To appear in *European Conference on Computer Vision 2012*.

Semantic Segmentation with Second-Order Pooling

João Carreira[1,2], Rui Caseiro[1], Jorge Batista[1], and Cristian Sminchisescu[2]

[1] Institute of Systems and Robotics, University of Coimbra, Portugal
{joaoluis,ruicaseiro,batista}@isr.uc.pt
[2] Faculty of Mathematics and Natural Sciences, University of Bonn, Germany
cristian.sminchisescu@ins.uni-bonn.de

Abstract. Feature extraction, coding and pooling, are important components on many contemporary object recognition paradigms. In this paper we explore novel pooling techniques that encode the second-order statistics of local descriptors inside a region. To achieve this effect, we introduce multiplicative second-order analogues of average and max-pooling that together with appropriate non-linearities lead to state-of-the-art performance on free-form region recognition, without any type of feature coding. Instead of coding, we found that enriching local descriptors with additional image information leads to large performance gains, especially in conjunction with the proposed pooling methodology. We show that second-order pooling over free-form regions produces results superior to those of the winning systems in the Pascal VOC 2011 semantic segmentation challenge, with models that are 20,000 times faster.

Key words: Semantic Segmentation, Feature Pooling

1 Introduction

Object recognition and categorization are central problems in computer vision. Many popular approaches to recognition can be seen as implementing a standard processing pipeline: (1) dense local feature extraction, (2) feature coding, (3) spatial pooling of coded local features to construct a feature vector descriptor, and (4) presenting the resulting descriptor to a classifier. Bag of words [1, 2], spatial pyramids [3] and orientation histograms [4] can all be seen as instantiations of steps (1)-(3) of this paradigm [5, 6].

The role of pooling is to produce a global description of an image region – a single descriptor that summarizes the local features inside the region and is amenable as input to a standard classifier. Most current pooling techniques compute first-order statistics [5]. The two most common examples are average-pooling and max-pooling [5], which compute, respectively, the average and the maximum over individual dimensions of the coded features. These methods were

shown to perform well in practice when combined with appropriate coding methods. For example average-pooling is usually applied in conjunction with a hard quantization step that projects each local feature into its nearest neighbor in a codebook, in standard bag-of-words methods [2]. Max-pooling is most popular in conjunction with sparse coding techniques [7].

In this paper we introduce and explore pooling methods that employ second-order information captured in the form of symmetric matrices. Much of the literature on pooling and recognition has considered the problem in the setting of image classification. Here we pursue the more challenging problem of joint recognition and segmentation, also known as semantic segmentation. Our contributions can be summarized as proposing the following:

1. Second-order feature pooling methods leveraging recent advances in computational differential geometry [8]. In particular we take advantage of the Riemannian structure of the space of symmetric positive definite matrices to summarize sets of local features inside a free-form region, while preserving information about their pairwise correlations. The proposed pooling procedures perform well without any coding stage and in conjunction with linear classifiers, allowing for great scalability in the number of features and in the number of examples.
2. New methodologies to efficiently perform second-order pooling over a large number of regions by caching pooling outputs on shared areas of multiple overlapping free-form regions.
3. Local feature enrichment approaches to second-order pooling. We augment standard local descriptors, such as SIFT [9], with both raw image information and the relative location and scale of local features within the spatial support of the region.

In the experimental section we establish that our proposed pooling procedure in conjunction with linear classifiers greatly improves upon standard first-order pooling approaches, in semantic segmentation experiments. Surprisingly, second-order pooling used in tandem with linear classifiers outperforms first-order pooling used in conjunction with non-linear kernel classifiers. In fact, an implementation of the methods described in this paper outperforms all previous methods on the Pascal VOC 2011 semantic segmentation dataset [10] using a simple inference procedure, and offers training and testing times that are orders of magnitude smaller than the best performing methods. Our method also outperforms other recognition architectures using a single descriptor on Caltech101 [11] (this approach is not segmentation-based).

We believe that the techniques described in this paper are of wide interest due to their efficiency, simplicity and performance, as evidenced on the PASCAL VOC dataset, one the most challenging in visual recognition. The source code implementing these techniques is publicly available on our websites.

1.1 Related Work

Many techniques for recognition based on local features exist. Some methods search for a subset of local features that best matches object parts, either within

generative [12] or discriminative [13] frameworks. These techniques are very powerful, but their computational complexity increases rapidly as the number of object parts increases. Other approaches use classifiers working directly on the multiple local features, by defining appropriate non-linear set kernels [14]. Such techniques however do not scale well with the number of training examples.

