HAL
open science

Classification of Urban Scenes from Geo-referenced Images in Urban Street-View Context

Corina Iovan, David Picard, Nicolas Thome, Matthieu Cord

To cite this version:

Corina Iovan, David Picard, Nicolas Thome, Matthieu Cord. Classification of Urban Scenes from Geo-referenced Images in Urban Street-View Context. Machine Learning and Applications (ICMLA), 2012 11th International Conference on, Dec 2012, Boca Raton, Florida, United States. pp.339–344. hal-00794980

HAL Id: hal-00794980

https://hal.science/hal-00794980

Submitted on 11 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Classification of Urban Scenes from Georeferenced Images in Urban Street-View Context

Corina Iovan
ECP - INRIA Saclay
Chtenay-Malabry, France
corina.iovan@ecp.fr

David Picard
Laboratoire ETIS, ENSEA
Cergy-Pontoise, France
picard@ensea.fr

Nicolas Thome, Matthieu Cord
LIP6, Université Pierre et Marie Curie
Paris, France
nicolas.thome@lip6.fr, matthieu.cord@lip6.fr

***Abstract*—This paper addresses the challenging problem of scene classification in street-view georeferenced images of urban environments. More precisely, the goal of this task is semantic image classification, consisting in predicting in a given image, the presence or absence of a pre-defined class (e.g. shops, vegetation, etc.). The approach is based on the BOSSA representation, which enriches the Bag of Words (BoW) model, in conjunction with the Spatial Pyramid Matching scheme and kernel-based machine learning techniques. The proposed method handles problems that arise in large scale urban environments due to acquisition conditions (static and dynamic objects/pedestrians) combined with the continuous acquisition of data along the vehicle's direction, the varying light conditions and strong occlusions (due to the presence of trees, traffic signs, cars, etc.) giving rise to high intra-class variability. Experiments were conducted on a large dataset of high resolution images collected from two main avenues from the 12th district in Paris and the approach shows promising results.**

***Keywords*-semantic image classification; street-level images; visual words; spatial pyramid matching; kernel-based machine learning;**

I. Introduction

In recent years, the emergence of street-level geoviewers (Google Street View, Microsoft Live Earth, Geoportail ...) led to a growing interest in exploiting the visual content of the acquired data. There are at least two particularities corresponding to this task, one being data acquisition and the second one being the applications developed. As such, numerous mobile mapping systems capable of capturing and delivering geospatial data of entire cities and metropolitan areas were conceived by companies such as Blue Dasher Technologies Inc. EveryScape Inc., Earthmine Inc., Google ™, or different Geographic Survey Agencies. Vehicles are criss-crossing the urban environment collecting stereo photographs and/or laser scanner data of every street, alley and freeway in the urban environment and creating highly detailed and accurate spatial datasets at a large scale. Collected data is globally positioned and oriented to form a seamless geospatial framework that accurately describes the urban environment.

Tremendously diverse target applications can be considered by exploiting such complex datasets, extending from 3D navigation through panoramic images, image-based search engines based on semantic and spatial queries ("Which offices are within the 50 meters from this point?") and 3D city modeling ones. The difficulty in processing such data arise from the challenging context of street-view image acquisition conditions generating occlusions, varying viewpoint and real traffic speed conditions (not constant speed for the acquisition system). This paper addresses the challenging task of street scene classification, which consists in predicting in a given image, the presence or absence of a pre-defined class (e.g. shops, vegetation, etc). This is done by exploiting local features extracted from a database of street-level high-resolution images acquired by a mobile mapping system. Data was collected from streets of the 12th district of Paris. It is a dense urban area combining natural scenes (park entances, street furniture, etc.) with highly commercial avenues overcrowded by pedestrians and more residential ones, containing a high number of parked vehicles. The realistic data acquisition conditions (static and dynamic objects/pedestrians) combined with the continuous acquisition of data along the vehicle's direction, the varying light conditions and strong occlusions due to the presence of trees, traffic signs, cars, etc. gives rise to high intra-class variability to the street scene classification task. The proposed system (c.f. Section II) follows the BOSSA approach [1] which is an extension of the Bag of Words (BoW) image retrieval formalism, in conjunction with the Spatial Pyramid Matching (SPM) scheme and kernel-based machine learning techniques. Details on the dataset and selected categories will be presented in section III), while experiments and results obtained for each category will be presented in section IV.

