Deep Semantic Matching with Foreground Detection and Cycle-Consistency

Yun-Chun Chen[1,4] Po-Hsiang Huang[1] Li-Yu Yu[1]
Jia-Bin Huang[2] Ming-Hsuan Yang[3] Yen-Yu Lin[4]

[1]NTUEE [2]Virginia Tech [3]UC Merced [4]Academia Sinica

Abstract. Establishing dense semantic correspondences between object instances remains a challenging problem due to background clutter, significant scale and pose differences, and large intra-class variations. In this paper, we present an end-to-end trainable network for learning semantic correspondences using only matching image pairs without manual keypoint correspondence annotations. To facilitate network training with this weaker form of supervision, we 1) explicitly estimate the foreground regions to suppress the effect of background clutter and 2) develop cycle-consistent losses to enforce the predicted transformations across multiple images to be geometrically plausible and consistent. We train the proposed model on the image pairs of the PF-PASCAL dataset and evaluate the learned model on the PF-PASCAL, PF-WILLOW, and TSS datasets. Extensive experimental results show that the proposed approach achieves favorably performance compared to the state-of-the-art.

1 Introduction

Semantic matching is an important and active research topic in computer vision. Previous methods such as optical flow estimation [5,10] and stereo matching [24,30] rely on per-pixel correspondence to match across images depicting the same scene or object instance. While correspondence estimation has been studied for years, there has been a growing trend to extend the idea of matching the same objects across images to matching images covering different instances of an object category. This progress not only attracts a lot of attention but also facilitates many real-world applications ranging from object recognition [16], object co-segmentation [23,27], to 3D reconstruction [19]. However, due to the presence of background clutter, ambiguity induced by large intra-class variations, and the limited scalability of obtaining large-scale datasets with manually annotated correspondences, semantic matching remains quite challenging.

Conventional methods for semantic matching heavily rely on hand-crafted descriptors such as SIFT [16], HOG [4], or DAISY [29] as well as an effective geometric regularizer. However, these hand-crafted descriptors are pre-defined and cannot adapt themselves to the given visual domains, leading to the sub-optimal performance of semantic matching. Driven by the recent success of convolutional neural networks (CNNs), several learning-based approaches, e.g., [3,7,13,21,22], have been proposed for addressing the problem of semantic matching. While

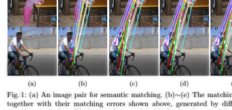

Fig. 1: (a) An image pair for semantic matching. (b)∼(e) The matching results, together with their matching errors shown above, generated by different approaches, including (b) the baseline model, (c) the baseline model with foreground detection, (d) the baseline model with cycle-consistency checking, and (e) the proposed approach.

promising results have been shown in many of the cases, these approaches still suffer from the following limitations. The methods in [3,7,13,21] require a vast amount of supervised data for training. Collecting a large-scale and diverse data, however, is expensive and labor-intensive. While weakly supervised methods such as [22] have been recently proposed to relax the issue, these approaches implicitly enforce the background features from both images to be similar. Thus, they still suffer from the unfavorable effect of background clutter.

In this work, we address the above-mentioned limitations by presenting an end-to-end trainable network for weakly supervised semantic matching, where training images covering objects of the same category are given without further manual annotations (e.g., keypoint correspondences). To alleviate the negative effect of background clutter, we integrate a foreground detection module into our semantic matching network. In this way, the effect of background clutter can be mitigated by excluding background matching. The proposed network then focuses on learning the features and geometric models for better matching the detected foreground, resulting in sizable improved performance. To address the matching difficulties caused by complex image appearance and large intra-class variations, we propose to narrow down the matching space by filtering out correspondences with geometric inconsistency. To this end, we exploit the property that correct correspondences should be *cycle-consistent* meaning that when matching a particular point from one image to the other and then performing a reverse mapping, we should arrive at the same spot. We further extend this idea to exploit transitivity consistency across multiple images.

Fig. 1 shows an image pair and the matching results generated by using the baseline model, the baseline model with foreground detection, the baseline model with cycle-consistency checking, and the proposed method (i.e., the base-

line model with both foreground detection and cycle-consistency checking). The numbers on the top of Fig. 1(b)-(e) are the errors measured by the average distance between the predicted keypoints and their corresponding ground truths. It can be observed in Fig. 1(c) that the unfavorable effect of background clutter is alleviated since most correspondences in the background are removed. Fig. 1(d) shows that exploiting cycle-consistency constraints helps exclude ambiguous matching, resulting in significant reduction of matching errors. Our method simultaneously integrates foreground detection and cycle-consistency checking into semantic matching. As shown in Fig. 1(e), the quality of matching, measured by the matching error, is further enhanced.

