Spatial-Visual Label Propagation for Local Feature Classification

Tarek El-Gaaly
Department of Computer Science
Rutgers University, USA
tgaaly@cs.rutgers.edu

Marwan Torki
Department of Computer and Systems Engineering
Alexandria University, Egypt
mtorki@alexu.edu.eg

Ahmed Elgammal
Department of Computer Science
Rutgers University, USA
elgammal@cs.rutgers.edu

Abstract—In this paper we present a novel approach to integrate feature similarity and spatial consistency of local features to achieve the goal of localizing an object of interest in an image. The goal is to achieve coherent and accurate labeling of feature points in a simple and effective way. We introduced our Spatial-Visual Label Propagation algorithm to infer the labels of local features in a test image from known labels. This is done in a transductive manner to provide spatial and feature smoothing over the learned labels. We show the value of our novel approach by a diverse set of experiments with successful improvements over previous methods and baseline classifiers.

I. Introduction

Object localization is a fundamental problem in computer vision. The detection and accurate localization of a given object under general settings with high class variation, different viewing conditions, presence of occlusion and clutter is a challenge. Local features descriptors, such as SIFT [1] and other similar descriptors, have been shown to be useful for object localization and recognition as they are highly discriminative and possess invariant properties. The spatial configuration of the local features is also important to decide the presence or absence of an object since it captures shape information which markedly reduces the rate of false positives. A good localization algorithm should find good object candidates with low false alarms.

Many researchers have addressed the localization problem by finding candidate patches that have high probability/score of lying on the object and at the same time rejecting patches that are likely to be false alarms [2], [3], [4], [5], [6], [7], [8], [9]. Most of these approaches use multiple cues and do not depend on local features alone. In [2] an aspect graph encodes the shape of the object and *shape masks* are learned to reduce the hypothesis space. In [4], [9] *segmentation* cues are augmented with local features to find accurate localization. In [5] a *hard matching* is established. Other approaches use different types of context cues [10].

Although it is reasonable to consider more cues beyond local feature descriptors and their locations to solve localization, it is also desirable to enhance localization without adding more cues. Enhancing the usage of local features is complementary to other state of the art achievements in localization. In this paper we only use local features defined by a feature descriptor and location in the image. We do not use any more cues.

Similar to our approach are [11], [12], [13] in which the local features are pruned heavily to find the good features to
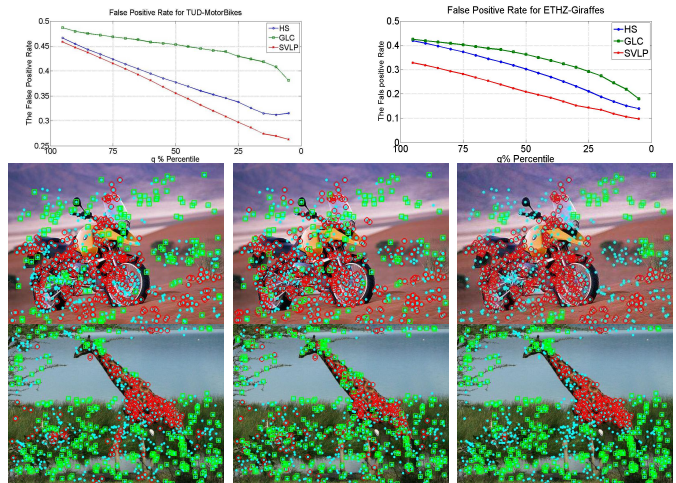

Fig. 1. The top row shows the false positive rate (FPR) comparison for 2 different datasets: TUD-MotorBikes and ETHZ-Giraffes. The bottom 2 rows show resulting feature classifications from each dataset. The images show the results of the 2 baselines: HS and GLS and then the top 80% localized features of our proposed SVLP method, respectively. Best viewed in color, with zooming.

be used in sophisticated localization algorithms. Our approach can be understood as a way of pruning local features so that only candidate features for the object class (*i.e.* foreground) and background class are considered for further higher level processing to accurately find the object of interest.

In this paper we pose the object localization problem as a transductive learning problem on a graph structure. *Graph-based methods* for both transductive and semi-supervised learning are widely used in many applications where the structure of unlabeled data can be combined with the structure of labeled data to learn better labeling [14]. A characteristic problem is that a feature may lie in more than one feature-space. For example, in the object localization problem using local features, local features can lie in two different spaces, namely, the feature descriptor space and the spatial x-y feature location on the image coordinates.

A successful approach of object class localization using local features must handle the feature descriptor and feature location spaces simultaneously. Under class variation (like many real objects) there might exist multiple manifold structures in the descriptor space. Simply, the manifold can be broken into several clusters where every cluster has its own manifold structure. This is what visual codebook methods try to capture.

