and $\hat{\mathcal{F}}^s$ in (9) have the same magnitude with opposite directions. Note that having multiple matches for individual points (i.e., a one-to-many matching) is impossible within our framework. Similar ideas have been explored in stereo fusion [49], [50] and optical flow [51], [52], but without appearance and shape variations. It is hard to incorporate this term in current semantic flow methods based on CNNs [17], [18], [21], [29], [43], mainly due to a lack of differentiability of the flow field. Recently, Zhou et al. [22] exploit cycle consistency between flow fields, but they regress correspondences directly from concatenated features from source and target images and do not consider background clutter. In contrast, our method establishes a differentiable flow field by computing feature similarities explicitly while considering background clutter.

Although the flow consistency term relieves the many-to-one matching problem, computing this term for a source or a target image only may cause a flow shrinkage problem (Fig. 6(a)). To address this problem, we compute this term w.r.t both source and target images in (9). This penalizes inconsistent matches, e.g., between the entire foreground region in the source image and small parts of the target image (Fig. 6(b)). Note that spreading the flow fields over the entire regions is particularly important to handle scale changes between objects (Fig. 6(c)).

#### 3.2.3 *Smoothness Term*

The differentiable flow field also allows us to exploit a smoothness term, as widely used in classical energy-based approaches [3], [7], [11]. We define this term using the first-order derivative of the flow fields $\mathcal{F}^s$ and $\mathcal{F}^t$ as

$$\mathcal{L}_{\text{smooth}} = \sum_{i\in\{s,t\}} \left( \frac{1}{|N_F^i|} \sum_{\mathbf{p}} ||\nabla\mathcal{F}^i(\mathbf{p}) \odot M^i(\mathbf{p})||_1 \right), \tag{11}$$

where $\|\cdot\|_1$ and $\nabla$ are the L1 norm and the gradient operator, respectively. This regularizes (or smooths) flow fields within foreground regions without being affected by (incorrect) correspondences at background.

## 4 Experiments

In this section, we give experimental details (Secs. 4.1), and present a detailed analysis and evaluation of our approach on the tasks of semantic correspondence (Section 4.2), mask transfer (Section 4.3) and pose keypoint propagation (Section 4.4). We then present ablation studies for different losses and network architectures, as well as a performance analysis for different training datasets (Section 4.5).

### 4.1 Experimental Details

#### 4.1.1 *Implementation*

Following [24], [25], [29], [43], [44], we use CNN features from ResNet-101 [36] trained for ImageNet classification [38]. Specifically, we use the networks cropped at `conv4-23` and `conv5-3` layers, respectively. This results in two feature maps of size $20 \times 20 \times 1024$ and $10 \times 10 \times 2048$, respectively, for a pair of input images of size $320 \times 320$, and gives a good compromise between localization accuracy and high-level semantics. Adaptation layers are trained with random initialization, separately for each feature map in a residual fashion [36]. To compute residuals, we add two blocks of convolutional, batch normalization [53] and ReLU [34] layers, with padding on top of each feature map, where the sizes of convolutional kernels for `conv4-23` and `conv5-3` features are $5 \times 5$ and $3 \times 3$, respectively. Each block outputs a residual, which is then added to the corresponding input features. Adaptation layers aggregate the features nonlinearly from large receptive fields (e.g., the receptive field of size $9 \times 9$ on a feature map of size $20 \times 20 \times 1024$), transforming them, guided by semantic correspondences and the corresponding loss terms in (4), to be highly discriminative w.r.t both appearance and spatial context. With the resulting two feature maps of size $20 \times 20 \times 1024$ and

$20 \times 20 \times 2048$[1], we compute pairwise matching scores and then combine them by element-wise multiplication, resulting in a correlation map of size $20 \times 20 \times 20 \times 20$. Following [23], [25], we fix the feature extractor, and train adaptation layers only. At test time, we upsample a flow field of size $20 \times 20$ using bilinear interpolation.

We empirically set the temperature parameter $\beta$ to 50 and standard deviation $\sigma$ of Gaussian kernel $k_{\mathbf{p}}$ to 5. We determine those values using a grid search over ($\beta$, $\sigma$) pairs, where the maximum search ranges for $\beta$ and $\sigma$ are 100 and 10 with intervals of 10 and 1, respectively. We select the parameters that give the best performance on the validation split of the PF-PASCAL dataset [24], [39]. Other parameters ($\lambda_{\text{mask}} = 3$, $\lambda_{\text{flow}} = 16$, $\lambda_{\text{smooth}} = 0.5$) are chosen similarly using the validation split of the PF-PASCAL dataset. We fix these parameters in all experiments.

#### *4.1.2 Training*

Training our network requires pairs of foreground masks for source and target images depicting different instances of the same object category. Although the TSS [8] and Caltech-101 [54] datasets provide such pairs, the number of masks in TSS [8] is not sufficient to train our network, and images in Caltech-101 [54] lack background clutter. Our model trained with these datasets suffers from overfitting and may not generalize well for other images containing clutter. Motivated by [20], [23], [25], [55], we generate pairs of source and target images synthetically from single images by applying random affine transformations and use the synthetically warped pairs as training samples. The corresponding foreground masks are transformed with the same parameters. Contrary to [20], [23], [25], our model does not perform parametric regression, and thus does not require ground-truth transformation parameters for training. We use the PASCAL VOC 2012 segmentation dataset [42] that consists of 1,464, 1,449, and 1,456 images for training, validation and test, respectively. We exclude 122 images from train/validation sets that overlap with the test split in PF-PASCAL [24], [39], and train our model with the corresponding 2,791 images. We augment the training dataset by horizontal flipping and color jittering. Note that we do not use the segmentation masks provided by the PASCAL VOC 2012 dataset, that specify the class of the object at each pixel. We instead generate binary foreground masks using all labeled objects, regardless of image categories and the number of objects, at training time. We train our model with a batch size of 16 for about 7k iterations, giving roughly 40 epochs over the training data. We use the Adam optimizer [56] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. A learning rate initially set to 3e-5 is divided by 5 after 30 epochs. All networks are trained end to end using `PyTorch` [57].

#### *4.1.3 Evaluation Metric*

We use the probability of correct keypoint (PCK) [58] to measure the precision of overall assignment, particularly at sparse keypoints of semantic relevance. We compute the Euclidean distance between warped keypoints using the estimated dense flow and ground truth, and count the number of keypoints whose distances lie within $\alpha_n \max(h_n, w_n)$ pixels, where $n \in \{\text{img}, \text{bbox}\}$, $\alpha$ typically set to $0.1$, is a tolerance value, and $h$ and $w$ are the height and width, respectively, of an image or an object bounding box. Following [23], [24], we divide keypoint coordinates by the height and width of the image size in case of $\alpha_{\text{img}}$, such that they are normalized in a range of $[0, 1]$ and $h_{\text{img}} = w_{\text{img}} = 1$.

