

Constructing Local Binary Pattern Statistics by Soft Voting

Juha Ylioinas, Xiaopeng Hong, and Matti Pietikäinen

Center for Machine Vision Research, P.O. Box 4500,
90014 University of Oulu, Finland

Abstract. In this paper we propose a novel method for constructing Local Binary Pattern (LBP) statistics for image appearance description. The method is inspired by the kernel density estimation designed for estimating the underlying probability function of a random variable. An essential part of the proposed method is the use of Hamming distance. Compared to the standard LBP histogram statistics where one labeled pixel always contributes to one bin of the histogram, the proposed method exploits a kernel-like similarity function to determine weighted votes contributing several possible pattern types in the statistic. As a result, the method yields a more reliable estimate of the underlying LBP distribution of the given image. In overall, the method is easy to implement and outperforms the standard LBP histogram description in texture classification and in biometrics-related face verification. We demonstrate that the method is extremely potential in problems where the number of pixels is limited. This makes the method very promising, for example, in low-resolution image description and the description of interest regions. Another interesting property of the proposed method is that it can be easily integrated with many existing LBP variants that use label statistics as descriptors.

1 Introduction

Image appearance descriptors aim to extract features that effectively describe the salient class information in the images. Besides, they are often required to cope with within-class variations due to degradations caused by, for example, varying imaging conditions. Image descriptors play a key role in almost all computer vision applications concerning detection, recognition and classification, and therefore, they have been one of the most important topics in the field. According to one aspect, the development of image descriptors can be seen as an experimental science guided by best practices more than by solid theory [1].

Different methods have been proposed in literature for image appearance description, such as Gabor wavelets, Histograms of Oriented Gradients (HOG), and Local Binary Patterns (LBP). Usually, the methods are first benchmarked on texture classification and then on the rest of the potential applications. Texture can be defined as the description of variation of data on scales smaller than the scales of interest. It is a fundamental property of natural images and has gained a lot of interest in the field of image analysis and computer vision.

Widely used local binary patterns is a type of image description. The method is based on a grayscale invariant texture operator, derived from a general definition of texture in a local neighborhood. The simplest form of LBP operates on image pixels by thresholding their 3×3 neighborhood with the center value and treating the subsequent 8-bit binary string as a decimal number. The histogram of these labels (or codes) is then used as an image description. Besides being computationally simple, local binary patterns have been shown to be highly discriminative. They are shown to be very successful in encoding image appearance in a number of computer vision problems [2].

The basic form of the LBP as well as many of its recently developed derivatives suffer from two fundamental problems. The first one arises due to vector quantization which may lead to instability in the sense that a small change in the input, due to noise for example, causes large errors in the output. Another serious issue is linked with the problems inherent to histograms, the commonly used statistic for LBP-based image description. The histogram is a simple form of density estimation, but indeed has several drawbacks. Regarding to LBP histograms, the main concern is the number of pixels as a small number of those may fail to provide enough information to build up a discriminative histogram description. Unarguably, low amount pixels leads to a low number of LBP occurrences which may further lead to sparse and unstable descriptions. In this paper, we tackle these two problems by introducing a novel method for constructing a more robust estimate of the underlying LBP distribution of the given image. Our approach is inspired by the function estimation theory, especially by the Kernel Density Estimation (KDE) methods. An enabling factor for using a KDE-like method in the context of local binary patterns is the use of Hamming distance. Compared with the conventional way of binning LBP occurrences for image description, our method uses a voting function to place votes according to the Hamming distance between the detected pattern and all the possible outcomes in the LBP space at hand. As a result, the method yields more stable and discriminative statistics for image description. Especially, the proposed voting scheme works better than the histogram method in situations where only a low amount of pixels is available.

The rest of this paper is organized as follows. Section 2 describes the LBP methodology and discusses some of the most important extensions and recently introduced derivatives. We introduce our proposed soft voting scheme in Section 3 and then provide extensive performance evaluation and comparative analysis in Section 4. Conclusions are drawn in Section 5.

2 Local Binary Patterns

Local Binary Patterns (LBP) [3] offer a powerful means for grayscale invariant texture description and has become highly popular in various computer vision applications.

