AnchorNet: A Weakly Supervised Network to Learn Geometry-sensitive Features For Semantic Matching

David Novotny[1,2] Diane Larlus[2] Andrea Vedaldi[1]

[1]Visual Geometry Group
Dept. of Engineering Science, University of Oxford
{david,vedaldi}@robots.ox.ac.uk

[2]Computer Vision Group
Xerox Research Centre Europe
diane.larlus@xrce.xerox.com

Abstract

Despite significant progress of deep learning in recent years, state-of-the-art semantic matching methods still rely on legacy features such as SIFT or HoG. We argue that the strong invariance properties that are key to the success of recent deep architectures on the classification task make them unfit for dense correspondence tasks, unless a large amount of supervision is used. In this work, we propose a deep network, termed AnchorNet, that produces image representations that are well-suited for semantic matching. It relies on a set of filters whose response is geometrically consistent across different object instances, even in the presence of strong intra-class, scale, or viewpoint variations. Trained only with weak image-level labels, the final representation successfully captures information about the object structure and improves results of state-of-the-art semantic matching methods such as the deformable spatial pyramid or the proposal flow methods. We show positive results on the cross-instance matching task where different instances of the same object category are matched as well as on a new cross-category semantic matching task aligning pairs of instances each from a different object class.

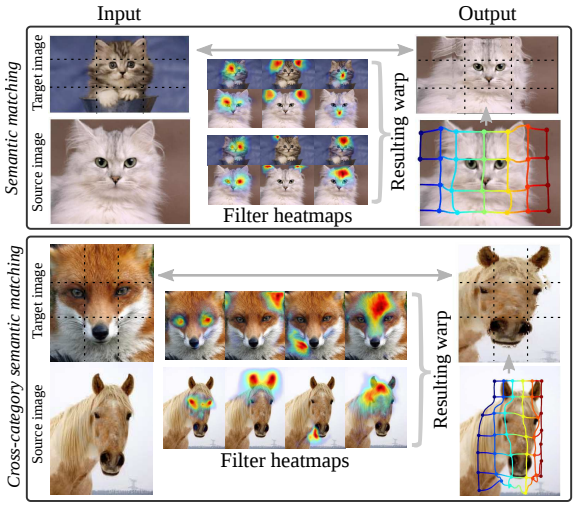


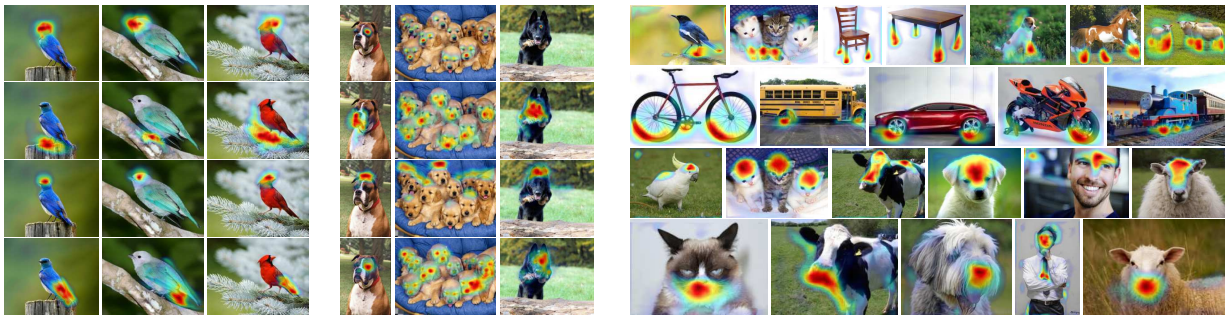
Figure 1: We propose AnchorNet, a novel deep architecture that produces an image representation which significantly improves state-of-the-art semantic matching methods. Key to its success is a set of filters with a sparse response that is geometrically consistent across different instances of a category or of two similar categories. Although these filters are learned in a weakly supervised manner (*i.e.* only image-level labels are used) they tend to anchor reliably on meaningful object parts.

1. Introduction

Matching, i.e. the problem of establishing correspondences between images, is one of the tent-poles of image understanding. It is well known that, given matches between images of the same object or scene, it is possible to estimate 3D geometry (stereo and structure from motion) and motion (visual odometry, optical flow, and tracking). But matching can be applied to much more abstract levels of understanding as well. For example, aligning different object instances of the same type [32, 21] allows to discover analogies between objects, inducting abstractions such as object categories.

While reliable techniques exist for low-level matching, high-level matching of different object instances remains a heavily-researched topic. Most of the work in this area has focused on finding powerful geometric regularizers, such as hierarchical correspondences [35] or deformable spatial pyramids [32], to compensate for the still brittle visual descriptors. Surprisingly, even powerful convolutional neural network (CNN) descriptors have been found lacking for cross-instance matching [37, 21, 63], and in fact comparable or even inferior to old hand-crafted features such as SIFT [38] and HoG [11] for this task.

It is unclear why CNN representations, which perform well for many challenging vision tasks, including object detection [16] and segmentation [36], image captioning [57],



(a) (b) (c)

Figure 2: Example responses of anchor filters discovered by the AnchorNet. (a), (b) show the class specific filters $F_k^{C_i}$ for bird and dog classes respectively while (c) depicts the class agnostic filters F_k^S across different categories (one filter per row).

and visual question answering [1], have not been found to work as well for cross-instance matching. Our hypothesis is that this is due to the fact that CNNs are trained on large datasets such as Imagenet ILSVRC [12] purely for the image classification task. By learning with the sole purpose of predicting a global image label, CNNs become insensitive to local details and geometry and hence work poorly for matching. This effect can be reversed by fine-tuning the model on substantial amounts of data strongly supervised with bounding box [16] or keypoint [9] annotations. While this allows to use CNNs as excellent object and keypoint detectors, it defeats the purpose of using CNN features as generic descriptors for *discovering* correspondences in an unsupervised manner, as matching requires.

