End-to-end weakly-supervised semantic alignment

Ignacio Rocco[1,2] Relja Arandjelović[3] Josef Sivic[1,2,4]
[1]DI ENS [2]Inria [3]DeepMind [4]CIIRC, CTU in Prague

Abstract

We tackle the task of semantic alignment where the goal is to compute dense semantic correspondence aligning two images depicting objects of the same category. This is a challenging task due to large intra-class variation, changes in viewpoint and background clutter. We present the following three principal contributions. First, we develop a convolutional neural network architecture for semantic alignment that is trainable in an end-to-end manner from weak image-level supervision in the form of matching image pairs. The outcome is that parameters are learnt from rich appearance variation present in different but semantically related images without the need for tedious manual annotation of correspondences at training time. Second, the main component of this architecture is a differentiable soft inlier scoring module, inspired by the RANSAC inlier scoring procedure, that computes the quality of the alignment based on only geometrically consistent correspondences thereby reducing the effect of background clutter. Third, we demonstrate that the proposed approach achieves state-of-the-art performance on multiple standard benchmarks for semantic alignment.

1. Introduction

Finding correspondence is one of the fundamental problems in computer vision. Initial work has focused on finding correspondence between images depicting the same object or scene with applications in image stitching [31], multi-view 3D reconstruction [11], motion estimation [6, 34] or tracking [4, 22]. In this work we study the problem of finding category-level correspondence, or *semantic alignment* [1, 20], where the goal is to establish dense correspondence between different objects belonging to the same category, such as the two different motorcycles illustrated in Fig. 1. This is an important problem with applications in object recognition [19], image editing [3], or robotics [23].

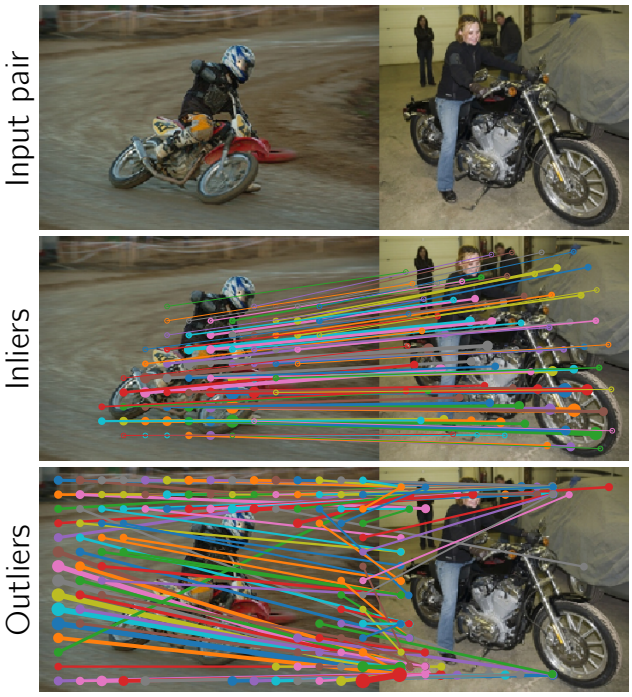


Figure 1: We describe a CNN architecture that, given an input image pair (top), outputs dense semantic correspondence between the two images together with the aligning geometric transformation (middle) and discards geometrically inconsistent matches (bottom). The alignment model is learnt from weak supervision in the form of matching image pairs without correspondences.

This is also an extremely challenging task because of the large intra-class variation, changes in viewpoint and presence of background clutter.

The current best semantic alignment methods [10, 17, 24] employ powerful image representations based on convolutional neural networks coupled with a geometric deformation model. However, these methods suffer from one of the following two major limitations. First, the image representation and the geometric alignment model are not trained together in an end-to-end manner. Typically, the image representation is trained on some auxiliary task such as image classification and then employed in an often ad-hoc geometric alignment model. Second, while trainable geometric

[1]Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.
[4]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

alignment models exist [2, 29], they require strong supervision in the form of ground truth correspondences, which is hard to obtain for a diverse set of real images on a large scale.

In this paper, we address both these limitations and develop a semantic alignment model that is *trainable end-to-end* from *weakly supervised* data in the form of matching image pairs without the need for ground truth correspondences. To achieve that we design a novel convolutional neural network architecture for semantic alignment with a differentiable soft inlier scoring module inspired by the RANSAC inlier scoring procedure. The resulting architecture is end-to-end trainable with only image-level supervision. The outcome is that the image representation can be trained from rich appearance variations present in different but semantically related image pairs, rather than synthetically deformed imagery [14, 29]. We show that our approach allows to significantly improve the performance of the baseline deep CNN alignment model, achieving state-of-the-art performance on multiple standard benchmarks for semantic alignment. Our code and trained models are available online [28].

