21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

BSIF: Binarized Statistical Image Features

Juho Kannala and Esa Rahtu
University of Oulu, Finland

Abstract

This paper proposes a method for constructing local image descriptors which efficiently encode texture information and are suitable for histogram based representation of image regions. The method computes a binary code for each pixel by linearly projecting local image patches onto a subspace, whose basis vectors are learnt from natural images via independent component analysis, and by binarizing the coordinates in this basis via thresholding. The length of the binary code string is determined by the number of basis vectors. Image regions can be conveniently represented by histograms of pixels' binary codes. Our method is inspired by other descriptors which produce binary codes, such as local binary pattern and local phase quantization. However, instead of heuristic code constructions, the proposed approach is based on statistics of natural images and this improves its modeling capacity. The experimental results show that our method improves accuracy in texture recognition tasks compared to the state-of-the-art.

1 Introduction

Local image descriptors have been under extensive investigation during the recent years and many important developments have been acquired. Nowadays local image descriptors are standard tools which provide image features for many computer vision applications. For example, local descriptors together with interest region detectors allow extraction and description of image regions which are used for wide baseline image matching, in e.g. multi-view reconstruction. Also, visual vocabularies, based on descriptors, are predominantly used in image retrieval and recognition tasks, like particular object retrieval and object class recognition. Further, local image descriptors are widely used in texture recognition and its applications.

There are various local image descriptors proposed in literature. For example, interest regions are typically represented using descriptors like SIFT [8], SURF [3], or BRIEF [4], and their variants. Further, there are several image descriptors originally designed for texture description and classification, such as local binary pattern (LBP) [9], local phase quantization (LPQ) [10, 2] and basic image features (BIF) [5]. Some of them, or their variants, have become increasingly popular also in other applications, like face identification [1, 2] and action recognition from videos [12].

In this paper, we propose an approach which is inspired by LBP and LPQ methodologies. These methods describe each pixel's neighborhood by a binary code which is obtained by first convolving the image with a set of linear filters and then binarizing the filter responses. The bits in the code string correspond to binarized responses of different filters. However, in contrast to earlier approaches, such as LBP and LPQ, we do not use a manually predefined set of filters but learn the filters by utilizing statistics of natural images.

Our texture and face recognition experiments show that the proposed approach gives a better overall performance than the popular and widely used comparison methods LBP and LPQ. Further, our results are obtained by using a fixed set of filters learnt from a small set of natural images, which shows that pre-learnt filters can be used for different applications. Nevertheless, unlike comparison methods, our approach provides an option of application-specific learning, which might be potentially useful for describing images that have unusual characteristics, such as certain medical images.

2 Method

Overview. Our method computes a binary code string for the pixels of a given image. The code value of a pixel is considered as a local descriptor of the image intensity pattern in the pixel's surroundings. Further, histograms of pixels' code values allow to characterize texture properties within image subregions. Thus, our descriptor can be used in texture recognition tasks in a similar manner as local binary patterns [9] or quantized local phase values [10].

The value of each element (i.e. bit) in our binary code string is computed by binarizing the response of a linear filter with a threshold at zero. Each bit is associated with a different filter and the desired length of the bit string determines the number of filters used. The set of filters is learnt from a training set of natural image patches by maximizing the statistical independence of the filter responses [6]. Hence, statistical properties of natural image patches determine the descriptors and therefore we call them *binarized statistical image features* (BSIF). The details of learning the linear filters follow [6] and they are briefly described below.

Details. Given an image patch X of size $l \times l$ pixels and a linear filter W_i of the same size, the filter response s_i is obtained by

$$s_i = \sum_{u,v} W_i(u,v)X(u,v) = \mathbf{w}_i^\top \mathbf{x}, \qquad (1)$$

where vector notation is introduced in the latter stage, i.e., vectors $\mathbf{w}$ and $\mathbf{x}$ contain the pixels of W_i and X. The binarized feature b_i is obtained by setting $b_i = 1$ if $s_i > 0$ and $b_i = 0$ otherwise. Given n linear filters W_i, we may stack them to a matrix $\mathbf{W}$ of size $n \times l^2$ and compute all responses at once, i.e., $\mathbf{s} = \mathbf{W}\mathbf{x}$ and we obtain the bit string $\mathbf{b}$ by binarizing each element s_i of $\mathbf{s}$ as above. Thus, given the linear feature detectors W_i, computation of the bit string $\mathbf{b}$ is straightforward. Also, it is clear that the bit strings for all image patches of size $l \times l$, surrounding each pixel of an image, can be computed conveniently by n convolutions.

