Weakly Supervised Manifold Learning for Dense Semantic Object Correspondence

Utkarsh Gaur
University of California Santa Barbara
utkarsh.gaur@cs.ucsb.edu

B. S. Manjunath
University of California Santa Barbara
manj@ece.ucsb.edu

Abstract

The goal of the semantic object correspondence problem is to compute dense association maps for a pair of images such that the same object parts get matched for very different appearing object instances. Our method builds on the recent findings that deep convolutional neural networks (DCNNs) implicitly learn a latent model of object parts even when trained for classification. We also leverage a key correspondence problem insight that the geometric structure between object parts is consistent across multiple object instances. These two concepts are then combined in the form of a novel optimization scheme. This optimization learns a feature embedding by rewarding for projecting features closer on the manifold if they have low feature-space distance. Simultaneously, the optimization penalizes feature clusters whose geometric structure is inconsistent with the observed geometric structure of object parts. In this manner, by accounting for feature space similarities and feature neighborhood context together, a manifold is learned where features belonging to semantically similar object parts cluster together. We also describe transferring these embedded features to the sister tasks of semantic keypoint classification and localization task via a Siamese DCNN. We provide qualitative results on the Pascal VOC 2012 images and quantitative results on the Pascal Berkeley dataset where we improve on the state of the art by over 5% *on classification and over* 9% *on localization tasks.*

1. Introduction

The semantic object correspondence problem has garnered considerable attention in the vision community in recent times. The objective of this problem is to densely match objects that belong to the same semantic category but are completely different in visual appearance space. Figure 1 shows an example of two semantically similar objects namely the California quail and the sharp-shinned hawk. The bottom row shows the dense correspondence map for these two images as computed by our method. To bring out the correctly matched object parts we have drawn lines between them across the two instances. All object parts such as head (red), tail(blue) back(purple/magenta), flank(green) are correctly matched by our method based on the images' class label alone. More examples of what we consider semantically similar objects are shown in figure 2 blocks 1-4.

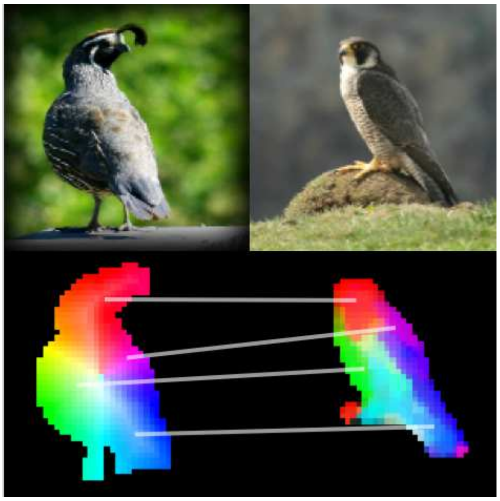

Figure 1. Dense semantic correspondence map for a pair of images visualized through color mapping. The foreground of the left image was artificially colored to mark different semantic regions. The color of the right image was generated by mapping the color of the best matched point on the left image.

Object instances in the natural world appear very different due to a combination of factors including different type (Siamese vs. tabby cat, monorail vs. bullet train, fighter vs. passenger aircraft), pose and viewpoint. Due to the vast variance in the appearance of these instances, traditional hand-crafted feature descriptors such as SIFT[19] or SURF[1] cannot readily match them at the object-part level.

Finding semantic object correspondence is an important problem. A variety of applications can benefit from the matching and recognizing of dense object parts and their geometric structure. Examples of such applications include semantic keypoint classification and localization problem, object part segmentation and complex activity recognition.