The idea of exploiting the manifold structure in the feature descriptor and spatial domain was recently addressed in [15]. Unlike [15] where they explicitly embed the feature manifold and perform inductive learning in that embedded space, we exploit the manifold structure in the data implicitly without embedding and within a transductive learning paradigm.

The spatial arrangement of features is essential for recognition. Spatial neighborhoods gives us local geometry and collectively provides shape information about a given object. Spatial neighborhoods also inherently provide smoothness over labels since we expect to see the same labels in close proximity to each other. This is used in MRFs for segmentation [16] where the points are typically defined on a grid.

The contribution of this paper is that we pose the object class localization problem as classifying the features of a test image using transduction on a graph composed of the training features as well as the test features. Every training feature has a label and using transductive learning we can infer the labels of the test features. We propose a new technique to capture similarity among data points which share two structures: the spatial structure, which refers to the spatial arrangement of local features within an image, and visual structure, which refers to the feature similarities between local features in the whole data set. We call our approach Spatial-Visual Label Propagation (SVLP) and can be used to detect objects and their parts in images.

Figure 1 shows the feature localization that we produce given our proposed SVLP method. Our proposed method gives superior improvements over baselines in both the false positive rate and the visual quality of the resulting localization. This confirms that the careful handling of both visual similarity and spatial proximity can lead to very accurate feature localizations.

II. PROBLEM DEFINITION

We denote the ith feature in the kth image by $f_i^k = (x_i^k, v_i^k)$, where $x_i^k \in \mathbb{R}^2$ is the feature coordinate in the image and $v_i^k \in \mathbb{R}^{Desc}$ is the feature descriptor. The feature descriptor can be an image patch or local descriptor such as SIFT, Geometric Blur, etc.. The labeled training data consisting of K sets of feature points, $X^1, X^2, \cdots X^K$ in K images where $X^k = \{(f_i^k, y_i^k)\}$. Here $y_i^k \in \mathbb{R}^C$ denotes the class label and C is the number of classes (*e.g.* foreground/background or object parts as classes). For the binary case $C = 2$ and for the kth image we have $y_i^k = [1, 0]$ if the feature f_i^k belongs to the object class and $y_i^k = [0, 1]$ if otherwise.

During testing, an unlabeled test image is given with its associated set of features and corresponding unknown labels $\{(x_i, v_i, y_i)\}$. The goal is to label these features in the test image. Once the labels are discovered we can localize the object (or parts of the object) of interest by its local feature labels. The labeling should reflect what we learned from the training data about the features and their local spatial arrangement as well as coherent regions in the test image.

A fundamental assumption in label propagation is label consistency: points in close proximity to each other are likely to have similar labels. This is often called the *cluster assumption* [17]. The key difference in our problem is that the consistency or cluster assumption in our case has two folds: spatial consistency: close by features on the same image should have the same label, feature consistency: similar features across the different images should have the same label. In addition, in our case we wish to capture and learn the spatial structure during training and apply it to the test images, and thus preserving spatial consistency over similar structured clusters of local features. The question is how to construct a graph that reflects spatial and feature similarity and allows label propagation in a way that preserves both similarities. Simple concatenation of the feature descriptor and its location in the image cannot be considered since this will give rise to the issue of how to do deal with a test image without knowledge about the location(s) of object(s) of interest.

The SVLP approach captures the local spatial arrangement between the feature points by computing a local kernel based on the spatial arrangement of the local features in each image (intra-image). SVLP also captures the similarities between features in the descriptor space across different images (inter-image). Augmenting these two types of similarities in one graph is important in order to find a meaningful, accurate and coherent labeling. Finally SVLP aims at finding long range (global) relations between the features by propagating local information through diffusion between the spatial and visual appearance information. SVLP uses a method of label propagation as a transductive solution to induce the desired labeling of the feature points in the test image.

III. APPROACH

Readers with enough knowledge on label propagation topic will find our approach easy to follow and verify. We refer other readers to the well written chapter 11 in [14].

A. Constructing the weighted graph for SVLP

We first define W as a block matrix, shown in equation 1. Block W_{uu} is computed as a Gaussian kernel $K_x(.,.)$ on the spatial structure of the local features spatial arrangement in the test image. The blocks W_{ul} and W_{lu} are computed as Gaussian kernels $K_v(.,.)$ on the visual appearance structure of the local features between test and training images. The block W_{ll} should be designed to reflect both the intra-image spatial structure within each training image as well as the inter-image visual appearance structure between features in different training images. W_{ll} is defined as a block matrix where the blocks on the diagonal represent the spatial structure within each training image and the off-diagonal blocks represent the visual structure between different images in the training set.

