Kernels for Visual Words Histograms

Radu Tudor Ionescu and Marius Popescu

Faculty of Mathematics and Computer Science
University of Bucharest, No. 14 Academiei Street, Bucharest, Romania
{raducu.ionescu,popescunmarius}@gmail.com

Abstract. Computer vision researchers have developed several learning methods based on the bag-of-words model for image related tasks, such as image retrieval or image categorization. For such an approach, images are represented as histograms of visual words from a codebook that is usually obtained with a simple clustering method. Next, kernel methods are used to compare such histograms. Popular choices, besides the linear SVM, are the intersection, Hellinger's, χ^2 and Jensen-Shannon kernels.

This paper aims at introducing a kernel for histograms of visual words, namely the PQ kernel. This kernel is inspired from a class of similarity measures for ordinal variables, more precisely Goodman and Kruskals gamma and Kendalls tau. A proof that PQ is actually a kernel is also given in this work. The proof is based on building its feature map.

Object recognition experiments are conducted to compare the PQ kernel with other state of the art kernels on two benchmark datasets. The PQ kernel has the best mean average precision (AP) on both datasets. In one of the experiments, PQ and Jensen-Shannon kernels are combined to improve the mean AP score even further. In conclusion, the PQ kernel can be used with success, alone or in combination with other kernels, for image retrieval, image classification or other related tasks.

Keywords: kernel method, rank correlation measure, ordinal measure, ordinal data, visual words histograms, bag-of-words, BoW model.

1 Introduction

The classical problem in computer vision is that of determining whether or not the image data contains some specific object, feature, or activity. Particular formulations of this problem are image classification, object class recognition, object detection. Computer vision researchers have recently developed sophisticated methods for such image related tasks. Among the state of the art models are discriminative classifiers using bag-of-words (BoW) representation [13, 19] and spatial pyramid matching [8], generative models [5] or part-based models [7]. The BoW models, which represent an image as a histogram of local features, have demonstrated impressive levels of performance for image categorization [19], image retrieval [11], or related tasks.

This paper focuses on learning methods based on the BoW model. A vocabulary (or codebook) of visual words is obtained by clustering local image descriptors extracted from images. An image is then represented as a histogram of

A. Petrosino (Ed.): ICIAP 2013, Part I, LNCS 8156, pp. 81–90, 2013.

visual words (or bag-of-visual-words). Next, kernel methods are used to compare such histograms. Popular choices, besides the linear SVM, are the intersection, Hellinger's, χ^2 and Jensen-Shannon (JS) kernels. There is no reason to limit the choice of kernels to these options, when other kernels are available. The final goal, that is to improve the results for image related tasks, can be achieved by trying different kernels that could possibly work better.

In this work, a kernel for histograms of visual words, namely the PQ kernel, is introduced. The PQ kernel is inspired from a class of similarity measures for ordinal variables, more precisely Goodman and Kruskals gamma and Kendalls tau. The idea is to treat the visual words histograms as ordinal data, in which data is ordered but cannot be assumed to have equal distance between values. In this case, a histogram will be considered as a rank of visual words according to their frequencies in that histogram. Usage of the ranking of visual words instead of the actual values of the frequencies may seem as a loss of information, but the process of ranking can actually make PQ more robust, acting as a filter and eliminating the *noise* contained in the values of the frequencies. This work proves that PQ is a kernel and it also shows how to build its feature map.

Experiments are conducted in order to assess the performance of different kernels, including PQ, on two benchmark datasets of images. The idea behind the evaluation is to use the same framework and variate only the feature maps computed with different kernels. The experiments show that the PQ kernel has the best mean average precision on both datasets.

The paper is organized as follows. Section 2 presents the learning framework used for image retrieval, image categorization and related tasks. The PQ kernel for histograms of visual words is discussed in section 3. Experiments conducted on two benchmark datasets are presented in section 4. Finally, the conclusions are drawn in section 5.