Unfortunately, there are no true ground-truth datasets

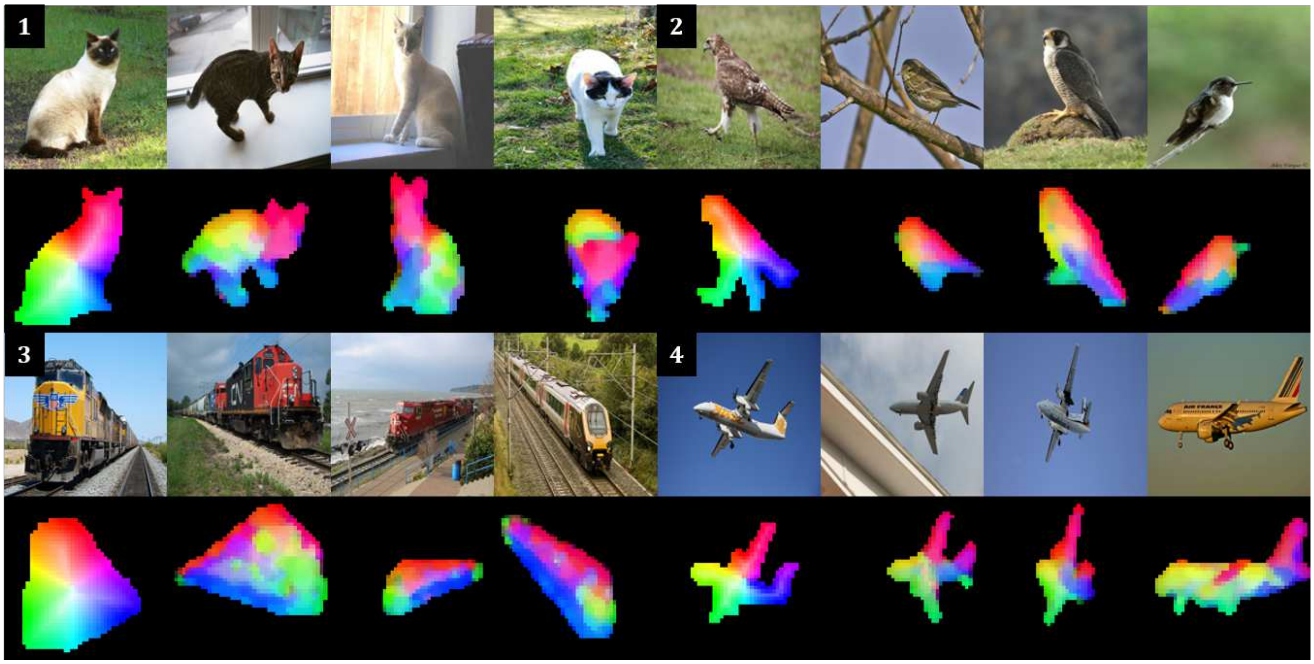

Figure 2. The leftmost image for each block is used as a template image and the three other images are densely matched with it. Dense semantic correspondence visualization scheme is the same as figure 1. Our method can effectively match rigid and non-rigid objects alike.

available for dense semantic associations, making it difficult to use existing deep learning strategies to learn such associations. Thus this problem requires knowledge transfer from other domains for any noteworthy matching performance.

A few research works have provided good results for this problem. The work in [33] presented a clever technique to use cyclic consistency between real images and CAD model images with known parameters to create a deep network supervisory signal. However, this method is limited to rigid objects with available 3D CAD models and still requires human intervention to orient model images to real images. Very recently [6] proposed using optical flow ground-truth to fine-tune a flow network to compute dense correspondence that works well for sparse semantic correspondence.

Contributions In this work we present a method to generate true dense semantic correspondence for rigid and non-rigid objects alike. We propose a novel optimization scheme to re-purpose deep *convolutional* features from a classification network to group semantically similar object parts. This is achieved by maximizing intra-class correlation in a learned embedded feature space regularized by inter-class penalties in the geodesic (feature neighborhood) space. We also introduce a mechanism to circumvent the NP-hardness inherent to the joint analysis of the above factors by breaking the problem down into two well-understood cyclic components, albeit at the cost of an exact solution.

As noted in [32] and [21], DCNN layers tend to implicitly learn object part models at various abstraction levels. Despite this representational ability, their deep features separate objects of different categories on the hyperplane as a result of being trained with a classification loss function. In this work our primary goal is to leverage this representational power and refocus it for recognizing semantic object parts (SOPs). We accomplish this task by utilizing the feature neighborhood context cues extracted from the feature maps of the convolutional layer in conjunction with the feature space similarities, as shown simplistically in figure 3.

The DCNN filters operate in local image neighborhoods (e.g. 3x3) so the spatially local patterns present in natural images extend to the deep convolutional feature maps as well. Intuitively, this implies that deep features in the same feature map locality are highly likely to belong to the same object part, impervious to their similarity in the feature space. For instance, a feature corresponding to the *tail* of the object *bird* should have other *tail* features in its close proximity. These spatially local patterns lend us additional feature connectivity information between deep features which is an excellent indicator of semantic similarity.

We extract feature maps from the convolutional layers and thus get access to the associated feature connectivity information. For this work, we make use of the convolutional feature maps from the $VGG16$ [26] network where a feature map for a given image is a 44x44 dimensional grid and each "pixel" on this grid is a 512 length deep feature. We use pre-trained features from ImageNet dataset and call it the source dataset. We provide qualitative results for the Pascal VOC 2012 dataset and refer to it as target dataset for the rest of the paper.

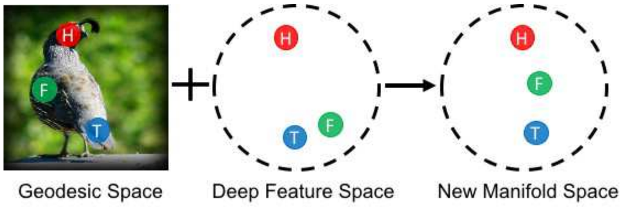

Figure 3. H=head, F=Flank, T=Tail features. Deep feature similarities do not correspond very well to object parts. With the help of geodesic space constraints, we project them to a manifold to obtain embedded features that recognize semantic object parts.

1.1. Related work

Solving dense correspondence is inherently challenging as it belongs to the class of inverse problems. The fundamental strategy for solving correspondence problems is to assume regularity on image properties such as viewpoint and brightness [12],[31]. Hand-crafted well-known features including HOG, SURF and DAISY have found widespread applications to numerous computer vision tasks [1, 28, 7]. However, such methods are susceptible to environmental and scene changes. Inspired by differential techniques for optical flow [20], SIFT-flow first introduced the idea of semantic correspondence by aligning an image to its nearest neighbors [17]. The resulting displacement fields from SIFT-flow are sparse and post processing to obtain dense fields does not yield favorable results. Long et al.[18] originally matched deep features computed at dense locations to perform image alignment. Their performance for the alignment or keypoint dataset is not too far from SIFT-flow.

