Weakly Supervised Object Recognition and Localization with Invariant High Order Features

Yimeng Zhang yz457@cornell.edu Tsuhan Chen tsuhan@ece.cornell.edu

School of Electronic and Computer Enginerring Cornell University Ithaca, NY, USA

Abstract

High order features have been proposed to incorporate geometrical information into the "bag of feature" representation. We propose algorithms to perform fast weakly supervised object categorization and localization with high order features. To this end, we first use Hough transform method to identify translation and scale invariant high order features co-occurring in two images. The co-occurrence is used to calculate a kernel for a SVM. Then, we propose an efficient algorithm for localization with high order features. A naive way would be to apply the SVM for all possible subwindows, which requires O(SM) kernel computations per image, where S is the number of support vectors, and M is the number of possible subwindows in an image. The algorithm collects the weights of high order features for the subwindows while calculating kernel value for the entire image, and thus reduces the kernel computations to O(S).

1 Introduction

Object categorization and localization with weakly labeled data have attracted increasing interest in computer vision. During training, we are only provided with the image level categories. This greatly reduces the human labeling work, and sometimes may even work better than supervised models [1]. One of the most important issues to address for object recognition is the great intra-category variability, such as the positions, scales, or poses. Especially for weakly supervised learning, in which the locations or scales of the objects are unknown during training. The model learning must be invariant to variability. In addition, the models must be discriminative enough to separate different categories.

The "bag of features" (BoF) representation of images [5, 24, 25] has been extremely popular for object categorization task due to its advantages of great invariance and computational simplicity. However, the representation discards any geometrical or shape information of objects, and thus its discriminative power is limited. Many works [5, 2, 15] have been done to model the spatial information by the locations of the local features or parts as opposed to the object center. These works have gain great success in supervised object detection, where the bounding boxes of objects are given. Constellation model [8] represents the spatial distribution as mutual relationship of the local parts, and therefore can be learned

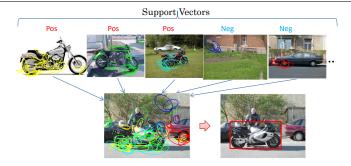


Figure 1: Illustration of the weakly supervised localization algorithm with invariant high order features. The input image is evaluated on the motorcycle classifier (SVM) learned with weakly labeled data. The images in the top line represent the support vectors with positive or negative weights. Each ellipse corresponds to a local feature. We show the largest order feature co-occurring in the input image and each support vector in different colors for different support vectors.

with weakly labeled data. However, the computation cost is usually large. Exemplar model [1] models the shapes variantly with Spatial Pyramid Kernel [12]. To learn from weakly labeled data, it iteratively processes the training images to find the optimal location. Multiple instance learning [12][23] has been successfully used for weakly supervised learning. It models images as bags of instances, which are usually segments or sub-windows. These works usually require iterative processing on the training images to find the positive instance in a positive bag.

Another extension for improving the BoF representation is to model the mutual geometrical relationship between the local features [\square , \square , \square]. High order features (HOFs) are created to represent a specific number of local features in a particular spatial relationship. According to the number of local features used, the HOFs are called doublet, triplet, or n^{th} order features. This type of method can be learned invariantly with weakly labeled data. However, as the order increases, the number of features increases exponentially to the order. In $[\square]$, the authors proposed an efficient algorithm for calculating the kernels with translation invariant high order features. The algorithm calculates the kernels for all order features efficiently in time linear to the number of local features. The kernel calculated is used with SVMs for object categorization.

In this paper, we extend the HOF idea for object localization. We learn a SVM with the HOF kernel on the entire images with the image labels. Since the classifier are learned invariantly, we can backproject the learned weights of the HOFs for localization. For most detection algorithms that include a SVM with non-linear kernels, the computation can be expensive, since all possible subwindows must be evaluated. There have been works [\square][\square] that use branch and bound algorithm to avoid evaluating every subwindow. We propose an efficient algorithm for localization with HOFs. The algorithm performs the kernel calculations for the entire image, and obtain the decision scores for the local patches at the same time. The decision scores for all possible subwindows can be calculated efficiently from the scores of the local patches. Thus, it reduces the number of times required for kernel computation from O(SM) to O(S) per image, where S is the number of support vectors, and M is the number of possible subwindows. The algorithm with HOFs also provides a novel way for the idea of classifying local patches using its context or spatial configuration. The idea

has been explored by many other works with various methods [2], [2].