The first form of LBP operates on 3×3 pixel neighborhoods. Later, it was extended to arbitrary circular neighborhoods with any radius and number of sampling points using bilinear interpolation at sub-pixel coordinates leading to the notation given by

$$\text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} t(g_p - g_c)2^p, \quad (1)$$

where $g_c = I(x_c, y_c)$, g_p refers to the gray values of P equally spaced sampling points in a circle of radius R , and t defines a thresholding function with $t(x) = 1$ if $x \geq 0$ and $t(x) = 0$ otherwise. LBP codes can be regarded as texture primitives including different types of curved edges, spots, flat areas, and so on. The occurrences of the LBP codes in the image are usually collected into a histogram. The classification can then be performed by computing simple histogram similarities. The efficiency of the LBP method has later inspired the research community to invent a wide variety of extensions and tuned derivatives for specific applications.

2.1 LBP Extensions and Derivatives

Among the most important early extensions are the rotation invariant version of the LBP operator and the feature selection process, better known as *uniform patterns*. Rotation invariant (*ri*) local binary patterns can be achieved by circularly rotating the binary code into its minimum value. The other extension, *uniform patterns (u2)*, was inspired by the fact that some binary patterns occur more commonly than others. Shortly, uniform patterns are those that contain at most two bitwise transitions from 0 to 1 or vice versa. Finally, by using these extensions, for example the original $\text{LBP}_{8,1}$ histogram with 256 dimensions can be reduced in three different ways, namely using *ri*, *u2*, or *riu2*, to a feature space with 36, 59, and 10 dimensions, respectively.

A major drawback of the LBP operator lies in the use of only signs of local pixel differences. As a result, the operator fails to take advantage of the contrast information available in the given image. To exploit contrast information, the original form of the LBP operator was slightly revised leading to a local contrast measure, and further, to a so called VAR measure [2,3]. The most recently proposed LBP-based method of measuring contrast is the magnitude coding of the so called completed modeling of LBP (CLBP) [9]. In that the magnitudes of the local pixel differences are coded replacing the threshold function in Eq. 1 to $t(m_p, c)$, where m_p is the magnitude of local pixel difference and c is a predetermined threshold value usually set as the mean value of local pixel differences in the whole image.

There are some other remarked issues in the original LBP concerning, for example, the inability to capture large-scale structures or anisotropic information that may be the dominant features of some textures. To solve the above problems, there are dozens of potential derivatives available. An extensive review of LBP methods can be found in [2]. Although forming a very promising compilation of methods to describe texture, it appears that many of these LBP variants

inherit problems due to vector quantization and the number of label occurrences, causing problems especially with the usual way of describing textures, namely histograms. This is partly due to the inherent characteristics of the histogram method, but also due to the finite number of pixels available in the given image.

One of the first works to mitigate the problem of sparse and unstable LBP histograms was presented by Ahonen and Pietikäinen [4]. The key idea of their soft histogram method was to use fuzzy membership functions to determine contributions of each distinct LBP neighborhood in the image to all bins in the final LBP histogram. Another work, proposed by the same authors [5], makes use of KDE for producing the LBP distribution per pixel in an image. However, the method was particularly designed for face recognition to tackle the downsides of the block-based facial description.

In [6], Yang *et al.* proposed to produce LBP histograms using the uniform $u2$ mapping with a Hamming distance constraint. Instead of assorting all uniform patterns to a single bin, they incorporated non-uniform patterns into uniform patterns by minimizing the Hamming distance between them. By the Hamming distance constraint, one is able to collect more information, but the downside is that the useful information provided by non-uniform patterns may still not be optimally utilized. As some textures (e.g. facial images) may retain more discriminative information in some non-uniform pattern than in some uniform ones put together, it makes unreasonable to connect them. However, acting on $u2$ -mapped LBP space by design the method has potential.

A fairly new method based on denser sampling of LBP codes can also be seen as a solution to the inadequate descriptors due to unstable histograms [7]. The idea there is to sample images so that the center of the LBP operator's neighborhood scans also pixel corners. Undoubtedly, this kind of denser sampling scheme enables one to extract more LBP codes from the given image, which may further yield statistically more robust estimates of the underlying LBP distribution. A feasible way for practically implementing the dense scheme, although not remarked in the original paper, would be to first upsample the original image of a size $M \times N$ pixels to $(2 * M - 1) \times (2 * N - 1)$ using bilinear interpolation (without any pre- or post-filtering), at the same time doubling the radius R of the used $LBP_{P,R}$ operator to $2 * R$. Actually, by this way one is able to extract even more LBP codes than in the original dense sampling scheme described in [7].