For many computer vision problems semantic parts bear crucial information, for e.g., keypoint localization and visibility prediction. Traditionally, to solve these problems, the spatial relationships between parts have been modeled as star-shaped using geometric constraints[16], using tree structured methods [35], estimating viewpoints[29] or poselets [3]. These models, however, have struggled against large deformations and shape variance.

Recently, joint image-set alignment has been used to tackle correspondence problem using supervised and fully unsupervised methods as shown by Learned-Miller's congealing procedure[13]. Congealing exhibited excellent results on MNIST data [15] and has been extended further to handle more difficult real-world images [13]. FlowWeb[34] uses compositions of flow fields for modeling a common image structure consistently among a large set of images of the same category. The results however vary widely depending on initialization quality. Carreira et al. [5] propose a middle ground by leveraging class information to infer dense correspondence. They construct a network connecting objects in similar viewpoints. Geodesics on this graph is used for pose prediction and object detection. Taniai et al. [27] compute dense correspondence jointly with a co-segmentation of an image set. Proposal flow[11] utilized object proposals[30] combined with geometric constraints to find dense correspondences for objects. Universal Correspondence Network [6] is a supervised, end-to-end network primarily trained for dense geometric correspondence which is shown to compute sparse semantic correspondences as well. Our method is weakly supervised and produces high quality dense semantic maps. Unlike the methods mentioned above, our method is very robust to strong affine transformations of the object instances.

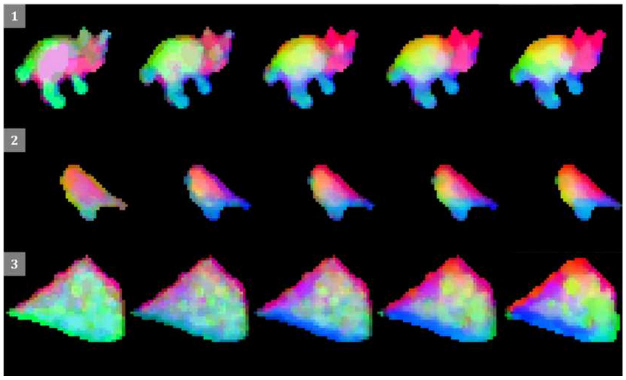

Figure 4. Semantic correspondence performance improvement over the optimization iterations as neighborhood context reshapes the manifold. Images correspond to second column of blocks 1, 2 and 3 from figure 2

Performance quantification Due to the lack of dense semantic correspondence ground-truth data, we look towards other natural applications of this problem to quantify the performance of our method. The keypoint localization and classification problem is closely related to our task since the keypoints are defined semantically on the classes of PASCAL VOC 2011 and thus require the classifier to have a semantic understanding of the dataset.

2. Method

As noted earlier, deep features from pre-trained DCNNs tend to separate inter-class *objects* instead of intra-class object *parts*. However, we aim to learn a manifold such that the features belonging to the same semantic object parts are projected closer to each other on the manifold. Let ϕ be an embedding representing such a manifold, parameterized by w on the deep features f such that $f^{\phi} = \phi(f; w)$, where f^{ϕ} are the embedded features. Directly solving for ϕ by jointly analyzing feature neighborhood connectivity and feature space similarities for all the images is NP-hard. This is due to the exponential growth of connectivity parameters with a linear increase in features.

At the cost of an exact solution, we circumvent the NP-hardness by introducing a transient variable ℓ which represents the semantic object part (SOP) labeling on the entire feature set. Here, an SOP label of a feature point is the index of the semantic object part to which it belongs (an SOP label may correspond to an eye, wing, tail etc. for category *bird*, however the exact label information is neither known nor necessary for the algorithm.). The labeling ℓ functions as a link that connects feature similarity-based analysis and the feature neighborhood context-based analysis in our model and enables us to breakdown the original objective into two cyclic terms as follows.

For simplicity of explanation, assume that we know the true SOP labels for the entire feature set represented by $\tilde{\ell}$. Then the manifold learning objective can be specifically defined as one where the embedded features when clustered by some function $\mathcal{H}$ in a deterministic manner, produce the labeling $\tilde{\ell}$. This can be formulated as:

$$\tilde{\phi} = \arg\min(|\mathcal{H}(f^{\phi}) - \tilde{\ell}|) \tag{1}$$

If there exists an optimal labeling $\tilde{\ell}$ that can correctly separate semantic object parts in the original feature space, then the optimal embedding $\tilde{\phi}$ can be computed exactly such that features of the same SOPs are projected closer on the manifold. Conversely, if $\tilde{\phi}$ is known that can completely separate features of different semantic object parts on some manifold, then we can exactly compute a labeling $\tilde{\ell}$ such that the similar SOPs are assigned the same label:

$$\tilde{\ell} = \arg\min_{l}(|\Psi(\ell, f^{\tilde{\phi}}) - \Phi_{geo}(\ell, f)|) \tag{2}$$

here Ψ encodes the label-cluster similarity in the embedded feature space whereas Φ represents mislabeling penalties in the geodesic (feature neighborhood) space. In this fashion, we can transform the otherwise NP-hard problem into one of label optimization with two well-understood components strung together. The optimization for the system proceeds by first keeping the labeling constant and learning an improved embedding to reorganize features on the manifold. In the next step, this embedding is held constant while neighborhood-based refinement readjusts the feature labeling based on their connectivity statistics. Figure 4 demonstrates the effect of the optimization iterations on semantic correspondence performance.

