Attentive Semantic Alignment
with Offset-Aware Correlation Kernels

Paul Hongsuck Seo[1], Jongmin Lee[1],
Deunsol Jung[1], Bohyung Han[2], and Minsu Cho[1]

[1] Pohang University of Science and Technology (POSTECH), Korea
[2] Dept. of ECE & ASRI, Seoul National University, Korea
{hsseo, ljm1121, hesedjds, mscho}@postech.ac.kr bhhan@snu.ac.kr

Abstract. Semantic correspondence is the problem of establishing correspondences across images depicting different instances of the same object or scene class. One of recent approaches to this problem is to estimate parameters of a global transformation model that densely aligns one image to the other. Since an entire correlation map between all feature pairs across images is typically used to predict such a global transformation, noisy features from different backgrounds, clutter, and occlusion distract the predictor from correct estimation of the alignment. This is a challenging issue, in particular, in the problem of semantic correspondence where a large degree of image variations is often involved. In this paper, we introduce an attentive semantic alignment method that focuses on reliable correlations, filtering out distractors. For effective attention, we also propose an offset-aware correlation kernel that learns to capture translation-invariant local transformations in computing correlation values over spatial locations. Experiments demonstrate the effectiveness of the attentive model and offset-aware kernel, and the proposed model combining both techniques achieves the state-of-the-art performance.

Keywords: Semantic correspondence, attention process, offset-aware correlation kernels, attentive semantic alignment, local transformation

1 Introduction

Semantic correspodence is the problem of establishing correspondences across images depicting different instances of the same object or scene class. Compared to conventional correspondence tasks handling pictures of the same scene, such as stereo matching [1,2] and motion estimation [3,4,5], the problem of semantic correspondence involves substantially larger changes in appearance and spatial layout, thus remaining very challenging. For this reason, traditional approaches based on hand-crafted features such as SIFT [6,7] and HOG [8,9,10] do not produce satisfactory results on this problem due to lack of high-level semantics in local feature representations.

While previous approaches to the problem focus on introducing an effective spatial regularizer in matching [7,11,9], recent convolutional neural networks have
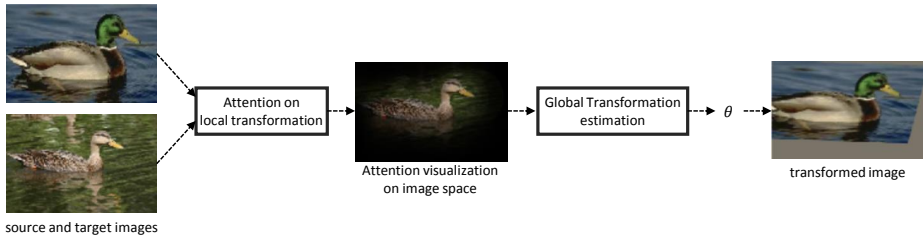
Fig. 1: The proposed attentive semantic alignment. Our model estimates dense correspondences of objects by predicting a set of global transformation parameters via an attention process. The attention process spatially focuses on the reliable local transformation features, filtering out irrelevant backgrounds and clutter.

advanced this area by learning high-level semantic features [12,13,14,15,16,17,18]. One of the main approaches [13] is to estimate parameters of a global transformation model that densely aligns one image to the other. In contrast to other approaches, it casts the whole correspondence problem for all individual features into a simple regression problem with a global transformation model, thus predicting dense correspondences through the efficient pipeline. On the other hand, however, the global alignment approach may be easily distracted; An entire correlation map between all feature pairs across images is used to predict such a global transformation, and thus noisy features from different backgrounds, clutter, and occlusion, may distract the predictor from correct estimation of the alignment. This is a challenging issue, in particular, in the problem of semantic correspondence where a large degree of image variations is often involved.

In this paper, we introduce an attentive semantic alignment method that focuses on reliable correlations, filtering out distractors as shown in Fig. 1. For effective attention, we also propose an offset-aware correlation kernel that learns to capture translation-invariant local transformations in computing correlation values over spatial locations. The resultant feature map of offset-aware correlation (OAC) kernels is computed from two input features, where each activation of the feature map represents how smoothly a source feature is transformed spatially to the target feature map. This use of OAC kernels greatly improves a subsequent attention process. Experiments demonstrate the effectiveness of the attentive model and offset-aware kernel, and the proposed model combining both techniques achieves the state-of-the-art performance.

Our contribution in this work is threefold:

- The proposed algorithm incorporates an attention process to estimate a global transformation from a set of inconsistent and noisy local transformations for semantic image alignment.
- We introduce offset-aware correlation kernels to guide the network in capturing local transformations at each spatial location effectively, and employ the kernels to compute feature correlations between two images for better representation of semantic alignment.

- The proposed network with the attention module and offset-aware correlation kernels achieves the state-of-the-art performances on semantic correspondence benchmarks.

The rest of the paper is organized as follows. We overview the related work in Section 2. Section 3 describes our proposed network with the attention process and the offset-based correlation kernels. Finally, we show the experimental results of our method and conclude the paper in Section 4 and 5.

2   Related Work

Most approaches to semantic correspondence are based on dense matching of local image features. Early methods extract local features of patches using hand-crafted feature descriptors [25] such as SIFT [11,7,26,27] and HOG [9,28,10,29]. In spite of some success, the lack of high-level semantics in the feature representation makes the approaches suffer from non-rigid deformation and large appearance changes of objects. While such challenges have been mainly investigated in the area of graph-based image matching [28,30,31,32], recent methods [15,16,17,18,19,20,21,22,23,24] rely on deep neural networks to extract high-level features of patches for robust matching. More recently, Han *et al.* [14] propose a deep neural network that learns both a feature extractor and a matching model for semantic correspondence. In spite of these developments, all these approaches detect correspondences by matching patches or region proposals based on their local features. In contrast, Rocco *et al.* [13] propose a global transformation estimation method that is the most relevant work to ours. Their model in [13] predicts the transformation parameters from a correlation map obtained by computing correlations of every pair of features in source and target feature maps. Although this model is similar to ours in that it estimates the global transformation based on correlations of feature pairs, our model is distinguished by the attention process suppressing irrelevant features and the use of the OAC kernels constructing local transformation features.