2. Related work

The problem of semantic alignment has received significant attention in the last few years with progress in both (i) image descriptors and (ii) geometric models. The key innovation has been making the two components trainable from data. We summarize the recent progress in Table 1 where we indicate for each method whether the descriptor (D) or the alignment model (A) are trainable, whether the entire architecture is trainable end-to-end (E-E), and whether the required supervision is strong (s) or weak (w).

Early methods, such as [1, 15, 19], employed hand-engineered descriptors like SIFT or HOG together with hand-engineered alignment models based on minimizing a given matching energy. This approach has been quite successful [9, 32, 33, 35] using in some cases [33] pre-trained (but fixed) convolutional neural network (CNN) descriptors. However, none of these methods train the image descriptor or the geometric model directly for semantic alignment.

Others [16, 17, 24] have investigated trainable image descriptors for semantic matching but have combined them with hand-engineered alignment models still rendering the alignment pipeline not trainable end-to-end.

Finally, recent work [10, 29] has employed trainable CNN descriptors together with trainable geometric alignment methods. However, in [10] the matching is learned at the object-proposal level and a non-trainable fusion step is necessary to output the final alignment making the method non end-to-end trainable. On the contrary, [29] estimate a parametric geometric model, which can be converted into dense pixel correspondences in a differentiable way, making the method end-to-end trainable. However, the method is trained with strong supervision in the form of ground truth correspondences obtained from synthetically warped images, which significantly limits the appearance variation in the training data.

Paper	Descriptor	Alignment method	Trainable D	A	E-E	S
Liu *et al.*'11 [19]	SIFT	SIFT Flow	✗	✗	✗	-
Kim *et al.*'13 [15]	SIFT+PCA	DSP	✗	✗	✗	-
Taniai *et al.*'16 [32]	HOG	TSS	✗	✗	✗	-
Ham *et al.*'16 [9]	HOG	PF-LOM	✗	✗	✗	-
Yang *et al.*'17 [35]	HOG	OADSC	✗	✗	✗	-
Ufer *et al.*'17 [33]	AlexNet	DSFM	✗	✗	✗	-
Novotny *et al.*'17 [24]	AnchorNet	DSP	✓	✗	✗	w
		PF-LOM	✓	✗	✗	w
Kim *et al.*'17 [16]	FCSS	SIFT Flow	✓	✗	✗	s
		PF-LOM	✓	✗	✗	s
Kim *et al.*'17 [17]	FCSS	DCTM	✓	✗	✗	s
Han *et al.*'17 [10]	VGG-16	SCNet-A	✓	✓	✗	s
		SCNet-AG	✓	✓	✗	s
		SCNet-AG+	✓	✓	✗	s
Rocco *et al.*'17 [29]	VGG-16	CNN Geo.	✓	✓	✓	s
	ResNet-101	CNN Geo.	✓	✓	✓	s
Proposed method	ResNet-101	CNN Geo.	✓	✓	✓	w

Table 1: **Comparison of recent related work.** The table indicates employed image descriptor and alignment method. The last four columns show which components of the approach are trained for the semantic alignment task: descriptor (D), alignment (A) or both in end-to-end manner (E-E); and the level of supervision (S): strong (s) or weak (w).

Contributions. We develop a network architecture where both the descriptor and the alignment model are trainable in an end-to-end manner from weakly supervised data. This enables training from real images with rich appearance variation and without the need for manual ground-truth correspondence. We demonstrate that the proposed approach significantly improves alignment results achieving state-of-the-art performance on several datasets for semantic alignment.

3. Weakly-supervised semantic alignment

This section presents a method for training a semantic alignment model in an end-to-end fashion using only weak supervision – the information that two images should match – but without access to the underlying geometric transformation at training time. The approach is outlined in Fig. 2.