In this paper, we address this issue by introducing a new deep architecture that can learn *representations that work well for cross-instance matching* (Figure 1), while using *exactly the same supervision* as traditional pre-training – namely image-level labels used to train categorizers on ILSVRC12 [12]. Using only image-level labels for matching amounts to weak supervision since the labels do not provide any information on the geometry of objects or scenes.

Our key insight is that a set of *diverse* and *sparse* filter responses provides a powerful representation for establishing matches. Convolutional features that respond sparsely on an image tend to automatically *anchor* to distinctive image structures such as semantic object parts. Further enforcing diversity of the filter bank responses results in a good coverage. This yields a unique description for *all* object fragments which is an essential property that enables reliable estimation of *dense* semantic correspondences.

We incorporate this idea by extracting from information-rich residual hypercolumns (section 3.1) a bank of distinctive and diverse filters with orthogonal responses (section 3.2; Figure 2). In this framework, which we call *AnchorNet*, geometric consistency is not imposed explicitly, but emerges spontaneously. We also show how to compress banks of class-specific filters into a class-agnostic bank (section 3.3) which works well for all classes.

Extensive experiments show that the proposed representation can be seamlessly leveraged by state-of-the-art semantic matching methods such as the Deformable Spatial Pyramid [32] or Proposal Flow [21] in order to improve their performance (section 4.1). For the first time, we also show that high-level correspondences can be established between objects of different categories, including new ones, unseen during the training of our network (section 4.2).

2. Related Work

Finding dense correspondences. The classical matching methods estimate very accurate pixel correspondences between two images of the same scene, in presence of moderate viewpoint variations [25, 39, 44]. Early methods use different hand-crafted features such as SIFT [38], HoG [11], SURF [4] or DAISY [52]. This task has many applications including stereo matching [44], optical flow [25, 59], or wide baseline matching [39, 61].

Recent works have generalized the notion of flow to image pairs that are only semantically related [34, 46, 32, 50, 21]. This requires handling a higher degree of variability in appearance. The semantic alignment task also finds many applications such as image completion [3], enhancement [20], or segmentation [34], and video depth estimation [30]. The SIFT Flow algorithm [35, 34] pioneered the idea of dense correspondences across different scenes and proposes a multi-resolution image pyramid and a hierarchical optimization algorithm for efficiency. This approach got extended by the Deformable Spatial Pyramid (DSP) algorithm [32] that introduced a multi-scale regularization with a hierarchically connected pyramid of graphs. The generalized deformable spatial pyramid [28] improves over DSP by enforcing additional spatial constraints at a significant computational cost. The Patch Match method [2] and its extension [3] target general purpose matching, including cross-instance matching. The method of [5] builds an exemplar-LDA classifier for every pixel to obtain dense correspondences that improve the performance of scene flows. Pro-

posal Flow [21] leverages the recent development in object proposals and uses local and geometric consistency constraints to establish dense semantic correspondences. Finally, WarpNet [29] learns correspondences by exploiting the relationships within a fine-grained dataset.

A few methods [26, 27, 45, 31, 41, 62] have posed the problem of finding correspondences as the joint alignment of multiple pairs of images, defining the task of collective alignment. These methods assume sets of images that share a category label and consistent viewpoints. The latest method in this field is FlowWeb [62], that builds a fully connected graph with images as nodes, and pairwise flow fields as edges. Yet, this method scales poorly with the size of the image collection, and it is not straightforward to establish pairwise alignments between new samples.

Deep features for correspondences. Long *et al*. [37] studied the application of CNN features pre-trained on large classification datasets for finding correspondences between object instances. They found that CNN features perform on par with hand-crafted alternatives such as SIFT for the weakly-supervised keypoint transfer problems, and can outperform them when keypoint supervision is available. This work paved the way to new deep architectures trained for finding dense correspondences between same object or scene instances [13, 58, 51]. Recently, Choy *et al*. [9] proposed a deep architecture that performs well at cross-instance alignment, but requires strong supervision in form of many keypoint matches.

The question of training deep features without keypoint annotations still remains unanswered, as state-of-the art semantic matching methods [32, 21] still rely on hand-engineered SIFT and HoG respectively.

3. Method

The output of a deep convolutional layer in a CNN is a tensor $\mathbf{x} \in \mathbb{R}^{H \times W \times D}$ of height H, width W, and with D feature channels. Thus, at each spatial location (u, v), one obtains a D-dimensional feature vector $\mathbf{d}_{uv} = (x_{uv1}, \ldots, x_{uvD})$. As noted by [10], such CNN feature vectors are analogous to hand-crafted dense descriptors like HoG and Dense-SIFT and can often be used as a plug-and-play replacement for the latter in applications. However, as noted in e.g. [37] and shown in the experiments, this substitution does not work well for cross-instance matching algorithms such as DSP [32] and Proposal Flow [21].

Since CNNs can be turned in excellent keypoint detectors by fine-tuning on data strongly annotated with keypoint labels [9, 53], the reason for this failure must be in the way most CNNs are pre-trained on image classification tasks. Note that collecting keypoint annotations for every category does not scale and defeats the purpose of cross-instance matching, which is to discover such correspondences automatically. As a solution, we propose a new architecture that, while using the same image-level supervision as the standard pre-training on the classification task, learns features with better geometric awareness.

Our method is motivated by a simple observation. Suppose that learning encourages a feature to respond very locally (ideally a point). A convolutional filter can do this only by responding to a visual structure that occurs uniquely in each image – hence the distinctive part or keypoint of an object. We call the latter the *anchoring principle*. A geometry-aware representation suitable for semantic matching should discover such a complete set of features that ultimately covers the whole object. We can do so by learning a bank of filters that respond to complementary image locations. We call this the *diversity principle*. Note that diversity indirectly encourages anchoring, as, if features respond to different parts of an image, they must also respond locally. Armed with these insights, we propose next an architecture termed AnchorNet that follows the two principles. We then show that these are sufficient to significantly boost the geometric awareness of the resulting features. A diagram of our network is presented in Figure 3.