There are some related studies on other tasks using feature correlations such as optical flow estimation [3] and stereo matching [33,34]. Dosovitskiy *et al.*[3] use correlations between features of two video frames to estimate optical flow, while Zbontar *et al.*[33] and Luo *et al.*[34] extract feature correlations from patches of images for stereo matching. Although all these methods utilize the correlations, they extract correlations from features in a limited set of candidate regions. Moreover, unlike ours, they do not explore the attentive process and the offset-based correlation kernels.

Lately, attention models have been widely explored for various tasks with multi-modal inputs such as image captioning [35,36], visual question answering [37,38], attribute prediction [39] and machine translation [40,41]. In these studies, models attend to the relevant regions referred and guided by another modality such as language, while the proposed model attends based on a self-guidance. Noh *et al.* [42] use an attention process for image retrieval to extract deep local features, where the attention is obtained from the features themselves as in our work.
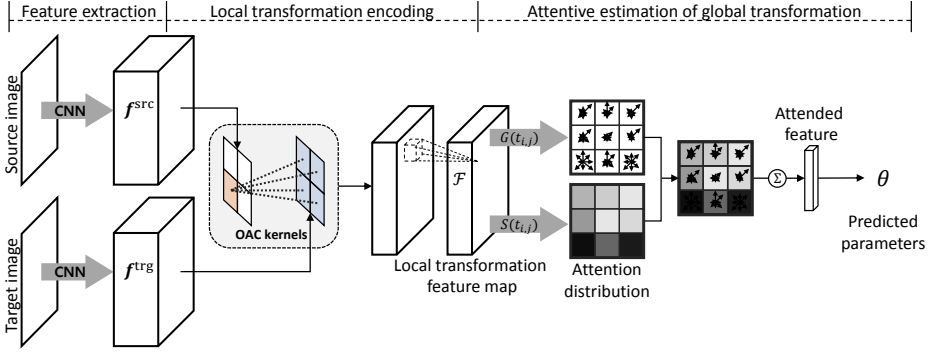
Fig. 2: Overall architecture of the proposed network. It consists of three main components: feature extractor, local transformation encoder, and attentive global transformation estimator. For details, see text.

3 Deep Attentive Semantic Alignment Network

We propose a deep neural network architecture for semantic alignment incorporating an attention process with a novel offset-aware correlation kernel. Our network takes as inputs two images and estimates a set of global transformation parameters using three main components: feature extractor, local transformation encoder, and attentive global transformation estimator as presented in Fig. 2. We describe each of these components in details.

3.1 Feature extractor

Given source and target images, we first extract their image feature maps $f^{\mathrm{src}}, f^{\mathrm{trg}} \in \mathbb{R}^{D \times H \times W}$ using a fully convolutional image feature extractor, where H and W are height and width of input images, respectively. We use a VGG-16 [43] model pretrained on ImageNet [44] and extract features from its $\texttt{pool4}$ layer. We share the weights of the feature extractor for both source and target images. Input images are resized into 240×240 and fed to the feature extractor resulting in 15×15 feature maps with 512 channels. After extracting the features, we normalize them using L_2 norm.

3.2 Local transformation encoder

Given source and target feature maps from the feature extractor, the model encodes local transformations of the source features with respect to the target feature map. The encoding is given by introducing a novel offset-aware correlation (OAC) kernel, which facilitates to overcome limitations of conventional correlation layers [13]. We briefly describe details of the correlation layer including its limitations and discuss the proposed OAC kernel.

Correlation layer The correlation layer computes correlations of all pairs of features from the source and the target images [13]. Specifically, the correlation layer takes two feature maps as its inputs and constructs a correlation map $c \in \mathbb{R}^{HW \times H \times W}$, which is given by

$$c_{i,j} = f_{i,j}^{\mathrm{src}\top} \hat{f}^{\mathrm{trg}}, \tag{1}$$

where $c_{i,j} \in c$ is a HW dimensional correlation vector at a spatial location (i,j), $f_{i,j}^{\mathrm{src}} \in f^{\mathrm{src}}$ is a feature vector at a location (i,j) of the source image, and $\hat{f}^{\mathrm{trg}} \in \mathbb{R}^{D \times HW}$ is a spatially flattened feature map of f^{trg} of the target image. In other words, each correlation vector $c_{i,j}$ consists of correlations between a single source feature $f_{i,j}^{\mathrm{src}}$ and all target features of f^{trg}. Although each element of a correlation vector maintains the correspondence likelihood of a source feature onto a certain location in the target feature map, the order of elements in the correlation vector is based on the absolute coordinates of individual target features regardless of the source feature location. This means that decoding the local displacement of the source feature requires not only the vector itself but also the spatial location of the source feature. For example, consider a correlation vector $c_{i,j} = [1, 0, 0, 0]^{\top}$ between 2×2 feature maps, each element of which is the correlation of $f_{i,j}^{\mathrm{src}}$ with $f_{0,0}^{\mathrm{trg}}$, $f_{0,1}^{\mathrm{trg}}$, $f_{1,0}^{\mathrm{trg}}$ and $f_{1,1}^{\mathrm{trg}}$. The displacement represented by the vector varies with the coordinate of the source feature (i,j). When $(i,j) = (0,0)$, it indicates that the source feature $f_{0,0}^{\mathrm{src}}$ remains at the same location $(0,0)$ in the target feature map. When $(i,j) = (0,1)$, it implies that $f_{0,1}^{\mathrm{src}}$ is moved to the left of its original location in the target feature map.

Given a correlation map, decoding the local displacement of a source feature requires incorporating the offset information from the source feature to individual target features. And, this local process is crucial for subsequent spatial attention process in the next section. Therefore, we first introduce an offset-aware correlation kernel that utilizes the offset of features during the kernel application.