2.1. Overview

Section 3 describes the manifold learning process and the construction of SOP models. For a set of target features and an initial labeling ℓ, an embedding is learned to minimize/maximize the intra and inter-class feature distances on the manifold. A clustering on the embedded features is performed and a model of semantic object parts (SOPs) is learned from the distribution of the feature clusters thus obtained. Subsequently, the likelihood of each feature point belonging to a SOP model is computed.

In section 4 we detail the feature neighborhood context analysis which takes as an input the labeling ℓ and feature SOP likelihood values from the above step and performs the neighborhood context-based label refinement. At this step spatial distances on the feature map are utilized to relabel frequently co-occurring features. This results in co-occurring features being projected closer on the manifold in the next manifold learning iteration. The above two steps are repeated until the SOP models remain unchanged.

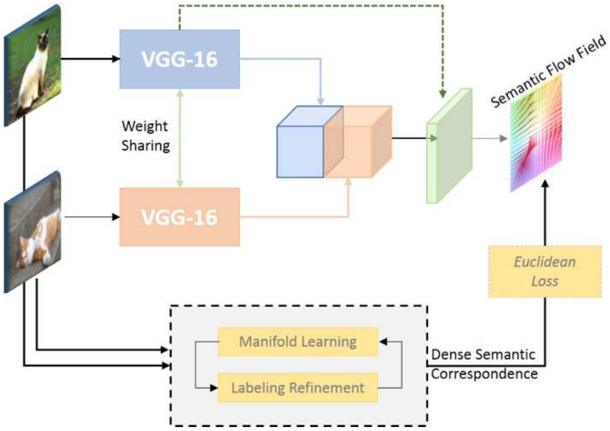

Figure 5. A schematic diagram depicting our Siamese DCNN and the interaction with the manifold learning module. The Euclidean loss computed against the embedded feature correspondences) is distributed to the network via backpropagation.

After the manifold has been refined, Euclidean distances between embedded features are used to compute dense correspondence to train a Siamese network. Motivation for and construction of a Siamese pixel predictor network along with its training from manifold correspondence maps as supervisory signal is explained in section 5. Finally, experiment details and a discussion of results is provided in section 6.

3. Learning Object Part Semantics

For a given category of a target dataset, a feature set is constructed by extracting the convolutional deep feature maps for all the images belonging to that category. We only concern ourselves with feature space similarity of feature points at this stage and not the neighborhood information. Accordingly, the feature maps' local neighborhood information is disregarded and all the features are aggregated into a single bag. Simple cluster analysis in this feature space is inadequate at bringing the SOP features together. This is due to the true distribution of each SOP cluster being highly anisotropic and non-linearly distributed in the current feature space. As discussed earlier, the reason for this is the classification loss function which emphasizes inter-class separation and does not account for intra-class object *part* separation.

In order to address this issue, we seek to find an expressive structure-preserving embedding that could effectively manage the anisotropic cluster shapes and be able to discriminate between the non-linear SOP distributions, on the manifold. Recall that in section 2 we introduced the SOP labeling ℓ for the entire feature set in order to transform the original objective. To accommodate this label for each feature point and to ensure that the embedded features belonging to the same SOP label fall close-by on the manifold, and farther away from the features of other models, we must use a supervised manifold learning method. For the manifold learning task, we employ Kernel Fisher discriminant (KFD) analysis [22] which is a supervised low-dimensional embedding technique. Since KFD uses the kernel trick to perform the linear operations of Fisher's linear discriminant in a reproducing kernel Hilbert space, it is able to account for the data non-linearity in the original deep feature space. In our case, the different SOP labels constitute the classes and class means are represented by computing between-class scatter matrix S_B^ϕ and the class variance by within-class scatter matrix S_W^ϕ for feature set f and labeling ℓ. The embedding is parameterized by vector w which is obtained by searching for a non-linear mapping Φ from the input space to some high-dimensional feature space $\mathcal{F}$, where the *Fisher criterion* is maximized:

$$J(w) = \frac{w^T S_B^\phi w}{w^T S_W^\phi w} \tag{3}$$

Due to the *implicit* feature space $\mathcal{F}$ being high-dimensional, direct computations in this space are avoided by rewriting the *Fisher criterion* in dot-products of the data only. The projection of new data point f^ϕ is computed as:

$$f^\phi = (w^T \cdot \Phi(f)) \tag{4}$$

For further details on KFD analysis and the closed form solution for the optimization in equation 3, the reader is referred to [22]. We normalize the features to unit L^2 norm and use the polynomial kernel function for the mapping. Since the convolutional deep features are very sparse in the original space due to the ReLU filtering, we use 32 discriminants for the final embedding which corresponds to the average number of non-zero activations.