3.1. Residual hypercolumns

We base our AnchorNet architecture on the powerful residual architectures of [24]. We select the ResNet50 model as a good compromise between speed and accuracy.

In order to improve the geometric sensitivity of the representation, we follow [22] and extract hypercolumns (HC). A HC $\mathbf{d}_{uv}$ at location (u, v) in the image is created by concatenating the convolutional feature responses at that location for different layers of the network. Recall that, in most CNN architectures, deeper features have reduced resolution; HC compensates for this by upsampling the responses to a common size before concatenation. We denote the resulting network $\mathbf{d} = \Phi(I)$, where I is the input image.

In more detail, we bilinearly upsample and concatenate the rectified outputs of the res2c, res4c and res5c layers [24] into a $56 \times 56 \times D$ hypercolumn tensor. Before concatenation, descriptors extracted at each layer are compressed by PCA to 256 dimensions (PCA is implemented as a 1×1 filter bank) and ℓ^2 normalized to balance their energies. This results in $D = 768$ dimensional HC vectors.

3.2. Learning anchoring features for an object type

The residual HC are high-capacity descriptors reflecting both high-level semantics as well as low-level image details. While this suggests that they should contain enough information for establishing matches, their direct utilization leads to suboptimal results. Thus, we train a set of 3×3 convolutional filters $F_1, ..., F_K$ that compress the HC responses into a compact set of *anchor filters* that are suitable for matching. To this end, we learn filters that satisfy two

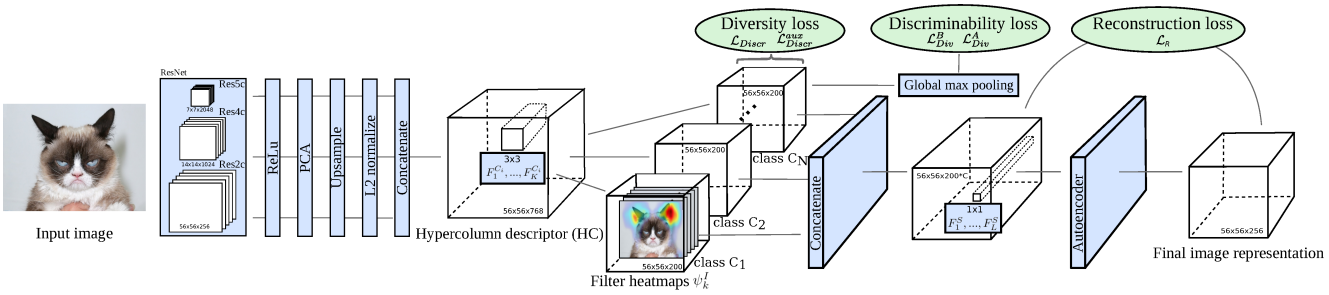


Figure 3: The proposed AnchorNet architecture. First, images are described using hypercolumn descriptors. Sparse filters are discovered for each category using a set of discriminability and diversity losses. Finally a denoising auto-encoder learns how to share these filters between categories, leading to a final category-agnostic representation generalizing to new classes.

properties: discriminability and diversity.

Discriminability constraints. We start by learning filters F_k predictive of an object category. As a result, the filters tend to focus on relevant foreground objects, and rarely on the background. Without loss of generality, we first consider a binary setting where images I are either containing object instances of a single object category ($y_I = 1$) or irrelevant background ($y_I = -1$). We later extend to multiple categories in section 3.3.

Learning uses a large dataset of images with cheap-to-obtain image-level class labels. We follow common deep networks [49, 24, 33] and use ILSVRC 12 [47] for training.

Discriminability is encouraged by minimizing the following loss function:

$$\mathcal{L}_{\text{Discr}}(I, y_I; \Phi, F) = -y_I \sum_{k=1}^{K} \text{gmax}\ \psi(F_k * \Phi(I)), \quad (1)$$

where $\Phi(I)$ denotes the HC tensor extracted from image I. The function $\psi(z) = \log(1+\exp(z))$ is the smooth version of ReLU [42] and gmax is the global max-pooling operator.

Minimizing $\mathcal{L}_{\text{Discr}}$ identifies the strongest response of each filter F_k in the image and then enhances or suppresses it depending on whether the image contains the object. A disadvantage is that, due to the global max-pooling, the backpropagated signal is extremely sparse, which makes learning slow. To speed-up the convergence rate, we introduce a secondary loss function that, for negative images only, generates much denser gradients by using global average pooling (gavg) instead of max pooling:

$$\mathcal{L}^{\text{aux}}_{\text{Discr}}(I, y_I; \Phi, F) = \delta_{[y_I=-1]} \sum_{k=1}^{K} \text{gavg}\max\{0, F_k * \Phi(I)\}. \quad (2)$$

Using global average pooling is meaningful for the negative images, where all responses should be suppressed, but not for the positive ones, where only selected responses should be enhanced.

Diversity constraints. Discriminability alone encourages filters to respond to the object; however different filters may learn to respond to redundant highly-distinctive object parts. In order to obtain good coverage (and ultimately good anchoring), we require the filters F_k of one class to be active on *diverse* regions.