2 Learning Model

In computer vision, the BoW model can be applied to image classification and related tasks, by treating image descriptors as words. A bag of visual words is a sparse vector of occurrence counts of a vocabulary of local image features. This representation can also be described as a histogram of visual words. The vocabulary is usually obtained by vector quantizing image features into visual words. The proposed learning model (framework) has two stages, one for training and one for testing. Each stage is divided in two important steps. The first step in both stages is for feature detection and representation. The second step is to train a kernel method (in the training stage) in order to predict the class label of new images (in the testing stage).

The feature detection and representation step in the training stage works as follows. Features are detected using a regular grid across the input image. At each interest point, a SIFT feature [10] is computed. This approach is known as dense SIFT [1,3]. Next, SIFT descriptors are vector quantized into visual words and a codebook of visual words is obtained. The vector quantization process

is done by k-means clustering [9], and visual words are stored in a randomized forest of k-d trees [11] to reduce search cost. The frequency of each visual word is then recorded in a histogram which represents the final feature vector for the image. The histograms of visual words enter the training step. A kernel method is used for training. Several kernels can be used, such as the linear SVM, the intersection kernel, the Hellinger's kernel, the χ^2 kernel or the Jensen-Shannon kernel. In this paper, a novel approach is proposed, that of using the PQ kernel described in section 3.

Feature detection and representation is similar during the testing stage. The only difference is to use the same vocabulary that was already obtained in the training stage by vector quantization. The histogram of visual words that represents the test image is compared with the histograms learned in the training stage. The system can return either a label (or a score) for the test image or a ranked list of images similar to the test image, depending on the application. For image categorization a label (or a score) is enough, while for image retrieval a ranked list of images is more appropriate.

As expected for an image retrieval system, the training stage can be done offline. For this reason, the time that is necessary for vector quantization and learning is not of great importance. What matters most is to return the result for a new (test) image as quick as possible.

Performance level of the described model depends on the number of training images, but also on the number of visual words. The number of visual words must be set a priori. The accuracy gets better as the number of visual words is greater.

Note that the described model ignores spatial relationships between image features. Despite this fact, visual words showed a high discriminative power and have been used for region or image level classification [2, 6, 19]. Although most approaches are based on sparse descriptors, others have used dense descriptors [6, 17]. A good way to improve performance is to include spatial information [8]. This can be done by dividing the image into spatial bins. The frequency of each visual word is then recorded in a histogram for each bin. The final feature vector for the image is a concatenation of these histograms. The aim of this paper is to improve the performance of the learning model by trying different kernel methods. Therefore, other methods of improving the performance level are disregarded, since they are beyond the purpose of this work. However, one should be aware of all the possibilities of improving the described model for a real application, where the level of performance is of great importance.

3 PQ Kernel for Visual Words Histograms

All common kernels used in computer vision treat histograms of visual words either as finite probability distributions, for example, the Jensen-Shannon kernel, either as random variables whose values are the frequencies of different visual words in the respective images, for example, the Hellinger's kernel (Bhattacharyya's coefficient) and the χ^2 kernel. Even the linear kernel can be seen as the Pearson's correlation coefficient if the two histograms are standardized.

But the histograms of visual words can also be treated as ordinal data, in which data is ordered but cannot be assumed to have equal distance between values. In this case, the values of histograms will be the ranks of visual words according to their frequencies in image rather than of the actual values of these frequencies.