II. Overview of the Scene Classification Pipeline

Scene classification is typically based on finite-dimensional representations of image regions, or feature vectors, describing the color, texture and/or other visual properties of images [2]. Effective image features (also called visual features or points of interest) are crucial to the performance of image classification tasks. Such tasks have been largely tackled by the Computer Vision community and can be summarized by Figure 1.

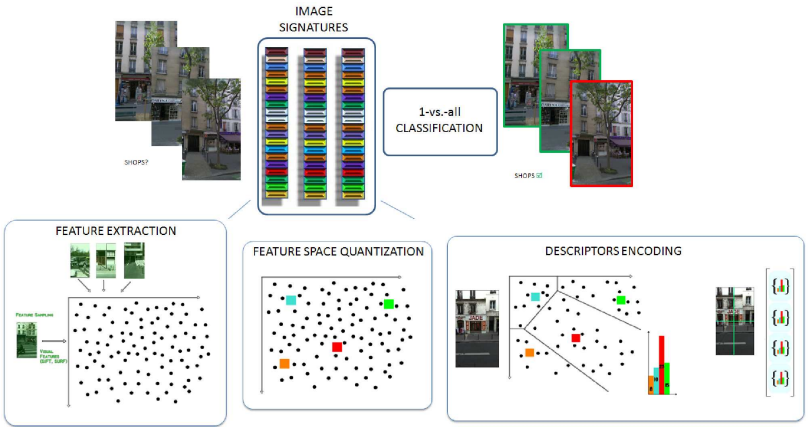


Figure 1. Image classification pipeline. Image signatures computed for each image in the dataset are used by the classification system to predict classes the image belongs to. Such unique vector representations are obtained by the following succesive steps: after dense sampling, visual feature space is quantized into visual words, which are further used to encode image characteristics. Spatial information is also taken in consideration and the final feature vector obtained by concatenation is the final image representation which can be used by any machine learning system to predict the class of an unknown sample.

In the following subsections, we will detail each of the steps of the scene classification pipeline, starting with state-of-the-art techniques encountered in such systems and continuing with the ones used for urban scene classification.

A. Detection and Description of Visual Features

The first step can be divided in two, the detection of a finite set of points containing rich local information and the description of the visual neighborhood of these points. Detection and description of visual features can either be considered together or independently. From the latter point of view, many types of primitive detectors exist: points (corners, blobs, etc.), edges, or rectangles. Ease in detection and description made point detectors the most popular in the computer vision community. They are classically based on signal processing principles: the interest points are maxima of saliency functions computed on the image signal. Other researches introduced blob detectors [3] based on maximal stable regions, or/and salient regions [4] making use of information theory. The most popular interest point detector is the Scale Invariant Feature Transform (SIFT) [5]. It combines an interest point detector based on the maxima of the scale space with an efficient descriptor that relies on histogram of gradients. From the same point of view of detector and descriptor taken together, a similar interest point detector and descriptor, SURF, was released by [6]. Detection is based on the Hessian matrix [7] while the descriptor consists of a distribution of 2D Haar wavelet responses around the point of interest. Once interest points have been sampled, local visual content around them can be described in a variety of manners. Depending on the level of invariance needed (viewpoint, scale, orientation, illumination, etc.) the choice of the descriptor may vary. In some applications, mere patches around the interest points are used as descriptors. For example Lepetit and Fua [8] use patches to train classifiers for object detection whereas Gabor filter banks (which are texture descriptors, performing a time-frequency analysis of the signal) are used in medical imaging [9]. The most widely used descriptor is the SIFT. Each interest region previously extracted is divided into sub-regions, each of which is associated an orientation histogram weighted by the maximum gradient orientation in the sub-region. The final descriptor is a concatenation of orientation histograms of each sub-region. The robustness of SIFT descriptors to small displacements and lighting changes as well as an efficient available implementation by Andrea Vedaldi [10] have made them the gold standard in Computer Vision tasks. In this work, interest regions were extracted in a dense sampling strategy. For each of the thus extracted regions, SIFT descriptors are computed on each of the color channels of the images in the dataset. The dimension of each descriptor is of 384 (3x128 SIFT dimension). This step outputs a set of local descriptors, denoted by $\mathcal{F} = \left\{ f_i = (p_i, d_i) \in \mathcal{K} \times \mathbb{R}^{3\times128} \right\}_{1\leq i\leq N}$, whereas N is the number of local regions $p_i \in \mathcal{K}$ obtained after the dense sampling step and associated with descriptors $d_i \in \mathbb{R}^{3\times128}$.