Offset-aware correlation kernels Similarly to the correlation layer, our OAC kernels also take two input feature maps and utilize correlations of all feature pairs between these feature maps. The kernels naturally capture the displacement of a source feature in the target feature map by aligning kernel weights based on the offset between the source and target features for each correlation as illustrated in Fig. A. Formally speaking, an OAC kernel captures feature displacement of a source feature $f_{i,j}^{\mathrm{src}}$ by

$$h_{i,j}^{(n)} = \sum_{k=1}^{H} \sum_{l=1}^{W} w_{i-k,j-l}^{(n)} c_{i,j;k,l} \tag{2}$$

$$= \sum_{k=1}^{H} \sum_{l=1}^{W} w_{i-k,j-l}^{(n)} f_{i,j}^{\mathrm{src}\top} f_{k,l}^{\mathrm{trg}}, \tag{3}$$

where $h_{i,j}^{(n)}$ is the kernel output with the kernel index n, $c_{i,j;k,l}$ is the correlation between a source feature $f_{i,j}^{\mathrm{src}}$ and a target feature $f_{k,l}^{\mathrm{trg}}$, and $\Phi^{(n)} = \{w_{s,t}^{(n)}\}$ is

(a) kernel alignment for $h_{0,0}$ (b) kernel alignment for $h_{0,1}$
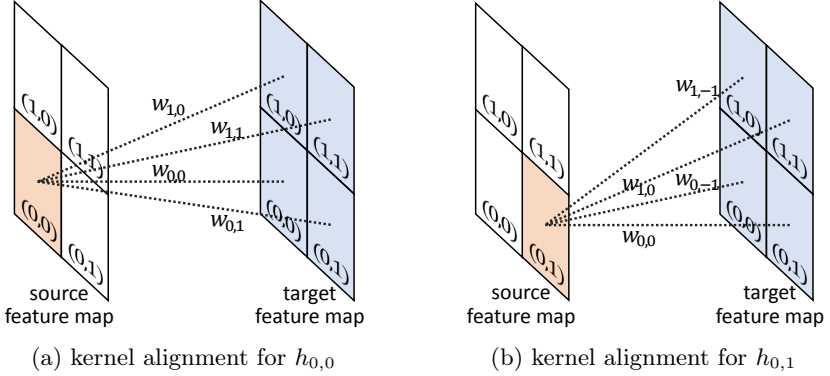
Fig. 3: Offset-aware correlation kernel at different source locations: (a) at $(0,0)$ and (b) at $(0,1)$. Each dotted line connects source and target features to compute correlation, and $w_{i,j}$ represents a kernel weight for the dotted line. Note that kernel weights are associated with different correlation pairs when source locations vary.

a set of the kernel weights. Note that the kernel weights are indexed by offset between the source and target features, and shared for correlations of any feature pair with the same offset. For example, in Fig. 3a, $w_{0,0}$ is associated with the target feature at $(0,0)$ because the source location is $(0,0)$. The same weight $w_{0,0}$ is associated with the target feature at $(0,1)$ when the source location is $(0,1)$ as in Fig. 3b because the offset between these features is $(0,0)$. Also note that each kernel output $h_{i,j}^{(n)}$ at a location (i,j) captures the displacement of its corresponding source feature $f_{i,j}^{\mathrm{src}}$ at the same location.

While a proposed kernel captures a single aspect of feature displacement, a set of the proposed kernels produce a dense feature representation of feature displacement for each source feature. We use 128 OAC kernels resulting in a feature displacement map $h \in \mathbb{R}^{128 \times 15 \times 15}$ encoding the displacement of each source feature. We set ReLU as the activation functions of the kernel outputs, and compute normalized correlations in OAC kernels since normalization further improves the scores as observed in [13].

In practice, the proposed OAC kernels are implemented by two sub-procedures. We first compute the normalized correlation map reordered based on the offsets between the locations of the source and target features. In this reordered correlation map, every correlation with the same relative displacement is arranged in the same channel. This reordering results in $(2H-1)(2W-1)$ possible offsets and thus the size of the output tensor becomes $(2H-1)(2W-1) \times H \times W$ where many of the values are zeros due to non-existing pairs for some offsets. Then, we use a 1×1 convolutional layer to compute the dense feature representation from the raw displacement information captured in the reordered correlation map. Note that this process significantly reduces the number of channels by compactly encoding various aspects of the local displacements into dense representations.
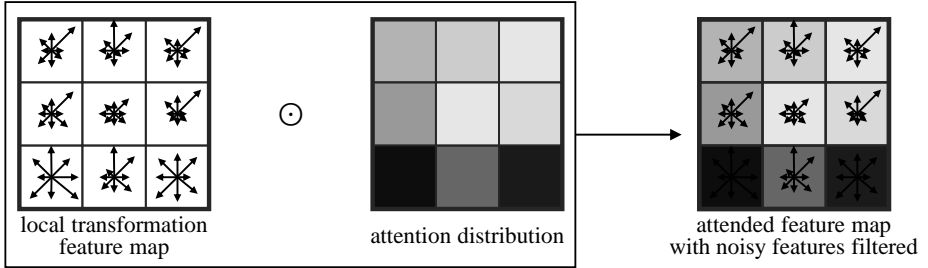
Fig. 4: Illustration of attention process. Noisy features in local transformation feature map are filtered by assigning lower probabilities to these locations. Arrows in boxes of local transformation feature map demonstrate features encoding local transformations, and grayscale colors in attention distribution represent magnitudes of probabilities where brighter colors mean higher probabilities.

Encoding local transformation features Since the feature displacement map conveys the movement of each source feature independently, each feature alone is not sufficient to predict the global transformation parameters. To allow the network predicts the global transformation from local features in the attention process, we construct a local transformation feature map by combining spatially adjacent feature displacement information captured by $\boldsymbol{h}$. That is, the proposed network feeds the feature displacement map $\boldsymbol{h}$ to a 7×7 convolution layer with 128 output channels applied without padding. This convolution layer results in a local transformation feature map $\mathcal{F} \in \mathbb{R}^{128 \times 9 \times 9}$. Note that each feature $t_{i,j} \in \mathcal{F}$ captures transformations occurred in a local region. We utilize this local transformation feature map to predict the global transformation through an attention process.