An entire set of correlation statistics for ordinal data are based on the number of concordant and discordant pairs among two variables. The number of concordant pairs among two variables (or histograms) $X, Y \in \mathbb{R}^n$ is:

$$P = |\{(i,j) \,:\, 1 \leq i < j \leq n,\, (x_i - x_j)(y_i - y_j) > 0\}|$$

In the same manner, the number of discordant pairs is:

$$Q = |\{(i,j) \,:\, 1 \leq i < j \leq n,\, (x_i - x_j)(y_i - y_j) < 0\}|$$

Goodman and Kruskal's gamma [14] is defined as:

$$\gamma = \frac{P - Q}{P + Q}$$

Kendall developed several slightly different types of ordinal correlation as alternatives to gamma. *Kendall's tau-a* [14] is based on the number of concordant versus discordant pairs, divided by a measure based on the total number of pairs (n is the sample size):

$$\tau_a = \frac{P - Q}{\frac{n(n-1)}{2}}$$

Kendall's tau-b [14] is a similar measure of association based on concordant and discordant pairs, adjusted for the number of ties in ranks. It is calculated as $(P - Q)$ divided by the geometric mean of the number of pairs not tied on X and the number of pairs not tied on Y, denoted by X_0 and Y_0, respectively:

$$\tau_b = \frac{P - Q}{\sqrt{(P + Q + X_0)(P + Q + Y_0)}}$$

All the above three correlation statistics are very related. If n is fixed and X and Y have no ties, then P, X_0 and Y_0 are completely determined by n and Q. Actually, all are based on the difference between P and Q, normalized differently.

The PQ kernel between two histograms X and Y is defined as:

$$k_{PQ}(X, Y) = 2(P - Q)$$

To prove that k_{PQ} is indeed a kernel, the explicit feature map induced by k_{PQ} is provided next.

Let $X, Y \in \mathbb{R}^n$ be two histograms of visual words. Let Ψ be defined as follows:

$$\Psi : \mathbb{R}^n \to \boldsymbol{M}_{n,n} \;\; \Psi(X) = (\Psi(X)_{i,j})_{1 \leq i \leq n, 1 \leq j \leq n}$$

with

$$\Psi(X) = \begin{cases} 1 & \text{if } x_i > x_j \\ -1 & \text{if } x_i < x_j \\ 0 & \text{if } x_i = x_j \end{cases}$$

Note that Ψ associates to each histogram a matrix that describes the order of its elements.

If matrices are treated as vectors, then the following equality is true:

$$k_{PQ}(X, Y) = 2(P - Q) = \langle \Psi(X), \Psi(Y) \rangle ,$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product. This proves that k_{PQ} is a kernel and provides the explicit feature map induced by k_{PQ}.

Another approach inspired from rank correlation measures is the WTA hash proposed in [18]. For K=2, the WTA hash is closesly related to the PQ kernel. However, there are two important differences. The first one is that WTA hash works with a random selection of pairs from the feature set. The second one is that, unlike PQ kernel, the WTA hash ignores equal pairs. In terms of feature representation, the PQ kernel represents a histogram with a feature vector containing $\{-1, 0, 1\}$ (0 for equal pairs), while the WTA hash with $K = 2$ uses only $\{1, 0\}$. In the experiments, one can observe that these differences have direct consequences to the performance level, creating an even greater gap between the two methods.

The authors of [16] state that histograms of γ-homogeneous kernels should be L_γ-normalized. Being linear in the feature space, PQ is a 2-homogeneous kernel and the histograms should be L_2-normalized. Therefore, in the experiments, the PQ kernel is based on the L_2-norm. An important advantage of PQ being linear is that it can be used with linear SVMs, such as the PEGASOS algorithm [12], that are much faster to learn and evaluate than the original non-linear SVMs.

Treating visual words frequencies as ordinal variables means that in the calculation of the distance (or similarity) measure, the ranks of visual words according to their frequencies in image will be used, rather than the actual values of these frequencies. Usage of the ranking of visual words in the calculation of the distance (or similarity) measure, instead of the actual values of the frequencies, may seem as a loss of information, but the process of ranking can actually make the measure more robust, acting as a filter and eliminating the *noise* contained in the values of the frequencies. For example, the fact that a specific visual word has the rank 2 (is the second most frequent feature) in one image, and the rank 4 (is the fourth most frequent feature) in another image can be more relevant than the fact that the respective feature appears 34 times in the first image, and only 29 times in the second. It is important to note that for big vocabularies (with more than 1.000 words), the kernel trick should be employed to obtain the kernel representation of PQ instead of computing its feature map, since there is a quadratic dependence between the feature map and the number of visual words.