B. Descriptors Encoding

The aim of this step is to obtain a global descriptor for each image based on local descriptors. One way to describe an image is to declare its contents using the previously extracted local descriptors, in a manner similar to the Bag-of-Words (BoW) model from text retrieval [11]: given a text and a predefined dictionary of K words, the Bag-of-Words of the text is a vector of K dimensions, where the kth entry indicates the number of times that a word k appears in the text. Analogously, an image can be represented as an unordered collection of visual words. The finite set of visual words is obtained by quantizing the space of local descriptors into informative regions whose internal structure can be

disregarded or parameterized linearly. The visual vocabulary is built during the training stage: training data is used to divide the descriptor space into clusters, each of them being labeled. The visual vocabulary is the list of cluster centers and associated identifiers. The clustering procedure is based on the *k-means* algorithm. Having the visual codebook and the dataset, each visual word appears in different amount of images and different times in each particular image. The visual codebook is denoted $C = \{c_m\}, m \in \{1; M\}$, M is the number of visual words.

C. Computing Image Signatures

The Bag-of-Words (BoW) representation is a collection of visual words representing the image content. Having the visual codebook, for each descriptor of local features from the image, the k visual words (nearest clusters) are found. The number of occurrences of each word is computed and used to increase the value of the image signature at the word's ID position. The image signature then can be seen as a histogram of occurred visual words. Given a set of descriptors $\{d_1, ..., d_N\}$ sampled from an image, let k_{mi} be the assignment of each descriptor d_i to the corresponding visual word c_m obtained through *k-means* clustering. Each local descriptor is assigned to the nearest visual word, according to $k_{mi} = \begin{cases} 1 & \text{if } m = \underset{m \in \{1;M\}}{\arg\min} \|d_i - c_m\|^2 \\ 0 & \text{otherwise} \end{cases}$

Thus the encoding of the set of local descriptors corresponds to a scalar h_m given by $[h]_m = card\,(d_i | k_{mi} = m)$. The final vector representation for an image, h is made up by concatenating values for each visual word $h = [h_1, \cdots, h_m, \cdots, h_M]^T$. Although numerous drawbacks, the BoW image representation became popular due to its simplicity and good performance. Its main limitation is the loss of spatial information, which can be overcome by computing one encoding (e.g. BoW) in different sub-regions of an image and then stacking the results (as proposed in [12] with the Spatial Pyramid Matching (SPM) technique). When computing the encoding for each spatial region, the contribution of local features can be considered either through sum-pooling, in which case the encodings of visual features in a given region are combined additively, or through max-pooling, in which case each bin in the encoding is assigned a value equal to the maximum across feature encodings in that region. Here, we use sum-pooling for the BoW encoding. Coding errors can be induced by the quantization of the descriptor space, which provides a very coarse approximation to the actual distance between two features - zero if assigned to the same visual word and infinite otherwise. Alternatives to such approaches (called hard assignment) have been proposed: soft-assignment (soft-weighting) techniques [13] which assign different weight to the visual word according to its distance or rank in the list or approaches explicitly minimizing reconstruction errors, e.g. Local Linear Coding [14]. Other approaches model the visual vocabulary through a probability density function (a Gaussian Mixture Model) such as the Fisher Vector representation [15] which describes in which direction the parameters of the model should be modified to best fit the data. In this work, we follow the BOSSA approach [1] that extends the BOW representation. It consists in modelizing the distribution of visual features around each visual word by computing for each local descriptor the distances to visual words. Given the distribution of visual features around the visual words, and the spread of this distribution, **histograms of occurrences of visual words** relative to the nearest prototype are built. Then the spatial pyramid approach is used to create local histograms. The approach is illustrated in Figure 2 through a toy example and compared to the creation of the BOW image representation. The distribution