In order to obtain a useful set of filters W_i we apply the ideas of [6] and estimate the filters by maximizing the statistical independence of s_i. In general, this approach provides good features for image processing [6]. Furthermore, in our case, the independence of s_i provides justification for the proposed independent quantization of the elements of the response vector $\mathbf{s}$. Thus, costly vector quantization, used e.g. in [14], is not necessary here for obtaining a discrete texton vocabulary.

However, in order to use standard independent component analysis (ICA) algorithms for estimating the independent components, one has to decompose the filter matrix $\mathbf{W}$ into two parts by

$$\mathbf{s} = \mathbf{W}\mathbf{x} = \mathbf{U}\mathbf{V}\mathbf{x} = \mathbf{U}\mathbf{z}, \qquad (2)$$

where $\mathbf{z} = \mathbf{V}\mathbf{x}$, and $\mathbf{U}$ is a $n \times n$ square matrix which will be estimated via ICA, and matrix $\mathbf{V}$ performs the canonical preprocessing, i.e. simultaneous whitening and dimension reduction of training samples $\mathbf{x}$ [6].

In short, the canonical preprocessing uses principal component analysis as follows. Given a training set of image patches randomly sampled from natural images, the patches are first made zero-mean (i.e. the mean intensity of each patch is subtracted) and then their dimension is reduced by keeping only the n first principal components which are further divided by their standard deviation to get whitened data samples $\mathbf{z}$. In detail, given the eigendecomposition $\mathbf{C} = \mathbf{E}\mathbf{D}\mathbf{E}^\top$ of the covariance matrix $\mathbf{C}$ of samples $\mathbf{x}$, the matrix $\mathbf{V}$ is defined by

$$\mathbf{V} = \left(\mathbf{D}^{-1/2}\mathbf{E}^\top\right)_{1:n}, \qquad (3)$$

where the main diagonal of $\mathbf{D}$ contains the eigenvalues of $\mathbf{C}$ in descending order, and $(\cdot)_{1:n}$ denotes the first n rows of the matrix in parenthesis.

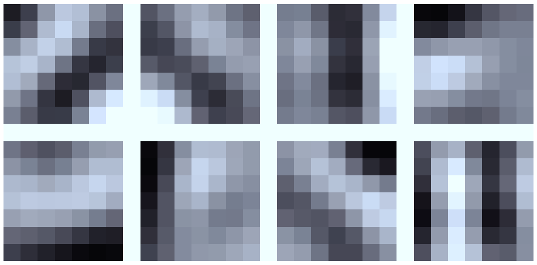


Figure 1. Learnt filters of size 9×9.

Then, given the zero-mean whitened data samples $\mathbf{z}$, one may use standard independent component analysis algorithms to estimate an orthogonal matrix $\mathbf{U}$ with which one yields the independent components $\mathbf{s}$ of the training data [7]. In other words, since $\mathbf{z} = \mathbf{U}^{-1}\mathbf{s}$, the independent components allow to represent the data samples $\mathbf{z}$ as a linear superposition of the basis vectors defined by the columns of $\mathbf{U}^{-1}$. Finally, given $\mathbf{U}$ and $\mathbf{V}$, one obtains the filter matrix $\mathbf{W} = \mathbf{U}\mathbf{V}$, which can be directly used for computing BSIF features.

Implementation. In all the experiments of this paper, we used the same filters learnt from a set of 13 natural images provided by the authors of [6]. Before random sampling of image patches for learning, the image intensities were normalized to have a zero mean and unit variance. As described above, there are two parameters in our BSIF descriptor: the filter size l and the length n of the bit string. We learnt the filters $\mathbf{W}$ using several different choices of parameter values, each set of filters was learnt using 50000 image patches. Learning was conducted by the three-stage process detailed in the previous subsection: (a) subtraction of the mean intensity of each patch, (b) dimension reduction and whitening via principal component analysis, and (c) estimation of independent components. The filters obtained with $l = 9$, $n = 8$ are illustrated in Figure 1. In addition, our implementation of BSIF features is available online.[1]

[1] http://www.cse.oulu.fi/Downloads/BSIF

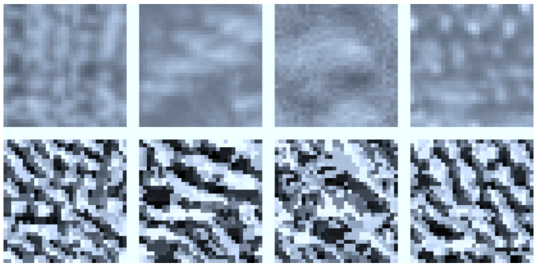


Figure 2. Samples from Outex database (top) and corresponding BSIF codes.