4 Experiments

4.1 Datasets Description

The Pascal Visual Object Classes (VOC) challenge [4] is a benchmark in visual object category recognition and detection, providing the vision and machine learning communities with a standard dataset of images and annotation, and standard evaluation procedures. In the experiments of this work, the Pascal VOC 2007 dataset is used. The reason for this choice is that this is the latest dataset for which testing labels are available for download, and the experiments can be done offline.

The second dataset was collected from the Web by the authors of [7] and consists of 100 images each of 6 different classes of birds: egrets, mandarin ducks, snowy owls, puffins, toucans, and wood ducks. This dataset of 600 images is used in order to assess kernels behavior when less training data is available. The Birds dataset is available at `http://www-cvr.ai.uiuc.edu/ponce_grp/data/`.

4.2 Implementation and Evaluation Procedure

The framework described in section 2 is used for object class recognition. Details about the implementation of the model are given next. In the feature detection and representation step, a variant of dense SIFT descriptors extracted at multiple scales, called PHOW features [1], are used. The number of visual words used in the experiments is 500. For better accuracy, up to 10.000 visual words or more can be used.

Several state of the art kernel methods are compared with the PQ kernel in both experiments. The baseline method is the linear SVM, for which the histograms are L_2-normalized. One of the state of the art methods is based on the Hellinger's kernel. Two variants with different norms of this kernel are used. The first one is based on L_1-normalized feature vectors, and the second one is based on L_2-normalized feature vectors. Another state of the art kernel is Jensen-Shannon, which is L_1-normalized. Finally, these kernels are to be compared with the PQ kernel described in this paper. The PQ kernel is L_2-normalized. For all kernel methods, feature maps are computed from the visual words histograms. The training is always done using a linear SVM on the computed feature maps. The linear SVM is based on a implementation of the PEGASOS algorithm described in [12]. Note the feature map of the JS kernel cannot be computed directly. In order to use the same learning setting, its feature map has to be approximated using the method proposed in [16]. To approximate the JS kernel, 10.500 features are used. The idea behind the evaluation is to use the same framework and variate only the feature maps computed with different kernels, since the final goal of the experiments is to evaluate the difference between these kernels, in terms of performance. The implementation of both the feature detection and representation step, and the learning step, is mostly based on the VLFeat library [15].

The evaluation procedure for both experiments follows the Pascal VOC benchmark. The qualitative performance of the learning model is measured by using

Table 1. Mean AP on Pascal VOC 2007 dataset for machine learning methods based on visual words histograms with different kernels. The best AP on each class is highlighted with bold.