The main contributions of this work are summarized as follows. First, we present an end-to-end trainable network that integrates foreground detection into semantic matching. With a module for explicit foreground detection, the proposed network suppresses the unfavorable effect of background clutter. Second, our model implicitly tackles the ambiguity induced by vast matching space by inferring bi-directional geometric transformations during matching. With these transformations, we explicitly enforce the inferred geometric transformations to be cycle-consistent by introducing the *forward-backward consistency loss*. In addition, we exploit the property of transitivity consistency and introduce the *transitivity loss* to further enhance the matching performance. We train our network with the image pairs of the PF-PASCAL dataset [6]. We then evaluate the proposed model on several standard benchmark datasets for semantic matching, including the PF-PASCAL [6], PF-WILLOW [6], and TSS [27] datasets. Extensive comparisons with existing semantic matching algorithms demonstrate that the proposed approach achieves the state-of-the-art performance.

2 Related Work

Semantic matching has been extensively studied in the literature. Here, we review several topics pertinent to our approach.

Semantic Correspondence. Conventional approaches to semantic matching [1,4,16,29] leverage hand-crafted descriptors such as SIFT or HOG along with geometric matching models. They seek the keypoint correspondences across images by optimizing a given energy function. The SIFT Flow [16] method shares a similar idea with optical flow, which aligns two images in a large corpus, to establish correspondences. It further employs the SIFT descriptor to extract semantic information from input images and introduces a coarse-to-fine search algorithm for efficient matching. Yang *et al.* [29] adopt DAISY as the descriptor to efficiently perform correspondence field estimation. Kim *et al.* [12] learn dense correspondence by proposing the deformable spatial pyramid. Ham *et al.* [6] introduce the Proposal Flow where the HOG features and object proposals are used as the matching primitives to learn semantic correspondence. With the use of object proposal, the Proposal Flow method is robust to scaling and background clutter. Taniai *et al.* [27] propose a model based on hierarchical Markov random field, where object co-segmentation and dense correspondences are jointly recovered.

However, the aforementioned methods are established based on hand-crafted descriptors, which may limit their generalization capability.

Semantic Correspondence via Deep Learning. Convolutional neural networks have been applied to semantic matching for the superior performance of feature extraction. Choy *et al.* [3] propose the universal correspondence network (UCN) along with a correspondence contrastive loss. Their method adopts a convolutional spatial transformer for feature transformations, which makes their method robust to scaling and rotations. In [13], Kim *et al.* propose a CNN-based descriptor called fully convolutional self-similarity (FCSS) and combine the descriptor with the Proposal Flow framework [6] for image matching. The SCNet developed by Han *et al.* [7] learns a geometrically plausible model for semantic correspondence by incorporating geometric consistency constraints into the loss function. While the methods in [7,13] employ trainable descriptors for semantic correspondence, feature matching is learned at the *object-proposal* level. Consequently, these methods are not end-to-end trainable because a fusion step is required to produce the final results. Rocco *et al.* [21] present a CNN-based architecture for geometric matching which estimates a parametric geometric model that can be converted to dense pixel correspondence in an end-to-end trainable fashion. Although these methods [3,6,7,13,21] perform better than those based on hand-crafted features, they need supervised data (in terms of manually labeled keypoint correspondence) for training. The dependency of manual supervision restricts the scalability.

Recently, a few CNN-based methods [11,20,22,28] have carried out weakly supervised semantic correspondence. Novotny *et al.* [20] propose the AnchorNet which learns a set of filters whose response is geometrically consistent across different object instances. However, this model is combined with hand-crafted alignment models, and therefore is not end-to-end trainable. The WarpNet [11] learns fine-grained image matching with small scale and pose variations via aligning objects across images through known deformation. James *et al.* [28] align a set of images by projecting image pixels to a common coordinate system. Although their method has been shown to be effective under well-controlled environments, it may not generalize well to real-world scenarios. Inspired by the inlier scoring procedure of RANSAC, Rocco *et al.* [22] propose an end-to-end trainable alignment network, which computes dense semantic correspondence while aligning two images. Our proposed method differs from [20,22] in two ways. First, our method further takes into account foreground detection. Our network learns feature embedding to enhance inter-image foreground similarity while alleviating the unfavorable effects caused by complex background. Second, we introduce bi-directional transformations and further leverage cycle consistency for enforcing geometrically consistent predictions.

Foreground Detection. We review a few methods related to foreground detection. For saliency detection, Xia *et al.* [31] develop a center surround inference network that detects salient regions of an image in an unsupervised way. Zhang *et al.* [36] propose a deep learning framework that leverages intra-saliency

prior transfer and deep inter-saliency mining to perform co-saliency detection among groups of images. For semantic segmentation, Long *et al.* [17] develop a fully convolutional network (FCN) which can be trained end-to-end and pixels-to-pixels to segment objects of interest. The PSPNet [37] by Zhao *et al.* extends the idea of feature pyramid for semantic segmentation in a coarse-to-fine fashion. He *et al.* [8] introduce a framework that can not only detect objects but generate high-quality masks for foreground objects. The proposed method leverages foreground detection to identify the regions for matching. Thus, it can work well on images with complex background.