Statistical SOP Modeling. Given the embedded features f^ϕ thus obtained, we perform hierarchical agglomerative clustering (HAClustering) with a fixed maximum cluster size of 15. Here each feature point starts out as an initial cluster and subsequently gets merged with other clusters during the construction of the hierarchy. The maximum cluster size here is a free parameter and corresponds to the number of expected semantic object parts.

Algorithm 1: Manifold Learning & Label Refinement

```
Input: Deep features $\{f\}_{c=1}^{C}$ for each object category
Output: Embedded features $\{f^\phi\}_{c=1}^{C}$, labeling $\{\tilde{\ell}\}_1^f$
Initialize $\ell$ = HAClustering($f$)
repeat
    Step1: Learn manifold & generate SOP clusters
        a) Compute embedding $\phi$
            $\tilde{\phi} \leftarrow \arg\max \left( \frac{w^T S_B^\phi w}{w^T S_W^\phi w} \right)$
        b) Project features
            $f^\phi = (w \cdot \phi(f^\phi))$
        c) Compute labeling via clustering
            $\tilde{\ell}$ = HAClustering $(f^\phi)$
        d) Model cluster distribution as Gaussian
            $p(f^\phi|c_i) = Ce^{-\frac{1}{2}(f^\phi-\mu_i)^T \sigma_i^{-1}(f^\phi-\mu_i)}$

    Step2: Refine labels wrt. geodesic distance
        a) Feature similarity-based likelihood term
            $\Psi(c_i|f_i^\phi) = -log(P(c_i|f_i^\phi))$
        b) Compute geodesic-based penalty term
            $\Phi(c_i, c_j \mid f_i^\phi, f_j^\phi) = \exp\left\{\frac{-\|f_i^\phi - f_j^\phi\|_{geo}}{\sigma^2}\right\}$
                if $c_i \neq c_j$
        c) Minimize energy to refine labels
            $\tilde{\ell} = \arg\min \sum_{f_i^\phi \in V} \Psi(c_i|f_i^\phi) +$
                $w \sum_{(f_i^\phi, f_j^\phi) \in E} \Phi(c_i, c_j|f_i^\phi, f_j^\phi)$
until SOP model unchanged
```

To represent the likelihood of a feature point belonging to an SOP, we need to statistically model the distribution of each cluster obtained earlier in the KFD feature space F. We collect the embedded feature values for each cluster c_i and their distributions are modeled by a simple multi-variate Gaussian thus producing SOP *models*:

$$p(f^\phi|c_i) = p(f^\phi, \mu_i, \sigma_i) = Ce^{-\frac{1}{2}(f^\phi-\mu_i)^T \sigma_i^{-1}(f^\phi-\mu_i)}, \tag{5}$$

Having modeled the SOP cluster distributions, we can now compute the likelihood of an embedded feature f^ϕ belonging to a SOP model c_i as:

$$p(c_i|f^\phi) \propto p(f^\phi|c_i)p(c_i), \tag{6}$$

where the prior $p(c_i)$ for a cluster is the membership fraction of all features belonging to that cluster. This likelihood term represents a confidence score of the feature's association with each of the semantic object parts model. We note here that to initialize the labeling ℓ for the first iteration, the above process is performed on the deep features albeit with an identity matrix as the embedding.

4. Local Context-based Model Refinement

As mentioned earlier, the spatially local patterns present in natural images extend to the deep convolutional feature

Table 1. Keypoint classification accuracies on the twenty categories of PASCAL 2011 from Berkeley PASCAL keypoint dataset. The parameters for SIFT-flow (radius) and Conv-flow (layer) are placed in the first column.

		aero	bike	bird	boat	bttl	bus	car	cat	chair	cow	table	dog	horse	mbike	prsn	plant	sheep	sofa	train	tv	mean
SIFT-flow[17]	20	37	50	39	35	74	67	47	40	36	43	68	38	42	48	33	70	44	52	68	77	50
(radius)	40	35	54	37	41	76	68	47	37	39	40	69	36	42	49	32	69	39	52	74	78	51
	80	33	43	37	42	75	66	42	30	**43**	36	70	31	36	51	27	70	35	49	69	77	48
Conv-flow[18]	4	44	53	49	42	78	70	45	55	41	48	68	51	51	**53**	41	76	49	52	73	76	56
(layer)	5	44	51	49	41	77	68	44	53	39	45	63	50	49	52	39	73	47	47	71	75	54
Ours		**52**	**59**	**57**	**45**	**80**	**74**	**58**	**68**	**43**	**49**	**72**	**59**	**56**	52	**44**	**78**	**52**	**53**	**77**	**81**	**61**

maps as well. Thus all the features belonging to an object part are highly likely to manifest in the same feature map locale regardless of their similarity in the feature space. For instance, a feature corresponding to the *eye* of a *bird* should not have a *tail* feature in its close proximity. Hence mining for feature co-occurrence statistics constrained to local feature neighborhoods across the target dataset serves as an excellent indicator of semantic similarity.

The convolutional deep features used in this work come from a DCNN trained on a classification loss function which is very different from our optimization objective. So the deep features which belong to very different parts of the same object oftentimes have high similarity in the deep feature space and thus a high likelihood of belonging to the same SOP, which is incorrect. Conversely, features that belong to the same object part and co-occur frequently exhibit very low feature-space similarity.