3.3   Attentive global transformation estimator

After local transformation encoding, a set of global transformation parameters is estimated with an attention process. Given a local transformation feature map $\mathcal{F} \in \mathbb{R}^{\hat{D} \times \hat{H} \times \hat{W}}$ extracted by OAC kernels with a convolution layer, the network focuses on reliable local transformation features by filtering out distracting regions as depicted in Fig. 4 to predict the parameters from the aggregation of those features. Although a feature map $\mathcal{F}$ gives sufficient information to predict the global transformation from source to target, local transformation features extracted from a real image pair is noisy due to image variations such as background clutter and intra-class variations. Therefore, we propose a model that suppresses unreliable features by the attention process and extracts an attended feature vector that summarizes local transformations from all reliable locations to estimate an accurate global transformation. In other words, the model computes

an attended transformation feature τ^{att} by

$$\tau^{\text{att}} = \sum_{i=1}^{\hat{H}} \sum_{j=1}^{\hat{W}} \alpha_{i,j} G(t_{i,j}), \tag{4}$$

where $G : \mathbb{R}^{\hat{D}} \to \mathbb{R}^{D'}$ is a projection function of $t_{i,j}$ into a D' dimensional vector space and $\boldsymbol{\alpha} = \{\alpha_{i,j}\}$ is an attention probability distribution over feature map. The model computes the attention probabilities by

$$\alpha_{i,j} = \frac{\exp\left(S(t_{i,j})\right)}{\sum_{k=1}^{\hat{H}} \sum_{l=1}^{\hat{W}} \exp\left(S(t_{k,l})\right)}, \tag{5}$$

where $S : \mathbb{R}^{\hat{D}} \to \mathbb{R}$ is an attention score function producing a single scalar given a local transformation feature. Note that the model learns to suppress noisy features by assigning low attention scores and reducing their contribution to the attended feature.

Once the attended feature τ^{att} over all local transformations is obtained, we compute the global transformation $\theta \in \mathbb{R}^{Q}$ by a simple matrix-vector multiplication as

$$\theta = W\tau^{\text{att}}, \tag{6}$$

where $W \in \mathbb{R}^{Q \times D'}$ is a weight matrix for linear projection of the attended feature τ^{att}.

In summary, we first compute local transformation between two images and perform a nonlinear embedding using a projection function $G(\cdot)$. The embedded vector is weighted by spatial attention to compute an attended feature τ^{att} as shown in Eq. (4). The global transformation vector is obtained by linear projection of the attended feature, which is parametrized by a matrix as presented in Eq. (6).

We use multi-layer perceptrons (MLPs) for G and S in Eq. (4) and (5). G is a two-layer MLP with 128 hidden and output ReLU activations. Since the feature representations produced by G is directly used for the final estimation as a linear mapping in Eq. (6), we additionally concatenate 5-dimensional index embedding to the feature $t_{i,j} \in \mathcal{F}$ to better estimate the global transformation from local transformation features. While S is another two-layer MLP with 64 hidden ReLU activations, its output is a scalar without non-linearity; this is due to the application of softmax normalization outside S. Note that we do not use the index embedding to avoid strong biases of attentions on certain regions. Since G and S are applied to all feature vectors across the spatial dimensions, we implement them by multiple 1×1 convolutions with batch normalizations.

Network training We build two of the proposed networks with different parametric global transformations: affine and thin-plate spline (TPS) transformations. To train the network, we adapt the average transformed grid distance loss proposed in [13], which indirectly measures the distance from the predicted

transformation parameters θ to the ground-truth transformation parameters θ_{GT}. Given θ and θ_{GT}, the transformed grid distance $\mathrm{TGD}(\theta, \theta_{\mathrm{GT}})$ is obtained by

$$\mathrm{TGD}(\theta, \theta_{\mathrm{GT}}) = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} d\left(\mathcal{T}_\theta(g), \mathcal{T}_{\theta_{\mathrm{GT}}}(g)\right)^2 \tag{7}$$

where $\mathcal{G}$ is a set of points in a regular grid, $\mathcal{T}_\theta$ is the transformation parameterized by θ and $d(\cdot)$ is a distance measure. We minimize the average TGD of training examples to train the network. Since every operation within the proposed network is differentiable, the network is trainable end-to-end using a gradient-based optimization algorithm. We use ADAM [45] with initial learning rate of 2×10^{-4} and batch size of 32 for 50 epochs. During training, the pretrained feature extractor is fixed and only the other parts of the network are finetuned.

4   Experiments

We evaluate the proposed method on public benchmarks for semantic correspondence estimation. The experiments demonstrate that the proposed attentive method and OAC kernels are effective in semantic alignment, substantially improving the baseline models. The codes are publicly released at http://cvlab.postech.ac.kr/research/A2Net/.

4.1   Experimental settings

Training with self-supervision While the loss function requires the full supervision of θ_{GT}, it is very expensive or even impractical to collect exact ground-truth transformation parameters for non-rigid objects involving intra-class variations. Therefore, it is hard to scale up to numerous instances and classes, restricting generalization. For example, the largest annotation dataset at this time, PF-PASCAL, only contains total 1,351 image pairs from 20 classes, and furthermore their dense annotations are extrapolated from sparse keypoints, thus being not fully exact. To work around this problem, we adopt the self-supervised learning for semantic alignment, which is free from the burden of any manual annotation, is an appealing alternative introduced in [13]. In this framework, given a public image dataset $\mathcal{D}$ without any annotations, we synthetically generate a training example $(I_{\mathrm{src}}, I_{\mathrm{trg}})$ by randomly sampling an image I_{src} from $\mathcal{D}$ and computing a transformed image I_{trg} by applying a random transformation θ_{GT}. We also use mirror padding and center cropping following [13] to avoid border artifacts. The synthetic image pairs generated by this process are annotated with the ground-truth transformation parameters θ_{GT} allowing us to train the network with full supervision. Note that, however, this training scheme can be considered unsupervised since no annotated real dataset is used during training.

For the synthetic dataset generation, we use PASCAL VOC 2011 [46], and build two variations of training datasets with either affine or TPS transformation each for its corresponding network. A set of PASCAL VOC images is kept separate to generate another set of synthetic examples for validation and the best performing models on the validation set is evaluated.