Currently, there is significant interest in methods that summarize the features inside a region, by using a combination of feature encoding and pooling techniques. These methods can scale well in the number of local features, and by using linear classifiers, they also have a favorable scaling in the number of training examples [15]. A good review can be found in [5]. While most pooling techniques compute first-order statistics, as discussed in the previous section, certain second-order statistics have also been proposed for recognition. For example, covariance matrices of low-level cues have been used with boosting [16]. Our work pursues different types of second-order statistics, more related to those used in first-order pooling. We also focus on features that are somewhat higher-level (e.g. SIFT) and popular for object categorization, and use a different tangent space projection. The Fisher encoding [17] also uses second-order statistics for recognition, but differently, our method does not use codebooks and has no unsupervised learning stage: raw local feature descriptors are pooled directly in a process that considers each pooling region in isolation (the distribution of all local descriptors is therefore not modeled).

Recently there has been renewed interest in recognition using segments [18–20], for the problem of semantic segmentation. However, little is known about which features and pooling methods perform best on such free-form shapes. Most papers [18, 20] propose a custom combination of bag-of-words and HOG descriptors, features popularized in other domains – image classification and sliding-window detection. At the moment, there is also no explicit comparison at the level of feature extraction, as often authors focus on the final semantic segmentation results, which depend on many other factors, such as the inference procedures. In this paper, we aim to fill this gap to some extent.

2 Second-Order Pooling

We assume a collection of m local features $D = (X, F, S)$, characterized by descriptors $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$, $\mathbf{x} \in \mathbb{R}^n$, extracted over square patches centered at general image locations $F = (\mathbf{f}_1, \ldots, \mathbf{f}_m)$, $\mathbf{f} \in \mathbb{R}^2$, with pixel width $S = (s_i, \ldots, s_m)$, $s \in \mathbb{N}$. Furthermore, we assume that a set of k image regions $R = (R_1, \ldots, R_k)$ is provided (e.g. obtained using bottom-up segmentation), each composed of a set of pixel coordinates. A local feature d_i is inside a region R_j whenever $\mathbf{f}_i \in R_j$. Then $F_{R_j} = \{\mathbf{f} | \mathbf{f} \in R_j\}$ and $|F_{R_j}|$ is the number of local features inside R_j.

We pool local features to form global region descriptors $P = (\mathbf{p}_1, \ldots, \mathbf{p}_k)$, $\mathbf{p} \in \mathbb{R}_q$, using second-order analogues of the most common first-order pooling operators. In particular, we focus on multiplicative second-order interactions

(e.g. outer products), together with either the average or the max operators. We define *second-order average-pooling* (2AvgP) as the matrix:

$$\mathbf{G}_{avg}(R_j) = \frac{1}{|F_{R_j}|} \sum_{i:(\mathbf{f}_i \in R_j)} \mathbf{x}_i \cdot \mathbf{x}_i^\top, \tag{1}$$

and *second-order max-pooling* (2MaxP), where the max operation is performed over corresponding elements in the matrices resulting from the outer products of local descriptors, as the matrix:

$$\mathbf{G}_{max}(R_j) = \max_{i:(\mathbf{f}_i \in R_j)} \mathbf{x}_i \cdot \mathbf{x}_i^\top. \tag{2}$$

We are interested in using classifiers that offer training time that is linear in the number of training examples [15]. The path we will pursue is not to make such classifiers more powerful by employing a kernel, but instead to pass the pooled second-order statistics through non-linearities that make them amenable to be compared using standard inner products.

Log-Euclidean Tangent Space Mapping. Linear classifiers such as support vector machines (SVM) optimize the geometric (Euclidean) margin between a separating hyperplane and sets of positive and negative examples. However $\mathbf{G}_{avg}$ leads to symmetric positive definite (SPD) matrices which have a natural geometry: they form a Riemannian manifold, a non-Euclidean space [21]. Fortunately, it is possible to map this type of data to an Euclidean tangent space while preserving the intrinsic geometric relationships as defined on the manifold, under strong theoretical guarantees. One operator that stands out as particularly efficient uses the recently proposed theory of Log-Euclidean metrics [8] to map SPD matrices to the tangent space at $\mathbf{I}_d$ (identity matrix). See [22] for a more technical presentation within computer vision.