Cycle Consistency. Leveraging cycle consistency to regularize learning has been studied. Sundaram *et al.* [26] exploit forward-backward consistency to tackle visual tracking. The CycleGAN [41] method proposed by Zhu *et al.* shares the same idea. It couples the network with an inverse mapping to deal with unpaired image-to-image translation. Meister *et al.* [18] propose an unsupervised learning framework, called UnFlow, which estimates bi-directional optical flow to explicitly reason about occlusion and make use of the census transform to increase robustness. While the idea of cycle consistency has been widely applied to various vision tasks, several methods [38–40] share the same idea in the context of semantic matching. Zhou *et al.* [40] tackle the problem of matching multiple images by jointly optimizing feature matching and enforcing cycle consistency. The FlowWeb [38] learns image alignment by establishing globally-consistent dense correspondences via exploiting cycle consistency constraints. However, these methods [38,40] employ hand-engineered descriptors which cannot adapt to an arbitrary object category given for matching. Zhou *et al.* [39] establish dense correspondences by using an additional 3D CAD model to form a cross-instance loop between synthetic data and real images. However, their cycle consistency loss requires four images at a time. On the contrary, we develop two loss functions to enforce cycle consistency and do not need additional data to guide the training. Experimental results demonstrate that by exploiting cycle consistency constraints, the proposed method improves the performance.

3 Proposed Algorithm

In this section, we first give an overview of our approach with the developed objective function. Then, each loss adopted in the objective function is described. Finally, the implementation details are provided.

3.1 Framework Overview

Let $\mathcal{D} = \{I_i\}_{i=1}^N$ denote a set of images covering instances of the same object category, where I_i is the i^{th} image and N is the number of images. Our goal is to learn a CNN-based model that can determine the keypoint correspondences between each image pair (I_A, I_B) in $\mathcal{D}$ *without knowing the object class in advance.* Our formulation for semantic matching is *weakly-supervised* since training our model requires only weak image-level supervision in the form of training image pairs containing common objects. No ground truth correspondences are used.

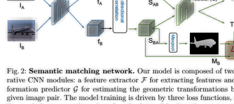

Fig. 2: **Semantic matching network.** Our model is composed of two collaborative CNN modules: a feature extractor $\mathcal{F}$ for extracting features and a transformation predictor $\mathcal{G}$ for estimating the geometric transformations between a given image pair. The model training is driven by three loss functions, including the masked correspondence loss $\mathcal{L}_M$, the forward-backward consistency loss $\mathcal{L}_F$, and the transitivity loss $\mathcal{L}_T$.

To accomplish this task, we present an end-to-end trainable network which is composed of two collaborative CNN modules: the feature extractor $\mathcal{F}$ and the transformation predictor $\mathcal{G}$. The former is a CNN model which learns and extracts the features for a given pair of images. The latter is a CNN-based regressor. For an image pair, it estimates the transformation that warps an image so that the warped image can better align the other image. Fig. 2 presents the two collaborative CNN modules in the proposed network architecture.

As shown in Fig. 2, the proposed network architecture takes an image pair for semantic matching. For each image pair (I_A, I_B) in $\mathcal{D}$, they are fed into the feature extractor $\mathcal{F}$ to extract their feature maps f_A and f_B, respectively. We then perform correlation from f_A to f_B to generate the correlation map S_{AB}. The other correlation map S_{BA} is symmetrically obtained. Then the transformation predictor $\mathcal{G}$ estimates the geometric transformation T_{AB} which warps I_A so that the warped image $\tilde{I}_A$ can align I_B. In the following section, we develop the objective function used to optimize the feature extractor $\mathcal{F}$ and the transformation predictor $\mathcal{G}$. After optimization, the matching between an image pair (I_A, I_B) can be performed via the predicted transformation T_{AB} or T_{BA}.

3.2 Objective Function

The overall training objective consists of three loss functions. First, the masked correspondence loss $\mathcal{L}_M$ minimizes the distance between the corresponding features based on the estimated geometric transformations. Unlike existing semantic matching methods [21,22], our model predicts foreground masks to suppress the effect of background clutter by excluding background matching. Second, the forward-backward consistency loss $\mathcal{L}_F$ and the transitivity loss $\mathcal{L}_T$ enforce the predicted transformations across multiple images to be geometrically plausible and consistent. Both losses regularize the network training. Specifically, the

training objective is defined by

$$\mathcal{L} = \mathcal{L}_M + \lambda_F \cdot \mathcal{L}_F + \lambda_T \cdot \mathcal{L}_T, \quad (1)$$

where λ_F and λ_T are hyper-parameters used to control the relative importance of the respective loss functions. The details of each loss function are described in the following.