The diversity constraint is implemented by two *diversity losses* $\mathcal{L}^A_{\text{Div}}$ and $\mathcal{L}^B_{\text{Div}}$, encouraging orthogonality of the filters and of their responses, respectively. $\mathcal{L}^A_{Div}$ makes filters orthogonal by penalizing their correlations, as follows:

$$\mathcal{L}^A_{\text{Div}}(F) = \sum_{i \neq j} \left| \sum_p \frac{\langle F_i^p, F_j^p \rangle}{\|F_i^p\|_F \, \|F_j^p\|_F} \right| \quad (3)$$

where F_i^p is the column of filter F_i at spatial location p[1]. Note that orthogonal filters are likely to respond to different image structures, but this is not necessarily the case. Thus, we introduce a second term $\mathcal{L}^B_{\text{Div}}$ that directly decorrelates the filters' *response maps* $\psi^I_k \doteq \psi(F_k * \Phi(I))$:

$$\mathcal{L}^B_{\text{Div}}(I; \Phi, F) = \sum_{i \neq j} \left\| \frac{\langle \psi_i^I, \psi_j^I \rangle}{\|\psi_i^I\|_F \|\psi_j^I\|_F} \right\|^2. \quad (4)$$

This term is further regularized by smoothing the response maps $\psi^I_k \doteq g_\sigma * \psi(F_k * \Phi(I))$ prior to computing the loss $\mathcal{L}^B_{\text{Div}}$, where g_σ is a Gaussian kernel; this encourages filter responses to spread farther apart by dilating their activations. Note that inducing diversity among classifier prediction has been explored before [15, 19, 18, 48, 6], however none of these works consider diversity as a loss to train a deep representation as we propose.

Discussion. By making a large number of filters F_k both discriminative and diverse, our method indirectly encourages them to become highly-specialized and hence to respond to unique parts of objects (the anchoring principle). This happens automatically, without enforcing such geometric properties explicitly. This intuition is strongly supported by our experiments. Examples of the filters learned

[1] *i.e.* for our 3×3 filters F_i, $p \in \{1, 2, \ldots, 9\}$

for the bird and dog classes are presented in Figure 2 (a) and (b). It is apparent that filters fire on consistent object parts despite large intraclass variations, demonstrating the power of our formulation and its applicability to matching.

3.3. Class-agnostic representation

In the previous section we have defined category specific anchoring filters. In this section, we extend them to be generic to any category. This allows to use the same representation for every image, irrespective of its label, to match instances across different categories (*e.g.* dog vs cat), and to even handle new categories.

First, a filter bank $F_1^{C_i}, ..., F_K^{C_i}$ is learned for each object category $C_1, \ldots, C_N$ using the method above. Each object is learned by considering only images C_i of that object class and a common background class B. Since filters are not learned to discriminate between objects, and since the diversity losses are applied only *within* each bank, different filter banks can develop correlations. Figure 2 illustrates this by showing that filters learned for the "dog" and "bird" classes capture similar concepts such as eyes or nose.

We take advantage of the overlap between different banks by introducing a new bank of 1×1 filters $F_1^S, ..., F_L^S$ that projects the class-specific responses of the filters $F_1^{C_1}, \ldots, F_K^{C_N}$ to L general-purpose response maps applicable to objects of any class.

In order to learn the projections F^S end-to-end, we add a *denoising autoencoder* (DAE) [56] to our architecture. DAE minimizes the *reconstruction loss* $\mathcal{L}_R(F^S, \hat{\Gamma})$

$$\mathcal{L}_R(F^S, \hat{\Gamma}) = \mathcal{D}(\hat{\Gamma}, (F^S)^\top * F^S * c(\hat{\Gamma})) \qquad (5)$$

where $\mathcal{D}(\mathbf{a}, \mathbf{b}) = \|\mathbf{a}/\|\mathbf{a}\| - \mathbf{b}/\|\mathbf{b}\|\|^2$ is the ℓ^2 distance between the ℓ^2 normalized tensors $\mathbf{a}$ and $\mathbf{b}$ and $(F^S)^\top$ is the *convolution transpose* operator [55]. Here $\hat{\Gamma} = \Gamma - \mu(\Gamma)$ denotes the stack of class-specific heatmaps $\Gamma = \text{stack}(\psi_{F_1^{C_1}}, \ldots, \psi_{F_K^{C_N}}) \in R^{W \times H \times (KN)}$ centered by removing their mean $\mu(\Gamma)$, estimated online during training. We have observed that centering followed by ℓ^2 normalization greatly improves the convergence properties of $\mathcal{L}_R$. Function $c(\mathbf{z})$ injects noise by randomly setting to zero 25% of the feature channels of the tensor $\mathbf{z}$.

The decorrelation loss eq. (3) is applied to the compression filters F^S as well in order to encourage their diversity.

Note that the reconstruction loss $\mathcal{L}_R$, when optimized end-to-end with the rest of the model, encourages the maps $\hat{\Gamma}$ to shrink (because, if $\hat{\Gamma} = 0$ everywhere, then the autoencoder has a trivial optimum). This is however prevented by the decorrelation losses $\mathcal{L}_{\text{Div}}^A$, $\mathcal{L}_{\text{Div}}^B$. $\mathcal{L}_R$ thus works as a regularizer enforcing part sharing. Examples of the learned class agnostic filters are in fig. 2 (c).

Denoising autoencoders have been used for domain adaptation before [7, 17]. In a similar spirit, the last part of our network transforms a set of class (domain) specific filters into a domain invariant representation that can accommodate for any class, even the one not seen during training.

Network training. AnchorNet is optimized with stochastic gradient descent (SGD) by minimizing the sum of the proposed losses $\mathcal{L}_{\text{Discr}}$, $\mathcal{L}_{\text{Discr}}^{\text{aux}}$, $\mathcal{L}_{\text{Div}}^A$, $\mathcal{L}_{\text{Div}}^B$ and $\mathcal{L}_R$, with mini batches of size 16, a learning rate of 10^{-2}, and a momentum of 0.0005. Parameters of the network are initialized with the ResNet50 model pre-trained on ILSVRC12. We use two-stage optimization to speed up the training process. First, the class-specific filters $F_i^{C_k}$ are trained on 4×10^4 training images independently for each object class C_k keeping the rest of the network parameters fixed. Then, we attach the autoencoder and the reconstruction loss to fine-tune all the network parameters end-to-end on 12×10^3 images. Further details are provided in the supplementary material.

