SFNet: Learning Object-aware Semantic Correspondence Supplement

Junghyup Lee[1,*] Dohyung Kim[1,*] Jean Ponce[2,3] Bumsub Ham[1,†]

[1]Yonsei University [2]DI ENS [3]INRIA

Here we present a detailed description of the kernel soft argmax and loss functions in Secs. 1 and 2, respectively, and show more quantitative comparisons with the state of the art on three benchmark datasets in Sec. 3: PF-PASCAL [6], PF-WILLOW [5], and TSS [21]. We show alignment examples of a dense flow field on the PF-PASCAL [6], PF-WILLOW [5], TSS [21], and Caltech-101 [4] datasets in Sec. 4. We then discuss issues including training with bounding boxes and with other datasets in Sec. 5.

1. Kernel soft argmax

The soft argmax [8, 9] is differentiable, but is susceptible to multi-modal distributions. It computes an output by a weighted average (*i.e.*, an expected value of all spatial coordinates weighted by corresponding probabilities $m_{\mathbf{p}}$). This approximates the discrete argmax only when the matching probability $m_{\mathbf{p}}$ is unimodal with one clear peak. The kernel soft argmax, on the other hand, makes the matching probability $m_{\mathbf{p}}$ have an (approximately) uni-modal distribution, by a 2-dimensional Gaussian kernel $k_{\mathbf{p}}$ centered on the position obtained by the discrete argmax. Note that the center position of the Gaussian kernel is changed every iterations at training time. Note also that the kernel soft argmax is differentiable, since we do not train the Gaussian kernel itself and no gradients are propagated through the discrete argmax. Figure 1 shows an example of estimating correspondences using soft and kernel soft argmax operators. We can clearly see that the soft argmax yields an incorrect correspondence in the presence of multiple highly correlated features, since a weighted average of matching probabilities having multi-modal distributions accumulates positional errors. For example, the soft argmax establishes a correspondence between the points in the background and ship. On the contrary, the Gaussian kernel in the kernel soft argmax suppresses matching probabilities except the ones around the highest value, providing a correct correspondence.

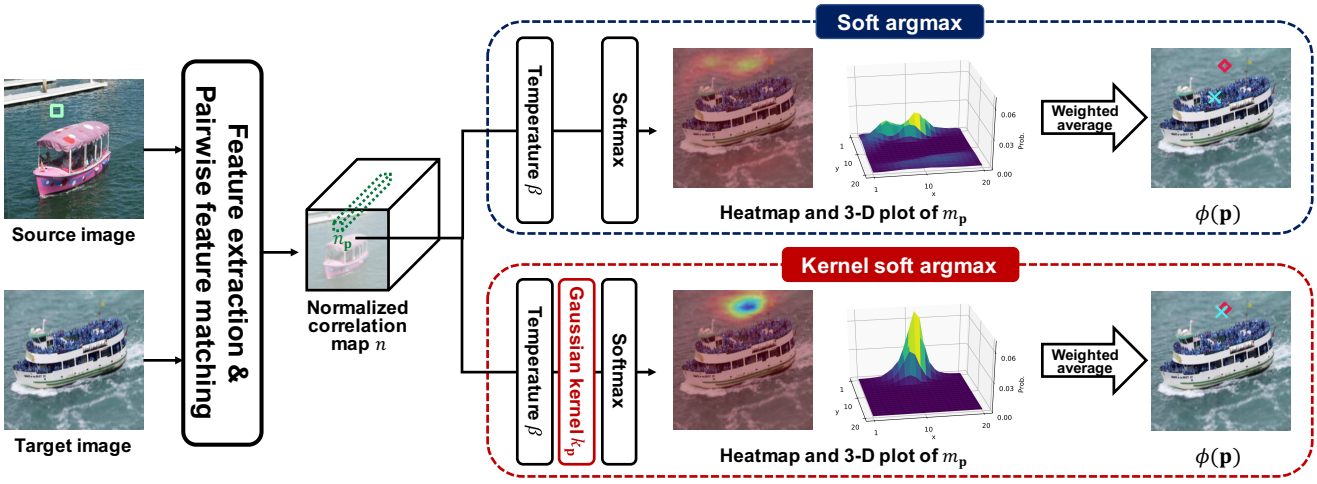


Figure 1: Visualization of soft and kernel soft argmax operations. A point to be matched in the source image is shown in the square. Correspondences computed by either soft or kernel soft argmax operators are shown in crosses. The points, denoted by diamonds, are correspondences established by the discrete argmax. When multiple features are highly correlated, the soft argmax often gives incorrect matches. The kernel soft argmax avoids this problem and approximates the discrete argmax well while maintaining differentiability. (Best viewed in color.)