Class	**Lin.** L_2	**Hel.** L_1	**Hel.** L_2	**WTA** L_2	**JS** L_1	**PQ** L_2	**JS+PQ**
Aeroplane	0,395%	0,555%	0,558%	0,534%	0,564%	0,526%	**0,574%**
Bicycle	0,189%	0,339%	0,337%	0,398%	0,367%	**0,409%**	0,386%
Bird	0,178%	0,248%	0,247%	0,274%	0,284%	0,281%	**0,305%**
Boat	0,334%	0,540%	0,551%	0,476%	0,549%	0,505%	**0,553%**
Bottle	0,122%	**0,143%**	0,139%	0,139%	0,127%	0,140%	0,129%
Bus	0,239%	0,334%	0,336%	0,404%	0,379%	**0,419%**	0,406%
Car	0,518%	0,599%	0,602%	0,659%	0,644%	**0,670%**	0,659%
Cat	0,281%	0,349%	0,351%	0,382%	0,378%	**0,402%**	0,393%
Chair	0,308%	0,399%	0,399%	0,398%	**0,414%**	0,405%	**0,414%**
Cow	0,117%	0,174%	0,172%	**0,209%**	0,169%	**0,209%**	0,198%
Dining Table	0,205%	0,238%	0,227%	0,237%	0,242%	0,253%	**0,255%**
Dog	0,212%	0,271%	0,266%	0,263%	0,293%	0,287%	**0,299%**
Horse	0,484%	0,518%	0,530%	0,601%	0,595%	0,609%	**0,614%**
Motorbike	0,213%	0,398%	0,389%	0,427%	0,413%	**0,451%**	0,450%
Person	0,639%	0,715%	0,717%	0,756%	0,759%	0,773%	**0,774%**
Potted Plant	0,099%	**0,125%**	0,110%	0,110%	0,112%	0,111%	0,115%
Sheep	0,220%	0,217%	0,237%	0,219%	0,222%	**0,259%**	0,243%
Sofa	0,184%	0,304%	0,320%	0,310%	0,325%	0,322%	**0,333%**
Train	0,363%	0,534%	0,528%	0,547%	0,554%	0,570%	**0,574%**
TV Monitor	0,196%	0,309%	0,295%	0,345%	0,336%	**0,351%**	0,342%
Overall	0,275%	0,365%	0,365%	0,384%	0,386%	0,398%	**0,401%**

the classifier score to rank all the test images. Next, the retrieval performance is measured by computing a precision–recall curve. In order to represent the retrieval performance by a single number (rather than a curve), the mean average precision (mAP) is often computed. The average precision as defined by TREC is used in the experiments. This is the average of the precision observed each time a new positive sample is recalled.

4.3 Pascal VOC Experiment

The first experiment is on the Pascal VOC 2007 dataset. There are 20 classes available in this dataset, and for each class the dataset provides a training set, a validation set and a test set. The training and validation sets have about 2.500 images each, while the test set has about 5.000 images. The validation set is used to validate the parameter C of the linear SVM algorithm. Table 1 presents the mean AP of the linear SVM, the Hellinger's kernel, the JS kernel, the WTA hash (with $K = 2$ and 10.000 random pairs) and the PQ kernel, on the Pascal VOC dataset. Looking at the results obtained by the JS kernel on one hand, and the PQ kernel on the other, one can observe that these methods are somehow complementary in terms of performance. This gives the idea of combing the two kernels to possibly obtain better results. Indeed, in this experiment another kernel based on the sum of JS and PQ kernels is presented. In order to obtain the feature map of this kernel combination, the feature maps of JS and PQ kernels are simply concatenated.

The accuracy of the state of the art kernels is well above the accuracy of the baseline linear SVM. In terms of AP, the state of the art kernels are about 10% better than the baseline method. The PQ kernel improves the accuracy of the

Table 2. The time for the second stage of the learning model and the number of features for each kernel. The time is measured in seconds.

	Lin. L_2	**Hel.** L_1	**Hel.** L_2	**WTA** L_2	**JS** L_1	**PQ** L_2	**JS+PQ**
Time	$1-2$	$2-3$	$2-3$	$15-16$	$15-16$	$830-860$	$850-880$
Features	500	500	500	10.000	10.500	250.000	260.500

learning model, when compared to the state of the art methods. The mAP of the PQ kernel is $3,3\%$ above the mAP of the Hellinger's kernels, $1,4\%$ above the mAP of the WTA hash, and $1,2\%$ above the mAP of the JS kernel. The combination of JS and PQ kernels improves the performance even further. The mAP of the JS+PQ kernel is $3,6\%$ above the mAP of the Hellinger's kernels, $1,7\%$ above the mAP of the WTA hash, and $1,5\%$ above the mean AP of the JS kernel. PQ kernel improves results over WTA hash by $1,4\%$, showing that the two methods are distinct. If the best AP per class is considered, the PQ kernel and the JS+PQ kernel win most of the classes (18 out of 20). The results presented in Table 1 come to support this statement. The L_1-normalized Hellinger's kernel seems to work best when classes are very difficult for all kernel methods.