Another contribution of the paper is that we propose an algorithm which can easily add more invariance to the HOF kernel [23]. The algorithm can be viewed as a generalized version of the algorithm in [23]. Our extension is based on the observation that the process for identifying the co-occurring translation invariant high order features of two images in [23] is analogous to the process of the generalized Hough Transform [13]. The Hough transform allows a pair of features from two images to vote for their transformation parameters. Thus we can easily define any transformation invariance we want to introduce, such as translation, scale, or rotation. Note that in [23], the authors have also given a suggested algorithm for dealing with scale invariance (without evaluation in experiments). This paper provides a more general and formulated algorithm for dealing with invariance for HOFs.

Fig. 1 illustrates the idea of weakly supervised localization with HOFs. A test image is evaluated by calculating the kernel values with each support vector of the SVM learned during training. Because of the invariance of the HOFs, objects that occur in different positions and at different scales in the training images can contribute properly to the scores for localization, although the training images are not aligned, and the scale is not normalized.

2 Learning

We describe the invariant high order feature kernel for recognition in this section. The algorithm can be viewed as a generalized version of [22] by adding more invariance to the high order features. To simplify the explanation, we only consider the transformation of scale and translation. But the algorithm can accommodate any linear transformations, such as rotation or affine.

An image I is represented as a collection of local patches, $I = \{p_1, ..., p_m\}$. Each patch is represented with its visual word assignment w, region size s and location x, y. $p_i = (w_i, s_i, x_i, y_i)$.

2.1 Invariant High Order Features

An n^{th} order feature f^n is composed of n visual words in a particular mutual scale and spatial relationship. If an n^{th} order feature can be created as a translation or/and scale transformation of the n words of the other n^{th} order feature, the two high order features are defined as the same feature. Fig. 2 gives an example of two occurrences of a 3rd order feature on two images with position and scale changes.

Specifically, the occurrence of a particular n^{th} order feature f^n in an image I is n patches $P^n = (p_1, p_2, ..., p_n)$ from the image. Let $(\hat{x}, \hat{y}, \hat{s})$ denote a transformation of the points, with (\hat{x}, \hat{y}) denoting translation, and \hat{s} denoting scale. Let $P^{n'}$ denote another n patches. We say P^n and $P^{n'}$ are the occurrences of the same features if and only if for $\forall i$, they satisfy the following conditions: 1) the same word assignment: $w_i = w_i'$; 2) The same transformation: $x_i = x_i' \times \hat{s} + \hat{x}$, $y_i = \hat{s}y_i' + \hat{y}$, $s_i = \hat{s}s_i'$.

The length of the feature vector is exponential to the order n. The n^{th} order feature space is $V^nX^{n-1}Y^{n-1}S^{n-1}$, where V is the vocabulary size of visual words, X and Y define the image space, and S represents the scales.

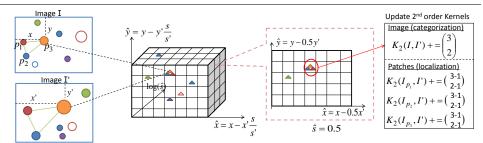


Figure 2: Illustration of the algorithm for finding co-occurrences of invariant high order features. Each circle in the images is a local patch. The different colors represents different visual word assignments. The size of the circle represents the scale of the patch. A triangle corresponds to a vote generated by a pair of patches.

2.2 Co-occurrence with the Hough Transform

We used the idea of the Generalized Hough Transform [III] for identifying the co-occurring high order features of two images. As illustrated in Fig. 2, For each pair of patches (p_i, p_i') with the same visual word assignments $(w_i = w_i')$ from the two images I and I', we make a vote on the transformation parameter space $(\hat{x}, \hat{y}, \hat{s})$ with the following values: $(x_i - \frac{s_i}{s_i'} x_i', y_i - \frac{s_i}{s_i'} y_i', \log(\frac{s_i}{s_i'}))$. If we have n votes at a particular point on the parameter space, we have n pair of patches with the same transformation $(\hat{x}, \hat{y}, \hat{s})$. From the definition for high order features, we have a co-occurrence of a certain n^{th} order feature. In addition, it also means that we have n choose k, $\binom{n}{k}$ number of k^{th} order features for any k < n. To allow for some deformation, we quantize the parameter space to cells. The transformations $(\hat{x}, \hat{y}, \hat{s})$ in the same cell will be treated as the same transformations. In the example of Fig. 2, at scale $\hat{s} = 1/2$, we have a co-occurring 3rd order feature, and $\binom{3}{2} = 3$ co-occurrences of 2nd order features. Thus, to compute the number of co-occurring n^{th} order features of two images, we can simply count from the parameter space.