*Equal contribution. †Corresponding author.
[1]School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea.
[2]Département d'Informatique de l'ENS, ENS, CNRS, PSL Research University, Paris, France.

2. Training loss

Mask and flow consistency losses. Although a mask consistency term encourages matches between features within foreground/background masks, it may cause a many-to-one matching problem. That is, it does not penalize the case when multiple points are matched to a single one (Fig. 2(a)), since binary masks do not give a positional certainty of correspondences. For example, the foreground mask in the source image can be reconstructed using a single foreground label in the target image. A flow consistency term alleviates this problem. In Fig. 2(b), all points except the center one are penalized by this term. Note that having multiple matches for individual points (*i.e.*, a one-to-many matching) is impossible within our framework. Accordingly, the flow consistency term favors a one-to-one matching (Fig. 2(c)).

Symmetric loss. Although the flow consistency loss gives a one-to-one matching, using this loss for a source or a target image only may cause a flow shrinkage problem (Fig. 3(a)). Computing this loss in a symmetric way alleviates this problem. We compute the flow consistency term w.r.t both source and target images, *i.e.*,

$$\mathcal{L}_{\text{flow}} = \sum_{\mathbf{p}} \left(||(\mathcal{F}^s(\mathbf{p}) + \hat{\mathcal{F}}^s(\mathbf{p})) \odot M^s(\mathbf{p})||_2^2 + ||(\mathcal{F}^t(\mathbf{p}) + \hat{\mathcal{F}}^t(\mathbf{p})) \odot M^t(\mathbf{p})||_2^2 \right). \quad (1)$$

The second term in (1) penalizes the inconsistent matches, *e.g.*, between the entire foreground region in the source image and small parts of the target image (Fig. 3(b)). Using the first or the second term in (1) only does not handle such cases. Note that spreading the flow fields over the entire regions is particularly important to handle scale changes between objects (Fig. 3(c)).

(a) Many-to-one matching (b) Flow consistency loss (c) One-to-one matching

Figure 2: Many-to-one matching. (a) Using the mask consistency term only may cause a many-to-one matching problem. 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.

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

Figure 3: Symmetric loss. (a) Considering the flow consistency loss w.r.t a source image may cause a flow shrinkage problem. (b) We penalize inconsistent matches by computing the loss w.r.t a target image as well. (c) This symmetric loss allows to establish an object-to-object matching. We denote by green and red arrows consistent and inconsistent matches, respectively.

3. Quantitative results

PF-WILLOW & PF-PASCAL. Table 1 shows per-class PCK scores on the PF-PASCAL dataset [6]. We compute the scores in [17, 18, 20] by the provided models (affine + TPS), and take others from [6]. Our model achieves state-of-the-art results for 19 object categories, and outperforms all methods on average by a significant margin. We compare in Table 2 the average PCK scores on the PF datasets. Following [7, 18], they are measured with keypoint coordinates normalized in the range of $[0, 1]$ by dividing them with height and width of the *image size*, respectively, which gives a threshold value larger than the one obtained by the object bounding box in Table 1, resulting in higher scores.