Equation 1 shows an example W matrix that has K training images and one test image. W_k^S is the spatial structure for image k, where W_{ij}^V is the visual structure kernel between features in image i and features in image j

$$W = \begin{pmatrix} \begin{array}{c|c} W_{ll} = \begin{pmatrix} W_1^S & W_{12}^V & \cdots & W_{1K}^V \\ W_{21}^V & W_2^S & \cdots & W_{2K}^V \\ \vdots & \ddots & & \vdots \\ W_{K1}^V & \cdots & \cdots & W_K^S \end{pmatrix} & W_{lu}^V \\ \hline W_{ul}^V & W_{uu}^S \end{array} \end{pmatrix} \quad (1)$$

TABLE I.   DEFINITIONS.

-p and q are the image indices.
-i and j are the feature indices.
-D is the diagonal matrix $D = \sum_j W_{ij}$.
-$S = D^{-1/2} W D^{-1/2}$ is the normalized affinity.
-$\hat{Y}^{(0)}$ is the initial labeling vector of all features.
-$\hat{Y}(i)$ is the estimated label of feature i.
-μ is a positive parameter.
-$\alpha = 1/(\mu + 1)$.

B. Objective Function for SVLP

Our objective function consists of the sum of three terms, shown in equation 2. The first term is the smoothness constraint on the intra-image spatial structures, The second term is the smoothness constraint on the inter-image visual structures. The third term is the fitting constraint.

In our formulation the first two terms mean that nearby points defined by the graph structure (either spatially or visually) should not change their labels very often to allow the neighborhood structure to control the labeling process. We use the definitions in table I. We define the objective function as

$$\Psi(\hat{Y}) = \sum_p \sum_{i,j} W_p^S(i,j) \| \frac{\hat{Y}_p(i)}{\sqrt{D_{ii}}} - \frac{\hat{Y}_p(j)}{\sqrt{D_{jj}}} \|^2$$
$$+ \sum_{p,q,p \neq q} \sum_{i,j} W_{pq}^V(i,j) \| \frac{\hat{Y}_p(i)}{\sqrt{D_{ii}}} - \frac{\hat{Y}_q(j)}{\sqrt{D_{jj}}} \|^2$$
$$+ \mu \sum_i \| \hat{Y}(i) - Y(i) \|^2 \qquad (2)$$

Once W is constructed, equation 2 can be rewritten as

$$\Psi(\hat{Y}) = \sum_{i,j} W(i,j) \| \frac{\hat{Y}(i)}{\sqrt{D_{ii}}} - \frac{\hat{Y}(j)}{\sqrt{D_{jj}}} \|^2$$
$$+ \mu \sum_i \| \hat{Y}(i) - Y(i) \|^2 \qquad (3)$$

Equation 3 reduces directly to the same cost function as [17] and the minimization can be computed in closed form:

$$\hat{Y}^{(\infty)} = (1 - \alpha)(I - \alpha S)^{-1} \hat{Y}^{(0)} \qquad (4)$$

Iterative solution can be used to avoid inverse matrix computation,

$$\hat{Y}^{(t+1)} = \alpha S \hat{Y}^{(t)} + (1 - \alpha) \hat{Y}^{(0)} \qquad (5)$$

C. Algorithm

We summarize our algorithm in algorithm 1.

In our approach we used k-nearest neighbors with $k = 20$ to create a sparse graph to ease computational load. The labels $\hat{Y}^{(0)}$ can be initialized using a binary SVM or k-nearest neighbor classifier.

D. Analysis: Illustrative Example

Two-Image Example: We illustrate the interaction between labeled and unlabeled features on a simple example where the features in the first image are all labeled and the features in the second image are all unlabeled.

The SVLP Solution in equation 4 utilizes the full S matrix, this actually means the whole graph structure of S will be utilized to induce the labels of the unlabeled features in the second image. We

---

Algorithm 1: Algorithm for SVLP

Data: Given K training images with labeled local features.
Result: SVLP feature Localization in a test image

Training (Constructing W_{ll})
for $k = 1 : K$ **do**
　Construct the block W_k^S as $W_k^S(i,j) = exp(\|x_i^k - x_j^k\|^2/(2\sigma_x^2))$
end
for p *and* $q = 1 : K$, $p \neq q$ **do**
　Construct the block W_{pq}^V as $W_{pq}^V(i,j) = exp(\|v_i^p - v_j^q\|^2/(2\sigma_v^2))$
end