2.3 High Order Feature Kernel

The n^{th} order feature vector $\phi^n(I)$ of image I is defined as a vector cascading the feature values of all n^{th} order features defined in section 2.1. The feature value of a feature f_i^n is defined as the number of occurrences of f_i^n , which fills in the i^{th} coordinate of $\phi^n(I)$. The kernel K_n calculates the inner product of the n^{th} order feature vectors of two images I and I'. In [23], the authors proved that the kernel value of two images is equal to the number of the co-occurrences of all n^{th} order features. We use the algorithm we proposed in section 2.2 to count the co-occurrences of all nth order features efficiently. Specifically, to calculate the n^{th} order kernel K_n , we first performing the Hough transform of two images. Then for each bin on the parameter space with N votes $(N \ge n)$, we increment the kernel value K_n with N choose n, $\binom{N}{n}$. For normalization, we use L1-norm.

$$\hat{K}_n(I,I') = K_n(I,I') / {\binom{|I|}{n} \binom{|I'|}{n}}$$

$$\tag{1}$$

where |I| and |I'| are the number of patches in images I and I'. In practise, a combined kernel K_n^* which combines the kernels from 1 to n can be used: $K_n^*(I,I') = \sum_{i=1}^n \mu^{1-i} \hat{K}_n(I,I')$, where

 μ is a constant value in (0,1).

Algorithm 1: Compute kernels $K_n(I,I_t)$ **and** $K_n(I_{p_t}^n,I_t)$

Input: Two images $I = p_1, ..., p_{|I|}$ and I_t

Output: $K_n(I,I_t)$ and $K_n(I,I_t)$ for i = 1,...,|I|

Algorithm:

- 1. Perform the Generalized Hough Transform for images I and I_t . Record which patch p_k from I contributes to each vote v_k .
- 2. On the quantized parameter space, for each bin that has $N(N \ge n)$ votes
 - (a) Increment $K_n(I, I_t)$ with N choose n, $\binom{N}{n}$.
 - (b) For each vote v_k , we increment $K_n(I_{p_k}^{n}, I_t)$ for the contributing patch p_k

3 Localization

The goal of weakly supervised localization is to discover and localize the parts of images (the objects of interest) that are visually common among the same categories and discriminative among different categories. We set our goal for localization as predicting a bounding box around the objects of interest. This has also been the standard scheme for supervised object detection. We learn a discriminative classifier (SVM) on the whole images with the high order kernel. Because of the invariance of the high order features (HOFs), we can utilize the weights learned for the HOFs for localization. The weights imply the discrimination of the features. The subwindows with more positive discriminative features would be more likely to be those we want to locate.

The general idea for our localization algorithm is to first assign scores to the local patches of a test image with an efficient kernel algorithm. The scores are defined in the way that the SVM decision score for a subwindow in the test image can be calculated as the summation of the scores of the local patches inside the window. Thus after obtaining the scores for the local patches, we can easily get the subwindow with the optimal score.

3.1 Local Patch Classification

We start by explaining how to obtain scores for local patches. The score of a local patch indicates whether the patch match to the object of interest, or the background. We make the decision for a local patch p_i based on all n patches from image I including p_i , which we denote as $I_{p_i}^n = \{P^n \in I | p_i \in P^n\}$. We denote the weights for the n^{th} order features learned by the SVM as \mathbf{v} , and the bias term of SVM as b. The decision score of a local patch p_i would be the sum of the weights of the n^{th} order features in I including p_i : $Score(p_i) = \mathbf{v}^T \phi^n(I_{p_i}^n) + b$, where $\phi^n(I_{p_i}^n)$ is the n order feature vector of $I_{p_i}^n$. When n is large, it is impractical to enumerate the weights \mathbf{v} for all high order features. The kernel methods are used. Let α be the coefficients of the support vectors learned by the SVM. A support vector is a certain image in the training set.

$$Score(p_i) = \mathbf{v}^T \phi^n(I_{p_i}^n) + b = \sum_t \alpha_t K_n(I_{p_i}^n, I_t) + b$$
 (2)

where K_n is the high order kernel defined in 2.3, and I_t is the t^{th} support vector.