TABLE 1: Quantitative comparison with the state of the art on the PF-WILLOW [27] and the test split of the PF-PASCAL [24], [39] in terms of average PCK. Numbers in bold indicate the best performance and underscored ones are the second best. The subscript for each method indicates the corresponding feature extractor. We denote by "F" and "A", respectively, semantic flow and semantic alignment methods. The characters in parentheses correspond to the type of supervisory signal used in training: T: transformation parameters; P: image pairs depicting different instances of the same object category; B: bounding boxes; C: ground-truth correspondences; M: foreground masks. All numbers in PF-WILLOW are taken from [29], [44]. The results of [27], [44], [59] in PF-PASCAL are taken from [44].

| Type | Methods | PF-WILLOW ($\alpha_{\text{bbox}} = 0.1$) | PF-PASCAL ($\alpha_{\text{img}} = 0.1$) |
|---|---|---|---|
| A | (T) A2Net$_{\text{res101}}$ [25] | 68.8 | 70.8 |
| A | (T) CNNGeo$_{\text{res101}}$ [23] | 69.2 | 71.9 |
| A | (T+P) WS-SA$_{\text{res101}}$ [24] | 70.2 | 75.8 |
| A | (P) RTN$_{\text{res101}}$ [44] | 71.9 | 75.9 |
| F | PF-LOM$_{\text{HOG}}$ [27] | 56.8 | 62.5 |
| F | (B+P) PF-LOM$_{\text{CAT-FCSS}}$ [59] | 58.4 | 68.9 |
| F | (P) NCN$_{\text{res101}}$ [43] | 67.0 | 78.9 |
| F | (C+P) HPF$_{\text{res101}}$ [29] | **74.4** | <u>80.4</u> |
| F | (M) Ours$_{\text{res101}}$ | <u>73.5</u> | **81.9** |

### 4.2 Semantic Correspondence

We compare our model to the state of the art on semantic correspondence including hand-crafted and CNN-based methods with the following four benchmark datasets: PF-WILLOW [27], PF-PASCAL [39], SPair-71k [29], and TSS [8]. Following the experimental protocol in [23], [24], [29], we use $\alpha_{\text{img}}$ for PF-PASCAL and TSS, and $\alpha_{\text{bbox}}$ for PF-WILLOW and SPair-71k, respectively. Results for all comparisons have been obtained from the source code or models provided by the authors, unless otherwise specified.

#### *4.2.1 PF-WILLOW and PF-PASCAL*

The PF-WILLOW [27] and PF-PASCAL [39] datasets provide 900 and 1,351 image pairs of 4 and 20 image categories, respectively, with corresponding ground-truth object bounding boxes and keypoint annotations. The PF-PASCAL dataset is more challenging than other datasets [8], [27] for semantic correspondence evaluation, featuring different instances of the same object class in the presence of large changes in appearance and scene layout, clutter and scale changes between objects. To evaluate our model, we use PF-WILLOW and the test split of PF-PASCAL provided by [24], [39] corresponding roughly 900 and 300 image pairs, respectively.

We show in Table 1 the average PCK scores for the PF-WILLOW and PF-PASCAL datasets, and compare our method with the state of the art. From this table, we observe four things: (1) Our model outperforms the state of the art by a significant margin in terms of PCK, especially for the PF-PASCAL dataset. In particular, it shows better performance than other object-aware methods [27], [59] that focus on establishing region correspondences between prominent objects. A plausible explanation is

[1] We upsample the features adapted from `conv5-3` using bilinear interpolation.

TABLE 2: Per-class PCK ($\alpha_{\text{img}} = 0.1$) on PF-PASCAL [39]. The results of [27] are taken from [24].

| Type | Methods | aero | bike | bird | boat | bot | bus | car | cat | cha | cow | tab | dog | hor | mbik | pers | plnt | she | sofa | trai | tv | all |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | (T) A2Net$_{\text{res101}}$ [25] | 83.2 | 82.8 | 83.8 | 44.4 | 57.8 | 81.3 | 89.4 | 86.1 | 40.1 | 91.7 | 21.4 | 73.2 | 33.8 | 76.3 | **74.3** | 63.3 | **100.0** | 45.5 | 45.3 | 60.0 | 70.8 |
| A | (T) CNNGeo$_{\text{res101}}$ [23] | 82.4 | 80.9 | **85.9** | 47.2 | 57.8 | 83.1 | 92.8 | 86.9 | 43.8 | 91.7 | 28.1 | 76.4 | 70.2 | 76.6 | 68.9 | 65.7 | 80.0 | 50.1 | 46.3 | 60.6 | 71.9 |
| A | (T+P) WS-SA$_{\text{res101}}$ [24] | 83.7 | 88.0 | 83.4 | 58.3 | 68.8 | 90.3 | 92.3 | 83.7 | 47.4 | 91.7 | 28.1 | 76.3 | **77.0** | 76.0 | 71.4 | **76.2** | 80.0 | 59.5 | 62.3 | 63.9 | 75.8 |
| F | PF-LOM$_{\text{HOG}}$ [27] | 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 |
| F | (P) NCN$_{\text{res101}}$ [43] | 86.8 | 86.7 | 86.7 | 55.6 | 82.8 | 88.6 | 93.8 | 87.1 | 54.3 | 87.5 | 43.2 | 82.0 | 64.1 | 79.2 | 71.1 | 71.0 | 60.0 | 54.2 | 75.0 | **82.8** | 78.9 |
| F | (C+P) HPF$_{\text{res101}}$ [29] | 86.5 | 88.9 | 81.6 | **75.0** | 81.3 | 89.7 | 93.7 | **87.6** | 62.2 | 87.5 | 52.6 | **87.5** | 74.2 | **83.5** | 73.5 | 66.2 | 60.0 | 66.2 | 68.5 | 66.7 | 80.4 |
| F | Ours$_{\text{res101}}$ | **89.5** | **89.2** | 83.1 | 73.6 | **85.9** | **92.6** | **95.0** | 83.7 | **65.6** | **93.8** | **53.6** | 81.3 | 71.6 | 80.6 | 72.3 | 71.0 | **100.0** | **69.3** | **80.0** | 79.5 | **81.9** |

TABLE 3: Per-class PCK ($\alpha_{\text{bbox}} = 0.1$) on SPair-71k [29]. All numbers but ours are taken from [29].