We use this operator, which requires only one principal matrix logarithm operation per region R_j, in our case:

$$\mathbf{G}_{avg}^{log}(R_j) = \log\left(\mathbf{G}_{avg}(R_j)\right), \tag{3}$$

We compute the logarithm using the very stable[3] Schur-Parlett algorithm [23] which involves between n^3 and n^4 operations depending on the distribution of eigenvalues of the input matrices [23]. We observed computation times of less than 0.01 seconds per region in our experiments. We do not apply this transformation with $\mathbf{G}_{max}$, which is not SPD in general [24].

Power Normalization. Linear classifiers have been observed to match well with non-sparse features. The power normalization, introduced by Perronnin *et al* [17] reduces sparsity by increasing small feature values and it also saturates high feature values. It consists of a simple rescaling of each individual feature value p by $sign(p) \cdot |p|^h$, with h between 0 and 1. We found $h = 0.75$ to work well

[3] This is the default algorithm for matrix logarithm computation in MATLAB.

in practice and used that value throughout the experiments. This normalization is applied after the tangent space mapping with $\mathbf{G}_{avg}$ and directly with $\mathbf{G}_{max}$.

We form the final global region descriptor vector $\mathbf{p}_j$ by concatenating the elements of the upper triangle of $\mathbf{G}(R_j)$ (since it is symmetric). The dimensionality q of $\mathbf{p}_j$ is therefore $\frac{n^2+n}{2}$. In practice our global region descriptors obtained by pooling raw local descriptors have in the order of 10.000 dimensions.

3 Local Feature Enrichment

Unlike with first-order pooling methods, we observed good performance by using second-order pooling directly on raw local descriptors such as SIFT (e.g. without any coding). This may be due to the fact that, with this type of pooling, information between all interacting pairs of descriptor dimensions is preserved.

Instead of coding, we enrich the local descriptors with their relative coordinates within regions, as well as with additional raw image information. Here lies another of our contributions. Let the width of the bounding box of region R_j be denoted by w_j, its height by h_j and the coordinates of its upper left corner be $[b_{jx}, b_{jy}]$. We then encode the position of d_i within R_j by the 4 dimensional vector $[\frac{f_{ix}-b_{jx}}{w_j}, \frac{f_{ix}-b_{jx}}{h_j}, \frac{f_{iy}-b_{jy}}{w_j}, \frac{f_{iy}-b_{jy}}{h_j}]$. Similarly, we define a 2 dimensional feature that encodes the relative scale of d_i within R_j: $\beta \cdot [\frac{s_i}{w_j}, \frac{s_i}{h_j}]$, where β is a normalization factor that makes the values range roughly between 0 and 1. We also augment each descriptor $\mathbf{x}_i$ with RGB, HSV and LAB color values of the pixel at $\mathbf{f}_i = [f_{ix}, f_{iy}]$ scaled to the range $[0, 1]$, for a total of 9 extra dimensions.

3.1 Multiple Local Descriptors

In practice we used three different local descriptors: SIFT [9], a variation which we call masked SIFT (MSIFT) and local binary patterns (LBP) [25], to generate four different global region descriptors. We pool the enriched SIFT local descriptors over the foreground of each region (eSIFT-F) and separately over the background (eSIFT-G). The normalized coordinates used with eSIFT-G are computed with respect to the full-image coordinate frame, making them independent of the regions, which is more efficient as will be shown in section 4. We also pool enriched LBP and MSIFT features over the foreground of the regions (eLBP-F and eMSIFT-F). The eMSIFT-F feature is computed by setting the pixel intensities in the background of the region to 0, and compressing the foreground intensity range between 50 and 255. In this way background clutter is suppressed and we allow for black objects to still have contrast along the region boundary. For efficiency reasons, we crop the image around the region bounding box and resize the region so that its width is 75 pixels.

In total the enriched SIFT descriptors have 143 dimensions, while the adopted local LBP descriptors [26] have 58 dimensions before and 73 dimensions after the enrichment procedure just described.

4 Efficient Pooling over Free-Form Regions

If the putative object regions are constrained to certain shapes (e.g. rectangles with the same dimensions, as used in sliding window methods), recognition can sometimes be performed efficiently. Depending on the details of each recognition architecture (e.g. the type of feature extraction), techniques such as convolution [4, 13], integral images [27], or branch and bound [28] allow to search over thousands of regions quickly, under certain modeling assumptions. When the set of regions R is unstructured, these techniques no longer apply.