3 Experiments

In this section we will asses the proposed method in two canonical texture recognition applications: texture classification and face recognition. For texture classification, we use Outex and CUReT benchmark datasets, for which we adopt the train-test splits defined in Outex test suite 00002 and in [14], respectively. Outex dataset contains 24 texture types and 368 images per class, while the CUReT database consists of 61 textures and 205 images per class. Some examples and corresponding BSIF code images are shown in Figures 2 and 3. The classification is performed using nearest neighbor classifier with χ^2 distance metric.

The baseline for the experiments is formed by Local Binary Patterns (LBP) [9], BIF-column (BIFc) [5], and Local Phase Quantization (LPQ) [10] methods. We use standard 8 bit coding for LBP and LPQ, which results in a feature vector with 256 elements. For BIFc we apply the parameters described in the original paper. Hence, BIFc applies 4 different scales with 6 codewords per scale, resulting in descriptor with 1296 elements.

The classification accuracies with different filter sizes are reported in Figures 5(a) and 5(b). The results indicate that the proposed descriptor is consistently better than LBP or LPQ over a range of different filter sizes. Interestingly, already 7 bit version of BSIF outperforms all baselines in Outex and LPQ in CUReT. Compared to BIFc descriptor, the new method performs clearly better in Outex. In the case of CUReT databbase, BIFc gives 1.5 percent better accuracy than 8 bit BSIF, but the difference vanishes when the descriptor length is increased to be equivalent to BIFc. The images in CUReT are taken from several viewing angles and hence they contain some rotations. Since BIFc is rotation invariant, it is understandable that it performs well in this experiment. However, BSIF reaches the same performance, even if it was not particularly designed to be rotation invariant.

In the face recognition experiment we apply the Face Recognition Grand Challenge (FRGC) test 1.0.4 [11]. The FRGC database is divided into probe and gallery sets, which represent 152 subjects. There are exactly 1 gallery image and 2-7 probe images per subject, totaling 152 images in the gallery and 608 images in the probe set. The images in the gallery are acquired with good quality camera under controlled conditions, while probe images are taken with pocket digital camera in uncontrolled conditions. Therefore, probe images contain considerable variations in lightning, facial expression, and blur. Some examples and the corresponding BSIF code images are shown in Figure 4.

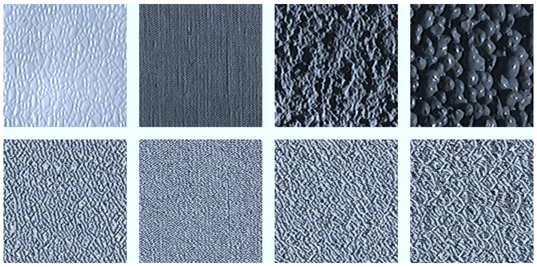


Figure 3. Samples from CUReT database (top) and corresponding BSIF codes.

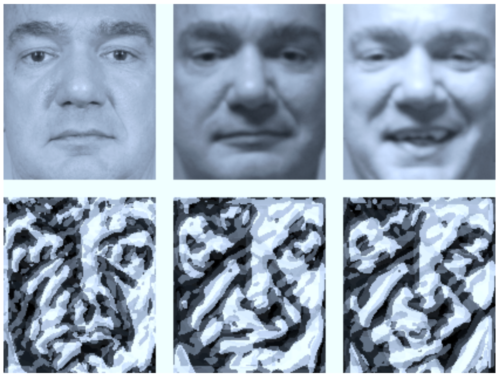
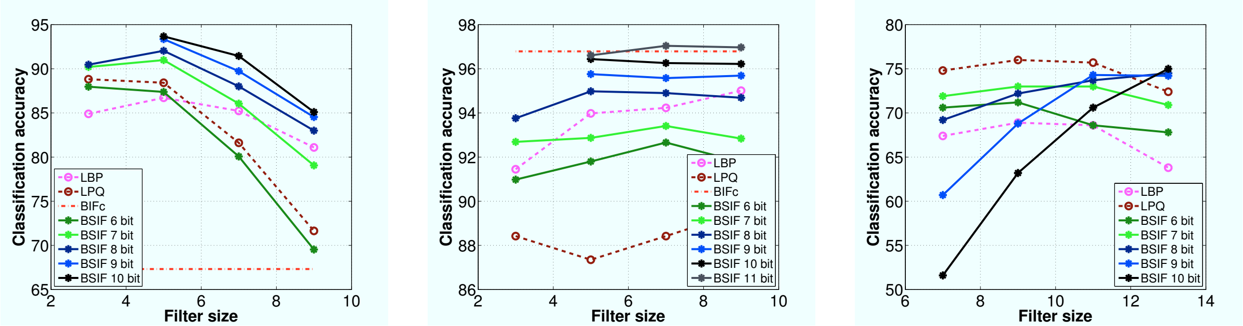