Method	aero	bike	bird	boat	bot	bus	car	cat	cha	cow	tab	dog	hor	mbik	pers	plnt	she	sofa	trai	tv	**Avg.**
DeepFlow [16]	0.55	0.31	0.10	0.19	0.24	0.36	0.31	0.12	0.22	0.10	0.23	0.07	0.11	0.32	0.10	0.08	0.07	0.20	0.31	0.17	0.21
GMK [2]	0.61	0.49	0.15	0.21	0.29	0.47	0.52	0.14	0.23	0.23	0.24	0.09	0.13	0.39	0.12	0.16	0.10	0.22	0.33	0.22	0.27
SIFTFlow [15]	0.61	0.56	0.20	0.34	0.32	0.54	0.56	0.26	0.29	0.21	0.33	0.17	0.23	0.43	0.18	0.17	0.17	0.31	0.41	0.34	0.33
DSP [10]	0.64	0.56	0.17	0.27	0.37	0.51	0.55	0.29	0.23	0.24	0.19	0.15	0.23	0.41	0.15	0.11	0.18	0.27	0.35	0.27	0.39
HOG+PF-LOM [5]	0.75	0.76	0.34	0.41	0.55	0.71	0.73	0.32	0.41	0.41	0.21	0.27	0.38	0.57	0.29	0.17	0.33	0.34	0.54	0.46	0.45
ResNet-101+CNNGeo [17]	0.83	0.78	0.73	0.54	0.55	0.77	0.92	0.82	0.41	0.85	0.24	0.74	0.61	0.73	0.64	0.73	0.80	0.47	0.41	0.43	0.68
ResNet-101+A2Net [20]	0.84	0.81	0.72	0.50	0.56	0.77	0.88	**0.83**	0.38	0.83	0.12	0.70	0.33	0.70	**0.72**	0.70	0.80	0.40	0.34	0.52	0.67
ResNet-101+WS-SA [18]	0.84	0.85	0.77	0.64	0.70	0.86	0.92	**0.83**	0.41	0.85	0.22	0.77	0.58	0.73	0.69	**0.76**	0.80	0.57	0.55	0.55	0.72
Proposed	**0.89**	**0.89**	**0.83**	**0.71**	**0.86**	**0.93**	**0.95**	**0.83**	**0.66**	**0.94**	**0.52**	**0.81**	**0.72**	**0.81**	**0.72**	0.71	**1.00**	**0.69**	**0.81**	**0.79**	**0.79**

Table 1: Per-class PCK on the PF-PASCAL dataset [6].

Type		Methods	PCK ($\alpha = 0.1$) WILLOW	PASCAL
CNN-based	F	(C) GoogLeNet+UCN [1]	0.42	0.56
	F	(C) VGG-16+SCNet-AG+ [7]	0.70	0.72
	A	(T) ResNet-101+CNNGeo [17]	0.81	0.72
	A	(T) ResNet-101+A2Net [20]	0.82	0.71
	A	(T+P) ResNet-101+WS-SA [18]	**0.84**	0.76
	F	(M) ResNet-101+RTNs [11]	-	0.76
	F	(P) ResNet-101+NCN [19]	-	0.79
	F	(M) ResNet-101+Ours	**0.84**	**0.82**

Table 2: Quantitative comparison with the state of the art on the PF-WILLOW [5] and the test split of the PF-PASCAL [6, 7] in terms of the average PCK. We measure the PCK scores with height and width of the image size instead of the bounding box size.

TSS. This dataset [21] 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 in [18], we compute the PCK scores densely over the foreground object where the distance threshold is set with $\alpha = 0.05$ and the height and width of the image size. Table 3 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. The PASCAL in the TSS contains many image pairs with different poses (*e.g.*, car objects captured with left- and right-side viewpoints). Current methods except for OADSC [22], that is specially designed for handling changes in viewpoints, have a limited capability of finding matches between images with different poses.

Type		Methods	FG3D.	JODS	PASC.
Hand-crafted	F	DSP [10]	0.487	0.465	0.382
	F	DFF [23]	0.493	0.303	0.224
	F	SIFTFlow [15]	0.634	0.522	0.453
	F	HOG+PF-LOM [5]	0.786	0.653	0.531
	F	HOG+TSS [21]	0.830	0.595	0.483
	F	HOG+OADSC [22]	0.875	0.708	**0.729**
CNN-based	A	(T) VGG-16+A2Net [20]	0.870	0.670	0.550
	A	(T) ResNet-101+CNNGeo [17]	0.901	0.764	0.563
	A	(T+P) ResNet-101+WS-SA [18]	0.903	0.764	0.565
	F	(B+P) FCSS+PF-LOM [12]	0.839	0.635	0.582
	F	(B+P) FCSS+DCTM [13]	0.891	0.721	0.610
	F	(M) ResNet-101+Ours	**0.906**	**0.787**	0.565