The kernel value $K_n(I_{p_i}^n, I_t)$ is equal to the number of co-occurring n^{th} order features of $I_{p_i}^n$ and I_t . We can compute this value from the parameter space after the Hough transform is performed on the whole image I with I_t . Algorithm 1 gives the algorithm that calculates

the kernel values for every patch p_i and the whole image I at the same time. The algorithm is illustrated in Fig. 2. The key idea is that when we have N votes in a bin on the parameter space, one of which is contributed by p_i from I (with a patch from I_s), we have $\binom{N-1}{n-1}$ number of co-occurring n^{th} order features of I_s and I including p_i .

The computational complexity for Algorithm 1 is the same as calculating the kernel for an image, which is in practise linear to the number of patches in an image, O(|I|). Therefore, with the same number of kernel computations for classifying an image O(S), where S is the number of support vectors), we also computed the scores for all local patches.

3.2 Subwindow Classification

We classify each subwindow B as a bounding box of the object of interest or not. We define the n^{th} order feature vector $\phi^n(B)$ of B as the number of occurrences of the features. When counting the occurrences of a n^{th} order feature, we count 1 when all n patches of the feature are inside B, otherwise, we count l/n, where l is the number of patches inside B. Therefore, to make the decision for a subwindow, we also consider the n^{th} order features that represent the context surrounding it. The value we give to a context feature is the ratio of the patches inside the window. Many works $[\Box, \Box]$ have shown that context can improve the detection performance.

The decision score of B can be calculated as follows.

$$Score(B) = \sum_{t} \alpha_{t} K(B, I_{t}) = \mathbf{v}^{T} \phi^{n}(B)$$
(3)

where α is the SVM coefficients of the support vectors, and **v** is the weights for the features. Combining with equation 2, it is easy to prove that the above equation can be calculated as the sum of the scores of the local patches inside *B*.

$$Score(B) = \frac{1}{n} \sum_{p_i \in B} (Score(p_i) - b) + b$$
 (4)

Finding the optimal subwindow

From the derivation above, after assigning the scores to every patch on an image with the algorithm in Section 3.1, the process of finding the optimal subwindow would be the same as the process for a linear SVM with the bag of words features. While the weights of the corresponding words learned by the SVM in the bag of words model would be replaced by the scores of the patches in our algorithm. Therefore, we can efficiently compute the decision scores of all possible subwindows with *integral image*. The efficient subwindow search algorithm can also be used to find the optimal subwindow. We can use the same upper bound designed for bag of words with linear SVM in the paper [12].

3.3 Normalization

Similar to the kernel for categorization (Section 2.3), we can use L1-norm for normalization. We denote the score for a subwindow B with the normalized kernel as Score(B).

$$\hat{Score}(B) = \sum_{t} \alpha_{t} \hat{K}_{n}(B, I_{t}) = \sum_{t} \alpha_{t} \frac{K_{n}(B, I_{t})}{\sum_{k: P_{k}^{n} \in I} I_{k} \binom{|I_{t}|}{n}}$$

$$(5)$$

where l_k is the number of patches of P_k^n inside window B, and $\sum_{k:P_k^n \in I} l_k$ is the L1-norm of the feature vector for subwindow B defined in Section 3.2. We define the score for a patch p_i

as $\tilde{Score}(p_i) = \sum_t \alpha_t K_n(I_{p_i}^n, I_t) / \binom{|I_t|}{n}$. This can be calculated with Algorithm 1 by modifying the Step 2(b). With this score for local patches, the normalized score for subwindow B can be easily calculated with the following equation.

$$\hat{Score}(B) = \frac{\frac{1}{n} \sum_{p_i \in B} \hat{Score}(p_i)}{\sum_{k: P_k^n \in I} l_k} = \frac{\sum_{p_i \in B} \hat{Score}(p_i)}{|B| \binom{|I| - 1}{n - 1}}$$
(6)

where |B| is the number of patches inside B. The values |B| for all subwindows B in an image can also be calculated with integral image. However, unlike the non-normalized case, the efficient subwindow search algorithm does not work directly with the upper bound designed for the bag of words model.

4 Experiments

We evaluate the proposed approach on several public datasets, including the Pascal VOC 2005 [5], Graz-01, Graz-02 [24], and Caltech-4 [5]. For all the datasets, we train the classifiers with only image level labels. We would like to verify two things: 1) Adding more invariance to the high order features can improve the recognition performance. 2) The classifier with high order kernels which is learned with only weakly labeled data can be used to localize the objects on new test images.

For all the datasets, we use harris-hessian interest region detectors [\square] and SIFT feature descriptor [\square]. We build the vocabulary with K-means. We use K = 1000 for Pascal (the same as the winner in the challenge [\square]), and K = 500 for Graz, and Caltech datasets. We quantize the parameter space with step size 16px for all datasets. Translation and scale invariance are considered during the experiments.