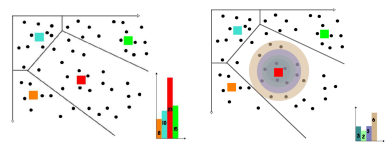


Figure 2. Image representations. (a) Standard Bag of Words encoding. (b) BOSSA representation. For each visual word, a histogram is computed. Distances around each visual word are discretized into a fixed number of bins. Local descriptors are indicated through black dots. The height of each bin of the histogram accounts for the number of local descriptors which are at the same distance from the visual word. Note that if the number of bins is set to 1, the BOSSA encoding resumes to the BoW representation.

of local descriptors around each visual word is estimated by discretizing distances over B bins and counting the number of local descriptors falling into each bin. Thus, for each visual word c_m we obtain a local histogram h_m given by $h_m = card(d_i | k_{mi} \in k_m^{max} \cdot [\frac{k}{B}, \frac{k+1}{B}])$, where B denotes the number of bins of each histogram h_m, k_m^{max} is the maximum distance to which the local histogram is computed. This parameter is given by the standard deviation σ_m of each visual word c_m and is obtained by applying the *k-means* algorithm, such as $k_m^{max} = \lambda \cdot \sigma_m$. To this local histogram representation is added a scalar f_m, encoding the information regarding the number of visual descriptors d_i corresponding to each visual word c_m. This is done for consistency reasons, in order to be robust to the l_1 normalization of the h_m histogram. As bin counts encode differently spatial information between different local histograms h_m representing the same image, each local histogram is normalized through: $h_m = \|h_m\|_1$. The final image representation is a vector of size $M \cdot (B + 1)$ which can be rewritten as $H = [[h_m], f_m]^T$.

D. Classification

Once image signatures have been computed, images can be classified using just any machine learning algorithm. In all experiments, a SVM classifier is used on top of each encoding. Training is performed for each class in a $one-vs.-all$ configuration. For each *region* r of the spatial

pyramid, a kernel k_r is defined by using the histograms $h_1^{(r)}$ and $h_2^{(r)}$ associated to the specific region, from the corresponding images x_1 and x_2. For each level l of the spatial pyramid grid, a kernel k_l is defined as a weighted sum of region kernels: $k_l(x_1, x_2) = \sum_{r \in l} \varpi_r k_r(h_1^{(r)}, h_2^{(r)})$, with ϖ_r being the weights. The **similarity** kernel K between two images x_1 and x_2 is the sum of kernels from each level of the spatial pyramid: $K(x_1, x_2) = \sum_l k_l(x_1, x_2)$. The weights are set such as the bigger the size of the region, the less its similarity is important in the final kernel, whereas the spatial pyramid grid is composed of one of the two configurations illustrated by Figure 3. Inspired by the spatial pyramid (SPM) technique [16] which consists in dividing an image according to a regular grid ($1x1$, $2x2$, $4x4$, or a total of 21 regions) we propose to divide the image according to the composition of an urban scene (in $1x1$, $3x1$ for a total of 4 regions). This approach is content-specific and more appropriate for the street-scene context studied here and is entitled in the following Street Context Slicing (SCS).