Testing (Construct full W and do the transduction) given a test image with unlabeled local features
-Construct the block W_{uu}^S as $W_{uu}^S(i,j) = exp(\|x_i^u - x_j^u\|^2/(2\sigma_x^2))$
-**for** $k = 1 : K$ **do**
　Construct the blocks W_{uk}^V as $W_{uk}^V(i,j) = exp(\|v_i^u - v_j^k\|^2/(2\sigma_v^2))$
end
-Construct $W_{ul}^V = \left[W_{u1}^V | W_{u2}^V | \cdots W_{uK}^V \right]$
-Construct $W_{lu}^V = (W_{ul}^V)^T$.
-Compute $S = D^{-1/2} W D^{-1/2}$.
-Iterate $\hat{Y}^{(t+1)} = \alpha S \hat{Y}^{(t)} + (1 - \alpha) \hat{Y}^{(0)}$ until convergence, where α is a parameter in the range $(0,1)$.
-Let $\hat{Y}^*$ denote the limit of the sequence $\{\hat{Y}^{(t)}\}$. Label each point $\hat{y}_i$ as a label $\hat{y}_i = arg \max_{j \leqslant c} \hat{Y}_{ij}^*$.

---

write down the expansion of equation 4 for the unlabeled features only as

$$\hat{Y}_u^{(\infty)} = \left(\alpha S_{(ul)}^1 + \alpha^2 S_{(ul)}^2 + \cdots \right) Y_l$$
$$+ \left(I_u + \alpha S_{(uu)}^1 + \alpha^2 S_{(uu)}^2 + \cdots \right) Y_u \qquad (6)$$

$\hat{Y}_u^{(\infty)}$ gets its labels from two terms. The first term depends on the labels of the ground truth labels Y_l and it also depends on $S_{(ul)}^p$. The terms $S_{(ul)}^p$ are the normalized similarities between labeled (training) and unlabeled (testing) features. The superscript p represents the order of the block matrices S which can be replaced by a summation of components consisting of S_{uu}, S_{ul} and S_{ll} which will be shown in equation 7. The second term in equation 6 depends on the unknown labels Y_u (which can be given some initial values using some external classifier, it also can be initialized as zeros) and it also depends on $S_{(uu)}^p$. The terms $S_{(uu)}^p$ are the normalized similarities between the unlabeled (testing) features.

The first order blocks $S_{(uu)}^1$ and $S_{(ul)}^1$ do not encode the spatial structure of the training image S_{ll}. On the other hand, the higher order blocks $S_{(uu)}^p$ and $S_{(ul)}^p$ do encode the spatial structure of the training image S_{ll}. This can be noticed if we further expand the terms $S_{(ul)}^2, S_{(uu)}^2, S_{(ul)}^3$ and $S_{(uu)}^3$ in terms of the original S blocks

$$S_{(ul)}^2 = S_{ul} S_{ll} + S_{uu} S_{ul}$$
$$S_{(uu)}^2 = S_{ul} S_{lu} + S_{uu} S_{uu}$$
$$S_{(ul)}^3 = S_{ul} S_{ll} S_{ll} + S_{ul} S_{lu} S_{ul} + S_{uu} S_{ul} S_{ll} + S_{uu} S_{uu} S_{ul}$$
$$S_{(uu)}^3 = S_{ul} S_{ll} S_{lu} + S_{ul} S_{lu} S_{uu} + S_{uu} S_{ul} S_{lu} + S_{uu} S_{uu} S_{uu} \quad (7)$$

The higher orders blocks $(S_{(uu)}^p, S_{(ul)}^p)$ already have the term S_{ll}. This shows that the unknown labels $\hat{Y}_u^{(\infty)}$ are not only affected by the similarity across the labeled and unlabeled data points, but in fact it is affected also by the similarity in the training points. In other words the spatial structure of the training points is reflected on the propagated labels.

Conclusion: The two-image example above leads us to a number of conclusions. First, the diffusion kernel $(I - \alpha S)^{-1}$ that is used in the SVLP solution is capturing the long-term relationships (*i.e.* between pre-convergence and post-convergence labels) in the **whole graph** constructed from the two sets of feature points, labeled and

unlabeled (coming from the single training image and single testing image).

Second, although it seems less intuitive to change the labels of the training set, we find that it is fundamental to change the labels in the training features so that we can benefit from the spatial structure in the training image. We understand that changing the labels for labeled data is sound when the labeled data has some overlap between the classes. In our addressed problem of object class localization from local features this is also sound, because the features that are close to the boundary of an object will have much confusion between its original label and the labels of surrounding features. This will lead to features that may change their label depending on their neighborhood structure.