3.3 Masked Correspondence Loss

To reduce the effect of background clutter and enforce only foreground regions to be similar, our model minimizes the masked correspondence loss. Given an image pair (I_A, I_B), the feature extractor $\mathcal{F}$ extracts their respective feature maps $f_A \in \mathbb{R}^{h_A \times w_A \times d}$ and $f_B \in \mathbb{R}^{h_B \times w_B \times d}$, where d is the number of channels. We correlate f_A with f_B to generate the correlation map $S_{AB} \in \mathbb{R}^{h_A \times w_A \times h_B \times w_B}$. Each element $S_{AB}(i, j, s, t) = S_{AB}(\mathbf{p}, \mathbf{q})$ records the normalized inner product between the feature vectors stored at two spatial locations $\mathbf{p} = [i, j]^\top$ in f_A and $\mathbf{q} = [s, t]^\top$ in f_B. The other correlation map $S_{BA} \in \mathbb{R}^{h_B \times w_B \times h_A \times w_A}$ can be yielded symmetrically. The correlation map S_{AB} is reshaped to a third-dimensional tensor with dimensions h_A, w_A, and $(h_B \times w_B)$, i.e., $S_{AB} \in \mathbb{R}^{h_A \times w_A \times (h_B \times w_B)}$. As such, it can be interpreted as a dense $h_A \times w_A$ grid with $(h_B \times w_B)$-dimensional local features. The reshape operation is applied to S_{BA} as well. With the reshaped S_{AB}, we use the transformation predictor $\mathcal{G}$ [21] to estimate a geometric transformation T_{AB} which warps I_A to $\tilde{I}_A$ so that $\tilde{I}_A$ aligns well to I_B.

Since the correlation map $S_{AB}(\mathbf{p}, \mathbf{q})$ records the normalized inner product between two feature vectors located at $\mathbf{p}$ in f_A and $\mathbf{q}$ in f_B. Our model estimates the foreground mask $M_A \in \mathbb{R}^{h_A \times w_A}$ by

$$M_A(\mathbf{p}) = \max_{\mathbf{q}}(S_{AB}(\mathbf{p}, \mathbf{q})). \quad (2)$$

Note that both the correlation maps S_{AB} and S_{BA} are compiled through a rectified linear unit (ReLU) to eliminate negative matching values in advance. Therefore, the value of the estimated foreground masks at each pixel will be bounded between 0 and 1. Intuitively, $M_A(\mathbf{p})$ has a low value (i.e., location $\mathbf{p}$ is likely to belong to background) if none of the feature vectors in f_B matches well with $f_A(\mathbf{p})$. Likewise, M_B can be obtained.

With the estimated geometric transformation T_{AB}, we can identify and filter out geometrically inconsistent correspondences. Consider a correspondence $(\mathbf{p} \in \mathcal{P}_A, \mathbf{q} \in \mathcal{P}_B)$, where $\mathcal{P}_A$ and $\mathcal{P}_B$ are the sets of all spatial coordinates of f_A and f_B, respectively. The distance $\|T_{AB}(\mathbf{p}) - \mathbf{q}\|$ represents the projection error of this correspondence with respect to transformation T_{AB}. Following [22], we introduce a correspondence mask m_A to determine if the correspondences are geometrically consistent with transformation T_{AB}. Specifically, m_A is defined by

$$m_A(\mathbf{p}, \mathbf{q}) = \begin{cases} 1, & \text{if } \|T_{AB}(\mathbf{p}) - \mathbf{q}\| \le \varphi, \\ 0, & \text{otherwise.} \end{cases}, \text{ for } \mathbf{p} \in \mathcal{P}_A \text{ and } \mathbf{q} \in \mathcal{P}_B, \quad (3)$$

where φ is the predefined threshold. Note that the correspondence mask m_A is introduced to remove the correspondences that are geometrically inconsistent with the transformation T_{AB}, namely those with the projection errors larger than φ. Empirically, we set φ to 1 in our experiments.

Given the geometric transformation T_{AB} and correspondence mask m_A, the likelihood of each spatial location $\mathbf{p} \in \mathcal{P}_A$ being matched is computed by

$$s_A(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{P}_B} m_A(\mathbf{p}, \mathbf{q}) \cdot S_{AB}(\mathbf{p}, \mathbf{q}). \quad (4)$$

To suppress the effect of background clutter, we incorporate the estimated foreground masks to focus on matching foreground regions. The masked correspondence loss is defined by

$$\mathcal{L}_M(I_A, I_B; \mathcal{F}, \mathcal{G}) = -\Big(\sum_{\mathbf{p} \in \mathcal{P}_A} s_A(\mathbf{p}) \cdot M_A(\mathbf{p}) + \sum_{\mathbf{q} \in \mathcal{P}_B} s_B(\mathbf{q}) \cdot M_B(\mathbf{q}) \Big). \quad (5)$$

Note that the negative sign in (5) is used in the objective function, since maximizing the matching score is equivalent to minimizing the loss $\mathcal{L}_M$.