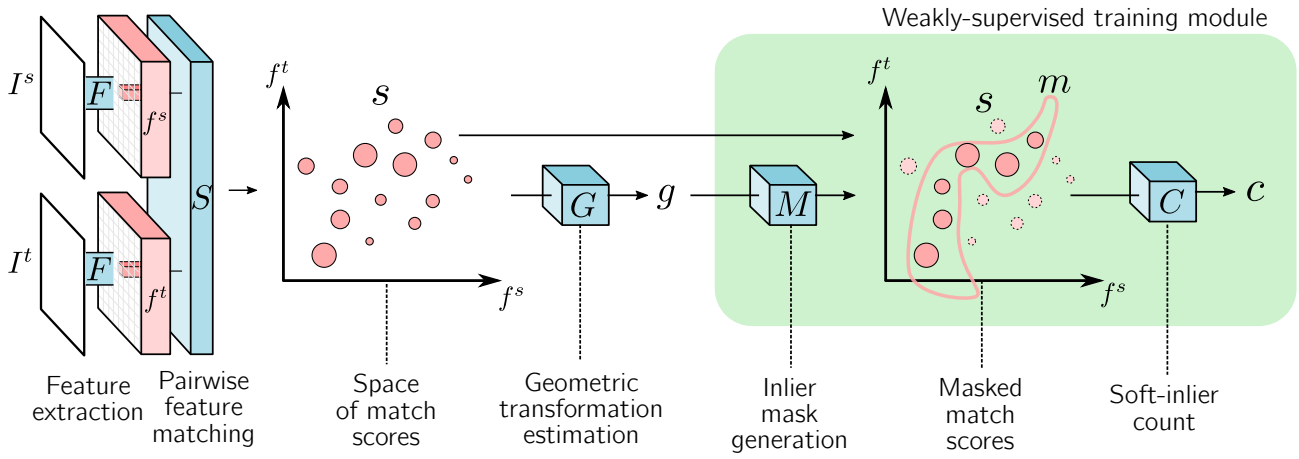


Figure 2: **End-to-end weakly-supervised alignment.** Source and target images (I^s, I^t) are passed through an alignment network used to estimate the geometric transformation g. Then, the soft-inlier count is computed (in green) by first finding the inlier region m in agreement with g, and then adding up the pairwise matching scores inside this area. The soft-inlier count is differentiable, which allows the whole model to be trained using back-propagation. Functions are represented in blue and tensors in pink.

Namely, given a pair of images, an alignment network estimates the geometric transformation that aligns them. The quality of the estimated transformation is assessed using the proposed *soft-inlier count* which aggregates the observed evidence in the form of feature matches. The training objective then is to maximize the alignment quality for pairs of images which should match.

The key idea is that, instead of requiring strongly supervised training data in the form of known pairwise alignments and training the alignment network with these, the network is "forced" into learning to estimate good alignments in order to achieve high alignment scores (soft-inlier counts) for matching image pairs. The details of the alignment network and the soft-inlier count are presented next.

3.1. Semantic alignment network

In order to make use of the error signal coming from the soft-inlier count, our framework requires an alignment network which is trainable end-to-end. We build on the Siamese CNN architecture described in [29], illustrated in the left section of Fig. 2. The architecture is composed of three main stages – feature extraction, followed by feature matching and geometric transformation estimation – which we review below.

Feature extraction. The input source and target images, (I^s, I^t), are passed through two fully-convolutional feature extraction CNN branches, F, with shared weights. The resulting feature maps (f^s, f^t) are $h \times w \times d$ tensors which can be interpreted as dense $h \times w$ grids of d-dimensional local features $f_{ij:} \in \mathbb{R}^d$. These individual d-dimensional features are L2 normalized.

Pairwise feature matching. This stage computes all pairwise similarities, or match scores, between local features in the two images. This is done with the normalized correlation function, defined as:

$$S : \mathbb{R}^{h\times w\times d} \times \mathbb{R}^{h\times w\times d} \to \mathbb{R}^{h\times w\times h\times w} \tag{1}$$

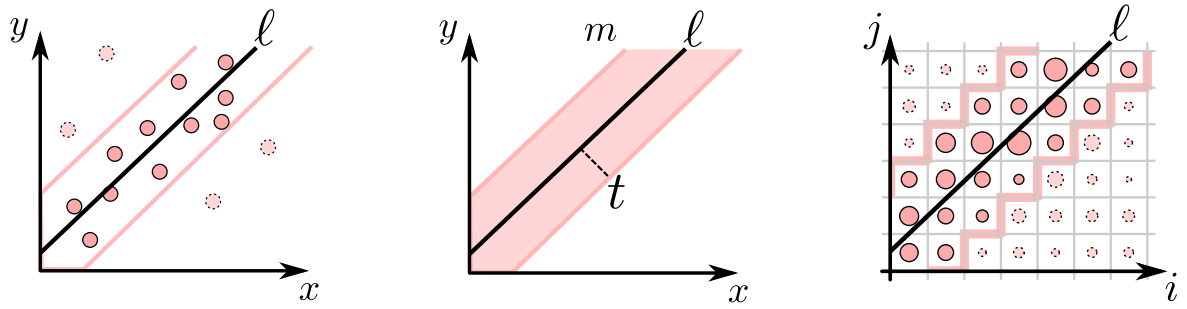
$$s_{ijkl} = S(f^s, f^t)_{ijkl} = \frac{\langle f^s_{ij:}, f^t_{kl:}\rangle}{\sqrt{\sum_{a,b}\langle f^s_{ab:}, f^t_{kl:}\rangle^2}}, \tag{2}$$

where the numerator in (2) computes the *raw* pairwise match scores by computing the dot product between features pairs. The denominator performs a normalization operation with the effect of down-weighing ambiguous matches, by penalizing features from one image which have multiple highly-rated matches in the other image. This is in line with the classical second nearest neighbour test of Lowe [21]. The resulting tensor s contains all normalized match scores between the source and target features.