4.1 Categorization

We do experiments on the second test dataset (difficult one) of Pascal 2005 dataset. In table 1(a), we compare the equal error rates of our method with the best score at the competition [\square 3], and other works after the competition [\square 3]. The best score at the competition is achieved with a χ^2 -kernel on bag of words representation applied to SVM. The PDK [\square 3] improves the bag of words model by modeling the distances of pair of words. We implemented the scale variant high order kernels in [\square 3] (variant HOF) with exact the same settings as ours. By modeling scale invariance, we outperformed the variant HOF by 4.4%, since the datasets include objects with great variability in scales. We outperform the bag of words model and the models (PDK) that only consider the distances of words by 5.9% and 5.5% respectively.

We further compare our approach with pervious works on Graz-01 and Graz-02 datasets. We adopt the same training and testing split as in [23]. On the Graz-01 dataset, we compare the scale variant HOF results reported in [23]. Table 1 (b) compare the EER of each category with previous works. We outperform other methods which use bag of words [23], or pair of words [23].

4.2 Localization

We first evaluate our localization algorithm on the Caltech-4 dataset [N] with the categories that have more rigid shapes: faces, motorbikes and airplanes. On this dataset, many previous

- 1	lable I	: Equal	Error R	kate (%)	on the	Pas	cal 05	(test2)	dataset	(a), and	the Gr	az datas	et (b)
		Winner		Variant	Invariant	ſ						Variant	Invariar

Class	Winner	PDK [18]	Variant	Invariant	
Class	BoW [28]	FDK [10]	HOF [29]	HOF	
Motor	79.8	76.9	82.1	83.8	
Bike	72.8	70.1	74.3	78.0	
Person	71.9	72.5	73.0	73.5	
Car	72.0	78.4	77.7	84.8	
Ave	74.1	74.5	76.8	80.0	

Dataset	Class	BoW [24]	Pair[17]	PDK[18]	Variant	Invariant
Dataset					HOF[26]	HOF
Graz01	Bicycle	86.5	84.0	95.0	94.0	96.0
Grazor	Person	80.8	82.0	88.0	86.0	90.0
	Bike	77.8	92.0	86.7	n/a	88.0
Graz02	Person	81.2	86.0	86.7	n/a	92.0
	Car	70.5	n/a	74.7	n/a	80.3

(a) Pascal (b) Graz

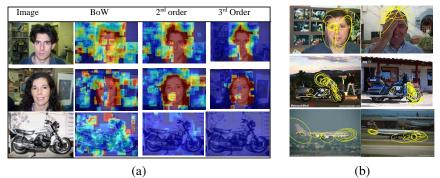


Figure 3: (a) Posterior maps for the *face* detector with 1 to 3 order features. Dark red means higher scores and dark blue means lower scores. Ideally, we should get high scores only on faces and low probability on background and other categories (motorbikes). (b) The patches (ellipses) with top scores obtained by the face detector (top images), the motorbike detector (middle), and the airplane detector (bottom).

works [3][23], including the bag of words model, can already achieve nearly 99% accuracy for object categorization. However the weakly supervised localization task on this dataset is not as perfect [3]. We use the classifier trained on the whole images of the training data, and evaluate the localization results on the test data. Since there is no much scale variance in this dataset, we calculate the decision scores for all subwindows of a fixed size for each category. We evaluate our results with the multi-class settings, where the detections from all other three categories (including the Leopard category) are false positives.

Fig. 3(a) illustrates the decision scores made by the face classifier on each local patch of images. The figures are obtained as follows. For each pixel, we average the scores of all regions including this pixel. The regions are modeled as rectangles. For bag of words model, high scores appear on both the objects, backgrounds and also other objects, since the same word may appear at many locations. As we increase the order of the features, high scores are much more likely to appear on the objects (faces). This is because the high order features are more discriminative and do not share so much between different categories and the backgrounds. Moreover, when we make the decision of each patch, we actually consider the interaction of this patch with all other patches in the image. Fig. 3 (b) shows the local patches which got highest scores with 3rd order features.

Fig. 4 presents the precision-recall curve of each category. We use the same evaluation scheme defined for detection task of the Pascal challenge. The bag of words model is not discriminative enough for the localization task when classifiers are learned with weakly labeled data, except the motorcycle category, for which most motorcycles cover the entire image.