Third, the two images example gives us an intuition of how to design the terms in the weight matrix W when we construct the graph, this will be reflected on the normalized weight matrix (S). We see that we need to define some spatial structure for the features from each image in the training set. We see that we need to define some structure that represents the visual appearance similarity between the image in the training set and the image in the test set. In our problem where the local features are defined by two different vectors (descriptor and spatial location), it is easy to see that the spatial structure can be utilized to assure coherent labeling in the spatial space. Also the visual structure can be inferred from the feature descriptor similarity in the descriptor space so that the features that have high similarity in descriptor space can be labeled similarly.

IV. EXPERIMENTS

We use Geometric Blur (GB) [18] and SIFT [1] as the local feature descriptors. Towards the end of this section we briefly compare between these two descriptors. The datasets used in the experiments are: Caltech-101 [19], TUD Motorbikes and Cows [20], ETHZ Shape Classes-Giraffes [11], GRAZ02-Bikes [21]. We set the α for our SVLP to .95 when we use SIFT descriptors and .9 when we use GB. For all experiments we used a weighted 20-nearest neighbors in the graph construction. We always use the output of SVM[1] classifier on feature descriptor to initialize the unknown labels.

A. Caltech-101

We performed object class localization on all the classes in Caltech-101 [19], each class separately. We carried out the localization for all the 101 classes to show that we can apply our method for object class localization *across very different kinds of objects* ranging from animals, man-made, indoor objects, etc..

Every training image has at most 300 local features (the number of local features actually vary significantly per class). These local features are described by GB descriptors and their spatial location in the images. The detected local features within the ground-truth contours are labeled as object class and the features outside the contours are marked as background class. We ran our algorithm 5 times on all classes for each of three different training settings (sizes: 10, 20 and 30). By using our SVLP method the labels of the test image feature points are inferred. This leads to localization of the object of interest. Similar to many other researchers in object class localization from local features [12], [13], [11], we report the q percentile of features that scored the highest in the object class or background class.

We applied SVLP on different number of training samples per class and fed the SVM estimated solution to our algorithm as an initial Y_u. The q percentile SVLP significantly improves over binary classification baselines SVM and KNN, even with a very large portion

---

[1]We use libsvm3.1 package with default parameters(radial basis function kernel) with **C** parameter=1

	F	M	A	W	K
SVLP	**0.0492**	**0.0141**	**0.0833**	**0.0270**	**0.0606**
SVM	0.2359	0.0487	0.2030	0.0902	0.1372
1-NN	0.3229	0.1667	0.2721	0.0732	0.2197
[12]	.30	0.11	0.21	.08	.19
[13]	.15	0.07	0.177	.03	0.08

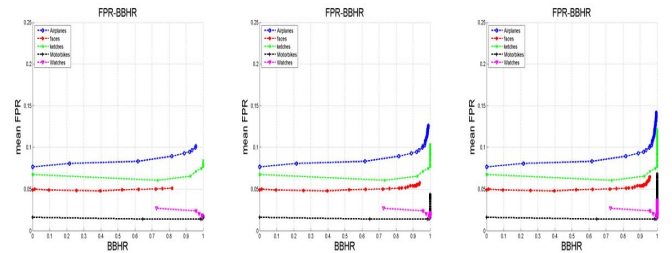TABLE II. FPR FOR DIFFERENT METHODS AT $q = 50\%$. F: FACES, M: MOTORBIKES, A: AIRPLANES, W: WATCHES AND K: KETCH.

Fig. 3. FPR at bounding-box hit rate (BBHR) =0.5 for the Caltech subsets. We varied the q percentile value as follows, left: q=0.01 to 0.10, middle: q=0.01 to 0.30, right: q=0.01 to 0.50. These plots can be compared to [12] and [13]

of features included in the accuracy measure, i.e. 80%. We note here that most other localization approaches use only the best 20% of the local features in measuring the accuracy. This improvement is very meaningful as the SVLP is always finding a spatially coherent feature labeling. As q decreases the localized features on the object of interest become more confidently localized features.

For comparative evaluations in table II we mainly consider the approaches [12], [13]. The reasons behind this selection are the following. Firstly, similar to [12], [13], our goal is to localize features into foreground/background classes. Due to this, we use the same evaluation measure (FPR) as [12], [13]. Using FPR is a more sensible choice over bounding box overlap ratio when evaluating sparse local feature localization. Secondly, the localization in [12], [13] is performed after clustering the images with very high accuracy (around 98%). These approaches localize the features that belong to the object in every individual cluster independently and hence the object is known to be in the image with high probability (around .98). In other words the unsupervised part (i.e. clustering) of their approaches does not increase the hardship of their feature ranking problem. Lastly, we only use $10 - 30$ training images which is markedly less than the 100 training images per class used in [12], [13]. The much larger number of training images they select balances the unsupervised ranking they perform on their features.