Geometric transformation estimation. The final stage of the alignment network consists of estimating the parameters of a geometric transformation g given the match scores s. This is done by a transformation regression CNN, represented by the function G:

$$G : \mathbb{R}^{h\times w\times h\times w} \to \mathbb{R}^K, \quad g = G(s) \tag{3}$$

where K is the number of degrees of freedom, or parameters, of the geometric model; *e.g.* $K = 6$ for an affine model. The estimated transformation parameters g are used to define the 2-D warping $\mathcal{T}_g$:



(a) Inliers and outliers (b) Inlier mask function (c) Discretized space

Figure 3: **Line-fitting example.** (a) The line hypothesis ℓ can be evaluated in terms of the number of inliers. (b) The inlier mask m specifies the region where the inlier distance threshold is satisfied. (c) In the discretized space setting, where the match score s_{ij} exists for every point (i, j), the soft-inlier count is computed by summing up match scores masked by the inlier mask m from (b).

$$\mathcal{T}_g : \mathbb{R}^2 \to \mathbb{R}^2, \quad (u^s, v^s) = \mathcal{T}_g(u^t, v^t) \tag{4}$$

where (u^t, v^t) are the spatial coordinates of the target image, and (u^s, v^s) the corresponding sampling coordinates in the source image. Using $\mathcal{T}_g$, it is possible to warp the source to the target image.

Note that all parts of the geometric alignment network are differentiable and therefore amenable to end-to-end training [29], including the feature extractor F which can learn better features for the task of semantic alignment.

3.2. Soft-inlier count

We propose the *soft-inlier count* used to automatically evaluate the estimated geometric transformation g. Making an effort to maximize this count provides the weak-supervisory signal required to train the alignment network, avoiding the need for expensive manual annotations for g. The soft-inlier count is inspired by the inlier count used in the robust RANSAC method [7], which is reviewed first.

RANSAC inlier count. For simplicity, let us consider the problem of fitting a line to a set of observed points p_i, with $i = 1, \dots N$, as illustrated in Fig. 3a. RANSAC proceeds by sampling random pairs of points used to propose line hypotheses, each of which is then scored using the inlier count, and the highest scoring line is chosen; here we only focus on the inlier count aspect of RANSAC used to score a hypothesis. Given a hypothesized line ℓ, the RANSAC inlier scoring function counts the number of observed points which are in agreement with this hypothesis, called the *inliers*. A point p is typically deemed to be an inlier iff its distance to the line is smaller than a chosen distance threshold t, *i.e.* $\mathrm{d}(p, \ell) < t$.

The RANSAC inlier count, c_R, can be formulated by means of an auxiliary indicator function illustrated in Fig. 3b, which we call the inlier mask function m:

$$c_R = \sum_i m(p_i), \text{ where } m(p) = \begin{cases} 1, & \text{if } \mathrm{d}(p, \ell) < t \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Soft-inlier count. The RANSAC inlier count cannot be used directly in a neural network as it is not differentiable. Furthermore, in our setting there is no sparse set of matching points, but rather a match score for every match in a discretized match space. Therefore, we propose a direct extension, the *soft-inlier count*, which, instead of counting over a sparse set of matches, sums the match scores over all possible matches.

The running line-fitting example can now be revisited under the discrete-space conditions, as illustrated in Figure 3c. The proposed soft-inlier count for this case is:

$$c = \sum_{i,j} s_{ij} m_{ij}, \tag{6}$$

where s_{ij} is the match score at each grid point (i,j), and m_{ij} is the discretized inlier mask:

$$m_{ij} = \begin{cases} 1 & \text{if } \mathrm{d}\big((i, j), \ell\big) < t \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Translating the discrete-space line-fitting example to our semantic alignment problem, s is a 4-D tensor containing scores for all pairwise feature matches between the two images (Section 3.1), and matches are deemed to be inliers if they fit the estimated geometric transformation g. More formally, the inlier mask m is now also a 4-D tensor, constructed by thresholding the transfer error:

$$m_{ijkl} = \begin{cases} 1 & \text{if } \mathrm{d}\big((i, j), \mathcal{T}_g(k, l)\big) < t \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $\mathcal{T}_g(k,l)$ are the estimated coordinates of target image's point (k,l) in the source image according to the geometric transformation g; $\mathrm{d}\big((i,j),\mathcal{T}_g(k,l)\big)$ is the transfer error as it measures how aligned is the point (i,j) in the source image, with the projection of the target image point (k,l) into the source image. The soft-inlier count c is then computed by summing the masked matching scores over the entire space of matches:

$$c = \sum_{i,j,k,l} s_{ijkl} m_{ijkl}. \tag{9}$$

Differentiability. The proposed soft-inlier count c is differentiable with respect to the transformation parameters g as long as the geometric transformation $\mathcal{T}_g$ is differentiable [13], which is the case for a range of standard geometric transformations such as 2D affine, homography or thin-plate spline transformations. Furthermore, it is also differentiable w.r.t. the match scores, which facilitates training of the feature extractor.