4. Experiments

We thoroughly compare our method with existing techniques for semantic matching (section 4.1). Then, we assess how well our features allow to establish matches across images of different categories (section 4.2) which, to the best of our knowledge, was never demonstrated before.

Note that for all reported results, *training only uses ILSVRC12* [12] images and labels, where the categories are merged according to the PASCAL-ILSVRC class mapping from [12] (*e.g. sofa* is a merge of "studio couch" and "day bed"). In this manner, 231 ILSVRC classes are used as positive examples spread over the 20 PASCAL VOC classes; the remaining 769 classes are used to form the set B of negative (background) images. Even when we report results on one of the $N = 20$ PASCAL VOC [14] classes, *none* of the PASCAL VOC training data is used.

4.1. Dense pairwise semantic matching

We follow the standard practice [62, 21] of using a dataset with manually annotated semantic keypoints or regions and assess how well a semantic matching method in combination with different types of features transfers the annotations from an image to another. We experiment on three datasets following their evaluation protocol.

Compared methods. The most successful cross-instance matching methods include DSP [32] and Proposal Flow [21] (PF). In their original formulation, these methods performed best with the Dense SIFT [38] feature for DSP, and the whitened version of HoG [23] for PF. In the following experiments, we replace these descriptors with our representation, as follows.

For DSP, the learned filter banks produce a dense field of feature vectors which are bilinearly upsampled to the original image size, ℓ_2 normalized and passed to DSP as a plug-and-play replacement of Dense SIFT. For PF, we mimic

	mean	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	mbike	person	plant	sheep	sofa	table	train	tv
Pairwise alignment methods																					
DSP + ANet-class	**0.45**	**0.31**	**0.49**	**0.32**	0.53	**0.75**	**0.51**	**0.47**	0.23	0.53	**0.37**	0.20	0.33	**0.41**	**0.22**	**0.46**	**0.45**	**0.77**	0.45	0.48	**0.74**
DSP + ANet	**0.45**	0.29	0.47	0.29	0.52	0.73	0.50	0.46	**0.25**	0.53	**0.37**	**0.21**	**0.34**	0.39	0.20	0.44	0.45	**0.77**	0.45	0.51	**0.74**
DSP + HC	0.41	0.29	0.45	0.24	0.51	0.73	0.48	0.44	0.20	0.52	0.32	0.16	0.28	0.35	0.19	0.39	0.37	0.74	0.44	0.48	0.67
DSP + SIFT [32]	0.39	0.25	0.46	0.21	0.48	0.63	0.50	0.45	0.19	0.48	0.30	0.14	0.26	0.35	0.13	0.40	0.37	0.66	0.37	0.48	0.62
Proposal Flow + ANet-class	0.43	0.26	0.43	0.28	**0.54**	0.71	0.50	0.45	0.24	**0.54**	0.32	**0.21**	0.28	0.35	0.21	0.45	0.40	0.74	**0.46**	0.50	0.70
Proposal Flow + ANet	0.42	0.26	0.41	0.26	0.53	0.70	0.49	0.45	**0.25**	**0.54**	0.31	0.19	0.28	0.31	0.17	0.43	0.39	0.74	0.44	**0.52**	0.69
Proposal Flow + HC	0.42	0.26	0.42	0.26	**0.54**	0.70	0.50	0.45	0.23	0.53	0.32	0.18	0.27	0.32	0.18	0.43	0.38	0.74	0.45	0.51	0.64
Proposal Flow + HoG [21]	0.41	0.25	0.45	0.23	**0.54**	0.70	0.49	0.44	0.19	0.53	0.30	0.16	0.25	0.35	0.16	0.41	0.35	0.74	0.44	0.50	0.63
Baseline: NoFlow	0.39	0.27	0.40	0.22	0.50	0.73	0.46	0.42	0.20	0.51	0.30	0.15	0.25	0.32	0.18	0.38	0.34	0.74	0.44	0.47	0.64
Collective alignment methods																					
FlowWeb [62]	0.43	0.33	0.53	0.24	0.51	0.72	0.54	0.51	0.20	0.52	0.32	0.15	0.29	0.45	0.19	0.41	0.39	0.73	0.41	0.51	0.68

Table 1: Weighted IoU for pairwise **semantic part matching** on PASCAL Parts. The proposed methods are in **bold**.

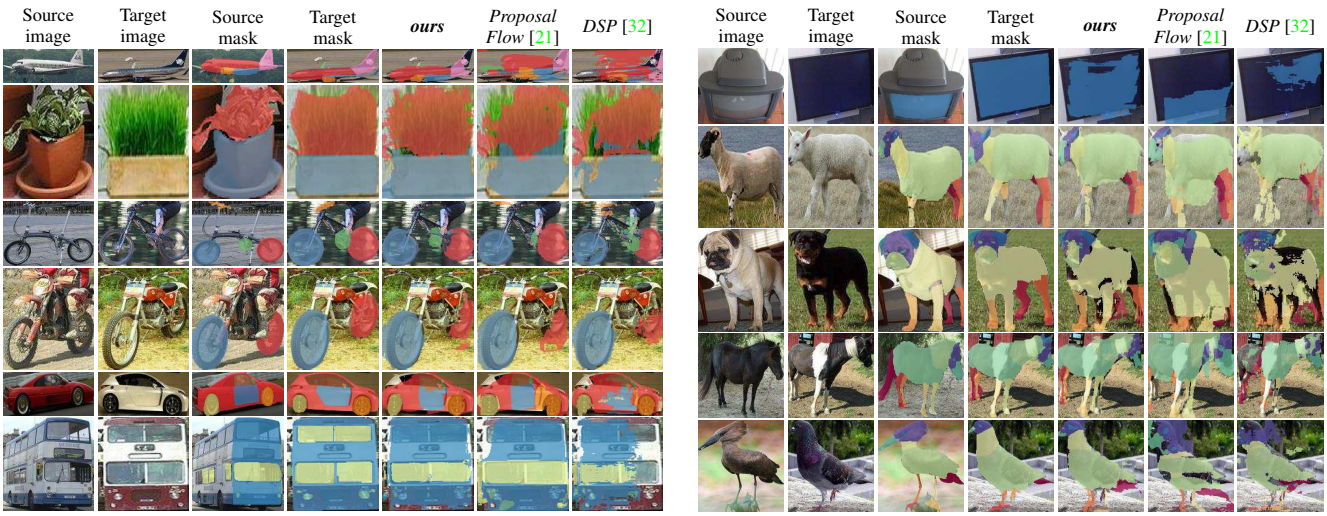


Figure 4: **Segmentation mask transfer** on PASCAL Parts for DSP+ANet (ours), Proposal Flow + HoG, and DSP + SIFT.