B. Generalization to Subsets of LabelMe

Caltech-101 is designed for single object categorization tasks. To evaluate the generalization of our proposed approach on different datasets which might have different distributions, we used training example from Caltech-101 and tested on images from the LabelMe datasets [22] with multiple object instances. We used subsets of LabelMe datasets that have been used by [13]. In this experiment we trained from four Caltech 101 classes namely {Motorbikes, Cars-rear, Faces, Airplanes}. Since the object scales are very different in Caltech-101 and LabelMe, we varied the scale on the test images. We show some results of the localized features in Figure 4. We generate bounding boxes using very simple heuristic by looking for bounding boxes of different sizes that maximize the normalized difference between detected foreground and background features in the neighborhood of the localized positive features. The average bounding box overlap ratio for the four classes is {$Faces : 0.432, CarsRear : 0.44, MotorBikes : 0.404, Airplanes : 0.29$}, we can see that our results are close to [13], we are better in MotorBikes and CarsRear, while [13] method is better for faces and airplanes.
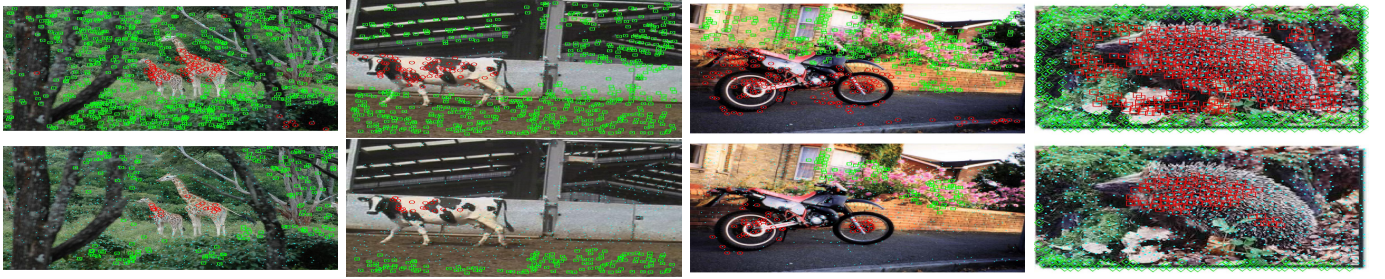
Fig. 2. Sample Results on ETHZ-Giraffes, TUD-Cows, TUD-Motorbikes and Caltech-101. The row shows the top 80% percentile of the features,and the bottom row shows the top 20%. Red features are foreground localized features, green are background localized features and detected features are shown in cyan. Best viewed in color with zooming.

Fig. 4. Generalization to subsets of LableMe dataset. Features with top 25% confidence are shown. Red for foreground localized features. Green for background localized features. Detected features shown in cyan. Best viewed in color with zooming.

C. TUD / ETHZ Datasets

We experimented on three other datasets to analyze the performance of our approach compared with an SVM and 1-NN baselines. The first dataset is TUD-Motorbikes which is part of the PASCAL collection [23] which is known to contain challenging images because of the fact that they have different resolutions, scales, background, heavy clutter and multiple instances per image. The second dataset is TUD-Cows which contains varying skin textures on the body of cows in the images. The third dataset is ETHZ-Giraffes which contains images of giraffes under different deformation conditions (i.e. the giraffes' necks vary in shape from fully extended to leaning downwards). The images in this dataset are also challenging as they exist in multiple scales, resolution, multiple instances per image and contain extensive clutter mostly in the form of vegetation.

For TUD-Cows and ETHZ-Giraffes we set the number of training images to 20. For TUD-Motorbikes we used 30 training images. The number of training images are approximately $21 - 26\%$ of the size of the respective datasets. The much larger portion of the dataset can then be used for testing. In all three datasets we used 300 SIFT descriptors for training. For testing, we sampled more features; up to 900 features per image. The reason why we used the SIFT descriptor in these datasets is because GB failed on images containing vegetation in the form of bushes, trees, grass, etc.. The reason behind this is that GB is not multi-scale and due to the large variance in the local structures of vegetation it is not able to generalize over the background class. SIFT on the other hand captures multiple scales of the local structures in the images and hence is able to discriminate between object and background classes with higher accuracy. The features were labeled and evaluated by using the ground truth bounding boxes for ETHZ-Giraffes and TUD-MotorBikes. The ground truth for TUD-Cows is in the form of masks.