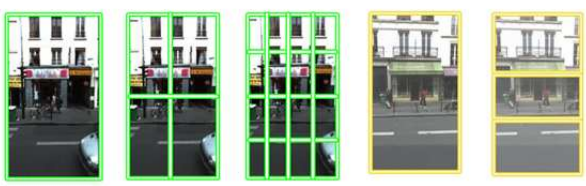


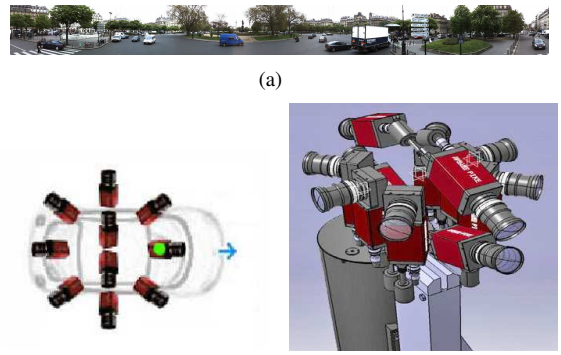
Figure 3. Taking into account spatial information in the image descriptor. Left: Illustration of the Spatial Pyramid Matching (SPM) scheme. Right: Illustration of the Street Context Slicing (SCS) technique.

III. Dataset and Implementation Details

This section describes the experimental setup, including implementation parameters used for each of the representations compared here. We apply the scene classification pipeline on a real-world dataset of a dense urban area, which will be described in the following.

A. Urban Area Mobile Mapping Dataset

Data is collected by a mobile mapping system (cf. Figure 4-(b)) composed of a set of ten full HD cameras mounted on a rigid frame (cf. Figure 4-(c)). Figure 4-(a) illustrates a panoramic assembly of images acquired by the mobile mapping system. The cameras are perfectly synchronized, mounted very closely, and have the same exposure times in order to produce seamless panoramic. They have been chosen to have a high radiometric dynamic and a high signal to noise ratio (200-300) in order to manage the variations in illumination between the shadowed and the lightened sides of the street. The cameras are triggered in a way to acquire images at regular distance intervals (one panoramic per 3 meters). The images are georeferenced in a global reference frame with the help of an Inertial Navigation Systems (integrating 2 GPS, an Inertial Measurement Unit



(a)

(b) (c)

Figure 4. The iTowns Mobile Mapping System.

and an odometer) providing overall a submetric absolute localization. All images were drawn from the same dataset, acquired on two main streets from the 12th district of Paris. Qualitatively the images contain a very wide range of viewing conditions, occlusions, and images where there is little bias toward images being of a particular object, e.g., there are images of boutiques in a street scene, rather than solely images of boutiques. Figure 5 depicts images from the dataset for each category. Experiments were conducted on

Figure 5. Example images for each of the four categories chosen for the street-scene classification task. First row presents examples from the *vegetation* category, second row presents samples from the *porch* class, the third one depicts the *commerce* class and the last row illustrates the generic *background* class.

images acquired by the side-cameras of the mobile mapping system, acquiring images from an orthogonal viewpoint with respect to the moving direction. Image sizes are 1920×1080 pixels and color information is exploited for all samples of the dataset. The dataset was equally divided into two sets, one used during the training stage and the second one during the test. We built a database of 1516 samples containing

four categories of scenes by manually labeling image data. The total number of examples labeled from each category is given in Table I, along with the number of samples used in the classification framework for each category.

Table I
DATASET SIZE FOR THE CLASSIFICATION FRAMEWORK. NUMBER OF SAMPLES FROM EACH CATEGORY USED DURING TRAINING AND TEST.

	Number of images		
Category	Total	Train	Test
Shops	569	284	285
Porch	194	97	97
Vegetation	404	202	202
Background	349	175	174
TOTAL	1516	758	758

B. Implementation Details

Visual features Interest regions are extracted in a dense sampling strategy, e.g. one region was extracted every 6th pixel on a scale of 5. The dimension of each descriptor is of 384 (3x128 sift dimension). The total number of descriptors per image is of 56.604. This is done for all images in the dataset by using the opponentcolorSIFT descriptor extracted using UVA's [17] software.