The feature detection and representation stage, that builds a vocabulary of visual words and obtains histograms, takes a few hours on this dataset. The time for the second stage of the learning framework, that includes computing feature maps, training and testing, depends on the number of features in the feature space for each kernel. The time for the second stage and the number of features for each kernel is given in Table 2. The time was measured on a computer with Intel Core i7 2.3 GHz processor and 8 GB of RAM memory using a single Core. While the feature maps can be computed only once for the entire experiment along with the feature detection and representation stage, training and testing has to be repeated for each class. Despite the time for the PQ kernel ($14-15$ minutes) is higher than the time for other kernels ($2-15$ seconds), it doesn't add an overhead to the overall time of the learning framework, since the overall time is about $4-6$ hours. The PQ kernel and the JS+PQ kernel are constantly better than the other methods. In conclusion, the PQ kernel, used either alone or in combination with the JS kernel, has the best performance on this experiment.

4.4 Birds Experiment

The second experiment is on the Birds dataset. The training set consists of 300 images and the test set consists of another 300 images. There are 6 classes in this dataset. For each class, the dataset contains 50 positive train images and 50 positive test images. Since there is no validation set this time, the parameter C of the linear SVM algorithm is cross-validated on the training set.

Table 3 presents the mAP of the linear SVM, the Hellinger's kernel, the JS kernel, the WTA hash (with $K = 2$ and 10.000 random pairs) and the PQ kernel, on the Birds dataset. A variant of the PQ kernel that ignores equal pairs (PQ ieq), which is more similar to the WTA hash, is also added to the experiment to emphase the difference between PQ kernel and WTA hash. The performance of the Hellinger's kernels is above the baseline linear SVM, as in the previous

Table 3. Mean AP on Birds dataset for machine learning methods based on visual words histograms with different mthods. The best AP on each class is highlighted with bold.

Class	**Lin.** L_2	**Hel.** L_1	**Hel.** L_2	**JS** L_1	**WTA** L_2	**PQ ieq** L_2	**PQ** L_2
Egret	0,552%	**0,760%**	0,747%	0,416%	0,735%	0,738%	0,753%
Mandarin Duck	0,446%	0,585%	0,607%	0,375%	0,784%	0,791%	**0,835%**
Owl	0,815%	0,895%	0,887%	0,490%	0,879%	0,889%	**0,915%**
Puffin	0,427%	0,696%	0,730%	0,369%	0,708%	0,703%	**0,764%**
Toucan	0,572%	0,715%	0,747%	0,558%	0,776%	0,787%	**0,845%**
Wood Duck	0,608%	0,795%	0,816%	0,361%	0,767%	0,769%	**0,849%**
Overall	0,570%	0,741%	0,756%	0,428%	0,775%	0,779%	**0,827%**

experiment. Both Hellinger's kernels are about 18% better than the baseline method. Unlike the previous experiment, the JS kernel has the worst accuracy on this dataset, when compared to the rest of the methods. The mAP of the JS kernel is 14,2% below the baseline AP. The bad performance of the JS kernel on this dataset can be explained by the fact that it is based on an informational measure that uses an estimation of the distribution of the data. The number of training samples may not be enough for a good estimation.

The results of the PQ kernel on this experiment are consistent with the previous experiment. The PQ kernel improves the performance of the learning model, when compared to the state of the art kernels. The mean AP of the PQ kernel is 8,6% above the mAP of the L_1-normalized Hellinger's kernel, 7,2% above the mAP of the L_2-normalized Hellinger's kernel, and 5,2% above the mAP of the WTA hash. Table 3 also shows that by ignoring equal pairs the mAP of the PQ kernel drops by 4,8%. By taking into account equal pairs and by considering the entire feature set, PQ has a significant improvement in terms of accuracy over WTA hash. There is no question that the two methods are distinct. If the best AP per class is considered, the PQ kernel wins most of the classes, again. The Hellinger's kernel based on the L_1-norm wins the *Egret* class. The PQ kernel wins the rest 5 classes. Note that the linear SVM, the L_2-normalized Hellinger's kernel and the JS kernel are not able to win any class. The PQ kernel is constantly better than the other methods. In conclusion, the PQ kernel has the best performance on the Birds dataset experiment.