Implementation as a CNN layer. The inlier mask m can be computed by warping an identity mask m^{Id} with the estimated transformation $\mathcal{T}_g$, where m^{Id} is constructed by thresholding the transfer error of the identity transformation:

$$m^{Id}_{ijkl} = \begin{cases} 1 & \mathrm{d}\big((i,j),(k,l)\big) < t \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The warping is implemented using a spatial transformer layer [13], which consists of a grid generation layer and a bilinear sampling layer. Both of these functions are readily available in most deep learning frameworks.

Optimization objective. For a given training pair of images that should match, the goal is to maximize the soft-inlier count c, or, equivalently, to minimize the loss $\mathcal{L} = -c$.

Analogy to RANSAC. Please also note that our method is similar in spirit to RANSAC [7], where (i) transformations are proposed (by random sampling) and then (ii) scored by their support (number of inliers). In our case, during training (i) the transformations are proposed (estimated) by the regressor network G and (ii) scored using the proposed soft-inlier score. The gradient of this score is used to improve both the regressor G and feature extractor F (see Fig. 2). In turn, the regressor produces better transformations and the feature extractor better feature matches that maximize the soft-inlier score on training images.

4. Evaluation and results

In this section we provide implementation details, benchmarks used to evaluate our approach, and quantitative and qualitative results.

4.1. Implementation details

Semantic alignment network. For the underlying semantic alignment network, we use the best-performing architecture from [27] which employs a ResNet-101 [12], cropped after `conv4-23`, as the feature extraction CNN F. Note that this is a better performing model than the one described in [29], mainly due to use of ResNet versus VGG-16 [30]. Given an image pair, the model produces a thin-plate spline geometric transformation $\mathcal{T}_g$ which aligns the two images; $\mathcal{T}_g$ has 18 degrees of freedom. The network is initialized with the pre-trained weights from [27], and we finetune it with our weakly supervised method. Note that the initial model has been trained in a self-supervised way from synthetic data, not requiring human supervision [29], therefore not affecting our claim of weakly supervised training[1].

Training details. Training and validation image pairs are obtained from the training set of PF-PASCAL, described in Section 4.2. All input images are resized to 240×240, and the value $t = L/30$ (where $L = h = w$ is the size of the extracted feature maps) was used for the transfer error threshold. The whole model is trained end-to-end, including the affine parameters in the batch normalization layers. However, the running averages of the batch normalization layers are kept fixed, in order to be less dependent on the particular statistics of the training dataset. The network is implemented in PyTorch [25] and trained using the Adam optimizer [18] with learning rate $5 \cdot 10^{-8}$, no weight decay and batch size of 16. The training dataset is augmented by horizontal flipping, swapping the source and target images, and random cropping. Early stopping is required to avoid overfitting, given the small size of the training set. This results in 13 training epochs, taking about an hour on a modern GPU.

4.2. Evaluation benchmarks

Evaluation is performed on three standard image alignment benchmarks: PF-PASCAL, Caltech-101 and TSS.

PF-PASCAL [9]. This dataset contains 1351 semantically related image pairs from 20 object categories, which present challenging appearance differences and background clutter. We use the split proposed in [10], which divides the dataset into roughly 700 pairs for training, 300 pairs for validation, and 300 pairs for testing. Keypoint annotations are provided for each image pair, which are used only for evaluation purposes. Alignment quality is evaluated in terms of the percentage of correct keypoints (PCK) metric [36], which counts the number of keypoints which have a transfer error below a given threshold. We follow the procedure employed in [10], where keypoint (x,y) coordinates are nor-

[1] The initial model is trained with a supervised loss, but the "supervision" is automatic due to the use of synthetic data.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	d.table	dog	horse	moto	person	plant	sheep	sofa	train	tv	**all**
HOG+PF-LOM [8]	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
VGG-16+SCNet-A [10]	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 [10]	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+ [10]	**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 [29]	75.2	80.1	73.4	59.7	43.8	77.9	84.0	67.7	44.3	89.6	33.9	67.1	60.5	72.6	54.0	41.0	60.0	45.1	58.3	37.2	65.0
ResNet-101+CNNGeo [29]	82.4	80.9	**85.9**	47.2	57.8	83.1	**92.8**	**86.9**	43.8	91.7	28.1	**76.4**	70.2	**76.6**	68.9	65.7	**80.0**	50.1	46.3	60.6	71.9
Proposed	83.7	**88.0**	83.4	58.3	68.8	**90.3**	92.3	83.7	47.4	91.7	28.1	76.3	**77.0**	76.0	**71.4**	**76.2**	**80.0**	59.5	62.3	63.9	**75.8**

Table 2: **Per-class PCK on the PF-PASCAL dataset.**

malized in the $[0, 1]$ range by dividing with the image width and height respectively, and the value $\alpha = 0.1$ is employed as the distance threshold.