Evaluation Two public benchmarks called PF-WILLOW and PF-PASCAL [9] are used for the evaluation. PF-WILLOW consists of about 900 image pairs generated from 100 images of 5 object classes. PF-PASCAL contains 1351 image pairs of 20 object classes. Each image pair in both datasets contains different instances of the same object class such as ducks and motorbikes, *e.g.*, left two images in Fig. 1. The objects in these datasets have large intra-class variations and many background clutters making the task more challenging. The image pairs of both PF-WILLOW and PF-PASCAL are annotated with sparse key points that establishe correspondences between two images. Following the standard evaluation metric, the probability of correct keypoint (PCK) [47] of these benchmarks, our goal is to correctly transform the key points in the source image to their corresponding ones in the target image. A transformed source key point is considered correct if its distance to its corresponding target key point is less than $\alpha \cdot \max(h, w)$, where $\alpha = 0.1$, and h and w are height and width of the object bounding box. Formally, PCK of a proposed model $\mathcal{M}$ is measured by

$$
\text{PCK}(\mathcal{M}) = \frac{\sum_{i=1}^{N} \sum_{(p_\text{s}, p_\text{t}) \in \mathcal{P}_i} \mathbb{1}\left[d\left(\mathcal{T}_{\theta_i}\left(p_\text{s}\right), p_\text{t}\right) < \alpha \cdot \max(h, w)\right]}{\sum_{i=1}^{N} |\mathcal{P}_i|}, \tag{8}
$$

where N is the total number of image pairs, $\mathcal{P}_i$ is a set of source and target key point pairs (p_s, p_t) for i^th example, θ_i is predicted transformation, and $\mathbb{1}$ is the indicator function which returns 1 if the expression inside brackets is true and 0 otherwise.

We evaluate three different versions of the proposed model as in [13]. The first two versions are the models with different transformations: affine and TPS transformations. The other version sequentially merges these two models. That is, the input image pair are first fed to the network with affine transformation, and the image pair transformed by its out is then fed to the network with TPS transformation.

4.2   Results

Comparisons to other models Table 1 shows the comparative results on both PF-WILLOW and PF-PASCAL benchmarks. It includes (i) previous methods using hand-crafted features: DeepFlow [5], GMK [31], SIFTFlow [7], DSP [11], and ProposalFlow [9], (ii) self-supervised alignment methods: GeoCNN [13] and the proposed attentive alignment network (A2Net), (iii) supervised methods: UCN [12], FCSS [17], and SCNet [14]. Note that the supervised methods are trained with either a weakly or strongly annotated data and that many of their PCKs are measured under a different criterion that are not directly comparable to the other scores. By contrast, our method is only trained using synthetic data with self-supervision. As shown in Table 1, the proposed method substantially outperforms all the other methods that are directly comparable. Using VGG-16 feature extractor, the proposed method improves 12.5 % and 8 % of PCK over

Table 1: Experimental results on PF-WILLOW and PF-PASCAL. PCK is measured with $\alpha = 0.1$. Scores for other models are brought from [9,13,14] while scores marked with an asterisk (*) are drawn from the reproduced models by released official codes. The PCK scores marked with a dagger ($\dagger$) are borrowed from [14] measuring them with respect to the entire image size instead of the bounding box size. Note that the PCK score with respect to the bounding box size is more conservative than the one with respect to the image size, resulting in lower scores. For example, PCK scores of A2Net on PF-PASCAL measured by image sizes are 0.59 (affine), 0.65 (affine+TPS) and 0.71 (affine+TPS; ResNet101).

| | Models | PCK ($\alpha = 0.1$) | |
		PF-WILLOW	PF-PASCAL
Hand-crafted	DeepFlow [5]	0.20	0.21
	GMK [31]	0.27	0.27
	SIFTFlow [7]	0.38	0.33
	DSP [11]	0.29	0.30
	ProposalFlow (NAM) [9]	0.53	–
	ProposalFlow (PHM) [9]	0.55	–
	ProposalFlow (LOM) [9]	0.56	0.45
Self Sup.	GeoCNN (affine) [13]	0.49	0.39*
	GeoCNN (affine+TPS) [13]	0.56	0.50*
	A2Net (affine)	0.52	0.46
	A2Net (affine+TPS)	0.63	0.54
	A2Net (affine+TPS; ResNet101)	**0.68**	**0.59**
Supervised	UCN [12]	0.42†	0.56†
	FCSS [17]	0.58	–
	SCNet-A [14]	0.73†	0.66†
	SCNet-AG [14]	0.72†	0.70†
	SCNet-AG+ [14]	0.70†	0.72†

the non-attentive alignment method [13] on PF-WILLOW and PF-PASCAL, respectively. This reveals the effect of the proposed attention model for semantic alignment. The quality of the model is further improved when incorporated with a more advanced feature extractor such as ResNet101. It is notable that the proposed model outperforms some of supervised methods, UCN [12] and FCSS [17], while it is trained without any real datasets.

Ablation study As our proposed model combines two distinct techniques we perform ablation studies to demonstrate their effects. We mainly compare the proposed model to GeoCNN as it directly predicts the global transformation parameters using the correlation layer. To see the effect of the proposed OAC kernels, we build a model, referred to as GeoCNN+OACK, by replacing the correlation layer of GeoCNN with the OAC kernels. As shown in Tabel 2, the use of the OAC kernels already improves the performances of GeoCNN for all three versions. Moreover, the OAC kernels reduce the number of parameters in

Table 2: PCKs of ablations on PF-WILLOW trained with PASCAL VOC 2011. Scores of GeoCNN are obtained from the code released by the authors. The numbers of network parameters exclude the feature extractors since all models share the same feature extractor.

Models	# of params	Affine	TPS	Affine+TPS
GeoCNN [13]	1.63M (x1.7)	0.430	0.539	0.560
GeoCNN+Attention	1.41M (x1.5)	0.423	0.478	0.476
GeoCNN+OACK	1.12M (x1.2)	0.491	0.555	0.609
Attention+OACK (A2Net)	**0.95M (x1.0)**	**0.521**	**0.563**	**0.626**

Table 3: PCKs of affine models on PF-WILLOW with different training datasets: PASCAL VOC 2011 and Tokyo Time Machine. Scores for GeoCNN are brought from [13].

Models	PASCAL VOC 2011	Tokyo Time Machine
GeoCNN [13]	0.45	0.49
A2Net	**0.52**	**0.51**

the network since it uses dense representations of local transformations allowing channel compression. Applying attention process on top of correlation layer (GeoCNN+Attention) drops the performance. This is because the correlation map does not encode local transformations in a translation invariant representations. On the other hand, the attention process with the OAC kernels, which is the proposed model, further improves the performances as the distracting regions can be suppressed during the transformation estimation thanks to the local transformation feature map obtained by the OAC kernels. It is also notable that applying the attention process reduces the number of model parameters because the model does not need extra layers that combine all local information to produce the global estimation; instead, the models simply aggregate local features with attention distribution. This additional parameter reduction results in 70 % fewer parameters than GeoCNN while the models maintain superior performance improvements.