To account for neighborhood context, we need to ensure that features that frequently co-occur together and do not exhibit a high likelihood association with any one SOP cluster get assigned the same label. Whereas features that coincide but exhibit a strong association to their cluster retain their label and become be the edge features. To accomplish this the labels computed in section 3 for all feature are mapped back to the original feature map grid from where these features originated, thus obtaining a $2D$ label matrix for each of the images of the given category.

Given the label matrix of a feature map, we define a graph $G = \{V, E\}$, where the nodes V are given by the individual features of the feature map and are connected by an edge $\{f_i^\phi, f_j^\phi\} \in E$ if the features f_i^ϕ and f_j^ϕ are 8-connected on the feature map. We can define a likelihood energy term Ψ for each node (i.e. feature) in G directly following section 3 equation 6:

$$\Psi(c_i|f_i^\phi) = -log(p(c_i|f_i^\phi)) \tag{7}$$

This term explains the likelihood of feature f_i^ϕ belonging to the SOP cluster c_i, learned via the statistical model in section 3. It ensures that the labeling ℓ is coherent with the observed data such that the label c_i to feature f_i^ϕ is penalized if it is too different with the observed data *in the feature-space*.

A pairwise energy term Φ can be defined for each pair of features constituting an edge in G to ensure that any two neighboring feature labels f_i^ϕ and f_j^ϕ are penalized if their labels do not co-occur frequently in the dataset. It can be defined as the average *geodesic* distance between the features of c_i and c_j across all the label matrices:

$$\Phi(c_i, c_j \mid f_i^\phi, f_j^\phi) = \begin{cases} 0 & \text{if } c_i = c_j \\ \exp\left\{\frac{-\|f_i^\phi - f_j^\phi\|_{geo}}{\sigma^2}\right\} & \text{otherwise,} \end{cases} \tag{8}$$

Here the distance between the two feature points $\|\cdot\|_{geo}$ is defined as the geodesic distance between the median location of c_i feature points and c_j feature points for all the images. The penalty imposed by Φ ensures that the cost of switching labels between two features that are far on the learned manifold yet very close in the geodesic space (distance on feature map) is low. Thus frequently co-occurring features can be assigned the same label with a low cost.

An energy function can now be defined on the graph G that essentially learns a labeling ℓ that maximizes intra-SOP correlation in the embedded space while being regularized by the inter-SOP similarities in the geodesic space:

$$\mathcal{E}(\ell) = \sum_{f_i^\phi \in V} \Psi(c_i|f_i^\phi) + w \sum_{(f_i^\phi, f_j^\phi) \in E} \Phi(c_i, c_j|f_i^\phi, f_j^\phi) \tag{9}$$

The regularization parameter w balances the influence between the confidence in the manifold learning and the neighborhood context-based refinement. The energy function in equation 9 can be solved through the standard multi-label graph-cuts optimization [4]. The labeling ℓ obtained as the result of equation 9 assigns the same label to frequently co-occurring features while accounting for label outliers.

In the next iteration of the manifold learning process, the features corresponding to the refined labels are projected even closer on the manifold by the embedding and a new set of labels are computed via the HAClustering. This process is repeated until the SOP models remain unchanged.

Table 2. Keypoint prediction results on PASCAL VOC 2011 from Berkeley PASCAL 2011 keypoint dataset. The average accuracy of localization is listed for percentage of correct keypoint (PCK) criteria with α = 0:1, similar to [18]

	aero	bike	bird	boat	bttl	bus	car	cat	chair	cow	table	dog	horse	mbike	prsn	plant	sheep	sofa	train	tv	mean
SIFT-flow[17]	17.9	16.5	15.3	15.6	25.7	21.7	22.0	12.6	11.3	7.6	6.5	12.5	18.3	15.1	15.9	21.3	14.7	15.1	9.2	19.9	15.7
[17]+prior	33.5	36.9	22.7	23.1	44.0	42.6	39.3	22.1	18.5	23.5	11.2	20.6	32.2	33.9	26.7	30.6	25.7	26.5	21.9	32.4	28.4
CONV5[18]	38.5	37.6	29.6	25.3	54.5	52.1	28.6	31.5	8.9	30.5	24.1	23.7	35.8	29.9	39.3	38.2	30.5	24.5	41.5	42.0	33.3
[18]+prior	50.9	48.8	35.1	32.5	66.1	62.0	45.7	34.2	21.4	41.1	27.2	29.3	46.8	45.6	47.1	42.5	38.8	37.6	50.7	45.6	42.5
Ours	**62.2**	**58.6**	**51.1**	**39.3**	**74.7**	**77.2**	**65.1**	**48.0**	**26.9**	**53.4**	**31.2**	**44.5**	**59.1**	**54.2**	**58.8**	**51.3**	**43.9**	**41.7**	**57.5**	**62.1**	**53.1**

5. Training a Siamese Network

Due to the lack of true dense semantic correspondence ground-truth data we quantify the performance of the learned features on standard semantic keypoint prediction and classification datasets. For this purpose we need a model that is easily fine-tunable to these sister tasks. We simply train a Siamese DCNN from the manifold correspondence maps since this network can be fine-tuned to predict and classify key points at a later stage. Siamese DCNNs are the de facto standard for flow computations ([8][34][33]). They enable comparison and differentiation between two data streams in the deep feature space without the need for explicit modeling. Please refer to figure 5 (right) for a schematic diagram of our Siamese DCNN.