| Methods | | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | dog | horse | moto | person | plant | sheep | train | tv | all |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Transferred models | (T) CNNGeo$_{\text{res101}}$ [23] | 21.3 | 15.1 | 34.6 | 12.8 | 31.2 | 26.3 | 24.0 | 30.6 | 11.6 | 24.3 | 20.4 | 12.2 | 19.7 | 15.6 | 14.3 | 9.6 | 28.5 | 28.8 | 18.1 |
| | (T) A2Net$_{\text{res101}}$ [25] | 20.8 | 17.1 | 37.4 | 13.9 | 33.6 | **29.4** | 26.5 | 34.9 | 12.0 | 26.5 | 22.5 | 13.3 | 21.3 | 20.0 | 16.9 | 11.5 | 28.9 | 31.6 | 20.1 |
| | (T+P) WS-SA$_{\text{res101}}$ [24] | 23.4 | 17.0 | 41.6 | 14.6 | **37.6** | 28.1 | **26.6** | 32.6 | 12.6 | 27.9 | 23.0 | 13.6 | 21.3 | **22.2** | 17.9 | 10.9 | 31.5 | 34.8 | 21.1 |
| | (P) NCN$_{\text{res101}}$ [43] | 24.0 | 16.0 | 45.0 | 13.7 | 35.7 | 25.9 | 19.0 | 50.4 | **14.3** | 32.6 | **27.4** | **19.2** | **21.7** | 20.3 | **20.4** | 13.6 | **33.6** | **40.4** | **26.4** |
| | (M) Ours$_{\text{res101}}$ | **27.3** | **17.2** | **47.2** | **14.7** | 36.7 | 21.4 | 16.5 | **56.4** | 13.6 | **32.9** | 25.4 | 17.4 | 19.9 | 19.5 | 15.9 | **15.9** | 33.2 | 35.1 | 26.0 |
| SPair-71k trained models | (T) CNNGeo$_{\text{res101}}$ [23] | 23.4 | 16.7 | 40.2 | 14.3 | 36.4 | 27.7 | 26.0 | 32.7 | 12.7 | 27.4 | 22.8 | 13.7 | 20.9 | 21.0 | 17.5 | 10.2 | 30.8 | 34.1 | 20.6 |
| | (T) A2Net$_{\text{res101}}$ [25] | 22.6 | 18.5 | 42.0 | **16.4** | 37.9 | **30.8** | 26.5 | 35.6 | 13.3 | 29.6 | 24.3 | 16.0 | 21.6 | 22.8 | 20.5 | 13.5 | 31.4 | **36.5** | 22.3 |
| | (T+P) WS-SA$_{\text{res101}}$ [24] | 22.2 | 17.6 | 41.9 | 15.1 | **38.1** | 27.4 | **27.2** | 31.8 | 12.8 | 26.8 | 22.6 | 14.2 | 20.0 | 22.2 | 17.9 | 10.4 | 32.2 | 35.1 | 20.9 |
| | (P) NCN$_{\text{res101}}$ [43] | 17.9 | 12.2 | 32.1 | 11.7 | 29.0 | 19.9 | 16.1 | 39.2 | 9.9 | 23.9 | 18.8 | 15.7 | 17.4 | 15.9 | 14.8 | 9.6 | 24.2 | 31.1 | 20.1 |
| | (C+P) HPF$_{\text{res101}}$ [29] | 25.2 | **18.9** | **52.1** | 15.7 | 38.0 | 22.8 | 19.1 | 52.9 | **17.9** | 33.0 | **32.8** | **20.6** | **24.4** | **27.9** | **21.1** | **15.9** | 31.5 | 35.6 | **28.2** |
| | (M) Ours$_{\text{res101}}$ | **26.9** | 17.2 | 45.5 | 14.7 | 38.0 | 22.2 | 16.4 | **55.3** | 13.5 | **33.4** | 27.5 | 17.7 | 20.8 | 21.1 | 16.6 | 15.6 | **32.3** | 35.9 | 26.3 |

that establishing correspondences between object proposals is susceptible to shape deformations. (2) We can clearly see that our model gives better results than semantic alignment methods on both datasets, but performance gain for the PF-PASCAL dataset, which typically contains pictures depicting a non-rigid deformation and clutter (e.g., in cow and sofa classes), is more significant. For example, the PCK gain over RTN [44] for the PF-PASCAL (81.9 vs. 75.9) is about four times more than that for the PF-WILLOW (73.5 vs. 71.9), indicating that our semantic flow method is more robust to non-rigid deformations and background clutter than semantic alignment approaches. (3) By comparing our model with CNN-based semantic flow methods, we can see that involving a spatial regularizer is significant. These techniques focus on designing fidelity terms (e.g., using a contrastive loss [18], [59]) to learn a feature space preserving semantic similarities. This is because of a lack of differentiability of the flow field. In contrast, our model gives a differentiable flow field, allowing to exploit a spatial regularizer while further leveraging high-level semantics from CNN features more specific to semantic correspondence. (4) We confirm once more a finding in [37] that CNN features trained for ImageNet classification [38] clearly show a better ability to handle intra-class variations than hand-crafted ones (HOG [15] in PF-LOM [27]).