Here, we propose two ways to speed up the pooling of local features over multiple overlapping free-form regions. The elements of local descriptors that depend on the spatial extent of regions must be computed independently for each region R_j, so it will prove useful to define the decomposition $\mathbf{x} = [\mathbf{x}^{ri}, \mathbf{x}^{rd}]$ where $\mathbf{x}^{ri}$ represents those elements of $\mathbf{x}$ that depend only on image information, and $\mathbf{x}^{rd}$ represents those that also depend on R_j. The speed-up will apply only for pooling $\mathbf{x}^{ri}$, the remaining ones must still be pooled exhaustively.

Caching over Region Intersections. Pooling naively using both (1) or (2) would require the computation of $k \cdot \sum_j |F_{R_j}|$ outer products and sum/max operations. In order to reduce the number of these operations, we introduce a two-level hierarchical strategy. The general idea is to cache intermediate results obtained in areas of the image that are shared by multiple regions. We implement this idea in two steps. First we reconstruct the regions in R by sets of fine-grained superpixels. Then each region R_j will require as many sum/max operations as the number of superpixels it is composed of, which can be orders of magnitude smaller than the number of local features contained inside it. The number of outer products also becomes independent of k. Regions can be approximately reconstructed as sets of superpixels by simply selecting, for each region, those superpixels that have a minimum fraction of area inside it.

We experimented with several algorithms to generate superpixels, including k-means, greedy merging of region intersections, or [29], all available in our public implementation. We adjusted thresholds to produce around 500 superpixels, a level of granularity leading to minimal distortion of R, obtained in our experiments by CPMC [30], with any of the algorithms.

Favorable Region Complements. Average pooling allows for one more speed-up by using $\sum_i \mathbf{x}_i^{ri}$, the sum over the whole image, and by taking advantage of favorable region complements. Given each region R_j, we determine whether there are more superpixels inside or outside R_j. We sum inside R_j if there are fewer superpixels inside, or sum outside R_j and subtract from the precomputed sum over the whole image, if there are fewer superpixels outside R_j. This additional speed-up has a noticeable impact for pooling over very large portions of the image, typical in feature eSIFT-G (defined on the background of bottom-up segments).

The last step is to assemble the pooled region-dependent and independent components. For example, for the proposed second-order variant of max-pooling, the desired matrix is formed as:

$$\mathbf{G}_{max}(R_j) = \begin{bmatrix} \mathbf{M}_i^{ri} & \max \mathbf{x}_i^{ri} \cdot \left(\mathbf{x}_i^{rd}\right)^\top \\ \max \mathbf{x}_i^{ri} \cdot \left(\mathbf{x}_i^{rd}\right)^\top & \max \mathbf{x}_i^{rd} \cdot \left(\mathbf{x}_i^{rd}\right)^\top \end{bmatrix}, \tag{4}$$

where max is performed again over $i : (\mathbf{f}_i \in R_j)$ and $\mathbf{M}_i^{ri}$ denotes the submatrix obtained using the speed-up. The average-pooling case is handled similarly. The proposed method is general and applies to both first and second-order pooling. It has however more impact in second-order pooling, which involves costlier matrix operations.

Note that when $\mathbf{x}^{ri}$ is the dominant chunk of the full descriptor $\mathbf{x}$, as in the eSIFT-F described above where 96% of the elements (137 out of 143) are region-independent, as well as for eSIFT-G where all elements are region-independent, the speed-up can be considerable. Differently, with eMSIFT-F all elements are region-dependent because of the masking process.

5 Experiments

We analyze several aspects of our methodology on the clean ground truth object regions of the Pascal VOC 2011 segmentation dataset. This allows us to isolate pure recognition effects from segment selection and inference problems and is easy to compare with in future work. We also assess recognition accuracy in the presence of segmentation "noise", by performing recognition on superpixel-based reconstructions of ground truth regions. Local feature extraction was performed densely and at multiple scales, using the the publicly available package VLFEAT [26] and all results involving linear classifiers were obtained with power normalization on. We invite the reader to consult our available implementation for additional details regarding these operations.