3.4 Cycle Consistency

For a pair of images I_A and I_B, the transformation predictor $\mathcal{G}$ estimates a geometric transformation T_{AB}, which transforms pixel coordinates from I_A to I_B. However, the large capacity of $\mathcal{G}$ often leads to a circumstance where various transformations can warp I_A to $\tilde{I}_A$ which aligns I_B very well. This implies that using the masked correspondence loss alone is not sufficient to reliably train $\mathcal{G}$ in the weakly supervised setting since there are no supervised correspondences to constrain the transformations. We address this issue by simultaneously estimating T_{AB} and T_{BA} and enforce the predicted transformations to be geometrically plausible and consistent across multiple images. It greatly reduces the feasible space of transformations and can serve as a regularization term in training the transformation predictor $\mathcal{G}$. To this end, we develop two loss functions where cycle consistency constraints are done in conjunction with the proposed method such that the model is end-to-end trainable. The developed loss functions are described below.

Forward-Backward Consistency Loss. Consider the correlation maps S_{AB} and S_{BA} generated from images I_A and I_B. The forward consistency states that property $T_{BA}(T_{AB}(\mathbf{p})) \approx \mathbf{p}$ holds for any $\mathbf{p} \in \mathcal{P}_A$. By the same token, the backward consistency means $T_{AB}(T_{BA}(\mathbf{q})) \approx \mathbf{q}$ for any $\mathbf{q} \in \mathcal{P}_B$. The resultant forward-backward consistency loss is then defined by

$$\mathcal{L}_F(I_A, I_B; \mathcal{F}, \mathcal{G}) = \sum_{\mathbf{p} \in \mathcal{P}_A} \|T_{BA}(T_{AB}(\mathbf{p})) - \mathbf{p}\| + \sum_{\mathbf{q} \in \mathcal{P}_B} \|T_{AB}(T_{BA}(\mathbf{q})) - \mathbf{q}\|, \quad (6)$$

where $\|T_{BA}(T_{AB}(\mathbf{p})) - \mathbf{p}\|$ is the reprojection error between coordinate $\mathbf{p}$ and the reprojected coordinate $T_{BA}(T_{AB}(\mathbf{p}))$.

Transitivity Loss. The idea of forward-backward consistency between a pair of images can be further extended to the transitivity consistency across multiple images. Considering the case of three images I_A, I_B, and I_C, we estimate three geometric transformations T_{AB}, T_{BC}, and T_{AC}. Transitivity consistency in this case states that the coordinate transformation from I_A to I_C should be path invariant. Namely, for any coordinate $\mathbf{p} \in \mathcal{P}_A$, the property, $T_{BC}(T_{AB}(\mathbf{p})) \approx T_{AC}(\mathbf{p})$, holds. The transitivity loss is then expressed as

$$\mathcal{L}_T(I_A, I_B, I_C; \mathcal{F}, \mathcal{G}) = \sum_{\mathbf{p} \in \mathcal{P}_A} \|T_{BC}(T_{AB}(\mathbf{p})) - T_{AC}(\mathbf{p})\|. \quad (7)$$

3.5 Network Selection and Initialization

We try several CNN-based architectures to serve as the feature extractor $\mathcal{F}$, and finally adopt the semantic matching network proposed in [22] due to its impressive results for image alignment. The network employs the ResNet-101 [9] model. The extracted features are those generated by layer `conv4-23`. The transformation predictor $\mathcal{G}$ we select is the same as that used in [21]. It contains convolutional layers followed by a fully connected layer to regress the parameters. The transformation predictor $\mathcal{G}$ is a cascade of two modules predicting an affine transformation and a thin plate spline (TPS) transformation. Given an image pair, the model first estimates an affine transformation with 6 degrees of freedom to obtain a rough alignment. The model then performs a second-stage geometric estimation based on the roughly aligned image pair to predict TPS transformation for alignment refinement. Similar to [21], we use a uniform 3×3 grid of control points for TPS, which corresponds to $3 \times 3 \times 2 = 18$ degrees of freedom. We initialize the feature extractor $\mathcal{F}$ and the transformation predictor $\mathcal{G}$ from the parameters pre-trained in [22] and fine-tune $\mathcal{F}$ and $\mathcal{G}$ by using the proposed objective function.

4 Experimental Results

Experiments are conducted in this section. We will describe the implementation details and the experimental setting, evaluate the proposed approach, compare it with the state-of-the-art, and analyze the results. More results are provided in the supplementary material. The source code and the pre-trained models will be made available to the public.