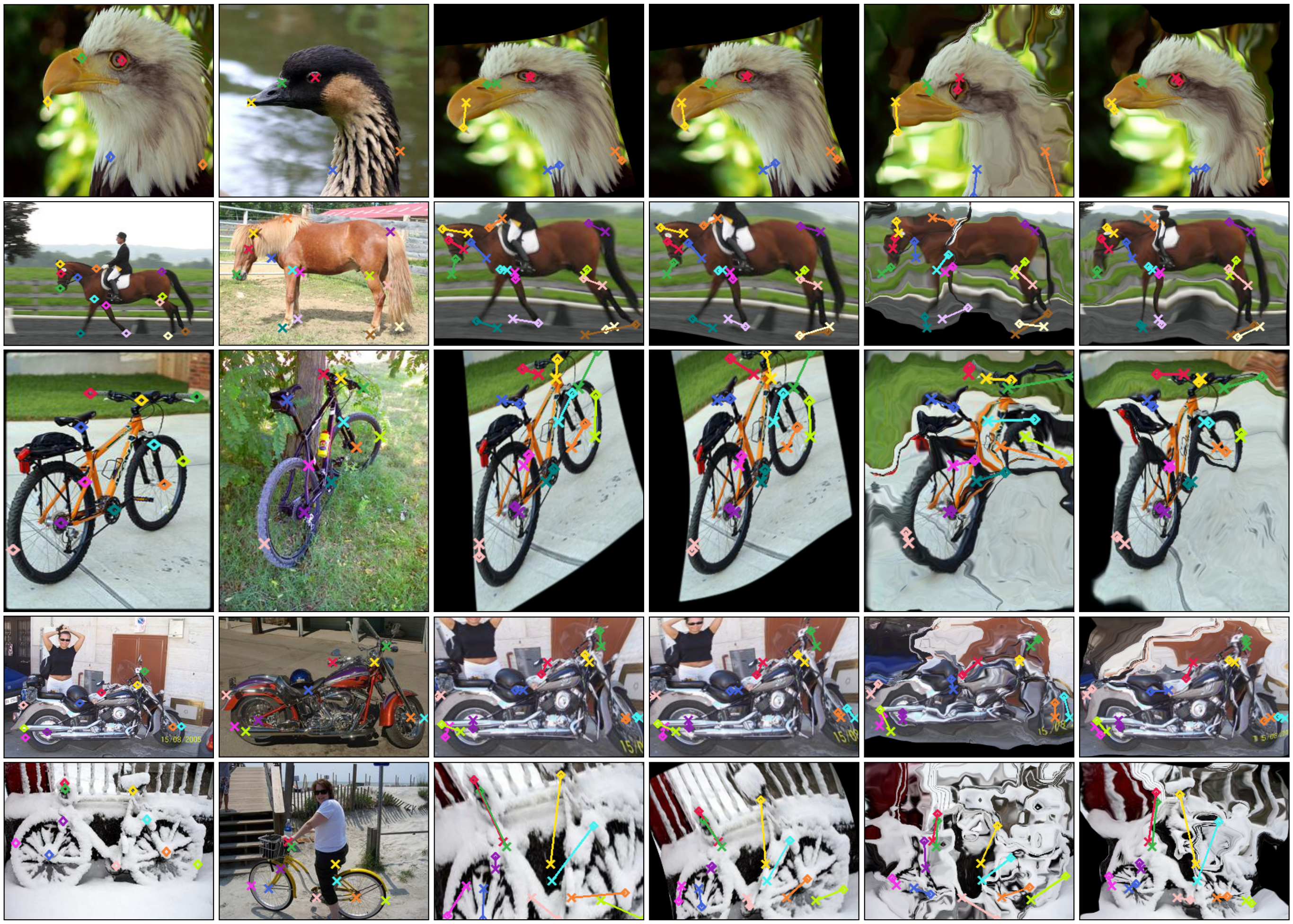
Table 2 shows per-class PCK scores on the PF-PASCAL dataset [39]. Our model achieves state-of-the-art results for 11 object categories, and outperforms all methods on average by a large margin. The performance gain is significant especially in the presence of non-rigid deformations (e.g., in cow and sheep classes) or distractions such as clutter (e.g., in table and sofa classes). This demonstrates once again that our method is able to establish reliable semantic correspondences of keypoints, even for images with large shape variations and clutter by which semantic alignment methods are easily distracted.

#### 4.2.2 *SPair-71k*

The SPair-71k dataset [29], a large-scale benchmark for semantic correspondence, provides 70,958 image pairs of 18 object categories with ground-truth annotations for object bounding boxes, segmentation masks, and keypoints. The image pairs in SPair-71k feature various changes in viewpoint, scale, truncation, and occlusion. Following the experimental protocol of [29], we evaluate our model on the test split of 12,234 image pairs, and compute PCK scores with $\alpha_{\text{bbox}} = 0.1$ We show in Table 3 the per-class and average PCK scores, and compare our model with the state of the art [23], [24], [25], [29], [43]. The first five rows show the PCK scores for models provided by authors, without retraining or finetuning on the SPair-71k dataset. We can see that our model achieves the second best performance, demonstrating that it generalizes well to unseen images. It is slightly outperformed by NCN [43] (0.4% in terms of the average PCK) but runs about 11 times faster at test time (Table 5). The last six rows show the PCK scores for the models trained with SPair-71k. For fair comparison, we train our model with the training set of SPair-71k (986 images). The results show that it performs best in the presence of non-rigid deformations (i.e., in cat and cow classes). For other object categories, our model outperforms other CNN-based methods except for HPF [29]. Note that HPF exploits ground-truth correspondences at training time, which gives strong constraints but is extremely labor-intensive. In contrast, our model uses binary foreground masks only, that are widely available and much cheaper to obtain.

#### 4.2.3 *TSS*

The TSS dataset [8] consists of three subsets (FG3DCar, JODS and PASCAL) that contain 400 image pairs of 7 object categories. It provides dense flow fields obtained by interpolating sparse keypoint matches with additional co-segmentation masks. Following the experimental protocol of [24], we compute the PCK scores ($\alpha_{\text{img}} = 0.05$) densely over the foreground object. Table 4 compares the average PCK on each subset in the TSS dataset. Our method shows better performance than the state of the art for FG3DCar and JODS. We do not do as well on the PASCAL part of TSS, which contains many image pairs with different poses (e.g., cars captured with left- and right-side viewpoints). Current methods, except for OADSC [31] that is specially designed for handling changes in viewpoint, have a limited capability of finding matches between images with different poses. Ours is no exception.



Source image. Target image. CNNGeo [23]. WS-SA [24]. HPF [29]. Ours.

Fig. 7: Visual comparison of alignment results between source and target images on the PF-PASCAL dataset [39]. Keypoints in the source and target images are shown as diamonds and crosses, respectively, with a vector representing the matching error. All methods use ResNet-101 features. Compared to the state of the art, our method is more robust to local non-rigid deformations, scale changes between objects, and clutter. See text for details. (Best viewed in color.)

TABLE 4: Quantitative comparison on the TSS dataset [8] in terms of the average PCK. We measure the PCK scores ($\alpha_{\text{img}} = 0.05$) on three subsets (FG3DCar, JODS and PASCAL). All numbers except ours are taken from [8], [24], [44].

| Type | | Methods | FG3D. | JODS | PASC. |
|---|---|---|---|---|---|
| Hand-crafted | F | DSP [7] | 48.7 | 46.5 | 38.2 |
| | F | DFF [13] | 49.3 | 30.3 | 22.4 |
| | F | SIFTFlow [3] | 63.4 | 52.2 | 45.3 |
| | F | PF-LOM$_{\text{HOG}}$ [27] | 78.6 | 65.3 | 53.1 |
| | F | TSS$_{\text{HOG}}$ [8] | 83.0 | 59.5 | 48.3 |
| | F | OADSC$_{\text{HOG}}$ [31] | 87.5 | 70.8 | **72.9** |
| CNN-based | A | (T) A2Net$_{\text{vgg16}}$ [25] | 87.0 | 67.0 | 55.0 |
| | A | (T) CNNGeo$_{\text{res101}}$ [23] | 90.1 | 76.4 | 56.3 |
| | A | (T+P) WS-SA$_{\text{res101}}$ [24] | 90.3 | 76.4 | 56.5 |
| | A | (P) RTN$_{\text{res101}}$ [44] | 90.1 | 78.2 | 63.3 |
| | F | (B+P) PF-LOM$_{\text{FCSS}}$ [21] | 83.9 | 63.5 | 58.2 |
| | F | (B+P) DCTM$_{\text{FCSS}}$ [60] | 89.1 | 72.1 | 61.0 |
| | F | (M) Ours$_{\text{res101}}$ | **90.6** | **78.7** | 56.5 |

TABLE 5: Runtime comparison per image pair on the test split of the PF-PASCAL dataset [17], [39] in milliseconds.