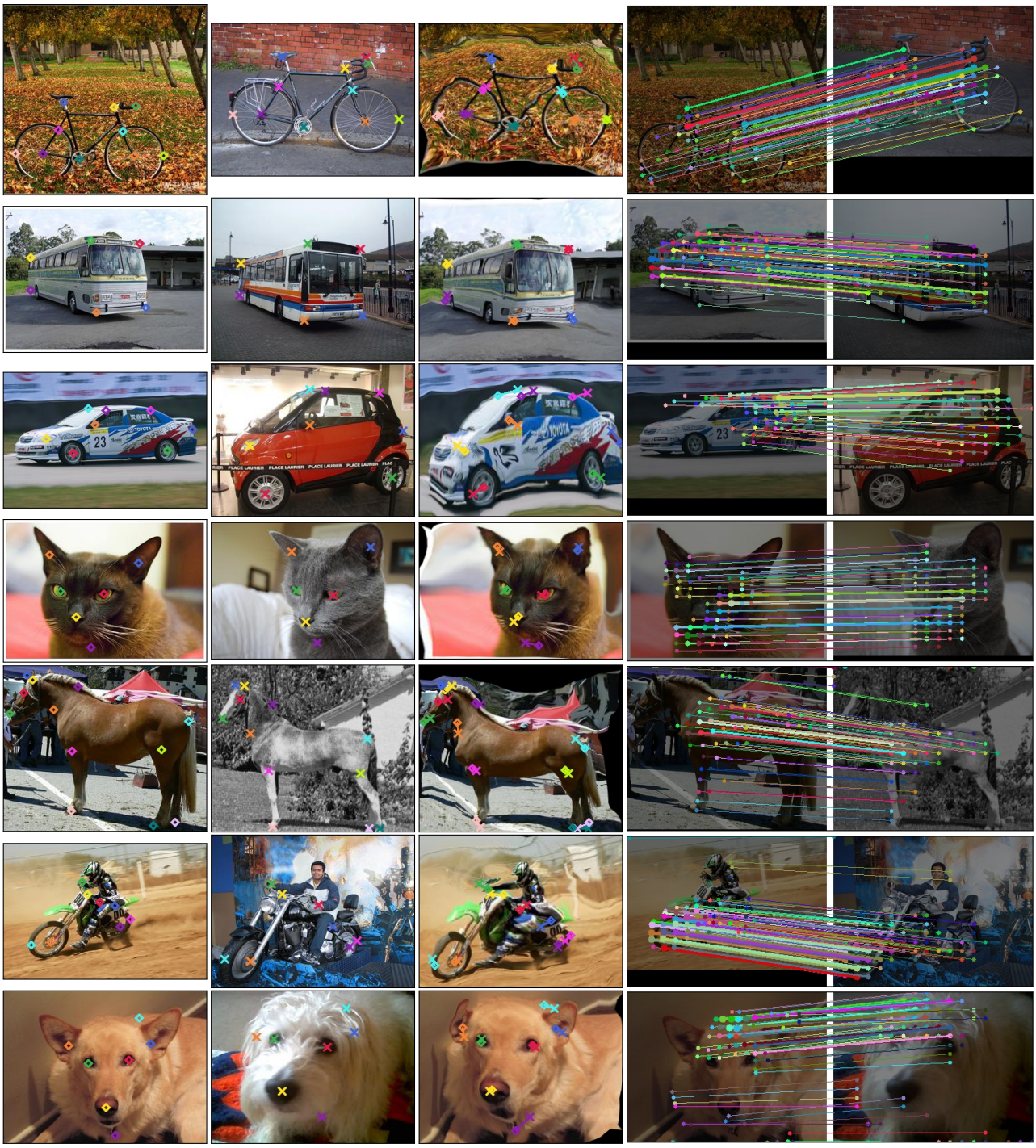
Table 3: Quantitative comparison on the TSS dataset [21]. All numbers are taken from [18, 21].

4. Aligned examples

Figures 4, 5, 6, 7 show alignment examples between source and target images on the PF-WILLOW [5], PF-PASCAL [6], TSS [21], and Caltech-101 [4] datasets, respectively. The source images are warped to the target images using dense flow fields established by our model. The alignment examples show that our method establishes semantic correspondences robust to scale changes between objects (*e.g.*, cars in the first two rows in Fig. 4), background clutter (*e.g.*, bikes in the first row in Fig. 5) and local non-rigid deformations (*e.g.*, bird's neck, body, and legs in the fifth and sixth rows in Fig. 7). In Figs. 4, 5 and 6, we also show top 60 matches chosen according to matching probabilities. We can see that most strong matches are established between prominent objects, and matches between foreground and background regions have low matching probabilities. In Fig. 7, we additionally show label transfer results overlaid on target images.