Sensitivity to training datasets While both our model and GeoCNN are generally applicable to any image datasets, we experiment the sensitivity of the models to changing training datasets. We train both models with the affine transformation on another image dataset, called Tokyo Time Machine [48], using the same synthetic generation process, and show how much the performances change depending on the datasets. Table 3 shows that the proposed model is less dependent on the choice of the training dataset compared to GeoCNN.

Qualitative results with attention visualizations Fig. 5 presents some qualitative examples of our model on PF-PASCAL. In our experimental setting,
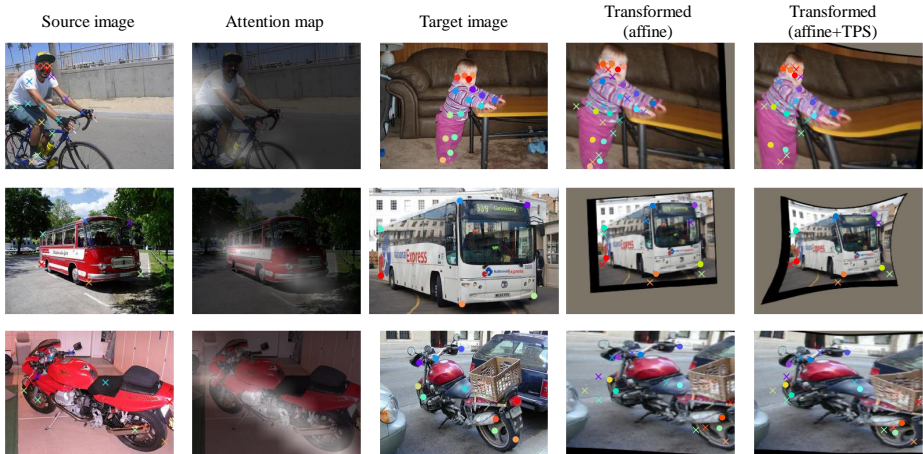
Fig. 5: Qualitative results of the attentive semantic alignment. Each row shows an example of PF-PASCAL benchmark. Given the source and target images shown in first and third columns, we visualize the attention maps of the affine model (second column), the transformed image by the affine model (fourth column) and the final transformed image by the affine+TPS model (last column). Since the models learn inverse transformation, the target image is transformed toward the source image while the attention distribution is drawn over the source image. The model attends to the objects to match and estimates dense correspondences despite intra-class variations and background clutters.

the models learn to predict inverse transformation. Therefore, we transform the target image toward the source image using the estimated inverse transformation whereas the attention distribution is drawn over the source image. The proposed model attends to the target objects with other regions suppressed and predicts the global transformation based on reliable local features. The model estimates the transformation despite large intra-class variations such as an adult vs. a kid.

We also investigate some failure cases of the proposed model in Fig. 6. The model is confused when there are multiple objects of the same class in an image or have a large obstacles occluding the matching objects. Also, objects in some examples are hard to visually recognize and lead mismatches. For instance, the model fails to correctly match a wooden chair to a transparent chair although the model attends to the correct region in the second example of Fig. 6. It is challenging even for human to recognize the transparent chair and its corresponding key points.

5   Conclusion

We propose a novel approach for semantic alignment. Our model facilitates an attention process to estimate global transformation from reliable local transfor-
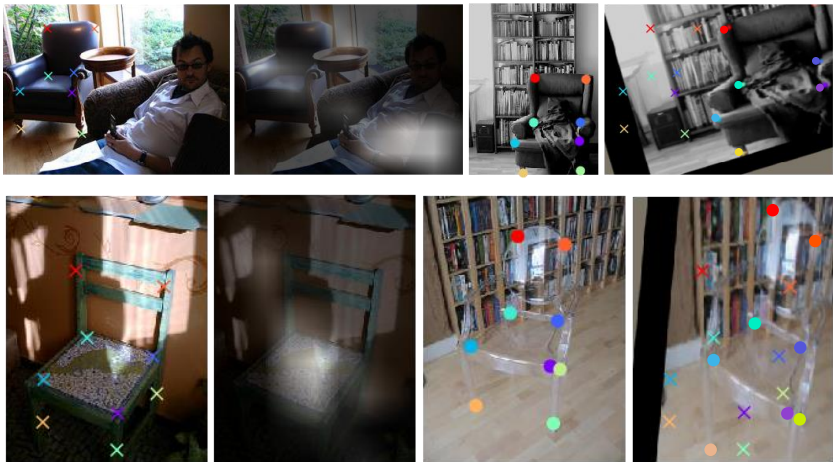
Fig. 6: Some failure cases of the proposed model with the affine transformation. Each row shows an example of PF-PASCAL. Each example contains 1) source image, 2) source image masked by attention distribution, 3) target image and 4) target image transformed by the predicted affine parameters. Even though the model attends to the matching objects, the model fails to find the correct correspondences due to multiple objects of the same class causing ambiguity or hard examples that are difficult to visually percept.

mation features by suppressing distracting features. We also propose offset-aware correlation kernels that reorder correlations of feature pairs and produce a dense feature representation of local transformations. The experimental results show the attentive model with the proposed kernels achieves the state-of-the-art performances with large margins over previous methods on the PF-WILLOW and PF-PASCAL benchmarks.