their use of HoG: every object proposal serves as a pooling region for the set of filter activations that are extracted once for every image. The pooling is performed by reading-off the filter activations inside the region and resizing them to 8×8 using bilinear interpolation. This tensor is then vectorized and ℓ^2 normalized to form the final descriptor of the proposal region. We use the variant of PF that extracts 1000 selective search boxes [54] per image. The rest of the matching procedure is identical to the original PF algorithm.

We compare both the class-agnostic (**ANet**) and class-specific (**ANet-class**) variants of our anchor filters. The class-agnostic variant uses the 256 dimensional features produced by the autoencoder filters F^S, whereas ANet-class uses the output of the class-specific filters F^{C_i} corresponding to a given PASCAL VOC object category C_i. Thus, ANet-class assumes knowledge of the object class label while ANet is universally applicable without requiring additional image-specific information. As baseline descriptors we consider SIFT, HoG and HC descriptors formed by concatenating the PCA projected layers of ResNet50 (res2c, res4c and res5c - section 3.1). We also report the NoFlow baseline that predicts zero-displacement for every pixel.

While we focus on pairwise matching, an alternative is to align many images together, known as co-alignment. Among various co-alignment methods, including [26, 45, 31], FlowWeb [62] is currently the state of the art. Due to its superior performance, we only report results for FlowWeb; however, while FlowWeb works very well, it is important to note that it is also substantially more expensive than pairwise matching, does not scale well and cannot accommodate for new image pairs.

Evaluation of segmentation masks transfer. We compare the various methods on the task of transferring semantic part segmentation masks, strictly following the protocol of [62]. Dense semantic matches, as determined by DSP or PF given a descriptor, are used to warp the part segmentation mask from a source to a target image. The matching quality is assessed as the average weighted intersection-over-union (IoU) between the predicted masks and the ground-truth ones for different semantic parts. The results are reported in Table 1, qualitative results are provided in Figure 4.

We make the following observations. First, the ResNet50 features, perform at most marginally better, than SIFT or HoG, while both ANet and ANet-class features improve performance for both DSP (+6% IoU) and PF (+1%

	mean	aero	bike	boat	bottle	bus	car	chair	mbike	sofa	table	train	tv
Pairwise alignment methods													
DSP + ANet-class	**0.24**	**0.23**	0.28	**0.06**	0.38	**0.44**	**0.39**	**0.14**	**0.19**	0.16	0.11	0.13	**0.41**
DSP + ANet	0.23	0.22	0.25	0.06	0.35	0.42	0.34	**0.14**	0.17	**0.17**	**0.13**	**0.14**	0.40
DSP + HC	0.20	0.20	0.23	0.05	**0.39**	0.36	0.25	0.10	0.15	0.12	0.10	0.12	0.28
DSP + SIFT [32]	0.18	0.17	**0.30**	0.05	0.19	0.33	0.34	0.09	0.17	0.12	0.09	0.12	0.18
Proposal Flow + ANet-class	0.17	0.17	0.21	0.05	0.25	0.26	0.27	0.10	0.14	0.12	0.07	0.10	0.24
Proposal Flow + ANet	0.16	0.16	0.19	0.05	0.22	0.26	0.25	0.10	0.12	0.11	0.05	0.12	0.23
Proposal Flow + HC	0.16	0.17	0.21	0.05	0.23	0.27	0.24	0.09	0.13	0.12	0.05	0.11	0.20
Proposal Flow + HoG [21]	0.17	0.20	0.26	0.05	0.20	0.31	0.29	0.10	0.17	0.13	0.05	0.13	0.21
Baseline: NoFlow	0.17	0.18	0.17	0.05	**0.39**	0.31	0.17	0.09	0.12	0.11	0.07	0.11	0.24
Collective alignment methods													
FlowWeb [62]	0.26	0.29	0.41	0.05	0.34	0.54	0.50	0.14	0.21	0.16	0.04	0.15	0.33

Table 2: PCK ($\alpha = 0.05$) for semantic keypoint transfer on the 12 rigid classes of the PASCAL Parts dataset.

IoU). Second, the class-specific features ANet-class perform on par with the class-agnostic features ANet, demonstrating the ability of our domain generalization approach to compress the class-specific filters into the class-agnostic ones. Third, our features, in combination with DSP, exhibit the best average performance among all the compared methods. Remarkably, both ANet and ANet-class outperform all co-alignment methods, including FlowWeb [62], achieving state-of-the-art results on this dataset. This is an interesting finding as the co-alignment methods exploit the small viewpoint and appearance variations in order to improve pairwise alignments.

Evaluation of keypoint matching. We also evaluate performance on matching semantic keypoints. Corresponding annotations are provided by [60] for the 12 rigid PASCAL VOC categories. Similar to the previous section, we use the dataset from [62], and, strictly following their evaluation protocol, we assess the matching accuracy using PCK, setting the misalignment tolerance parameter α to 0.05.

Table 2 contains the results of this experiment. Our features improve the original DSP results by a large margin (+6% PCK), obtaining state-of-the-art results on this dataset among the pairwise alignment methods. Pairwise matching becomes in fact competitive with the results obtained by FlowWeb in co-alignment, although the latter use more information. Proposal Flow is generally weaker on this task and is not helped by the better features.