4.1 Implementation Details

We implement our model using PyTorch. The training and validation data are both obtained from the PF-PASCAL dataset [6]. All images are resized to the resolution 240×240. We perform data augmentation by horizontal flipping, random cropping the input images, and swapping the order of images in the image pair. Our model is trained with ADAM optimizer [15] with an initial learning rate of 5×10^{-8}. Our model is learned with the forward-backward consistency loss and transitivity loss first to obtain a good initialization and then is fine-tuned with the full objective function. For transitivity loss, the input triplets are

randomly selected within a mini-batch. We sample $10 \times 10 = 100$ spatial coordinates for computing the forward-backward consistency loss and the transitivity loss. The training process takes about 2 hours on a single NVIDIA GeForce GTX 1080 graphics card.

4.2 Evaluation Metric and Datasets

We conduct the evaluation on the PF-PASCAL [6], PF-WILLOW [6], and TSS [27] benchmark datasets.

Evaluation Metric. We evaluate the performance of the proposed method on a semantic correspondence task. To assess the performance, we adopt the percentage of correct keypoints (PCK) metric [33] which measures the percentage of keypoints whose reprojection errors are below the given threshold. The reprojection error is the Euclidean distance $d(\phi(\mathbf{p}), \mathbf{p}^*)$ between the locations of the warped keypoint $\phi(\mathbf{p})$ and the ground truth keypoint $\mathbf{p}^*$. The threshold is defined as $\tau \cdot \max(h, w)$ where h and w are the height and width of the annotated object bounding box on the image, respectively.

PF-PASCAL [6]. The PF-PASCAL dataset is selected from the PASCAL 2011 keypoint annotations [2] containing 1,351 semantically related image pairs from 20 object categories. For images of a category, they contain different object instances of that category with similar poses but different appearances. In addition, the presence of background clutter makes it a challenging dataset on semantic matching. We divide the dataset into 735 pairs for training, 308 pairs for validation, and 308 pairs for testing. Manually annotated correspondences are provided for each image pairs. However, under the weakly supervised setting, the keypoint annotations are only used for evaluation. We compute the PCK for each object category with τ equals to 0.1.

PF-WILLOW [6]. The PF-WILLOW dataset is composed of 100 images with 900 image pairs divided into four semantically related subsets: car, duck, motorbike, and wine bottle. Each subset contains images with large intra-class variations and background clutters. For each image, there are 10 keypoint annotations. We follow [7] and compute the PCK at three different thresholds with τ equals to 0.05, 0.1, and 0.15, respectively.

TSS [27]. The TSS dataset comprises 400 semantically related image pairs divided into three groups, including FG3DCar, JODS, and PASCAL. FG3DCar contains 195 image pairs of automobiles. JODS is composed of 81 image pairs of airplanes, cars, and horses. 124 image pairs of trains, cars, buses, bikes, and motorbikes form the group of PASCAL. Ground truth flows for each image pair are provided. Following [27], we densely compute the PCK over foreground object by setting τ to 0.05.

4.3 Experimental Results on the PF-PASCAL Dataset

In the following, we compare the performance of the proposed method with the state-of-the-art approaches. Note that many of the existing methods require manually annotated correspondences while our model can be trained using only image-level supervision.

Table 1: **Per-class PCK on the PF-PASCAL dataset with $\tau = 0.1$.**

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	d.table	dog	horse	moto	person	plant	sheep	sofa	train	tv	mean
HOG+PF-LOM [6]	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 [3]	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
VGG-16+SCNet-A [7]	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
VGG-16+SCNet-AG [7]	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
VGG-16+SCNet-AG+ [7]	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
VGG-16+CNNGeo [21]	79.5	80.9	69.9	61.1	57.8	77.1	84.4	55.5	48.1	83.3	37.0	54.1	58.2	70.7	51.4	41.4	60.0	44.3	55.3	30.0	62.6
ResNet-101+CNNGeo(S) [21]	83.0	82.2	81.1	50.0	57.8	79.9	92.8	77.5	44.7	85.4	28.1	69.8	65.4	77.1	64.0	65.2	**100.0**	50.8	44.3	54.4	69.5
ResNet-101+CNNGeo(W) [22]	84.7	88.9	80.9	55.6	76.6	89.5	**93.9**	79.6	52.0	85.4	28.1	71.8	67.0	75.1	66.3	70.5	**100.0**	62.1	62.3	61.1	74.8
Ours	**85.6**	**89.6**	**82.1**	**83.3**	**85.9**	**92.5**	**93.9**	**80.2**	52.2	85.4	55.2	**75.2**	64.0	**77.9**	67.2	**73.8**	**100.0**	65.3	69.3	61.1	**78.0**

Performance Evaluation. We compare our approach with the ProposalFlow [6], the UCN [3], different versions of the SCNet [7], the CNNGeo with different feature extractors [21], and a weakly supervised approach proposed by Rocco *et al.* [22]. Table 1 presents the experimental results for the PF-PASCAL dataset. Our results show that the proposed approach compares favorably against state-of-the-art methods, achieving an overall PCK of 78.0% (outperforming the previous best method [22] by 3.2%). The advantage of incorporating foreground detection and enforcing cycle consistency constraints can be observed by comparing our approach with ResNet-101+CNNGeo(W) [22] since both methods utilize the same feature descriptor and are trained with image-level supervision only.