We reuse the DCNN from section 3 and duplicate it so as to parallel process two image streams. These two copies of the network share weights so that the feature extractors operating on the two images are identical. The last fully-connected layer ($FC9$) of our VGG-16 network is a fully convolutional layer in the current DCNN. The two networks are combined at this layer and the feature maps from the previous layer are simply stacked. Finally, to upsample the correspondences in the decoder, we use the convolution-transpose layers and concatenate it with corresponding feature maps of the encoder part.

To train the network, we utilize the truncated Euclidean loss similar to [33]. Let $\overline{F}_{i,j}$ be the semantic correspondence between images i and j as computed by our DCNN, and $\tilde{F}_{i,j}$ be the *pseudo* ground-truth semantic correspondence computed via nearest neighbor search on embedded feature similarities, then the truncated Euclidean loss between the two flows is given by :

$$\mathcal{L}_{flow}(\tilde{F}_{i,j}, \overline{F}_{i,j}) = \sum_{p \in I_{i,j}} min(\|\tilde{F}_{i,j}(p) - \overline{F}_{i,j}(p)\|^2, T^2), \tag{10}$$

The loss function is based on artificial ground-truth obtained by finding correlations in the space of embedded features obtained from section 3 and 4 and the value of T^2 is kept equal to the long edge of the image. This network directly computes dense correspondences and is trained end-to-end via back propagation.

6. Experimental Results

For our experiments we use deep convolutional features from the VGG16 classification network trained on the ImageNet 1000-class dataset [24]. We provide qualitative dense correspondence for images in the Pascal VOC 2012 dataset. The ImageNet dataset and the Pascal VOC dataset do not share any class labels, however, many classes are semantically similar across the datasets. For additional experimental results on the Taniai and proposal flow benchmarks, please visit *vision.ece.ucsb.edu/research/dense-semantic-object-correspondence*

In order to learn the SOP manifold, a per-class *training* image set is formed by selecting images containing a single object instance for each of the 20 classes from the Pascal VOC 2012 [10] dataset and deep features are extracted from the DCNN. We note here that during testing time, it is not required to know the class of any image pair to compute dense correspondence between them. As a preprocessing step, we process the features with a rectified linear unit (ReLU) non-linearity so that the feature maps contain positive activations only. Since Pascal VOC is a simpler dataset, it does not activate the full range of VGG16 neurons. This step leads to very sparse features that can be easily projected to a low-dimensional embedding without loss of its representative power.

Since our feature extractor DCNN (section 3) was trained for classification, high-magnitude feature activations belonging to the most discriminative part of the object class often end up overpowering other features. To correct this imbalance, we normalize features to unit $L2$ norm to match agreements between different neurons. This is also in accordance with the findings of [25], [23] who recommend doing this in order to stabilize the gradients for training. For clustering, we divide the embedded features for each object class into 15 clusters, using the same number as the expected semantic object parts.

After the manifold learning process concludes, we compute similarities between the embedded features for each pair of images in the per-class training image set. The dense correspondences thus obtained are treated as synthetic dataset for the training of the Siamese DCNN. In order to avoid over-fitting the network, a widely used strategy is to augment the training data via various geometric trans-

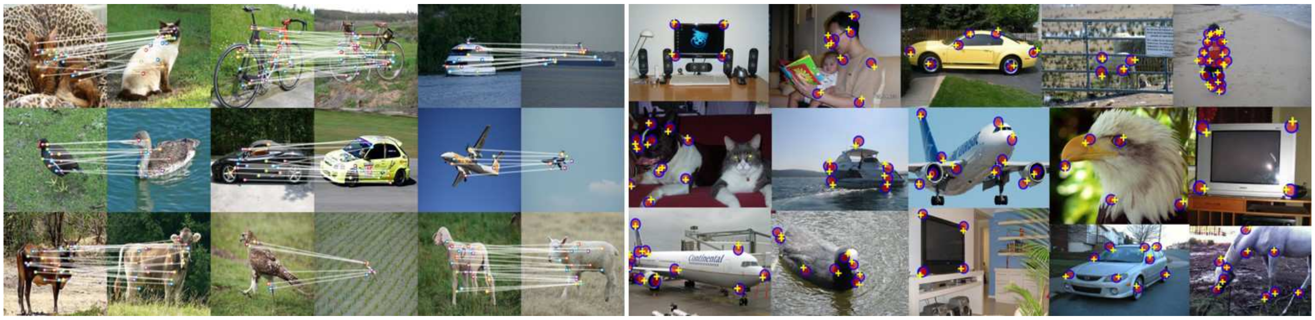

Figure 6. Left: Randomly sampled correspondence points obtained via matching embedded deep features across various environmental and scale variations for different PASCAL VOC 2012 categories. Right: Results of fine-tuning our DCNN for the task of keypoint localization results on the Berkeley PASCAL keypoint dataset for different object classes. Ground-truth points are represented by red circles whereas corresponding predictions made by our method are shown in yellow crosses.

formations [14][9]. During the training epochs, we randomly apply rotations sampled from $[-16°, 16°]$ and scaling from $[0.9, 2.0]$ to all the image pairs in the mini-batch, as well as their corresponding ground-truth data.