Figure 4. A sample gallery image and two probe images images from FRGC (top), and corresponding BSIF codes. The blur in the probe images is clearly observable.

For the recognition we apply the procedure described in [1]: the face image is first divided into 8 × 8 non-overlapping rectangular regions and a given descriptor is computed independently within each of these regions. Finally, the descriptors from different regions are concatenated to a global description of the face. The classification is performed using nearest neighbor classifier with χ^2 distance metric. Also, as in [2], we apply the illumination normalization proposed in [13].

The average face recognition accuracies are shown in Figure 5(c). The effect of blurring is clearly observable in the BSIF results. If the number of filters is increased while keeping the size fixed, more high frequency information will be included into the descrip-



(a) Outex

(b) CUReT

(c) FRGC

Figure 5. The average classification accuracies for different benchmark databases. Note: Since BIFc is a multiresolution method, the corresponding results are indicated by horizontal lines.

tor. Since high frequencies are particularly disturbed by blurring, it will affect the performance of such combinations. Hence, we need to enlarge the filter size with respect to the code length in order to cope with blurring. The 6 and 7 bit version of BSIF give good results already with 7 × 7 filter, losing only 3 percents to the blur invariant LPQ. When increasing the descriptor length to 8 bit and above, we reach approximately the same performance as LPQ with 13 × 13 filters. Furthermore, already 6 bit BSIF outperforms LBP over the wide range of filter sizes.

4 Conclusion

In this paper we presented a method for constructing local texture descriptors, based on independent component analysis and efficient scalar quantization scheme. The proposed algorithm results in a binary code for each pixel, which can be subsequently used to construct a convenient histogram representation for image areas. The key idea in the approach is to apply learning, instead of manual tuning, to obtain statistically meaningful representation of the data, which enables efficient information encoding using simple element-wise quantization. Learning provides also an easy and flexible way to adjust the descriptor length and to adapt to applications with unusual image characteristics.

In texture classification experiments, the new BSIF descriptor clearly outperformed the state-of-the-art baseline methods with equivalent descriptor length. Interestingly, also more compact versions of BSIF resulted in better accuracy than some of the baselines. Moreover, the proposed method was tested in a face recognition application, where it resulted in equal performance to the state-of-the-art descriptors. Although some of the test images were blurred and imperfectly aligned, BSIF resulted in similar performance as specifically designed rotation and blur invariant methods. This indicates the tolerance of BSIF descriptor to image degradations appearing in practice.

References

[1] T. Ahonen et al. Face description with local binary patterns: Application to face recognition. *TPAMI*, 2006.

[2] T. Ahonen et al. Recognition of blurred faces using local phase quantization. In *ICPR*, 2008.

[3] H. Bay et al. Speeded-up robust features (SURF). *CVIU*, 2008.

[4] M. Calonder et al. BRIEF: Binary robust independent elementary features. In *ECCV*, 2010.

[5] M. Crosier and L. Griffin. Using basic image features for texture classification. *IJCV*, 3(88):447–460, 2010.

[6] A. Hyvärinen et al. *Natural Image Statistics*. Springer, 2009.

[7] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 2000.

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

[9] T. Ojala et al. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 7(24):971–987, 2002.

[10] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In *ISP*, 2008.

[11] P. Phillips et al. Overview of the face recognition grand challenge. 2005.

[12] M. Pietikäinen et al. *Computer Vision Using Local Binary Patterns*. Springer, 2011.

[13] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *AMFG*, 2007.

[14] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1):61–81, 2005.