Acknowledgement

References

1. Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(2) (2013) 504–511

2. Okutomi, M., Kanade, T.: A multiple-baseline stereo. IEEE Transactions on pattern analysis and machine intelligence **15**(4) (1993) 353–363

3. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV. (2015)

4. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: ICCV. (2013)

5. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Deepmatching: Hierarchical deformable dense matching. International Journal of Computer Vision **120**(3) (2016) 300–323

6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2) (2004) 91–110

7. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. In: Dense Image Correspondences for Computer Vision. Springer (2016) 15–49

8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)

9. Ham, B., Cho, M., Schmid, C., Ponce, J.: Proposal flow. In: CVPR. (2016)

10. Taniai, T., Sinha, S.N., Sato, Y.: Joint recovery of dense correspondence and cosegmentation in two images. In: CVPR. (2016)

11. Kim, J., Liu, C., Sha, F., Grauman, K.: Deformable spatial pyramid matching for fast dense correspondences. In: CVPR. (2013)

12. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. In: Advances in Neural Information Processing Systems. (2016) 2414–2422

13. Rocco, I., Arandjelovic, R., Sivic, J.: Convolutional neural network architecture for geometric matching. In: CVPR. (2017)

14. Han, K., Rezende, R.S., Ham, B., Wong, K.Y.K., Cho, M., Schmid, C., Ponce, J.: Scnet: Learning semantic correspondence. In: ICCV. (2017)

15. Ufer, N., Ommer, B.: Deep semantic feature matching. In: CVPR. (2017)

16. Kim, S., Min, D., Lin, S., Sohn, K.: Dctm: Discrete-continuous transformation matching for semantic flow. In: ICCV. (2017)

17. Kim, S., Min, D., Ham, B., Jeon, S., Lin, S., Sohn, K.: Fcss: Fully convolutional self-similarity for dense semantic correspondence. In: CVPR. (2017)

18. Novotny, D., Larlus, D., Vedaldi, A.: Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In: CVPR. (2017)

19. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: CVPR. (2015)

20. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: CVPR. (2015)

21. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: CVPR. (2015)

22. Long, J.L., Zhang, N., Darrell, T.: Do convnets learn correspondence? In: NIPS. (2014)

23. Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning dense correspondence via 3d-guided cycle consistency. In: CVPR. (2016)

24. Kanazawa, A., Jacobs, D.W., Chandraker, M.: Warpnet: Weakly supervised matching for single-view reconstruction. In: CVPR. (2016)

25. Yang, H., Lin, W.Y., Lu, J.: Daisy filter flow: A generalized discrete approach to dense correspondences. In: CVPR. (2014)

26. Hur, J., Lim, H., Park, C., Chul Ahn, S.: Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation. In: CVPR. (2015)

27. Bristow, H., Valmadre, J., Lucey, S.: Dense semantic correspondence where every pixel is a classifier. In: ICCV. (2015)
28. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR. (2005)
29. Yang, F., Li, X., Cheng, H., Li, J., Chen, L.: Object-aware dense semantic correspondence. In: CVPR. (2017)
30. Cho, M., Lee, J., Lee, K.M.: Reweighted random walks for graph matching. In: ECCV. (2010)
31. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: ICCV. (2011)
32. Cho, M., Alahari, K., Ponce, J.: Learning graphs to match. In: ICCV. (2013)
33. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research **17**(1-32) (2016) 2
34. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5695–5703
35. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML. (2015)
36. Mun, J., Cho, M., Han, B.: Text-guided attention model for image captioning. In: AAAI. (2017)
37. Xu, H., Saenko, K.: Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In: ECCV. (2016)
38. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: CVPR. (2016)
39. Seo, P.H., Lin, Z., Cohen, S., Shen, X., Han, B.: Progressive attention networks for visual attribute prediction. arXiv preprint arXiv:1606.02393 (2016)
40. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR. (2015)
41. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP. (2015)
42. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: ICCV. (2017)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
44. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009)
45. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
46. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html
47. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE transactions on pattern analysis and machine intelligence **35**(12) (2013) 2878–2890
48. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: CVPR. (2016)
49. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence **28**(4) (2006) 594–611

50. Lin, Y.L., Morariu, V.I., Hsu, W., Davis, L.S.: Jointly optimizing 3d model fitting and fine-grained classification. In: ECCV. (2014)
51. Rubinstein, M., Joulin, A., Kopf, J., Liu, C.: Unsupervised joint object discovery and segmentation in internet images. In: CVPR. (2013)
52. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV. (2011)

Supplementary Material

A   Evaluation on Other Datasets

We have evaluated the proposed model on two more datasets: Taniai's dataset [10] and Caltech 101 [49].

Results on on Taniai's dataset Taniai's dataset [10] contains 400 image pairs in three subsets: FG3DCar (195 pairs from [50]), JODS (81 pairs from [51]) and PASCAL (124 pairs from [52]). Each image pair in this dataset is annotated with dense flows based on key points. Following [10], we measure flow accuracy which is the percentage of correctly transferred flows. Each flow is considered correct if the distance between the estimated flow and the ground-truth flow is less than 5 pixels.

Table A shows the results on Taniai's dataset. Our model shows higher flow accuracies over all other models on every subset of the dataset. Especially, we emphasize that the proposed model shows significant gains over GeoCNN which uses a single end-to-end neural network for global transformation estimation as in our model.

Results on Caltech-101 Caltech-101 [49] consists of 1515 image pairs of 101 object classes. Unlike other datasets, these image pairs are not annotated with dense correspondences. Instead, we utilize the annotated segmentation masks for the evaluation. Following [13], we measure label transfer accuracy (LT-ACC) and intersection over union (IoU) of transformed masks. LT-ACC measures the accuracy of pixel-level segmentation labels between the transformed masks and the ground-truth masks, and IoU measures the intersection over union of those masks. We believe that localization error (LOC-ERR) used in [13,9,14] is not appropriate for evaluating dense correspondences as it measures the error based on the bounding box information extracted from masks losing the precise details, but we report this score for completeness.

Table B shows the results on Caltech-101. Our model shows the best score in IoU and LT-ACC over all the methods.

B   Visualizations of OAC Kernel Weights

We visualize the learned weights of OAC kernels in the proposed model. Fig. A presents three examples of kernel weights arranged by the offsets along with X and Y axes. Kernel weights are focused on a certain region with similar offsets capturing a displacement to that direction.

C   More Qualitative Results

We present additional qualitative results from Caltech-101 dataset. In Fig. B, segmentation masks of source images are transformed and visualized on target images.

Table A: Flow accuracies with error threshold with 5 pixels on Taniai's dataset. Scores of DCTM is obtained with VGG-16 feature extractor as in our method since the FCSS feature extractor in the original paper is trained with a weak supervision.