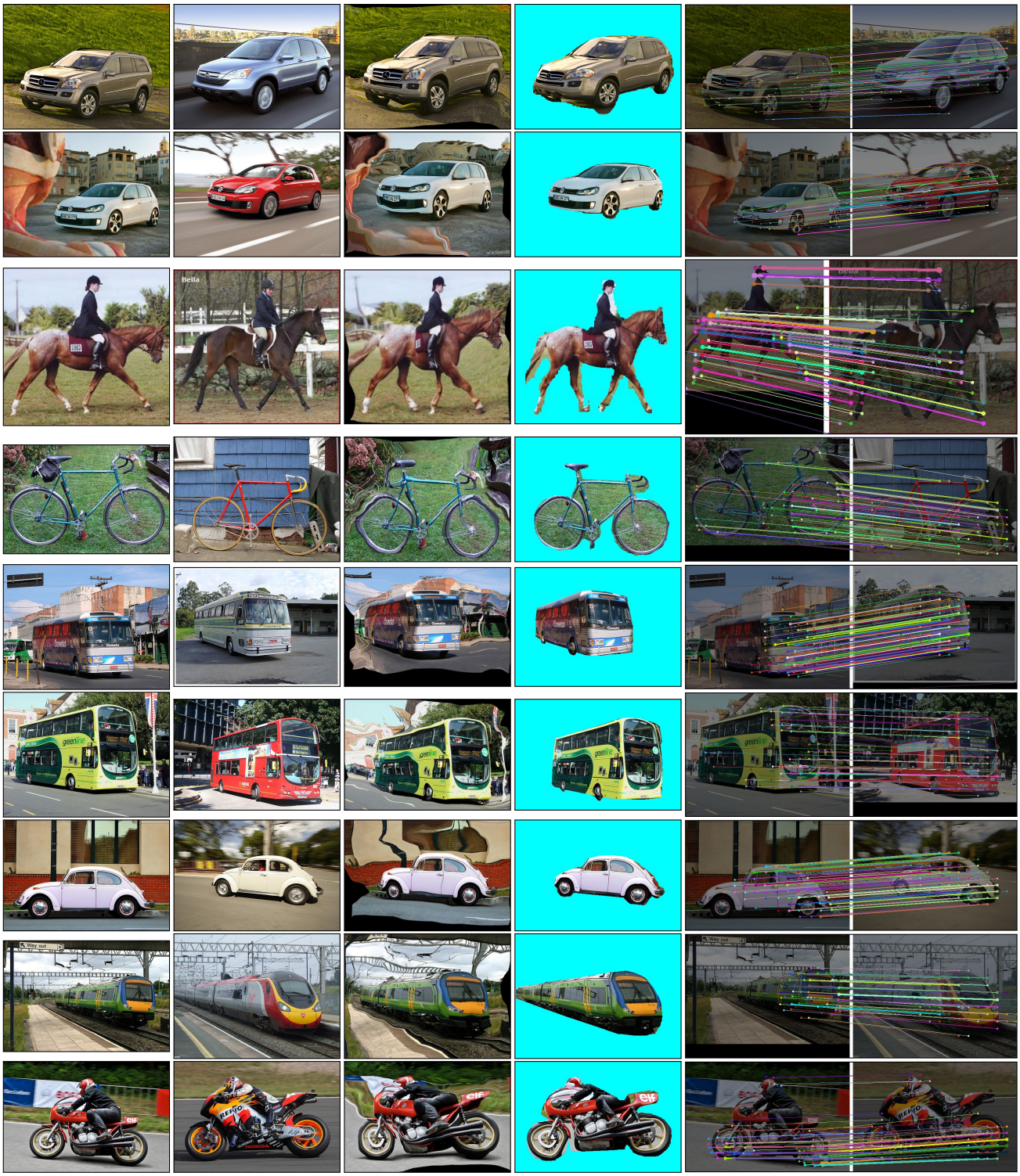
Source image. Target image. Alignment. Top 60 matches.

Figure 4: Alignment examples on the PF-WILLOW dataset [5]. Keypoints of the source and target images are shown in diamonds and crosses, respectively, with a vector representing the matching error. We visualize top 60 matches according to matching probabilities. (Best viewed in color.)