Fig. 3 presents the warping results and Fig. 5 presents the qualitative results for the PF-PASCAL dataset. To further highlight the importance of each component of the proposed method, we present an ablation study of our method in the following.

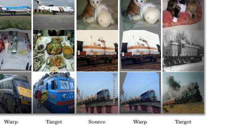

Fig. 3: **Warping.** We present the qualitative results of image warping that warps the source image to the target image.

Ablation Study. To analyze the importance of each loss function, we conduct ablation experiments on the PF-PASCAL [6] dataset. We present the PCK

Table 2: **Ablation experiments.**

Method	mean
Baseline [22]	74.8
Baseline [22] + $\mathcal{L}_M$	75.5
Baseline [22] + $\mathcal{L}_F$	77.4
Baseline [22] + $\mathcal{L}_T$	77.6
Baseline [22] + $\mathcal{L}_M$ + $\mathcal{L}_F$ + $\mathcal{L}_T$	**78.0**

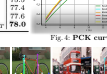

Fig. 4: **PCK curves.**

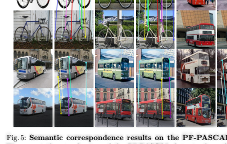

Fig. 5: **Semantic correspondence results on the PF-PASCAL dataset.** We present the visualization of the PF-PASCAL dataset where the matched coordinates are linked with color lines.

curves depicting the performance of the ablations under different thresholds. Table 2 presents the mean PCK value of variants of our approach evaluated on the PF-PASCAL dataset with τ equals to 0.1 where "Baseline" represents ResNet-101+CNNGeo(W) [22] and "Baseline + $\mathcal{L}_M$ + $\mathcal{L}_F$ + $\mathcal{L}_T$" denotes our proposed method. Our results show that both $\mathcal{L}_F$ and $\mathcal{L}_T$ substantially improve the performance when compared with the Baseline [22]. To demonstrate the effectiveness of forward-backward consistency property, we visualize three examples in Fig. 7 where the red points indicate the key points and the green points represent the reprojected points. The length of the yellow line represents the distance (loss) between the corresponding points. We observe that enforcing cycle consistency property effectively encourages the network to produce consistent predictions. However, the performance gain of using only $\mathcal{L}_M$ is modest. We believe that the reason is due to the evaluation protocol of datasets considers only the matching on the foreground region. Namely, matching a background pixel in the source

Table 3: **Results on PF-WILLOW.**

Method	$\tau = 0.05$	$\tau = 0.1$	$\tau = 0.15$
SIFT Flow [16]	0.247	0.380	0.504
DAISY w/SF [29]	0.324	0.456	0.555
DeepC w/SF [35]	0.212	0.364	0.518
LIFT w/SF [34]	0.224	0.346	0.489
VGG w/SF [25]	0.324	0.456	0.555
FCSS w/SF [13]	0.354	0.532	0.681
LOM HOG [6]	0.284	0.568	0.682
UCN [3]	0.291	0.417	0.513
SCNet-A [7]	0.390	0.725	0.873
SCNet-AG [7]	0.394	0.721	0.871
SCNet-AG+ [7]	0.386	0.704	0.853
ResNet-101+CNNGeo(S) [21]	0.448	0.777	0.899
ResNet-101+CNNGeo(W) [22]	0.477	0.812	0.917
Ours	**0.491**	**0.819**	**0.922**

Table 4: **Results on TSS.** Marker * indicates that the method uses extra images from the PASCAL VOC 2007 dataset.

Method	FG3DCar	JODS	PASCAL	Avg.
HOG+PF-LOM [6]	0.786	0.653	0.531	0.657
HOG+TSS [27]	0.830	0.595	0.483	0.636
FCSS+SIFT Flow [13]	0.830	0.656	0.494	0.660
FCSS+PF-LOM [13]	0.839	0.635	0.582	0.685
HOG+OADSC [32]*	0.875	0.708	0.729	0.771
FCSS+DCTM [14]	0.891	0.721	**0.610**	0.740
VGG-16+CNNGeo [21]	0.835	0.656	0.527	0.673
ResNet-101+CNNGeo(S) [21]	0.886	0.758	0.560	0.735
ResNet-101+CNNGeo(W) [22]	0.892	0.758	0.562	0.737
Ours	**0.898**	**0.768**	0.560	**0.742**

image to a foreground pixel in the target image will not be penalized. To demonstrate the effect of masked correspondence loss $\mathcal{L}_M$, we compute the percentage of correctly warped pixels (i.e., pixels in the foreground/background regions that are correctly warped into foreground/background region). As shown in Fig. 6, our method effectively reduces the errors in matching pixels from foreground to background and vice versa. The improvement here is important in real-world applications but is not reflected in the metric used in the standard datasets.