| Type | Methods | Time (*ms*) |
|---|---|---|
| A | (T) CNNGeo$_{\text{res101}}$ [23] | 34.2 |
| A | (T+P) WS-SA$_{\text{res101}}$ [24] | 34.4 |
| A | (T) A2Net$_{\text{res101}}$ [25] | 61.2 |
| F | (P) NCN$_{\text{res101}}$ [43] | 284.2 |
| F | (C+P) HPF$_{\text{res101}}$ [29] | 48.9 |
| F | (M) Ours$_{\text{res101}}$ | **24.9** |

### *4.2.4 Runtime Analysis*

Table 5 shows runtime comparisons of state-of-the-art methods. For comparison, we run the original source codes implemented using `PyTorch` [57]. The average runtime is measured on the same machine with a NVIDIA Titan RTX GPU. The table shows that our model is fastest among the state of the art. Semantic alignment methods [23], [24], [25] estimate parameters of affine and thin place spline sequentially, which degrades runtime performance. Semantic flow methods involve 4-D convolutions [43] or a Hough voting process [29] on top of the correlation volume, requiring additional computations to establish pixel-level correspondences. Our model, on the other hand, simply assigns the best matches from the correlation volume in a single stage. Most computation time is spent extracting features (23.7 milliseconds). Computing matching scores and establishing correspondences using the kernel

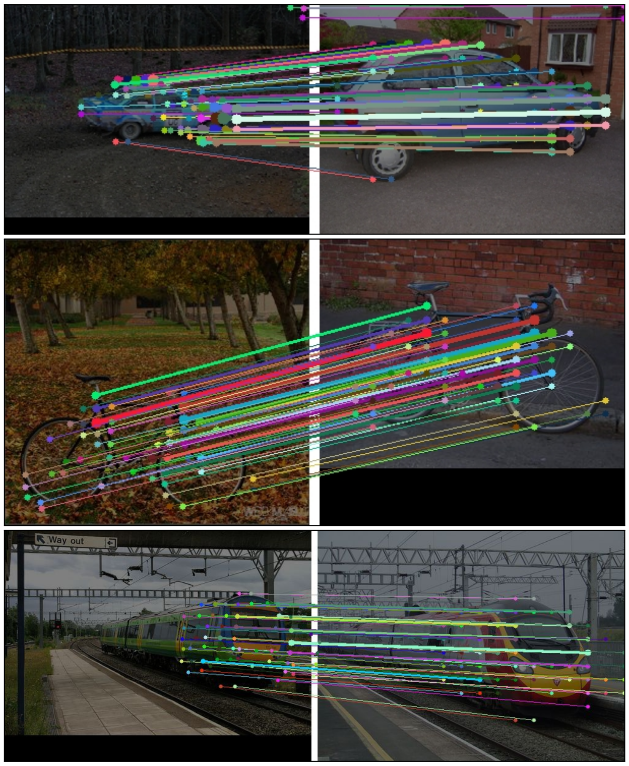


Fig. 8: Top matches on standard benchmarks. We visualize the top 60 matches according to matching probabilities. Each row shows a result on PF-WILLOW [27], PF-PASCAL [39] and TSS [8], respectively. (Best viewed in color.)

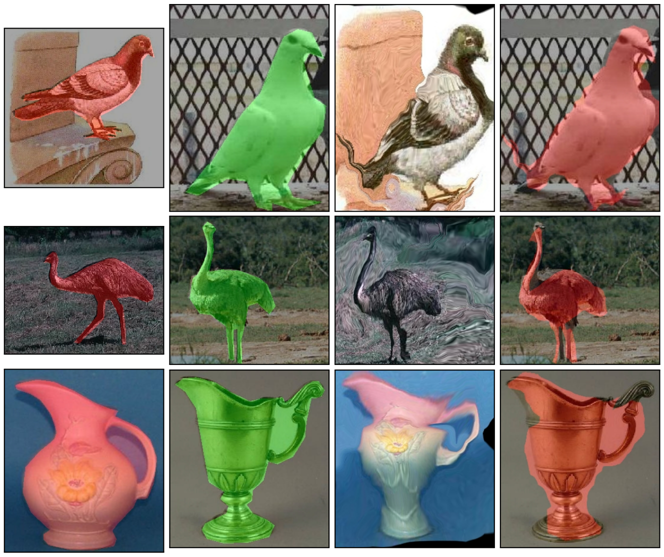
soft argmax just take 1.2 milliseconds.

### 4.2.5 *Qualitative Results*

Figure 7 shows a visual comparison of alignment results between source and target images with the state of the art on the test split of PF-PASCAL [24], [39]. To this end, the source images are warped to the target images using the dense flow fields computed by each method. We can see that our method is robust to local non-rigid deformation (e.g., bird beaks and horse legs in the first two rows), scale changes between objects (e.g., front wheels in the third row), and clutter (e.g., wheels in the last row), while semantic alignment methods [23], [24] are not. In particular, the fourth example clearly shows that our method gives more discriminative correspondences, cutting off matches for non-common objects. For example, it does not establish correspondences between a person and background regions in the source and target images, respectively, while CNNGeo [23] and WS-SA [24] fail to cut off matches on these regions. We can also see that all methods do not establish correspondences for occluded regions (e.g., a bicycle saddle in the last row). We also show in Fig. 8 the top 60 matches chosen according to matching probabilities on the PF-WILLOW [27], PF-PASCAL [39], and TSS [8] datasets. We can see that most strong matches are established between prominent objects, and matches between foreground and background regions have low matching probabilities.

## 4.3 Mask Transfer

We apply our model to the task of mask transfer on the Caltech-101 [54] dataset. This dataset, originally introduced for image classification, provides pictures of 101 image categories with ground-truth object masks. Unlike the PF [27], [39] and TSS [8] datasets, it does not provide ground-truth keypoint annotations. For fair comparison, we use 15 image pairs, provided by [17], [24], for each object category, and use the corresponding 1,515 image pairs for evaluation. Following the experimental protocol in [7], we compute matching accuracy with two metrics using the ground-truth masks: Label transfer accuracy (LT-ACC) and the intersection-over-union (IoU) metric. Both metrics count the number of correctly labeled pixels between ground-truth and transformed masks using dense correspondences, where the LT-ACC evaluates the overall matching quality while the IoU metric focuses more on foreground objects. Following [24], [25], we exclude the LOC-ERR metric, since it measures the localization error of correspondences using object bounding boxes due to a lack of keypoint annotations, which