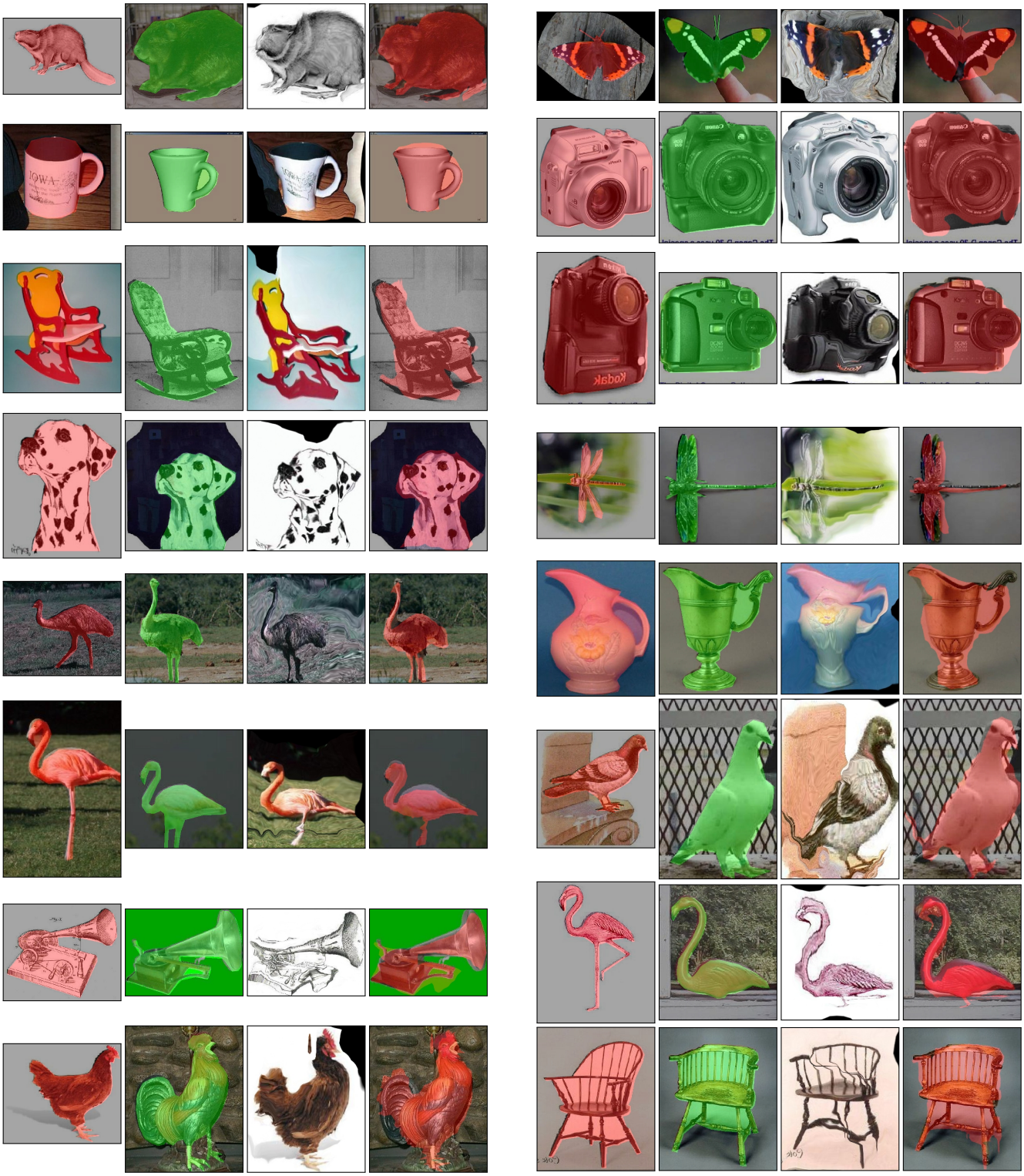
Source image. Target image. Alignment. Top 60 matches.

Figure 5: Alignment examples on the PF-PASCAL dataset [6]. Keypoints of the source and target images are shown in diamonds and crosses, respectively, with a vector representing the matching error. We visualize top 60 matches according to matching probabilities. (Best viewed in color.)



Source image. Target image. Alignment. Ground truth. Top 60 matches.

Figure 6: Alignment examples on the TSS dataset [21]. We visualize top 60 matches according to matching probabilities. (Best viewed in color.)



Source image. Target image. Alignment. Label transfer. Source image. Target image. Alignment. Label transfer.

Figure 7: Alignment examples on the Caltech-101 dataset [4]. The source and target masks are overlaid on corresponding images. We transfer pixel labels of the source images to the target ones using established correspondences. (Best viewed in color.)