5 Conclusion and Further Work

This paper discussed learning methods based on the BoW model. Usually, kernel methods, such as the linear SVM, the Hellinger's kernel or the JS kernel, are used to compare such histograms. This work showed that results for image classification, image retrieval or related tasks, can be improved by changing the kernel. Object recognition experiments compared the PQ kernel with other state of the art kernels on two benchmark datasets. The PQ kernel, used either alone or in combination with the JS kernel, has the best accuracy on the Pascal VOC 2007 experiment. The mAP of the JS+PQ kernel is at least 1,5% above the best mAP of the state of the art kernels. On the Birds experiment, the PQ kernel improved the performance again. The mAP of the PQ kernel is at least 5,2% above the best mAP of the state of the art kernels. The PQ kernel is constantly better than the other methods.

A possible way of improving the results for the PQ kernel may be that of using a TF-IDF measure for visual words as in [11]. Furthermore, eliminating visual words that have a low TF-IDF score can lead to an approximation of the PQ kernel that works faster and possibly better. In future work, other methods inspired from ordinal measures can be investigated.

References

1. Bosch, A., Zisserman, A., Munoz, X.: Image Classification using Random Forests and Ferns. In: ICCV, pp. 1–8. IEEE Computer Society Press (2007)
2. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)
3. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, vol. 1, pp. 886–893. IEEE Computer Society, Washington, DC (2005)
4. Everingham, M., van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge. IJCV 88(2), 303–338 (2010)
5. 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 106(1), 59–70 (2007)
6. Fei-Fei, L., Perona, P.: A Bayesian Hierarchical Model for Learning Natural Scene Categories. In: CVPR, vol. 2, pp. 524–531. IEEE Computer Society (2005)
7. Lazebnik, S., Schmid, C., Ponce, J.: A Maximum Entropy Framework for Part-Based Texture and Object Recognition. In: ICCV 2005, vol. 1, pp. 832–838. IEEE Computer Society, Washington, DC (2005)
8. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR 2006, vol. 2, pp. 2169–2178. IEEE Computer Society, Washington, DC (2006)
9. Leung, T., Malik, J.: Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. IJCV 43(1), 29–44 (2001)
10. Lowe, D.G.: Object Recognition from Local Scale-Invariant Features. In: ICCV, vol. 2, pp. 1150–1157. IEEE Computer Society, Washington, DC (1999)
11. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR 2007, pp. 1–8 (2007)
12. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In: ICML, pp. 807–814. ACM (2007)
13. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering Objects and their Localization in Images. In: Proceedings of ICCV, pp. 370–377 (2005)
14. Upton, G., Cook, I.: A Dictionary of Statistics. Oxford University Press (2004)
15. Vedaldi, A., Fulkerson, B.: VLFeat: An Open and Portable Library of Computer Vision Algorithms (2008), `http://www.vlfeat.org/`
16. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. In: CVPR, pp. 3539–3546. IEEE Computer Society, San Francisco (2010)
17. Winn, J., Criminisi, A., Minka, T.: Object Categorization by Learned Universal Visual Dictionary. In: ICCV, vol. 2, pp. 1800–1807. IEEE Computer Society (2005)
18. Yagnik, J., Strelow, D., Ross, D.A., Lin, R.S.: The power of comparative reasoning. In: ICCV, pp. 2431–2438. IEEE (2011)
19. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. IJCV 73(2), 213–238 (2007)