Qualitative results for the correspondence process are shown in figure 6. Our network is able to map semantic parts across a wide variety of foreground and background variations. The semantic correspondences are robust to viewpoint changes as well. We note that some features of an object (e.g. head for most animals) match more precisely and with a higher similarity score, as opposed to features belonging to other parts (e.g. breast or flank of a bird). We believe this to be the result of these dominant features being more discriminative for the original task of classification hence better recognized by our network. Due to the lack of dense semantic correspondence ground-truth data, we look towards the sister tasks of keypoint localization and classification to quantify the performance of our method. This problem is a natural application of correspondence problem since the keypoints are defined semantically on the classes of PASCAL VOC 2011 and thus require the classifier to have a semantic understanding of the dataset[2].

Similar to [18], we use 80% of the data for training and keep the rest for validation. The network fine-tuning and training data augmentation is performed in the manner described earlier. For a given query image, top K images are extracted from keypoint training dataset based on normalized L^2 distance between their deep features ($K = 25$). Next, dense correspondences are computed between the query image and each of these K images via our Siamese DCNN. Thus, a keypoint estimate on the query image is computed through mapping the known keypoints of the K images onto the query image and averaging the classification scores and the location coordinates. Mapping process is linear time in number of keypoints of the K images, which on average are less than a dozen per image.

The localization performance is measured via percentage of correct keypoints (PCK) metric originally used in [18][34]. Note that our results are not post-processed in any way. Figure 6 (right) shows example results from different categories. Our method is able to correctly predict the absence of keypoints in the query image as well. Each *cross* represents the ground-truth keypoints, whereas a circle represents the corresponding point found by our algorithm. Our method evidently performs better than conv-flow and outperforms SIFT-flow by a large margin (table 2, 1). Compared with other contemporary techniques, our method works well even for keypoints belonging to small object parts (e.g. eyes and nose). This is due to the ability of our network to obtain high-resolution correspondences via multiple upconvolutional layers, similar to FCN-8 [18].

7. Discussion

We presented a method to extract the implicit semantic object part knowledge present in existing image classification DCNNs by the means of utilizing feature neighborhood context. A manifold was learned where the embedded feature distances reflected the combination of geodesic space similarities and feature space similarities. The information contained in this manifold was imparted to a Siamese DCNN by training it to compute dense semantic flow field against a loss function based on the embedded feature similarities. This DCNN can now be adapted to any suitable computer vision task that can benefit from implicit understanding of SOPs. In order to demonstrate the efficacy of our method, we fine-tuned the Siamese DCNN for the task of semantic keypoint prediction and localization. For both these tasks, our network outperformed the classical SIFT-flow, which is the dominant method for semantic correspondences, thus validating our knowledge transfer process from pre-trained classification networks. Our method also outperformed the conv-flow method based on simple application of Conv features for semantic task, thus corroborating the efficacy of our SOP modeling framework.

Acknowledgements This work was supported by award #HD059217 from the National Institutes of Health.

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. European Conference on Computer Vision*, 2006.

[2] L. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *Proc. International Conference on Computer Vision*, 2011.

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. International Conference on Computer Vision*.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Trans. Pattern Analysis and Machine Intelligence*, 2001.

[5] J. Carreira, A. Kar, S. Tulsiani, and J. Malik. Virtual view networks for object reconstruction. In *Proc. Computer Vision and Pattern Recognition*, 2015.

[6] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, 2016.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. Computer Vision and Pattern Recognition*, 2005.

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. International Conference on Computer Vision*, 2015.

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010.

[11] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *Proc. Computer Vision and Pattern Recognition*, 2016.

[12] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 1981.

[13] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *Proc. International Conference on Computer Vision*, 2007.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 1998.

[16] B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. In *Toward category-level object recognition*. 2006.

[17] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Trans. Pattern Analysis and Machine Intelligence*, 2011.

[18] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, 2014.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[20] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, 1981.

[21] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proc. Computer Vision and Pattern Recognition*, 2015.

[22] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX*, 1999.

[23] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. Computer Vision and Pattern Recognition*, 2016.

[24] 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. *International Journal of Computer Vision)*, 2015.

[25] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. Computer Vision and Pattern Recognition*, 2015.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository*, 2014.

[27] T. Taniai, S. N. Sinha, and Y. Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *Proc. Computer Vision and Pattern Recognition*, 2016.

[28] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Trans. Pattern Analysis and Machine Intelligence*, 2010.

[29] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proc. Computer Vision and Pattern Recognition*, 2015.

[30] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[31] J. Yan, Z. Ren, H. Zha, and S. Chu. A constrained clustering based approach for matching a collection of feature sets. In *Proc. International Conference on Pattern Recognition*, 2016.

[32] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. European Conference on Computer Vision*, 2014.

[33] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proc. Computer Vision and Pattern Recognition*, 2016.

[34] T. Zhou, Y. J. Lee, X. Y. Stella, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proc. Computer Vision and Pattern Recognition*, 2015.

[35] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. Computer Vision and Pattern Recognition*, 2012.