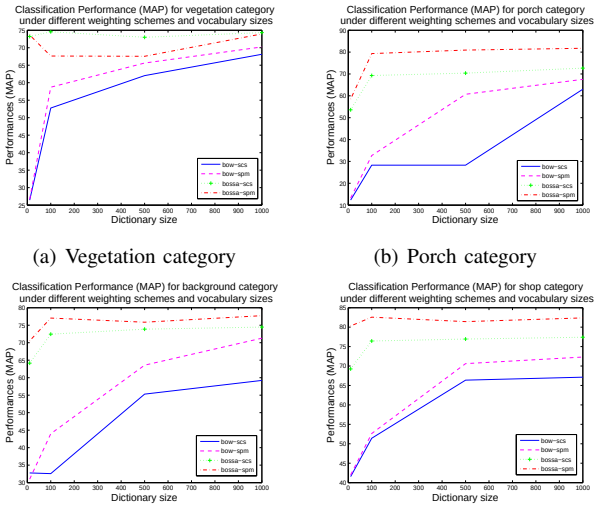
Visual dictionary 5.000 descriptors were randomly sampled from each image (i.e. some 14 million descriptors) to create a visual codebook using the *k-means* clustering algorithm with Euclidean distance. k is set to 10, 100, 500, 1000 visual prototypes.

Image signatures Two approaches have been studied to create image signatures. The first one is the baseline Bag-of-Words (BOW) approach, and the second one is the Bag Of Statistical Sampling Analysis (BOSSA). Each of the approaches was studied using two types of grid-partitions, the standard SPM one, dividing an image in 21 (1x1, 2x2, and 4x4) regions and the SCS one, dividing the image into 4 (1x1 and 3x1) parts. Representations based on BOSSA were constructed using a *hard* type of assignment of descriptors to the histograms. This consists in adding 1 to the histograms' bin corresponding to the most likely cluster (this is computed like a Gaussian centered on the clusters' center and with standard deviation computed during the *k-means* algorithm from the descriptors assigned to the cluster). The parameter values for the proposed representation of BOSSA are given in the following: B (the number of bins in each histogram) took values in the range $[5, 10]$, while the λ parameter giving k_m^{max} was in the range $[1, 5]$.

Similarity measure and SVM classifier Classification is done with a support vector machine (SVM) classifier trained in one-vs.-all paradigm: a classifier is learned to separate each class from the rest. We used the *JKernelMachines* library [18] and two types of Gaussian kernels, $chi2$ and $L2$. The weights are chosen such as each level of the pyramid has the same importance in the global similarity. The γ parameter of the Gaussian kernel is identical for each layer of the pyramid. For each classifier and for each type of image signature, the γ was tuned by cross-validation.

IV. RESULTS

The results of the experiments are shown in this section. First, we present the influence of the **dictionary size** on classification performances, for each class and different weighting schemes. Figure 6 presents classification performance results for varying codebook sizes of 10, 100, 500 and 1000 prototypes. Table IV lists classification performances



(a) Vegetation category (b) Porch category

(c) Background category (d) Shops category

Figure 6. Classification Performances (MAP) for each class under different weighting schemes and vocabulary sizes.

achieved for each class, in terms of Mean Average Precision (MAP), for the two representation approaches, BOW and BOSSA. In order to be comparative in terms of size of the image representation obtained, results for a dictionary size of 100 for the BOSSA approach and for 1000 for the BOW approach should be compared.

Table II
SCENE CLASSIFICATION RESULTS FOR STREET-LEVEL DATABASE. THE HIGHEST RESULTS FOR EACH CONFIGURATION ARE SHOWN IN BOLD.

		BOW k=1000		BOSSA k=100		BOSSA k=1000	
	Kernel	SPM	SCS	SPM	SCS	SPM	SCS
shop	$chi2$	72.3	67.1	**82.5**	81.8	82.3	77.4
shop	$L2$	72.3	67.1	**81.9**	76.4	80.4	79.1
porch	$chi2$	67.5	63	**79.3**	69.2	81.7	72.6
porch	$L2$	67.5	63	94.2	**95.2**	94.2	93.7
vegetation	$chi2$	**70.2**	68.1	68.8	68.7	73.9	74.3
vegetation	$L2$	70.2	68.1	67.6	**74.5**	69.0	70.8
background	$chi2$	71.2	59.2	**72.4**	60.9	77.7	74.4
background	$L2$	71.2	59.2	**77.0**	60.9	67.5	68.2