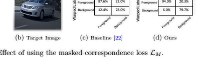

Fig. 6: Effect of using the masked correspondence loss $\mathcal{L}_M$.

The ablation study shows that all of the proposed components play crucial roles in producing accurate matching results. From Fig. 4, we observe that the proposed method outperforms the best competitor [22] with a significant margin at multiple thresholds.

4.4 Experimental Results on the PF-WILLOW and TSS Datasets

To evaluate the generalization capability, we apply the learned model trained on the PF-PASCAL dataset to test directly on the PF-WILLOW and TSS datasets. Namely, our model is not fine-tuned on each of the two datasets.

Results on the PF-WILLOW Dataset. Table 3 reports the quantitative results for the PF-WILLOW dataset. We compare the performance with several recent methods [3,7,13,21,22] as well as other methods [16,25,29,34,35] using hand-crafted features. The results are directly taken from [7] except [21,22]. For [21] and [22], we run their released code to obtain the figures. From Table 3, we observe that our model achieves the state-of-the-art performance with all three thresholds.

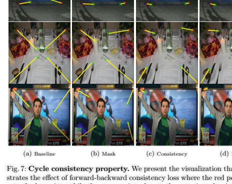

Fig. 7: **Cycle consistency property.** We present the visualization that demonstrates the effect of forward-backward consistency loss where the red points indicate the keypoints while the green points denote the reprojected points. Yellow line represents the distance (loss) between the linked points.

Results on the TSS Dataset. We also evaluate the performance on the TSS dataset. The quantitative results are presented in Table 4. It can be observed that the proposed method achieves the state-of-the-art performance on two of the three groups of the TSS dataset: FG3DCar and JODS. Our results are slightly worse than that in [32] in the PASCAL group. However, the method in [32] also uses additional images from the PASCAL VOC 2007 dataset. We report their results for completeness. Under the same experimental settings, the proposed method performs favorably against existing approaches.

5 Conclusions

We have addressed the problem of semantic matching by presenting a *weakly-supervised* and *end-to-end trainable* network. The core technical novelty lies in the explicit modeling of a *foreground detection* module to suppress the effect of background clutter and exploiting the *cycle consistency* constraints so that the predicted geometric transformations are geometrically plausible and consistent across multiple images. The network training requires only training image pairs with image-level supervision and thus significantly alleviates the cost of constructing and labeling large-scale training datasets. Experimental results demon-

strate that our approach performs favorably against existing semantic matching algorithms on several standard benchmarks. Moving forward, we believe that the semantic matching network can be further integrated to other computer vision tasks, e.g., supporting 3D semantic object reconstruction and fine-grained visual recognition.

References

1. H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.
2. L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.
3. C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *NIPS*, 2016.
4. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
5. 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 *ICCV*, 2015.
6. B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow: Semantic correspondences from object proposals. *TPAMI*, 2017.
7. K. Han, R. S. Rezende, B. Ham, K.-Y. K. Wong, M. Cho, C. Schmid, and J. Ponce. Scnet: Learning semantic correspondence. In *ICCV*, 2017.
8. K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.
9. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
10. 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 *CVPR*, 2017.
11. A. Kanazawa, D. W. Jacobs, and M. Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016.
12. J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013.
13. S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *CVPR*, 2017.
14. S. Kim, D. Min, S. Lin, and K. Sohn. Dctm: Discrete-continuous transformation matching for semantic flow. In *CVPR*, 2017.
15. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.
16. C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 2011.
17. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
18. S. Meister, J. Hur, and S. Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018.
19. A. Mustafa and A. Hilton. Semantically coherent co-segmentation and reconstruction of dynamic scenes. In *CVPR*, 2017.
20. D. Novotny, D. Larlus, and A. Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *CVPR*, 2017.

21. I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017.
22. I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. In *CVPR*, 2018.
23. M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.
24. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.
25. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.
26. N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010.
27. T. Taniai, S. N. Sinha, and Y. Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016.
28. J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NIPS*, 2017.
29. E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *TPAMI*, 2010.
30. Z.-F. Wang and Z.-G. Zheng. A region based stereo matching algorithm using cooperative optimization. In *CVPR*, 2008.
31. C. Xia, F. Qi, and G. Shi. Bottom-up visual saliency estimation with deep autoencoder-based sparse reconstruction. *TNNLS*, 2016.
32. F. Yang, X. Li, H. Cheng, J. Li, and L. Chen. Object-aware dense semantic correspondence. In *CVPR*, 2017.
33. Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *TPAMI*, 2013.
34. K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016.
35. S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.
36. D. Zhang, J. Han, J. Han, and L. Shao. Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining. *TNNLS*, 2016.
37. H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017.
38. T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, 2015.
39. T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *CVPR*, 2016.
40. X. Zhou, M. Zhu, and K. Daniilidis. Multi-image matching via fast alternating minimization. In *ICCV*, 2015.
41. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *CVPR*, 2017.