Caltech-101 [5]. Although originally introduced for the image classification task, the dataset was adopted in [15] for assessing semantic alignment, and has been then extensively used for this purpose [9, 10, 16, 29]. The evaluation is performed on 1515 semantically related image pairs, 15 pairs for each of the 101 object categories of the dataset. The semantic alignment is evaluated using three different metrics: (i) the label transfer accuracy (LT-ACC); (ii) the intersection-over-union (IoU), and; (iii) the object localization error (LOC-ERR). The label transfer accuracy and the intersection-over-union both measure the overlap between the annotated foreground object segmentation masks, with former putting more emphasis on the background class and the latter on the foreground object. The localization error computes a dense displacement error. However, given the lack of dense displacement annotations, the metric computes the ground-truth transformation from the source and target bounding boxes, thus assuming that the transformation is a simple translation with axis-aligned anisotropic scaling. This assumption is unrealistic as, amongst others, it does not cover rotations, affine or deformable transformations. Therefore, we believe that LOC-ERR should not be reported any more, but report it here for completeness and in order to adhere to the currently adopted evaluation protocol.

TSS [32]. The recently introduced TSS dataset contains 400 semantically related image pairs, which are split into three different subsets: FG3DCar, JODS and PASCAL, according to the origin of the images. Ground-truth flow is provided for each pair, which was obtained by manual annotation of sparse keypoints, followed by automatic densification using an interpolation algorithm. The evaluation metric is the PCK computed densely over the foreground object. The distance threshold is defined as $\alpha \max(w^s, h^s)$ with (w^s, h^s) being the dimensions of the source image, and $\alpha = 0.05$.

Assessing generalization. We train a single semantic alignment network with the 700 training pairs from PF-PASCAL *without* using the keypoint annotations, and stress that our weakly-supervised training objective only uses the information that the image pair should match. *The same* model is then used for all experiments – evaluation on the test sets of PF-PASCAL, Caltech-101 and TSS datasets. This poses an additional difficulty as these datasets contain images of different object categories or of different nature. While PF-PASCAL contains images of common objects such as car, bicycle, boat, *etc*., Caltech-101 contains images of much less common categories such as accordion, buddha or octopus. On the other hand, while the classes of TSS do appear in PF-PASCAL, the pose differences in TSS are usually smaller than in PF-PASCAL, which modifies the challenge into obtaining a very precise alignment.

4.3. Results

In the following, our alignment network trained with *weak supervision* is compared to the state-of-the-art alignment methods, many of which require *manual annotations* or *strong supervision* (*c.f*. Table 1).

PF-PASCAL. From Table 2 it is clear that our method sets the new state-of-the-art, achieving an overall PCK of 75.8%, which is a 3.6% improvement over the best competitor [10]. This result is impressive as the two methods are trained on the same image pairs, with ours being weakly supervised while [10] make use of bounding box annotations.

The benefits of weakly supervised training can be seen by comparing our method with ResNet-101+CNNGeo [27, 29]. The two use the same base alignment network (*c.f*. Section 4.1), but ResNet-101+CNNGeo was trained only on synthetically deformed image pairs, while ours employs the proposed weakly supervised fine-tuning. The 3.9% boost clearly demonstrates the advantage obtained by training on real image pairs and thus encountering rich appearance variations, as opposed to using synthetically transformed pairs in ResNet-101+CNNGeo [29].

Caltech-101. Table 3 presents the quantitative results for this dataset. The proposed method beats state-of-the-art results in terms of the label-transfer accuracy and intersection-over-union metrics. Weakly supervised training again improves the results, by 2%, over the synthetically trained ResNet-101+CNNGeo. In terms of the localization-error metric, our model does not attain state-of-the-art per-

Method	LT-ACC	IoU	LOC-ERR
HOG+PF-LOM [9]	0.78	0.50	0.26
FCSS+SIFT Flow [16]	0.80	0.50	0.21
FCSS+PF-LOM [16]	0.83	0.52	0.22
VGG-16+SCNet-A [10]	0.78	0.50	0.28
VGG-16+SCNet-AG [10]	0.78	0.50	0.27
VGG-16+SCNet-AG+ [10]	0.79	0.51	0.25
HOG+OADSC [35]	0.81	0.55	**0.19**
VGG-16+CNNGeo [29]	0.80	0.55	0.26
ResNet-101+CNNGeo [29]	0.83	0.61	0.25
Proposed	**0.85**	**0.63**	0.24

Table 3: **Evaluation results on the Caltech-101 dataset.**

Method	FG3D.	JODS	PASC.	avg.
HOG+PF-LOM [9]	0.786	0.653	0.531	0.657
HOG+TSS [32]	0.830	0.595	0.483	0.636
FCSS+SIFT Flow [16]	0.830	0.656	0.494	0.660
FCSS+PF-LOM [16]	0.839	0.635	0.582	0.685
HOG+OADSC [35]	0.875	0.708	**0.729**	**0.771**
FCSS+DCTM [17]	0.891	0.721	0.610	0.740
VGG-16+CNNGeo [29]	0.839	0.658	0.528	0.675
ResNet-101+CNNGeo [29]	0.901	**0.764**	0.563	0.743
Proposed	**0.903**	**0.764**	0.565	0.744

Table 4: **Evaluation results on the TSS dataset.**

formance, but we argue that this metric is not a good indication of the alignment quality, as explained in section 4.2. This claim is further backed up by noticing that the relative ordering of various methods based on this metric is in direct opposition with the other two metrics.