In addition to the binary classifiers baselines *i.e.* SVM and 1-NN, we use two additional baselines. We use the original Global and Local Consistency (GLC) [17] algorithm in which there is no spatial proximity encoded. The second baseline is the Harmonic function solution (HS) [24], *but* we use the spatial proximity in the test image to form the unlabeled-to-unlabeled weights in the HS algorithm. The
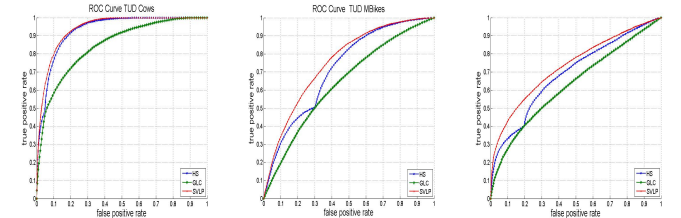

Fig. 5. The ROC curves for ETHZ-TUD subsets. The curves are generated by using combining all features from all test images in the subsets. Left: TUD-Cows. Middle: TUD Mbikes. Right:ETHZ-Giraffes. Clearly SVLP outperforms other label propagation baselines GLC and HS

reason behind comparing against these baselines is to show that the GLC without spatial proximity encoded would not work well in practice because the spatial proximity plays an important role in describing the spatial structure (*i.e.* shape) of a given object. In addition the HS with spatial proximity in the test image *only* will not benefit from the spatial relationships between features in the training sets. The comparison to these baselines in figure 5 favors our SVLP approach which accomplishes coherent labeling of the local features based on both spatial relationships in train and test image as well as the visual similarity of the features. We notice that close ROC curves for HS and SVLP for TUD-Cows in which we use the contours around the object as ground truth for training. In the remaining two classes we use bounding boxes for training, as we can see there is clear advantage using SVLP over HS or GLC which confirms our analysis in III-D.

We also show in figure 1 the false positive rate (FPR) of our SVLP in comparison to GLC and HS baselines. We show different accuracies and false positive rates based on the percentile of scoring features (i.e. q=80% means only the top 80% of the scoring features are the output of the alogrithm). Figure 2 shows sample results of feature localization on different datasets.

We show in figure 1 the false positive rate of our SVLP in comparison to GLC and HS baselines. We show different accuracies and false positive rates based on the percentile of scoring features (i.e. q=80% means only the top 80% of the scoring features are the output of the alogrithm). Figure 2 shows sample results of feature localization on different datasets.

D. Object Parts Localization

For qualitative evaluation of our approach on part localization we carried out object part localization for some classes: {Caltech-Motorbikes, TUD-cows [25]}. The parts of the objects are manually annotated via bounding boxes in the train images. We used TUD-cows to test how our part localization works in the case of non-rigid objects with articulation.

We used 20 images for training, each has 300 GB features. As shown in figure 6 we defined three parts motorbike using bound boxes by gathering the front wheel and some part of the attached handle,

	Accuracy				FPR				Recall			
	SVM	KNN	$q = 80\%$	$q = 20\%$	SVM	KNN	$q = 80\%$	$q = 20\%$	SVM	KNN	$q = 80\%$	$q = 20\%$
ETHZ-G	.5980	.5878	.6822	.7639	.4036	.3972	.2931	.1341	.6079	.5988	.6007	.6950
TUD-C	.8550	.8259	.9339	.9933	.2217	.3119	.1357	.0536	.8781	.8298	.7102	.9980
TUD-MB	.5776	.5655	.6703	.7601	.4829	.4914	.3763	.2835	.6463	.6127	.5858	.6627

TABLE III.   COMPARISON OF PRESENTED APPROACHES USING DIFFERENT PERCENTILES q AS WELL AS TWO BASELINE CLASSIFIERS: SVM AND 1-NEAREST NEIGHBORS
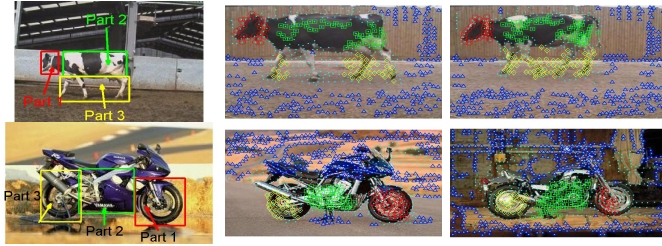

Fig. 6.   Object part localization. Left: bboxes defining the parts used during training. Mid. and Right: some part localization results on TUD-cows and Caltech-Mbikes. Features with top 60% confidence are labeled. **R** for part 1 localized features. **G** for part 2 localized features. **Y** for part 3 localized features. **B** for background localized features. Detected features shown in cyan. Better Viewed in color and zooming.

the second part is the engine area and the third part is the rear wheel and some part of the seat. We defined three parts on the Cow object using bound boxes as the head, body and legs. In both cases the remaining features are considered as back ground class. Notice that in the motorbike example, the front and back wheels have similar appearance and in the cow example the head and body have similar texture. Successfully localizing the parts in these example shows that the approach is in fact learning about the feature spatial arrangement. We can see (Figure 6) that the part labels are retrieved efficiently, here we use the top 60% percentile to show the localized features for each class.