We begin with a comparison of first and second-order max and average-pooling using SIFT and enriched SIFT descriptors. We train one-vs-all SVM models for the 20 Pascal classes using LIBLINEAR [31], on the training set, optimize the C parameter independently for every case, and test on the validation set. Table 1 shows large gains of second-order average-pooling based on the Log-Euclidean mapping. The matrices presented to the matrix log operation have sometimes poor conditioning and we added a small constant on their diagonal (0.001 in all experiments) for numerical stability. Max-pooling performs worse but still improves over first-order pooling. The power normalization improves accuracy by 1.5% with log(2AvgP) on ground truth regions and by 2.5% on their superpixel approximations, while the 15 additional dimensions of eSIFT help very significantly in all cases, with the 9 color values and the 6 normalized coordinate values contributing roughly the same. As a baseline, we tried the popular HOG feature [4] with an 8x8 grid of cells adapted to the region aspect ratio, and this achieved (41.79/33.34) accuracy.

	1MaxP	1AvgP	2MaxP	2AvgP	log(2AvgP)
SIFT	16.61/12.36	33.92/25.41	38.74/30.21	48.74/39.26	**54.17/47.27**
eSIFT	26.00/18.97	43.33/31.91	50.16/40.50	54.30/45.35	**63.85/56.03**

Table 1. Average classification accuracy using different pooling operations on raw local features (e.g. without a coding stage). The experiment was performed using the ground truth object regions of 20 categories from the Pascal VOC2011 Segmentation validation set, after training on the training set. The second value in each cell shows the results on less precise superpixel-based reconstructions of the ground truth regions. Columns 1MaxP and 1AvgP show results for first-order max and average-pooling, respectively. Column 2MaxP shows results for second-order max-pooling and the last two columns show results for second-order average-pooling. Second-order pooling outperforms first-order pooling significantly with raw local feature descriptors. Results suggest that log(2AvgP) performs best and the enriched SIFT features lead to large performance gains over basic SIFT. The advantage of 2AvgP over 2MaxP is amplified by the logarithm mapping, inapplicable with max.

Given the superiority of log(2AvgP), the remaining experiments will explore this type of pooling. We now evaluate the combination of the proposed global region descriptors eSIFT-F, eSIFT-G, eMSIFT-F and eLBP-F, described in sec. 3 and instantiated using log(2AvgP). The contribution of the multiple global regions descriptors is balanced by normalizing each one to have L_2 norm 1. It is shown in table 2 that this fusion method, referred to by O_2P (as in order 2 pooling), in conjunction with a linear classifier outperforms the feature combination used by SVR-SEGM [18], the highest-scoring system of the the VOC2011 Segmentation Challenge [10]. This system uses 4 bag-of-word descriptors and 3 variations of HOG (all obtained using first-order pooling) and relies for some of its performance on exponentiated-χ^2 kernels that are computationally expensive during training and testing. We will evaluate the computational cost of both methods in the next subsection.

	O_2P	-eSIFT	-eMSIFT	-eLBP	Feats. in [18]	
	(linear)	(linear)	(linear)	(linear)	(linear)	(non-linear)
Accuracy	**72.98**	69.18	67.04	72.48	57.44	65.99

Table 2. Average classification accuracy of ground truth regions in the VOC2011 validation set, using our feature combination here denoted by O_2P, consisting of 4 global region descriptors, eSIFT-F, eSIFT-G, eMSIFT-F and eLBP-F. We compare with the features used by the state-of-the-art semantic segmentation method SVR-SEGM [18], with both a linear classifier and their proposed non-linear exponentiated-χ^2 kernels. Our feature combination within a linear SVM outperforms the SVR-SEGM feature combination in both cases. Columns 3-5 show results obtained when removing each descriptor from our full combination. The most important appears to be eMSIFT-F, then the pair eSIFT-F/G while eLBP-F contributes less.

5.1 Semantic Segmentation in the Wild - Pascal VOC 2011

In order to fully evaluate recognition performance we experimented with our best pooling method on the Pascal VOC 2011 Segmentation dataset without ground truth masks. We followed a feed-forward architecture similar to that of SVR-SEGM. First we compute a pool of up to 150 top-ranked object segmentation candidates for each image, using the publicly available implementation of Constrained Parametric Min-Cuts (CPMC) [30]. Then we extract on each candidate the feature combination detailed previously and feed these to linear support vector regressors (SVR) for each category. The regressors are trained to predict the highest overlap between each segment and the objects from each category [18, 19].