Models	FG3DCar	JODS	PASCAL	Average
DFF [25]	0.50	0.30	0.22	0.31
DSP [26]	0.49	0.47	0.38	0.45
SIFT Flow [7]	0.63	0.51	0.36	0.50
Zhou *et al.* [23]	0.72	0.51	0.44	0.56
Taniai *et al.* [10]	0.83	0.60	0.48	0.64
Proposal Flow [9]	0.79	0.65	0.53	0.66
DCTM (VGG-16) [16]	0.79	0.61	0.53	0.63
GeoCNN [13]	0.85	0.64	0.53	0.67
Proposed	**0.87**	**0.67**	**0.55**	**0.70**

Table B: Results on Caltech-101. *The scores for GeoCNN are reproduced using publicly released code by the authors and show slightly lower LT-ACC.

Models	LT-ACC	IoU	LOC-ERR
DeepFlow [5]	0.74	0.40	0.34
GMK [31]	0.77	0.42	0.34
DSP [26]	0.77	0.47	0.35
SIFT Flow [7]	0.75	0.48	0.32
Proposal Flow [9]	0.78	0.50	**0.25**
GeoCNN [13]*	0.79	0.56	0.28
Proposed	**0.80**	**0.57**	0.28

D   Effect of Pretrained Network

To investigate the importance of the pretrained recognition capability of the feature extractor, we evaluated our model on different test subsets using Caltech101 dataset. The first subset consists of test pairs containing objects that appear in ImageNet, which is used to pretrain our feature extractor, and the second subset contains pairs whose objects are unseen at the pretraining. We use Caltech101 dataset for this experiment because almost all objects in PF-WILLOW and PF-PASCAL also appear in ImageNet. Table C summarizes IoU results of our model on the test subsets. We can see that pairs of objects that appear in ImageNet show slightly better performance than the other pairs. However, the gap is relatively small, indicating that the model is still capable of predicting transformations of objects that are unseen during the pretraining. We conjecture that this is because our alignment prediction (1) relies on the feature correlations, which are less class-specific than the raw features themselves, and (2) uses intermediate-level visual features (extracted at `pool4`), which are rather class-agnostic compared to higher-level features.
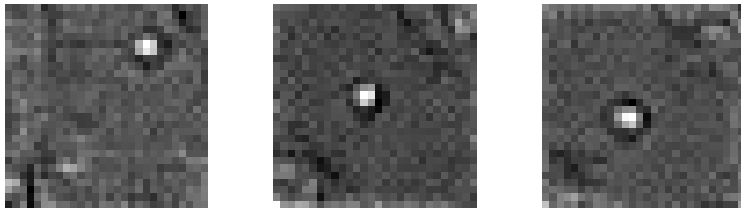
Fig. A: Visualizations of some offset-aware correlation kernels. Each kernel captures displacements by learning pointy weights focused on a region with similar offsets.

Table C: IoU on different test subsets of Caltech-101. See text for detail.

	Affine	TPS	Affine+TPS
ImageNet objects	0.54	0.56	0.58
Other objects	0.52	0.53	0.56
All objects	0.53	0.55	0.57

E   Experiments on Cross-class Matching

We also conduct the cross-class matching experiments as done in the work of [18]. As the authors have neither provided their code online nor responded to our request yet, we did our best to reproduce the experimental settings of [18] for ourselves. However, some of the descriptions are unclear and thus there might be some differences. In the experiment of measuring weighted IoU (wIoU) of part segmentations for cross-class objects on the PASCAL Parts dataset, where the best model in [18] produces wIoU of 37.5 %, our method shows a competitive wIoU of 35.0 %. As mentioned earlier, this may be due to the use of the feature correlations, rather than the raw features themselves, in alignment prediction. If the features of cross-class objects share similar representations, our method is still capable of predicting the alignment between them.
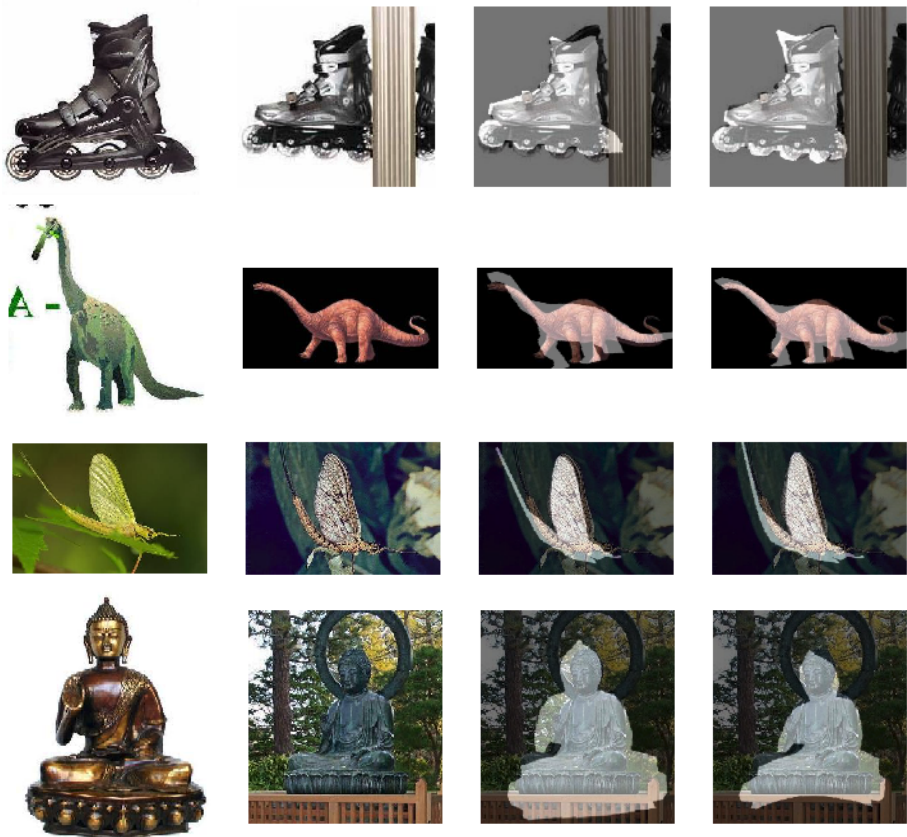
Fig. B: Qualitative examples from Caltech-101. Each row presents different examples including source image (first column), target image (second column), target image with affine transformed source mask (third column), and target image with TPS transformed source mask (fourth column).