Evaluation of region matching. As a third benchmark dataset, we use the PF dataset and corresponding protocol as described in detail in [21]. The dataset contains 10 image sets of 4 object types and the task is to establish matches between annotated semantic regions within the image sets. We report region matching precision using the definitions specified in [21]. Table 3 contains the results obtained by using the code and data made available by [21].

We evaluate our deep features in combination with the two matching methods presented in [21]: the best performing local offset matching (LOM), and the naive appearance matching (NAM). ANet is compared with the best performing feature from [21], *i.e.* HoG [23]. We observe that

Matching \ Feature	AuCs for PCR		
	ANet-class	**ANet**	HOG [21]
NAM: baseline	**0.41**	0.36	0.29
LOM: Proposal Flow	**0.46**	0.43	0.43

Table 3: **Region matching** on the PF dataset.

Matching Alg.	DSP			Proposal Flow			NoFlow
Feature	**ANet**	HC	SIFT	**ANet**	HC	HoG	-
PCK ($\alpha = 0.05$)	0.11	0.08	0.06	**0.13**	0.09	0.06	0.04
PCK ($\alpha = 0.1$)	0.24	0.18	0.12	**0.32**	0.25	0.18	0.12

Table 5: **Semantic matching** on the AnimalParts dataset. For each method, we report the average PCK over all possible 12x12 domain pairs. An overview of individual cross-category results can be found in Figure 5

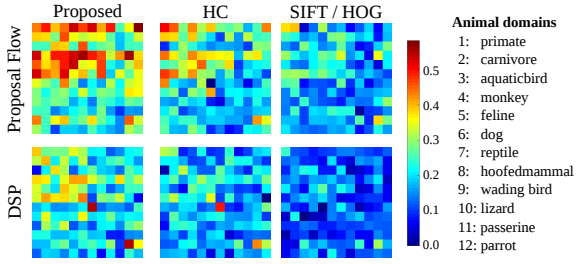


Figure 5: **Per-domain semantic matching** on the AnimalParts dataset. Cells are colored proportionally to the matching performance on a given animal class pair. Columns denote the source domains, rows the targets.

using ANet-class features in combination with both matching methods (LOM, NAM) brings a significant performance improvement. Note in particular that ANet-class is sufficiently powerful to make the NAM baseline, which does not use any sophisticated geometric reasoning, competitive with the LOM+HoG, which uses geometric reasoning but handcrafted features (LOM+ANet-class is even better).

4.2. Generalization across categories

The previous section experimented on the task of aligning different object instances of the same category. Here, we depart from this scenario and consider instead *cross-*

Source class Target class	**mean**	bicycle ↓ mbike	mbike ↓ bicycle	bus ↓ car	car ↓ bus	bus ↓ car	dog ↓ cat	cat ↓ dog	sheep ↓ dog	dog ↓ sheep	horse ↓ cow	cow ↓ horse	sheep ↓ cow	cow ↓ sheep
DSP + ANet	**0.37**	**0.35**	**0.45**	**0.52**	**0.35**	0.36	**0.25**	0.25	**0.34**	**0.27**	**0.31**	**0.47**	**0.37**	**0.58**
DSP + HC	0.32	0.27	0.44	0.48	0.32	0.34	0.20	0.21	0.22	0.23	0.27	0.40	0.28	0.54
DSP + SIFT [32]	0.29	0.28	0.40	0.40	0.27	0.30	0.16	0.16	0.20	0.19	0.26	0.31	0.28	0.50
Proposal Flow + ANet	0.35	0.32	0.38	0.50	0.32	**0.37**	0.23	**0.27**	0.30	0.25	0.29	0.41	0.32	0.53
Proposal Flow + HC	0.33	0.31	0.34	0.49	0.29	0.35	0.22	0.24	0.28	0.23	0.28	0.41	0.32	0.53
Proposal Flow + HOG [21]	0.31	0.30	0.43	0.48	0.30	0.35	0.19	0.21	0.22	0.19	0.25	0.37	0.29	0.50
Baseline: NoFlow	0.27	0.26	0.44	0.35	0.26	0.25	0.17	0.18	0.22	0.17	0.22	0.29	0.26	0.49

Table 4: Weighted IoU for **cross instance semantic part matching** on PASCAL Parts.

Figure 6: **Cross-class alignments** on the AnimalParts dataset. Given a target (top row) and source images (bottom row) we establish semantic correspondences between parts of animal classes. The alignment warps the source image into the target image. We compare Proposal Flow + ANet (ours - 2nd row) and Proposal Flow + HoG [21] (3rd row).

category matching, where correspondences are established between objects of different categories. To the best of our knowledge, this is the first time this task is considered.

For evaluation, we first use the PASCAL Parts [8] data from [62]. Parts with different location qualifiers are merged into one (e.g. "left-leg" and "right-leg" are merged into "leg") to ensure shareability across categories. Overall, there are 9 object categories and 13 shared part types.

Second, we consider the AnimalParts [43] dataset, introduced as a test-bed to study the transferability of semantic part detectors. Here, we reuse the dataset in order to assess transferability of ANet filters trained without explicit supervision. AnimalParts includes only a few part types ("eye" and "foot"), but a large number of different categories – 100 animals from the ILSVRC12 dataset. In order to present results compactly, animals are grouped in 12 families, based on the WordNet [40] hierarchy. For each pair of super-classes, 40 image pairs are randomly sampled for evaluation, resulting in ∼7K image pairs in total. PCK is computed for each pair of super-classes, and the results are averaged over such pairs. The class-specific ANet-class does not apply since the goal is to match across categories and most of these categories were not seen during training.