Learning. We used all 12,031 available training images in the "Segmentation" and "Main" data subsets for learning, as allowed by the challenge rules, and the additional segmentation annotations available online [32], similarly to recent experiments by Arbelaez *et al* [20]. Considering the CPMC segments for all those images results in a grand total of around 1.78 million segment descriptors, the CPMC descriptor set. Additionally we collected the descriptors corresponding to ground truth and mirrored ground truth segments, as well as those CPMC segments that best overlap with each ground truth object segmentation to form a "positive" descriptor set. We reduced dimensionality of the descriptor combination from 33,800 dimensions to 12,500 using non-centered PCA [33], then divided the descriptors of the CPMC set into 4 chunks which individually fit on the 32 GB of available RAM memory. Non-centered PCA outperformed standard PCA noticeably (about 2% higher VOC segmentation score given a same number of target dimensions), which suggests that the relative average magnitudes of the different dimensions are informative and should not be factored out through mean subtraction. We learned the PCA basis on the reduced set of ground truth segments plus their mirrored versions (59,000 examples) which takes just about 20 minutes.

We pursued a learning approach similar to those used in object detection [13], where the training data also rarely fits into main memory. We trained an initial model for each category using the "positive" set and the first chunk of the CPMC descriptor set. We stored all descriptors from the CPMC set that became support vectors and used the learned model to quickly sift through the next CPMC descriptor chunk while collecting hard examples (outside the SVR ϵ-margin). We then retrained the model using the positive set together with the cache of hard negative examples and iterated until all chunks had been processed. We warm-started the training of a new model by reusing the previous $\boldsymbol{\alpha}$ parameters of all previous examples and initializing the values of $\boldsymbol{\alpha}$, for the new examples to zero. We observed a 1.5-4x speed-up.

Efficiency of Feature Extraction. Using 150 segments per image, the highly shape-dependent eMSIFT-F descriptor took 2 seconds per image to compute. We evaluated the proposed speed-ups on the other 3 region descriptors, where they are applicable. Naive pooling from scratch over each different region took

11.6 seconds per image. Caching reduces computational time to just 3 seconds and taking advantage of favorable segment complements reduces time further to 2.4 seconds, a 4.8x speed-up over naive pooling. The timings reported in this subsection were obtained on a desktop PC with 32GB of RAM and an i7-3.20GHz CPU with 6 cores.

Inference. A simple inference procedure is applied to compute labelings biased to have relatively few objects. It operates by sequentially selecting the segment and class with highest score above a "background" threshold. This threshold is linearly increased every time a new segment is selected so that a larger scoring margin is required for each new segment. The selected segments are then "pasted" onto the image in the order of their scores, so that higher scoring segments are overlaid on top of those with lower scores. The initial threshold is set automatically so that the average number of selected segments per image equals the average number of objects per image on the training set, which is around 2.2, and the linear increment was set to 0.02. The focus of this paper is not on inference but on feature extraction and simple linear classification. More sophisticated inference procedures could be plugged in [18, 19, 34, 35].

Results. The results on the test set are reported in table 4. The proposed methodology obtains mean score 47.6, a 10% and 15% improvement over the two winning methods of the 2011 Challenge, which both used the same non-linear regressors, but had access to only 2,223 ground truth segmentations and to bounding boxes in the remaining 9,808 images during training. In contrast, our models used segmentation masks for all training images. Besides the higher recognition performance, our models are considerably faster to train and test, as shown in a side-by-side comparison in Table 3. The reported learning time of the proposed method includes PCA computation and feature projection (but not feature extraction, similarly in both cases). After learning, we project the learned weight vector to the original space, so that at test time no costly projections are required. We observed that reprojecting the learned weight vector did not change recognition accuracy at all.

5.2 Caltech101

Semantic segmentation is an important problem, but it is also interesting to evaluate second-order pooling more broadly. We use Caltech101 [11] for this purpose, because despite its limitations compared to Pascal VOC, it has been an important testbed for coding and pooling techniques so far. Most of the literature on local feature extraction, coding and pooling has reported results on Caltech101. Many approaches use max or average-pooling on a spatial pyramid together with a particular feature coding method [3, 36, 37]. Here, we use the raw SIFT descriptors (e.g. no coding) and our proposed second-order average-pooling on a spatial pyramid. The resulting image descriptor is somewhat high-dimensional (173.376 dimensions using SIFT), due to the concatenation of the global descriptors of each cell in the spatial pyramid, but because linear classifiers

	Feature Extr.	Prediction	Learning
Exp-χ^2 [18] (7 descript.)	7.8s / img.	87s / img.	59h / class
O_2P (4 descript.)	4.4s / img.	0.004s / img.	26m / class

Table 3. Efficiency of our regressors compared to those of the best performing method [18] on the Pascal VOC 2011 Segmentation Challenge. We train and test on the large VOC dataset orders of magnitude faster than [18] because we use linear support vector regressors, while [18] requires non-linear (exponentiated-χ^2) kernels. While learning is 130 times faster with the proposed methodology, the comparative advantage in prediction time per image is particularly striking: more than 20,000 times quicker. This is understandable, since a linear predictor computes a single inner product per category and segment, as opposed to the 10,000 kernel evaluations in [18], one for each support vector. The timings reflect an experimental setting where an average of 150 (CPMC) segments are extracted per image.