The above described methods (soft histograms, KDE-based LBP distributions at specific xy -locations, Hamming distance constraint, and dense sampling) provide all potential means to tackle the issue present in using histograms for estimating LBP distributions. Still, they bring some new disadvantages, whether they are computationally too demanding, too application-specific, or otherwise not discriminative enough. As an alternative to these we next introduce a novel method for constructing more reliable estimates of the underlying LBP distribution.

3 Soft Voting

The major drawback of all LBP methods using vector quantization is the instability in the sense that even small variations in the input can cause large variations in the output. This undesirable property, not taken into consideration while using histograms, is such that even a one-level-decrease in the intensity of a single pixel may result in several alterations in the adjacent LBP labels in the same region. Another concern is the image size, as a low number of pixels results in a low number of LBPs which finally may not yield stable histograms.

To tackle these issues, we propose a novel method for producing more robust LBP statistics. Instead of voting to a single bin as the standard LBP histogram does, our method assigns weighted votes, not for a single bin, but for several candidates. The weighted votes are placed based on the similarity between the detected and candidate patterns covering all possible outcomes of the used $LBP_{P,R}$ operator.

Let us first have an LBP labeled image $I_{LBP_{P,R}}$ and one labeled pixel value l at a position (x, y) so that $l = LBP_{P,R}(x, y)$. If we then consider the label as a binary word so that $l \in B^P$ (P -dimensional binary space with $B \in \{0, 1\}$), the voting weight, given l , to the pattern type l_j , $j = 0, \dots, 2^P - 1$, is defined by

$$w(l_j|l; \beta) = K_\beta(d(l_j, l)), \quad (2)$$

where $K_\beta(\cdot)$ is a kernel function including the appropriate parameters β . Further, from the observation that LBP words lie in the P -dimensional binary space the similarity d is set as the Hamming distance defined by

$$d(l_j, l) = \sum_{i=1}^P (l_j^i \oplus l^i), \quad (3)$$

where l^i corresponds to the i th bit of the code l and \oplus denotes bitwise exclusive OR operation. The chosen kernel can be any of the ones commonly in use to estimate the probability density function of a random variable including Gaussian, uniform, triangular, Epanechnikov, Cosine, or the one proposed in [11]. Moreover, as the range of possible distances, using a particular $LBP_{P,R}$, is given by a set of positive integers $\{0, 1, 2, \dots, P\}$, the number of generated weights w is bounded to P . Finally, instead of a histogram, an alternative estimate of the relative frequency of a pattern type l_j in an image is the summation of all weights given by

$$\hat{f}_\beta(l_j) = \sum_{x,y} w(l_j|l; \beta), \quad j = 0, \dots, 2^P - 1. \quad (4)$$

Evidently, the resulting statistic can be characterized as a kernel density estimate of the underlying LBP distribution enabled by the Hamming distance between two LBP codes.

Kernel density estimation methods belong to the class of non-parametric models to estimate the probability density function of a random variable, based on a finite data sample [10]. Our assumption is that the potential solution to the

sparse and unstable LBP histograms, especially in a situation where the number of pixels is low, can be found from a KDE-based method, just like our voting scheme is.

In our method, the selection of the kernel specifies the weighting in the voting function. Constructing the statistic it simply means the more similar patterns the bigger the given vote. Basically, the selection can be made from all kernels commonly in use to estimate the probability density function, but the votes can also be handcrafted. In our experiments, we use Gaussian kernels, so that the kernel becomes $K_\sigma(d) = (\sigma\sqrt{2\pi})^{-1} \exp(-d^2/(2\sigma^2))$. Fig. 1 illustrates some possible voting functions generated by a Gaussian kernel with a chosen σ .

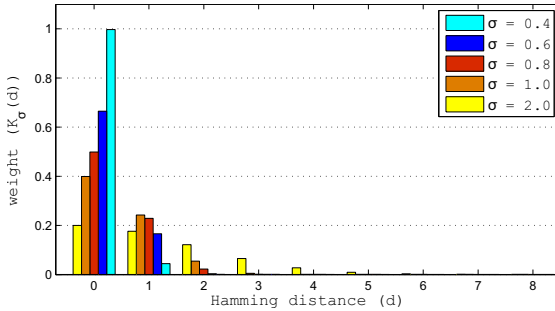


Fig. 1. Example votes given by a Gaussian kernel using a specific σ . This kind of voting function is suitable for 8 bit binary codes, such as ones generated by the LBP_{8,2} operator.