Globally, the proposed representation of images through the BOSSA approach improves the classification performance over the standard BoW approach. With a smaller codebook size, the BOSSA with kernel $L2$ performs better than the baseline encoding with $L2$ kernel, for all categories. The BoW representation with a $chi2$ kernel performs better than BOSSA for the vegetation category,

while the performance degrades dramatically with linear kernel for the other categories. For larger codebook sizes (k=1000), BOSSA performs better than the baseline BoW and BOSSA for k=100 for the vegetation and background categories and a $chi2$ kernel. However, it is interesting to note that the BOSSA encoding using the linear kernel achieves comparable performance to the $chi2$ kernel across different vocabulary sizes and outperforms the results using the $chi2$ kernel for porch, vegetation and background categories. This suggests that the linear kernel is sufficient to achieve good performance with the encoding, avoiding the computational expense of applying a non-linear kernel. Experiments clearly demonstrate that larger vocabularies lead to higher accuracy, as can be observed in Figure 6. It should be noted that in the case of the BOW encoding, even at a vocabulary size of $1,000$ the performance appears to be still increasing suggesting that further gains could be achieved by increasing the vocabulary size even further. Nonetheless, gains of the BOSSA encoding seem to be saturating even for higher dictionary sizes but are most of the time superior to performances achieved by the baseline approach. Spatial information obtained by applying different partitioning techniques does matter, with a higher number of regions yielding higher accuracy (the performance of the context slicing technique (SCS) considered here slightly improves over the baseline SPM for only two categories). The baseline BoW method gains considerably from large vocabularies resulting in a correspondingly large encoding size opposite than the BOSSA encoding, which results in a smaller yet faster to compute representation (since it searches neighbors/compute distances within a much smaller vocabulary).

V. Conclusion

We have presented an evaluation of two encoding methods for semantic image classification in a database of georeferenced street view images. We examined the performance of the BOSSA representation and compared it to the state-of-the-art bag-of-words (BOW) representation, both in a standard spatial-matching scheme (SPM) and a street-context one (SCS). The most encouraging result of this paper is the non-parametric histogram representation BOSSA, compact and simple to compute, which works well with SVM and improves classification accuracy. Our experiments on a variety of categories for image classification prove the effectiveness of this approach. Based on the quantitative evaluations for image classification on the real street-scene database, the proposed representation seems to retain more information than state-of-the-art approaches which it significantly outperformed. We consider that BoW approaches and extensions presented here are well adapted to the image classification task and we intend to share our database and ground truth with the community in order to allow the benchmarking of other such approaches on it. Further research of this study and theoretical understanding is an interesting direction which needs to be undertaken.

References

[1] S. E. F. de Avila, N. Thome, M. Cord, E. Valle, and A. de Albuquerque Arajo, "Bossa: Extended bow formalism for image classification," in *ICIP*, 2011.

[2] J. Yang, Y. Jiang, A. Hauptmann, and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *ACM Workshop on Multimedia information Retrieval*, 2007.

[3] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, 2004.

[4] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," *ECCV*, 2004.

[5] D. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision (ICCV)*, 1999.

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*. Springer, 2006.

[7] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, 2005.

[8] V. Lepetit and P. Fua, "Keypoint Recognition Using Randomized Trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, 2006.

[9] Y. Zhan and D. Shen, "Automated segmentation of 3D US prostate images using statistical texture-based matching method," *Medical Image Computing and Computer-Assisted Intervention*, 2003.

[10] A. Vedaldi and B. Fulkerson, "{VLFeat}: An Open and Portable Library of Computer Vision Algorithms," 2008, http://www.vlfeat.org/.

[11] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. Addison-Wesley New York, 1999, vol. 1st edition.

[12] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[13] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 7, pp. 1271–1283, 2010.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[15] G. Csurka and F. Perronnin, "Fisher vectors: Beyond bag-of-visual-words image representations," *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, 2011.

[16] K. Grauman and T. Darrell, "The pyramid match kernel: Efficient learning with sets of features," *J. Mach. Learn. Res.*, vol. 8, 2007.

[17] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[18] D. Picard, N. Thome, and M. Cord, "Jkernelmachines," 2012, http://mloss.org/software/view/409/.