Fig. 1. Examples of our semantic segmentations including failures. There are typical recognition problems: false positive detections such as the tv/monitor in the kitchen scene, and false negatives like the undetected cat. In some cases objects are correctly recognized but not very accurately segmented, as visible in the potted plant example.

are used and the number of training examples is small, learning takes only a few seconds. We also experimented using SVM with an RBF-kernel but did not observe any improvement over the linear kernel.

Our proposed pooling leads to the best accuracy among aggregation methods with a single feature, using 30 training examples and the standard evaluation protocol. It is also competitive with other top-performing, but significantly slower alternatives. Our method is very simple to implement, efficient, scalable and requires no coding stage. The results and additional details can be found in table 5.

6 Conclusion

We have presented a framework for second-order pooling over free-form regions and applied it in object category recognition and semantic segmentation. The

	O_2P	BERKELEY	BONN-FGT	BONN-SVR	BROOKES	NUS-C	NUS-S
background	**85.4**	83.4	83.4	84.9	79.4	77.2	79.8
aeroplane	**69.7**	46.8	51.7	54.3	36.6	40.5	41.5
bicycle	22.3	18.9	23.7	**23.9**	18.6	19.0	20.2
bird	45.2	36.6	**46.0**	39.5	9.2	28.4	30.4
boat	**44.4**	31.2	33.9	35.3	11.0	27.8	29.1
bottle	46.9	42.7	**49.4**	42.6	29.8	40.7	47.4
bus	**66.7**	57.3	66.2	65.4	59.0	56.4	61.2
car	**57.8**	47.4	56.2	53.5	50.3	45.0	47.7
cat	**56.2**	44.1	41.7	46.1	25.5	33.1	35.0
chair	13.5	8.1	10.4	**15.0**	11.8	7.2	8.5
cow	46.1	39.4	41.9	**47.4**	29.0	37.4	38.3
diningtable	32.3	**36.1**	29.6	30.1	24.8	17.4	14.5
dog	**41.2**	36.3	24.4	33.9	16.0	26.8	28.6
horse	**59.1**	49.5	49.1	48.8	29.1	33.7	36.5
motorbike	**55.3**	48.3	50.5	54.4	47.9	46.6	47.8
person	**51.0**	50.7	39.6	46.4	41.9	40.6	42.5
pottedplant	**36.2**	26.3	19.9	28.8	16.1	23.3	28.5
sheep	50.4	47.2	44.9	**51.3**	34.0	33.4	37.8
sofa	**27.8**	22.1	26.1	26.2	11.6	23.9	26.4
train	**46.9**	42.0	40.0	44.9	43.3	41.2	43.5
tv/monitor	44.6	43.2	41.6	37.2	31.7	38.6	**45.8**
Mean	**47.6**	40.8	41.4	43.3	31.3	35.1	37.7

Table 4. Semantic segmentation results on the VOC 2011 test set [10]. The proposed methodology, O_2P in the table, compares favorably to the 2011 challenge co-winners (BONN-FGT [19] and BONN-SVR [18]) while being significantly faster to train and test, due to the use of linear models instead of non-linear kernel-based models. It is the most accurate method on 13 classes, as well as on average. While all methods are trained on the same set of images, our method (O_2P) and BERKELEY [20] use additional external ground truth segmentations provided in [32], which corresponds to *comp6*. The other results were obtained by participants in *comp5* of the VOC2011 challenge. See the main text for additional details.

proposed pooling procedures are extremely simple to implement, involve few parameters and obtain high recognition performance in conjunction with linear classifiers and without any encoding stage, working on just raw features. We also presented methods for local descriptor enrichment that lead to increased performance, at only a small increase in the global region descriptor dimensionality, and proposed a technique to speed-up pooling over arbitrary free-form regions.