5. Discussion

Parameter setting. We use a grid search with pairs of the temperature parameter β and standard deviation σ where the maximum search ranges of β and σ are 100 and 10 with intervals of 10 and 1, respectively. We then select a pair of parameters that gives the best performance on the validation split of the PF-PASCAL dataset [6, 18]. Other parameters (λ_{mask}, λ_{flow}, λ_{flow}) are similarly chosen with the validation split of the PF-PASCAL dataset [6, 18].

Training with bounding boxes. We try using the bounding boxes themselves as binary masks during training. 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 [3] for training, for an average PCK ($\alpha = 0.1$) of 0.779 on PF-PASCAL [6], which is comparable with the score of 0.787 obtained using ground-truth masks. Using flow consistency terms of (8) w.r.t both source and target images penalizes matches between background and foreground regions, making our method robust to noisy labels.

Training on PF-PASCAL. CNN-based methods use different training sets, *e.g.*, the train split of PF-PASCAL [6] (about 700 image pairs) for [7, 18] and Pascal VOC 2011 (11,540 images) for [17, 18, 20]. In [17, 18], the Tokyo Time Machine dataset (20,000 images) is also used. For fair comparison with [7], we train a network on the training split in PF-PASCAL. We excluded 302 images in this split that overlap with either target or source images in the test dataset. Note that [7, 18] ignore this bias. We use object bounding boxes due to the lack of ground-truth foreground masks in the training split. The corresponding average PCK ($\alpha = 0.1$) of 0.778 on PF-PASCAL still outperforms the state of the art (*e.g.*, 0.72 for [18] and 0.68 for [17]) by a large margin.

Training on larger datasets. To test this, we use the MS COCO 2014 training dataset [14]. Among 80 object categories, we select 16,624 images of 20 object classes of Pascal VOC 2012 [3] using segmentation masks, which is about $6\times$ the number used in the paper (2,791 images). We test our model on PF-PASCAL [6], since MS COCO does not provide benchmarks for semantic correspondence. Despite using a larger number of training samples, the average PCK ($\alpha = 0.1$) decreases slightly from 0.787 to 0.771, mainly due to domain differences between MS COCO and Pascal VOC. This, however, demonstrates once more the generalization ability of our approach to samples outside the training domain.

References

[1] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *NIPS*, 2016. 3

[2] Olivier Duchenne, Armand Joulin, and Jean Ponce. A graph-matching kernel for object categorization. In *ICCV*, 2011. 3

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) challenge. *IJCV*, 88(2), 2010. 8

[4] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE TPAMI*, 28(4), 2006. 1, 4, 7

[5] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *CVPR*, 2016. 1, 3, 4

[6] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE TPAMI*, 40(7), 2018. 1, 3, 4, 5, 8

[7] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. SCNet: Learning semantic correspondence. In *ICCV*, 2017. 3, 8

[8] Sina Honari, Pavlo Molchanov, Stephen Tyree, Pascal Vincent, Christopher Pal, and Jan Kautz. Improving landmark localization with semi-supervised learning. In *CVPR*, 2018. 1

[9] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 1

[10] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013. 3

[11] Seungryong Kim, Stephen Lin, Sangryul Jeon, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *NIPS*, 2018. 3

[12] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. FCSS: Fully convolutional self-similarity for dense semantic correspondence. In *CVPR*, 2017. 3

[13] Seungryong Kim, Dongbo Min, Stephen Lin, and Kwanghoon Sohn. DCTM: Discrete-continuous transformation matching for semantic flow. In *ICCV*, 2017. 3

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 8

[15] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE TPAMI*, 33(5), 2011. 3

[16] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. DeepMatching: Hierarchical deformable dense matching. *IJCV*, 120(3), 2016. 3

[17] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017. 3, 8

[18] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *CVPR*, 2018. 3, 8

[19] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NIPS*, 2018. 3

[20] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *ECCV*, 2018. 3, 8

[21] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016. 1, 3, 4, 6

[22] Fan Yang, Xin Li, Hong Cheng, Jianping Li, and Leiting Chen. Object-aware dense semantic correspondence. In *CVPR*, 2017. 3

[23] Hongsheng Yang, Wen-Yan Lin, and Jiangbo Lu. DAISY filter flow: A generalized discrete approach to dense correspondences. In *CVPR*, 2014. 3