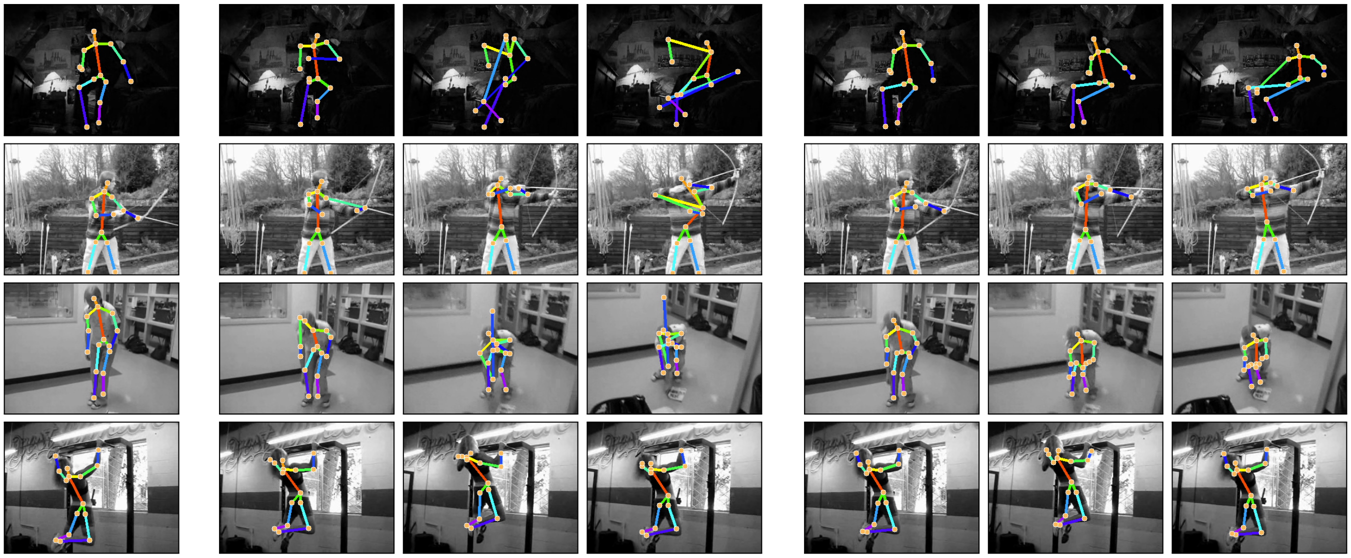


Source image. Target image. Alignment. Label transfer.

Fig. 9: Alignment and label transfer examples on the Caltech-101 dataset [54]. The source and target masks are overlaid on the corresponding images. We transfer pixel labels of the source images to the target ones using established correspondences. We show label transfer results overlaid on target images. (Best viewed in color.)

TABLE 6: Quantitative comparison on the Caltech-101 dataset [54]. All numbers but ours are taken from [24], [25], [29], [39].

| Type | | Methods | LT-ACC | IoU |
|---|---|---|---|---|
| Hand-crafted | F | DeepFlow [61] | 0.74 | 0.40 |
| | F | GMK [6] | 0.77 | 0.42 |
| | F | SIFTFlow [3] | 0.75 | 0.48 |
| | F | DSP [7] | 0.77 | 0.47 |
| | F | PF-LOM$_{\text{HOG}}$ [39] | 0.78 | 0.50 |
| | F | OADSC [31] | 0.81 | 0.55 |
| CNN-based | A | (T) A2Net$_{\text{vgg16}}$ [25] | 0.80 | 0.57 |
| | A | (T) CNNGeo$_{\text{res101}}$ [23] | 0.83 | 0.61 |
| | A | (T+P) WS-SA$_{\text{res101}}$ [24] | 0.85 | 0.63 |
| | F | (B+P) PF-LOM$_{\text{FCSS}}$ [21] | 0.83 | 0.52 |
| | F | (C+P) SCNet-AG$_{\text{vgg16}}$ [17] | 0.79 | 0.51 |
| | F | (P) NCN$_{\text{res101}}$ [43] | 0.85 | 0.60 |
| | F | (C+P) HPF$_{\text{res101}}$ [29] | 0.87 | 0.63 |
| | F | (M) Ours$_{\text{res101}}$ | **0.88** | **0.67** |



Input. TimeCycle$_{\text{res50}}$ [62]. Ours$_{\text{res50}}$.

Fig. 10: Visual comparison of keypoint propagation results on the JHMDB dataset [30]. Given an input labeled with the ground-truth pose keypoints, we propagate them through video sequences. Compared to the state of the art, our method is more robust to background clutter, large displacements, and occlusion. The keypoints are shown in circles. (Best viewed in color.)

TABLE 7: Quantitative comparison on the JHMDB dataset [30]. All numbers except for HPF [29] are taken from [62]. Identity: copying labels in the first frame; †: models trained from scratch; V: unlabeled video frames.

| Methods | PCK ($\alpha_{\text{bbox}} = 0.1$) | PCK ($\alpha_{\text{bbox}} = 0.2$) |
|---|---|---|
| Identity | 43.1 | 64.5 |
| SIFTFlow [3] | 49.0 | 68.6 |
| (V+C) Optical Flow$_{\text{flownet2}}$† [63] | 45.2 | 62.9 |
| (V) Video Colorizaiton$_{\text{3d-res18}}$† [64] | 45.2 | 69.6 |
| (V) TimeCycle$_{\text{res18}}$† [62] | 57.3 | 78.1 |
| (V) TimeCycle$_{\text{res50}}$† [62] | 57.7 | 78.5 |
| (V) TimeCycle$_{\text{res50}}$ [62] | 58.4 | 78.4 |
| (C+P) HPF$_{\text{res101}}$ [29] | 58.7 | 76.8 |
| (M) Ours$_{\text{res50}}$ | 59.9 | 78.9 |
| (M) Ours$_{\text{res101}}$ | **61.0** | **80.6** |

does not cover rotations, affine or deformable transformations.

The LT-ACC and IoU comparisons on the Caltech-101 dataset are shown in Table 6. Although this dataset provides ground-truth object masks, we do not retrain or fine-tune our model to evaluate its generalization ability on other datasets. From this table, we can see that (1) our model generalizes better than other CNN-based methods for other images outside the training dataset; and (2) it outperforms the state of the art in terms of the LT-ACC and IoU, verifying once more that our model focuses on regions containing objects while filtering out background clutter, even without using object proposals [17], [21], [31], [39] or inlier counting [24]. In Fig. 9, we show alignment and label transfer examples on the Caltech-101 [54] dataset. We can see that our method is robust against local non-rigid deformations (e.g., bird's neck, body, and legs).

### 4.4 Pose Keypoint Propagation

We apply our model to the task of keypoint propagation on the JHMDB [30] dataset. We propagate ground-truth pose keypoints in the first frame to subsequent ones by estimating semantic correspondences between them. The JHMDB [30] dataset contains 928 clips of 21 action categories with pose keypoints, segmentation masks of humans in action, obtained by a 2D articulated human puppet model [65], and provides three splits, where each split consists of training and test sets. Following the experimental protocol of [62], [64], we test our model on the test set in the split 1 corresponding to 268 clips of action categories, without retraining or fine-tuning on the dataset. We normalize keypoint coordinates in the range of [0,1] by dividing them with the height and width of the human bounding box size, respectively, and use the PCK score with two threshold values ($\alpha_{\text{bbox}} = 0.1, 0.2$) for evaluation.