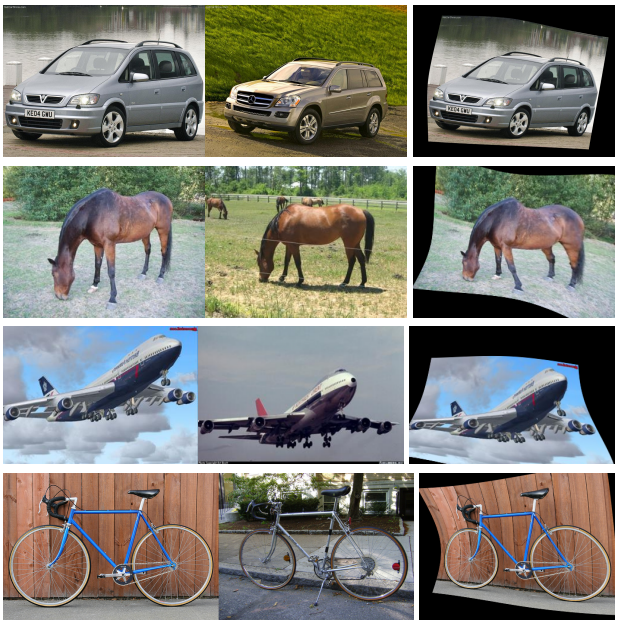
TSS. The quantitative results for the TSS dataset are presented in Table 4. We set the state-of-the-art for two of the three subsets of the TSS dataset: FG3DCar and JODS. Although our weakly supervised training provides an improvement over the base alignment network, ResNet-101+CNNGeo, the gain is modest. We believe the reason is a very different balancing of classes in this dataset compared to our training. Recall our model is trained *only once* on the PF-PASCAL dataset, and is then applied without any further training on TSS and Caltech-101.

Qualitative results. Figures 4a, 4b and 5 show qualitative results on the Caltech-101, TSS and PF-PASCAL datasets, respectively. Our method is able to align images across prominent viewpoint changes, in the presence of significant clutter, while simultaneously tolerating large intra-class variations. For additional qualitative examples, please refer to [26].

5. Conclusions

We have designed a network architecture and training procedure for semantic image alignment inspired by the robust inlier scoring used in the widely successful RANSAC fitting algorithm [7]. The architecture requires supervision only in the form of matching image pairs and sets the new state-of-the-art on multiple standard semantic alignment benchmarks, even beating alignment methods that require geometric supervision at training time. However, handling multiple objects and non-matching image pairs still remains an open challenge. These results open-up the possibility of learning powerful correspondence networks from large-scale datasets such as ImageNet.