Experimental results suggest that our methodology outperforms the state-of-the-art on the Pascal VOC 2011 semantic segmentation dataset, using regressors that are 4 orders of magnitude faster than those of the most accurate methods [18]. We also obtain state-of-the-art results on Caltech101 using a single descriptor and without any feature encoding, by directly pooling raw SIFT descriptors.

In future work, we plan to explore different types of symmetric pairwise feature interactions beyond multiplicative ones, such as *max* and *min*. Source

Aggregation-based methods						Other	
SIFT-O_2P	**eSIFT-O_2P**	SPM [3]	LLC [36]	EMK [37]	MP [6]	NBNN [38]	GMK [39]
79.2	80.8	64.4	73.4	74.5	77.3	73.0	80.3

Table 5. Accuracy on Caltech101 using a single feature and 30 training examples per class, for various methods. Regions/segments are not used in this experiment. Instead, as typical for this dataset (SPM, LLC, EMK), we pool over a fixed spatial pyramid with 3 levels (1x1, 2x2 and 4x4 regular image partitionings). Results are presented based on SIFT and its augmented version eSIFT, which contains 15 additional dimensions.

code implementing the techniques presented in this paper is publicly available online from our websites.

Acknowledgments. We would like to thank the anonymous referees for helpful suggestions. This work was supported, in part, by the Portuguese Science Foundation (FCT) under project "Differential Geometry for Computer Vision and Pattern Recognition" (PTDC/EEA-CRO/122812/2010), and by the European Commission under a Marie Curie Excellence Grant MCEXT 025481.

References

1. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. TPAMI (1997)
2. Dance, C., Willamowski, J., Fan, L., Bray, C., Csurka, G.: Visual categorization with bags of keypoints. In: ECCV SLCV Workshop. (2004)
3. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. (2006)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
5. Boureau, Y., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in vision algorithms. In: ICML. (2010)
6. Boureau, Y., Le Roux, N., Bach, F., Ponce, J., LeCun, Y.: Ask the locals: multi-way local pooling for image recognition. In: ICCV. (2011)
7. Ranzato, M., Boureau, Y., LeCun, Y.: Sparse feature learning for deep belief networks. In: NIPS. (2007)
8. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Geometric means in a novel vector space structure on symmetric positive-definite matrices. In: SIAM JMAA. (2006)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2011 (http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html)
11. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. CVIU (2007)
12. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. IJCV (2007)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI (2010)

14. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: ICCV. (2005)
15. Joachims, T.: Training linear svms in linear time. In: ACM KDD, ACM (2006)
16. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. TPAMI (2008)
17. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: ECCV. (2010)
18. Carreira, J., Li, F., Sminchisescu, C.: Object Recognition by Sequential Figure-Ground Ranking. IJCV (2012)
19. Ion, A., Carreira, J., Sminchisescu, C.: Probabilistic joint segmentation and labeling. In: NIPS. (2011)
20. Arbelaez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., Malik, J.: Semantic segmentation using regions and parts. In: CVPR. (2012)
21. Bhatia, R.: Positive Definite Matrices. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, USA (2007)
22. Caseiro, R., Henriques, J., Martins, P., Batista, J.: A nonparametric riemannian framework on tensor field with application to foreground segmentation. In: ICCV. (2011)
23. Davies, P.I., Higham, N.J.: A schur-parlett algorithm for computing matrix functions. (2003)
24. Caputo, B., Jie, L.: A performance evaluation of exact and approximate match kernels for object recognition. ELCVIA (2010)
25. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. TPAMI (2002)
26. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/ (2008)
27. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. CVPR (2001)
28. Lampert, C., Blaschko, M., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR. (2008)
29. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR. (2009) 2294–2301
30. Carreira, J., Sminchisescu, C.: CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. TPAMI (2012)
31. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR (2008)
32. Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV. (2011)
33. Jolliffe, I.: Principal Component Analysis. Springer Verlag (1986)
34. Ladicky, L., Sturgess, P., Alaharia, K., Russel, C., Torr, P.H.: What, where & how many ? combining object detectors and crfs. In: ECCV. (2010)
35. Gonfaus, J.M., Boix, X., van de Weijer, J., Bagdanov, A.D., Serrat, J., Gonzàlez, J.: Harmony potentials for joint classification and segmentation. In: CVPR. (2010)
36. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR. (2010)
37. Bo, L., Sminchisescu, C.: Efficient Match Kernel between Sets of Features for Visual Recognition. In: NIPS. (2009)
38. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR. (2008)
39. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: ICCV. (2011)