We show in Table 7 the average PCK scores for the keypoint propagation task, and compare our method with the state of the art including self-supervised methods [62], [64]. From this table, we can see that our model based on ResNet-50 [36] outperforms the state of the art, even without using video datasets for training. For example, TimeCycle [62] is trained with the VLOG [66] dataset that contains 114k videos with the total length of 344 hours. Training networks with such video datasets requires lots of computational resources and training time. We can also see that our model outperforms HPF [29], demonstrating once again its generalization ability to unseen images during training. Figure 10 shows a visual comparison of keypoint propagation results with the TimeCycle [62] method on the JHMDB [30] dataset. The qualitative results for the comparison have been obtained from the original model [62] provided by the authors. We predict the keypoints in the rest of the videos, by propagating ground truth in the first frame. We can see that our method is more robust to background clutter (e.g., body parts in the first row) and large displacements (e.g., elbows and wrists in the second row). Moreover, it is not seriously affected by occlusion (e.g., ankles and

TABLE 8: Average PCK comparison of different loss functions.

| Mask consistency | Flow consistency | Smoothness | PCK ($\alpha_{\text{bbox}} = 0.1$) |
|---|---|---|---|
| ✓ | ✗ | ✗ | 67.5 |
| ✗ | ✓ | ✗ | 71.8 |
| ✓ | ✓ | ✗ | 78.2 |
| ✓ | ✓ | ✓ | **78.7** |

TABLE 9: Average PCK comparison of variants of our model. We denote by "D", "S", and "KS" discrete, soft, and kernel soft argmax operators, respectively.

| Adaptation | Multi-level | Argmax | | PCK |
|---|---|---|---|---|
| layer | feature | Train | Test | ($\alpha_{\text{bbox}} = 0.1$) |
| ✗ | ✓ | - | D | 45.8 |
| ✗ | ✓ | - | S | 8.8 |
| ✗ | ✓ | - | KS | 28.4 |
| ✓ | ✗ | S | D | 72.5 |
| ✓ | ✗ | S | S | 71.7 |
| ✓ | ✗ | KS | KS | 75.0 |
| ✓ | ✓ | S | D | 76.8 |
| ✓ | ✓ | S | S | 76.2 |
| ✓ | ✓ | KS | KS | **78.7** |

wrists in the last two rows) as the smoothness term regularizes flow fields within prominent objects.

## 4.5 Ablation Study and Effect of Training Data

We show an ablation analysis on different components and losses in our model. We measure a PCK score ($\alpha_{\text{bbox}} = 0.1$), which is a more strict metric compared to $\alpha_{\text{img}}$, and report the results of semantic correspondence on the test split of PF-PASCAL [24], [39]. We also study the effect of using different training datasets on performance.

### 4.5.1 Training Loss

We show the average PCK for three variants of our model in Table 8. The mask consistency term encourages establishing correspondences between prominent objects. Our model trained with this term only, however, may not yield spatially distinctive correspondences, resulting in the worst performance. The flow consistency term, which spreads flow fields over foreground regions, overcomes this problem, but it does not differentiate correspondences between background and objects. Accordingly, these two terms are complementary each other and exploiting both significantly boosts the performance of our model from 67.5/71.8 to 78.2. An additional smoothness term further boosts performance to 78.7.

### 4.5.2 Network Architecture

Table 9 compares the performance of networks with different components in terms of average PCK. The baseline models in the first three rows compute matching scores using multi-level features from `conv4-23` and `conv5-3` layers, and estimate correspondences with different argmax operators. They do not involve any training similar to [37] that uses off-the-shelf CNN features for semantic correspondence. We can see that applying the soft argmax directly to the baseline model degrades performance severely, since it is highly susceptible to multi-modal distributions.

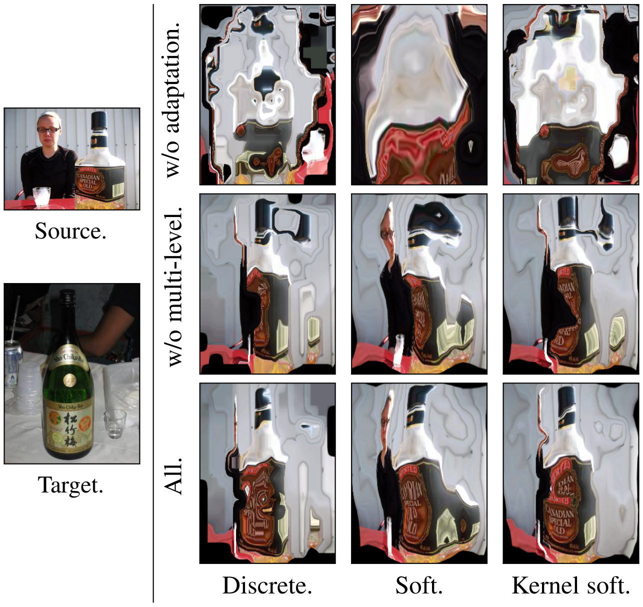


Fig. 11: Visual comparison of different network architectures and argmax operators. We show alignment examples for the variants of our model in Table 9.

The results in the next three rows are obtained with a single adaptation layer on top of `conv4-23`. This demonstrates that the adaptation layer extracts features more adequate for pixel-wise semantic correspondences, boosting performance of all baseline models significantly. In particular, we can see that the kernel soft argmax outperforms others by a large margin, since it enables training our model end to end including adaptation layers at a sub-pixel level and is less susceptible to multi-modal distributions. The last three rows suggest that exploiting deeper features is important, and using all components with the kernel soft argmax performs best in terms of the average PCK. We show in Fig. 11 alignment examples for the variants of our model in Table 9. This confirms once more the results in Table 9 that the adaptation layers and exploiting multi-level features boost the matching performance drastically, regardless of types of argmax operators, and the soft argmax is highly susceptible to multi-modal distributions, e.g., caused by ambiguous matches between a bottle and a glass in the source and target images, respectively.

### 4.5.3 Training with Bounding Boxes

We train our model using object bounding boxes themselves as binary masks. The generated masks are noisy, but are less expensive to annotate than ground-truth foreground masks. We use the same 2,791 images from the PASCAL VOC 2012 segmentation dataset [42] for training, and obtain an average PCK ($\alpha_{\text{bbox}} = 0.1$) of 77.9 on the PF-PASCAL dataset [39], which is comparable with the score of 78.7 using ground-truth masks. This suggests that using bounding boxes might be a less accurate but cheaper alternative.