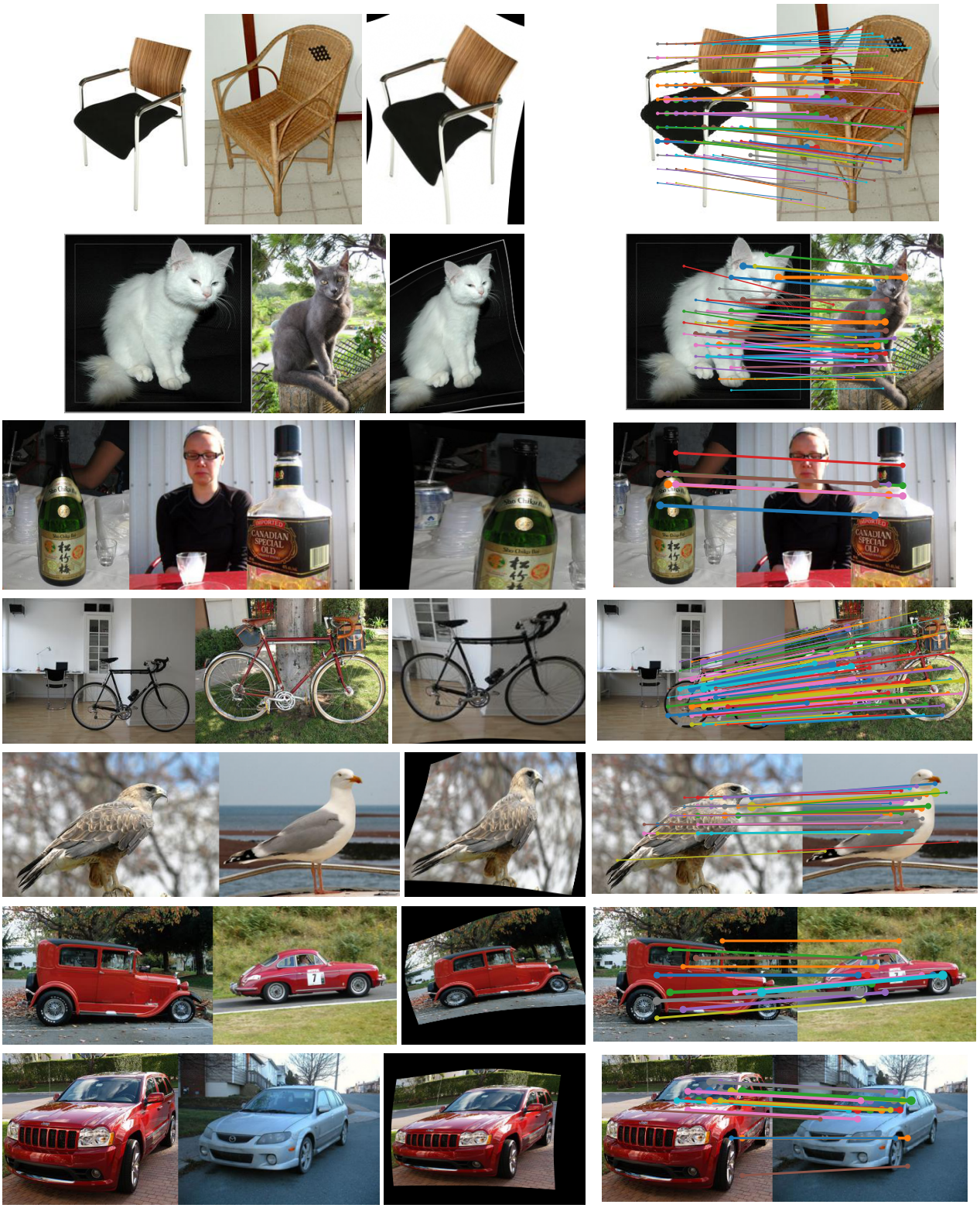
(a) Caltech-101



(b) TSS

Figure 4: **Alignment examples on the Caltech-101 and TSS datasets.** Each row shows the (left) source and (middle) target images, and (right) the automatic semantic alignment.

Acknowledgements. This work has been partly supported by ERC grant LEAP (no. 336845), the Inria CityLab IPL, CIFAR Learning in Machines & Brains program and the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468).



(a) Semantic alignment (b) Strongest inlier matches

Figure 5: **Alignment examples on the PF-PASCAL dataset.** Each row corresponds to one example. (a) shows the (right) automatic semantic alignment of the (left) source and (middle) target images. (b) shows the strongest inlier feature matches.

References

[1] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, 2005.

[2] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC - Differentiable RANSAC for camera localization. In *CVPR*, 2017.

[3] K. Dale, M. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister. Image restoration using online photo collections. In *CVPR*, 2017.

[4] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *ECCV*, 2014.

[5] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE PAMI*, 28(4):594–611, 2006.

[6] 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.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *CVPR*, 2016.

[9] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE PAMI*, 2017.

[10] 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.

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[14] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016.

[15] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013.

[16] 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.

[17] S. Kim, D. Min, S. Lin, and K. Sohn. DCTM: Discrete-continuous transformation matching for semantic flow. In *ICCV*, 2017.

[18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE PAMI*, 33(5):978–994, 2011.

[20] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT flow: Dense correspondence across different scenes. In *ECCV*, 2008.

[21] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[22] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, 2011.

[23] E. Nikandrova and V. Kyrki. Category-based task specific grasping. *Robotics and Autonomous Systems*, 70(Supplement C):25 – 35, 2015.

[24] D. Novotny, D. Larlus, and A. Vedaldi. AnchorNet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *CVPR*, 2017.

[25] A. Paszke, S. Gross, S. Chintala, and G. Chanan. PyTorch. `http://pytorch.org/`.

[26] I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. *arXiv preprint arXiv:1712.06861*.

[27] I. Rocco, R. Arandjelović, and J. Sivic. Webpage: Convolutional neural network architecture for geometric matching. `http://www.di.ens.fr/willow/research/cnngeometric/`.

[28] I. Rocco, R. Arandjelović, and J. Sivic. Webpage: End-to-end weakly-supervised semantic alignment. `http://www.di.ens.fr/willow/research/weakalign/`.

[29] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[31] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[32] T. Taniai, S. N. Sinha, and Y. Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016.

[33] N. Ufer and B. Ommer. Deep semantic feature matching. In *CVPR*, 2017.

[34] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[35] F. Yang, X. Li, H. Cheng, J. Li, and L. Chen. Object-aware dense semantic correspondence. In *CVPR*, 2017.

[36] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE PAMI*, 2013.