Tables 4 and 5 and Figure 5 show that ANet works substantially better than other matching methods. For the AnimalParts, the best results are obtained with Proposal Flow in combination with our features, with a 7% PCK improvement over the PF + HoG baseline ($\alpha = 0.05$). The fact that AnimalParts contains categories unseen at train time (*e.g.* reptiles) demonstrates the scalability and generalization of the proposed approach. For PASCAL Parts, similar to the intra-class matching experiment (section 4.1), DSP performs best. Here ANet attains a 16% relative improvement over the best previously published method (Proposal Flow + HoG). Figure 6 provides qualitative results.

5. Conclusion

In this paper we have examined the problem of dense semantic matching. Employing the concept of filter anchoring, we have designed a novel deep architecture, termed AnchorNet. Supervised with only image-level labels, AnchorNet automatically learns a set of filters which respond in a sparse and geometrically consistent manner across object instances. Thanks to these filters, our architecture produces powerful representations for image matching. We experimentally validate these features in conjunction with state-of-the art semantic matching methods attaining state-of-the-art performance on the segmentation transfer and keypoint matching tasks. Versatility of our representation has been demonstrated on the new task of cross-category matching where we report positive results on two test-beds.

Acknowledgments. We would like to thank Xerox Research Center Europe and ERC 677195-IDIU for supporting this research.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 2

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. 2009. 2

[3] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *Proc. ECCV*, 2010. 2

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, 2008. 2

[5] H. Bristow, J. Valmadre, and S. Lucey. Dense semantic correspondence where every pixel is a classifier. In *Proc. ICCV*, 2015. 2

[6] T. T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE IT*, 57:4680–4688, 2011. 4

[7] M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. In *Proc. ICML*, 2012. 5

[8] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proc. CVPR*, 2014. 8

[9] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *Proc. NIPS*. 2016. 2, 3

[10] M. Cimpoi, S. Maji, and A. Vedaldi. Deep convolutional filter banks for texture recognition and segmentation. In *Proc. CVPR*, 2015. 3

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 1, 2

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*, 2009. 2, 5

[13] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, 2015. 3

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. 2010. 5

[15] A. Gane, T. Hazan, and T. S. Jaakkola. Learning with maximum a-posteriori perturbation models. In *Proc. AISTATS*, 2014. 4

[16] R. Girshick. Fast r-cnn. In *Proc. ICCV*, 2015. 1, 2

[17] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proc. ICML*, 2011. 5

[18] A. Guzman-Rivera, D. Batra, and P. Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Proc. NIPS*, 2012. 4

[19] A. Guzman-Rivera, P. Kohli, D. Batra, and R. A. Rutenbar. Efficiently enforcing diversity in multi-output structured prediction. In *Proc. AISTATS*, 2014. 4

[20] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. 2011. 2

[21] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *Proc. CVPR*, 2016. 1, 2, 3, 5, 6, 7, 8

[22] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. CVPR*, 2015. 3

[23] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Proc. ECCV*, 2012. 5, 7

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proc. CVPR*, 2016. 3, 4

[25] B. K. P. Horn and B. G. Schunck. Determining optical flow: A retrospective. *Artif. Intell.*, 59(1-2):81–87, 1993. 2

[26] G. B. Huang, V. Jain, and E. G. Learned-Miller. Unsupervised joint alignment of complex images. In *Proc. ICCV*, 2007. 3, 6

[27] G. B. Huang, M. A. Mattar, H. Lee, and E. G. Learned-Miller. Learning to align from scratch. In *Proc. NIPS*, 2012. 3

[28] J. Hur, H. Lim, C. Park, and S. C. Ahn. Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation. In *Proc. CVPR*, 2015. 2

[29] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *Proc. CVPR*, 2016. 3

[30] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *Proc. ECCV*, 2012. 2

[31] I. Kemelmacher-Shlizerman and S. M. Seitz. Collection flow. In *Proc. CVPR*, 2012. 3, 6

[32] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proc. CVPR*, 2013. 1, 2, 3, 5, 6, 7, 8

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 4

[34] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011. 2

[35] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proc. ECCV*, 2008. 1, 2

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 1

[37] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Proc. NIPS*, 2014. 1, 3

[38] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 5

[39] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, 2002. 2

[40] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41, 1995. 8

[41] H. Mobahi, C. Liu, and W. T. Freeman. A compositional model for low-dimensional image set representation. In *Proc. CVPR*, 2014. 3

[42] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, 2010. 4

[43] D. Novotny, D. Larlus, and A. Vedaldi. I have seen enough: Transferring parts across categories. In *Proc. BMVC*, 2016. 8

[44] M. Okutomi and T. Kanade. A multiple-baseline stereo. *PAMI*, 15(4):353–363, 1993. 2

[45] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *Proc. CVPR*, 2010. 3, 6

[46] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu. Scale-space sift flow. In *Proc. WACV*, 2014. 2

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 4

[48] M. Schiegg, F. Diego, and F. A. Hamprecht. Learning diverse models: The coulomb structured support vector machine. In *Proc. ECCV*, 2016. 4

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. 4

[50] M. Tau and T. Hassner. Dense correspondences across scenes and scales. *PAMI*, 38(5):875–888, 2016. 2

[51] J. Thewlis, S. Zheng, P. Torr, and A. Vedaldi. Fully-trainable deep matching. In *Proc. BMVC*, 2016. 3

[52] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *PAMI*, 32(5):815–830, 2010. 2

[53] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proc. CVPR*, 2015. 3

[54] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104:154–171, 2013. 6

[55] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proc. ACM Int. Conf. on Multimedia*, 2015. 5

[56] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. ICML*, 2008. 5

[57] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proc. CVPR*, 2015. 1

[58] J. Žbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. 17(1):2287–2318, 2016. 3

[59] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *Proc. ICCV*, 2013. 2

[60] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, 2014. 7

[61] H. Yang, W.-Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *Proc. CVPR*, 2014. 2

[62] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proc. CVPR*, 2015. 3, 5, 6, 7, 8

[63] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proc. CVPR*, 2016. 1