### 4.5.4 Training on PF-PASCAL

Semantic correspondence methods based on CNNs use different training sets. For example, the methods of [23], [24] use the PASCAL VOC 2011 (11,540 images) and Tokyo Time Machine datasets (20,000 images). In [17], [24], [26], [43], [44], the training split of PF-PASCAL [39] (about 700 image pairs for 1,001 images) is used. Following these approaches, we train a network on the training split in the PF-PASCAL dataset. We exclude 302 images in this split that overlap with either target or source images in the test dataset. Note that current methods ignore this bias. We use object bounding boxes due to the lack of ground-truth foreground masks in the training split. We obtain the average PCK ($\alpha_{\mathrm{bbox}} = 0.1$) of 77.8 on PF-PASCAL, which is comparable with the score of 78.7 for the model trained using 1,464 images on the PASCAL VOC 2012 segmentation dataset. This indicates that our model is robust to the size of training data.

### 4.5.5 Training on Larger Datasets

We use the training split of MS COCO 2014 [67] to train our model on a larger dataset. Among 80 object categories, we select 16,624 images of 20 object classes of PASCAL VOC 2012 [42] using segmentation masks, which is about 6 times the number used in Section 4.2 (2,791 images). We test our model on the PF-PASCAL dataset [39], since MS COCO does not provide benchmarks for semantic correspondence. Despite using a larger number of training samples, the average PCK ($\alpha_{\mathrm{bbox}} = 0.1$) decreases slightly from 78.7 to 77.1, mainly due to domain differences between MS COCO and PASCAL VOC datasets. This, however, demonstrates once more the generalization ability of our approach to samples outside the training domain.

## 5 Conclusion

We have presented a CNN model dubbed SFNet for learning an object-aware semantic flow end to end, with a novel kernel soft argmax layer that outputs differentiable matches at a sub-pixel level. We have proposed to use binary foreground masks that are widely available and can be obtained easily compared to pixel-level annotations to train a model for learning pixel-to-pixel correspondences. The ablation studies clearly demonstrate the effectiveness of each component and loss in our model. Finally, we have shown that the proposed method is robust to distracting details and focuses on establishing dense correspondences between prominent objects, outperforming the state of the art on standard benchmarks for the tasks of semantic correspondence, mask transfer, and pose keypoint propagation in terms of accuracy and speed.

## Acknowledgments

This research was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2019R1A2C2084816), the Louis Vuitton/ENS chain on artificial intelligence, the Inria/NYU collaboration agreement, and the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

## References

[1] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 353–363, 1993.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 25–36.

[3] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.

[4] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, 2013.

[5] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 41–48.

[6] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1792–1799.

[7] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2307–2314.

[8] T. Taniai, S. N. Sinha, and Y. Sato, "Joint recovery of dense correspondence and cosegmentation in two images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 4246–4255.

[9] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister, "Image restoration using online photo collections," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2217–2224.

[10] 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. Comput. Vis. Pattern Recog.*, 2015, pp. 1191–1200.

[11] J. Hur, H. Lim, C. Park, and S. Chul Ahn, "Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1392–1400.

[12] H. Bristow, J. Valmadre, and S. Lucey, "Dense semantic correspondence where every pixel is a classifier," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4024–4031.

[13] H. Yang, W.-Y. Lin, and J. Lu, "DAISY filter flow: A generalized discrete approach to dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3406–3413.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 886–893.

[16] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, 2010.

[17] K. Han, R. S. Rezende, B. Ham, K.-Y. K. Wong, M. Cho, C. Schmid, and J. Ponce, "SCNet: Learning semantic correspondence," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1831–1840.

[18] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2414–2422.

[19] D. Novotnỳ, D. Larlus, and A. Vedaldi, "AnchorNet: A weakly supervised network to learn geometry-sensitive features for semantic matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 5277–5286.

[20] A. Kanazawa, D. W. Jacobs, and M. Chandraker, "WarpNet: Weakly supervised matching for single-view reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3253–3261.

[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. Comput. Vis. Pattern Recog.*, 2017, pp. 6560–6569.

[22] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros, "Learning dense correspondence via 3D-guided cycle consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 117–126.

[23] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 6148–6157.

[24] ——, "End-to-end weakly-supervised semantic alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6917–6925.

[25] P. H. Seo, J. Lee, D. Jung, B. Han, and M. Cho, "Attentive semantic alignment with offset-aware correlation kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 349–364.

[26] S. Jeon, S. Kim, D. Min, and K. Sohn, "PARN: Pyramidal affine regression networks for dense semantic correspondence estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 351–366.

[27] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3475–3484.

[28] J. Lee, D. Kim, J. Ponce, and B. Ham, "SFNet: Learning object-aware semantic correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2278–2287.

[29] J. Min, J. Lee, J. Ponce, and M. Cho, "Hyperpixel flow: Semantic correspondence with multi-layer neural features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3395–3404.

[30] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3192–3199.

[31] F. Yang, X. Li, H. Cheng, J. Li, and L. Chen, "Object-aware dense semantic correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2777–2785.

[32] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[33] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu, "Scale-space SIFT flow," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 1112–1119.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[37] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1601–1609.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[39] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1711–1725, 2018.

[40] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1365–1372.

[41] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3d object detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 75–82.

[42] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[43] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1651–1662.

[44] S. Kim, S. Lin, S. Jeon, D. Min, and K. Sohn, "Recurrent transformer networks for semantic correspondence," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 6126–6136.

[45] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralbwa, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2921–2929.

[46] S. Honari, P. Molchanov, S. Tyree, P. Vincent, C. Pal, and J. Kautz, "Improving landmark localization with semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1546–1555.

[47] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 66–75.

[48] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[49] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1592–1599.

[50] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 270–279.

[51] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. 32-nd AAAI Conf. Artif. Intell.*, 2018.

[52] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 36–53.

[53] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[54] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, 2006.

[55] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self-supervised learning of geometrically stable features through probabilistic introspection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3637–3645.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[57] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[58] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2878–2890, 2012.

[59] S. Kim, D. Min, B. Ham, S. Lin, and K. Sohn, "FCSS: Fully convolutional self-similarity for dense semantic correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 581–595, 2019.

[60] S. Kim, D. Min, S. Lin, and K. Sohn, "DCTM: Discrete-continuous transformation matching for semantic flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4529–4538.

[61] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "DeepMatching: Hierarchical deformable dense matching," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 300–323, 2016.

[62] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2566–2576.

[63] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2462–2470.

[64] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, "Tracking emerges by colorizing videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 391–408.

[65] S. Zuffi, O. Freifeld, and M. J. Black, "From pictorial structures to deformable structures," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2012, pp. 3546–3553.

[66] D. F. Fouhey, W.-c. Kuo, A. A. Efros, and J. Malik, "From lifestyle vlogs to everyday interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4991–5000.

[67] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.