V.   CONCLUSION

We have presented a novel approach for object class localization using local features alone. We use the training labels of feature points in relatively small training sets to propagate their labels to the features in a test image. Towards this end we defined our SVLP algorithm which utilizes the spatial structure of the local features within an image (train or test) and the visual appearance structure between pairs of images (train or test). We defined an objective function that is suitable for the localization problem with a closed form solution.

Several advantages result from our novel approach. First, finding bounding boxes can be done over the localized features using simple heuristics as we did for the generalization task on LabelMe subsets IV-B. Second, classification of the local features into positive and negative samples can be helpful in many settings like feeding the initial seeds of an interactive segmentation algorithm such lazy snapping [26], but without user intervention. Third, there is no need to perform hard matching for the object localization task since state of the art methods use graph matching to compute hard matching. Graph matching is usually modeled by many researchers as a quadratic assignment problem [27], [18] and every training image should be matched separately to the query image. This adds too much overhead.

We experimented on all Caltech-101 classes, TUD (Cows & Motorbikes classes) and ETHZ-Giraffes. We show clear improvements over binary baseline classifiers. SVLP outperforms the standard label propagation method GLC where the spatial consistency is not encoded. It also outperformed a modified HS where we use the spatial proximity in test image. We also reported improvement over recent works on object class localization from local features [12], [13] on subsets of Caltech-101 dataset. We have reported qualitative results on the generalization of our learned classifiers on other datasets where the objects appear at different scales, resolutions, with multiple instances and in severely cluttered backgrounds and we contrasted that to the results in [13].

REFERENCES

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

[2] M. Marszalek and C. Schmid, "Accurate object localization with shape masks," *CVPR*, 2007.

[3] M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in *In ECCV*, 2008.

[4] C. Pantofaru, G. Dorko, C. Schmid, and M. Hebert, "Combining regions and patches for object class localization," in *CVPR Workshops*, 2006.

[5] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple object class detection with a generative model," in *In CVPR*, 2006, pp. 26–36.

[6] B. Leibe, K. Mikolajczyk, and B. Schiele, "Segmentation based multi-cue integration for object detection," in *BMVC*, 2006.

[7] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *ICCV*, 2009.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *In CVPR*, 2005, pp. 886–893.

[9] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *IJCV*, 2008.

[10] C. Galleguillos, B. McFee, S. Belongie, and G. Lanckriet, "Multi-class object localization by combining local contextual interactions," in *CVPR10*, 2010.

[11] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool, "Efficient mining of frequent and distinctive feature configurations," *ICCV*, 2007.

[12] G. Kim, C. Faloutsos, and M. Hebert, "Unsupervised modeling of object categories using link analysis techniques," in *CVPR*, 2008.

[13] Y. J. Lee and K. Grauman, "Shape discovery from unlabeled image collections," *CVPR*, 2009.

[14] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*.   Cambridge, MA: MIT Press, 2006. [Online]. Available: http://www.kyb.tuebingen.mpg.de/ssl-book

[15] M. Torki and A. Elgammal, "Putting local features on a manifold," in *CVPR*, 2010.

[16] F. Wang, X. Wang, and T. Li, "Efficient label propagation for interactive image segmentation," *ICMLA*, 2007.

[17] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf, "Learning with local and global consistency," in *NIPS*, 2004.

[18] A. C. Berg, "Shape matching and object recognition," Ph.D. dissertation, University of California, Berkeley, 2005.

[19] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVIU*, vol. 106, no. 1, pp. 59–70, April 2007.

[20] B. C. M. Fritz, B. Leibe and B. Schiele, "Integrating representative and discriminant models for object category detection," in *ICCV*, 2005.

[21] A. Opelt and A. Pinz, "Object localization with boosting and weak supervision for generic object recognition," in *Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA)*, 2005.

[22] LabelMe, "The open annotation tool," http://labelme.csail.mit.edu/.

[23] PASCAL, "The pascal object recognition database collection," http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html.

[24] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.

[25] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *ECCV*, 2004.

[26] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, 2004.

[27] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *ECCV*, 2008.