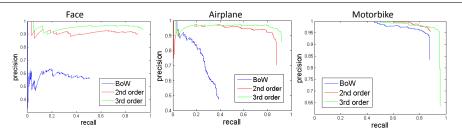


Figure 4: Precision-recall curves for detection task on Caltech-4 dataset. We compare bag of words (green), 2nd order feature (red), and 3rd order feature (green). Comparing to the Precision-recall curves present in [N], which uses a constellation model, we got better performance when using the high order features.

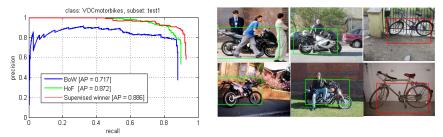


Figure 5: Precision-recall curves on the test1 set of Pascal 2005 for the motorcycle category (vs. all). Note that our detector is trained with only weakly labeled data. Example detections (green) and false positives (red) with high scores are shown in the right images

The model of high order features achieves much better performance. The Average Precision scores (as defined in Pascal challenge) using the 3rd order features are 0.884, 0.883, 0.903 for airplane, face and motorbike respectively.

Scale Invariant Localization

We further evaluate the proposed algorithm on the motorcycle category of Pascal 2005 dataset, which include more scale variance than the Caltech dataset. All detections on images of other categories and the background are treated as false positives. We test on the test1 set. There are 689 images, among which 220 motorcycles are presented.

For scale invariance, we make decisions for subwindows of different scales with a fixed aspect ratio. It is worth noting that the algorithm which performs scale variant detection after rescaling the image into different sizes does not work, because objects in training images appear in different scales, and we do not know the scales during training. Therefore, the models or features must be able to do scale invariant matching.

Fig. 5 shows the precision recall curves which compare the bag of words model, our model, and the supervised winner in the challenge. The average precision with the scale invariant detection model is 0.872, which is close to the winner in the challenge (AP: 0.886) where model is trained using bounding boxes.

Computational time

On a Core 2 Duo 2.0G machine, it takes around 1ms for a kernel calculation of two images. Except feature extraction, localization takes around 0.1 second per image (for evaluating all subwindows) on the Caltech-4 dataset. The implementation is done in Matlab with

some C helps.

5 Conclusion

We proposed efficient algorithms for weakly supervised object categorization and localization with the high order features. A classifier is trained on the whole images with invariant high order features. The learned weights of the features are backprojected for localization. An efficient algorithm is proposed for localization, which avoid calculating kernel values for all possible windows and consider context of the window at the same time. For future work, we plan to apply our localization algorithm for supervised detection task.

References

- [1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 714–725, 1987.
- [2] M. B. Blaschko and C. H. Lampert. Object localization with global and local context kernels. In *BMVC*, 2009.
- [3] D. Chaitanya, R. Deva, and F. Charless. Discriminative models for multi-class object layout. In *ICCV*, 2009.
- [4] O. Chum and A. Zisserman. An exemplar model for learning object classes. In CVPR, 2007.
- [5] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV*, 2004.
- [6] M. Everingham, L. Gool, C. K. Williams, J. Winn, van and visual Α. Zisserman. The pascal object classes challenge 2005. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2005/workshop/index.html.
- [7] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.
- [8] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3): 273–303, 2007.
- [9] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *ECCV*, 2008.
- [10] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *ECCV*, 2008.
- [11] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.
- [12] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.

- [13] S. Lazebnik and M. Raginsky. An empirical bayes approach to contextual region classification. In *CVPR*, 2009.
- [14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.
- [15] A. Lehmann, B. Leibe, and L. van Gool. Feature-centric efficient subwindow search. In *ICCV*, 2009.
- [16] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259– 289, 2008.
- [17] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, June 2007.
- [18] H. Ling and S. Soatto. Proximity distribution kernels for geometric context in category recognition. In *ICCV*, 2007.
- [19] D. Liu, G. Hua, P. Viola, and T. Chen. Integrated feature selection and higher-order spatial feature extraction for object categorization. In *CVPR*, 2008.
- [20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
- [21] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [22] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006.
- [23] M. H. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *ICCV*, 2009.
- [24] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.
- [25] S. E. Palmer. The effect of contextual scenes on the identification of objects. *Memory and Cognition*, 3:519–526, 1975.
- [26] T. Quack, V. Ferrari, B. Leibe, and L.V. Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV*, 2007.
- [27] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR*, 2006.
- [28] J. Zhang, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, 73, 2007.
- [29] Y. Zhang and T. Chen. Efficient kernels for identifying unbounded-order spatial features. In *CVPR*, 2009.