Different from soft histograms [4], our method approximates LBP distributions using weighted votes determined by a suitable kernel and the pattern similarity. To begin with, our method does not make use of any complicated (e.g. fuzzy membership functions or other mappings) methods integrated into the LBP operator to calculate contributions to different bins, but rather calculates the original labels given by the pure LBP operator to distribute weighted votes finally producing a smoother estimate of the true underlying LBP distribution. Furthermore, unlike in [5], where pixelwise LBP distributions using KDE is proposed, our method does not estimate distributions in specific xy -locations, but rather outputs a total distribution of LBP labels for the given image.

The use of the Hamming distance is needed as local binary patterns are not actually defined in the Euclidean space but rather in the multidimensional binary space. Unlike in [6], we use the Hamming distance to define points among which the weighted votes are placed. Clearly, our method makes good use of the whole LBP space defined by the number of samples P and the radius of the neighborhood R .

It must be noted, that the standard LBP histogramming (or binning) is a special case of our proposed voting based estimation of the true underlying LBP distributions. From the perspective of voting, in the standard LBP histogram

a weight of 1 is given to the pattern type similar as the detected pattern l and 0 to the rest, more formally, using a kernel $K_\delta(d(l_j, l)) = 1$ if $d(l_j, l) = 0$ and $K_\delta(d(l_j, l)) = 0$ otherwise. The soft voting method is also universal in the sense that it can be used with different LBP variants that use label statistics as descriptions.

4 Experimental Analysis

To evaluate the performance of our proposed soft voting method for constructing LBP statistics, we conducted experiments on two different problems using publicly available databases commonly used by the research community. We carried out experiments in texture classification and in biometrics-related face verification. In texture classification, the aim is to assign an unseen texture sample into one of the predefined classes, whereas in face verification, given two facial images, one needs to determine if they contain the same individual.

In both problems, we analyzed the effect of the proposed soft voting scheme on the completed modeling of LBP operator [9]. We compared separate and concatenated CLBP sign and magnitude histograms against the ones constructed by soft voting. We also experimented on combining the dense LBP scheme [7] and soft voting. In both problems, we experimented Gaussian kernels as a basis for determining the votes. Moreover, in texture classification, we further made an experiment where we illustrate the functionality of our method in a so called limited-sample-size scenario where the available information, i.e. the size of the image, dramatically drops.

From now on, we refer to CLBP sign and magnitude components as *signs* and *magns*, whereas the dense sampling scheme is referred as dense(\cdot).

4.1 Texture Classification

In texture classification, we conducted experiments on the CURET database containing 61 texture classes collected from real-world surfaces with varying properties. In the database, each texture class contains 205 images captured at several viewpoints and under different illumination orientations. We considered the publicly available cropped dataset [12] having a total of 5,612 images with 92 samples for each class. Fig. 2 depicts some examples from the CURET database. In our experiments, we use only gray-scale versions of the original cropped ones.

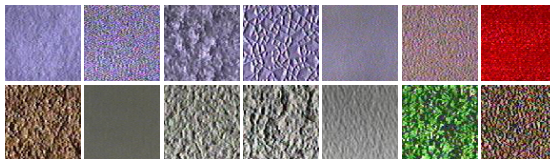


Fig. 2. Texture image samples from the CURET dataset

In our experiments, we randomly selected 46 images from each class for training whereas we used the remaining 46 samples for testing. We tested the soft voting method in two scenarios. In the first one we took the original CURET samples, size of 200×200 pixels, and computed the LBP histogram and voting based descriptions out of them. Considering a rather big size of the original samples we further performed the second experiment. In that we wanted to validate our hypothesis about the feasibility of our KDE-motivated voting scheme especially in limited-sample-size scenarios where the number of available information, i.e. number of pixels, is much lower. Therefore, for the second experiment, we took only the inner part of the cropped CURET images (`img(80:120,80:120)`), a patch size of 41×41 pixels, computed the LBP descriptions and performed the same experiment. In both experiments, we used the nearest neighbor (NN) classifier with the Euclidean distance metric. For obtaining the results, we analysed the average classification performance of ten permutations.

In our results, we compare the classification results using raw LBPs, but also *u2* and *ri*-mapped versions. For *magns* we ended up using only the *ri*-mapping as we think that measures of contrast should always be rotation invariant. However, it is highly important that while using whatever mapping it is to be applied only after constructing the statistic in the whole LBP space. Moreover, while using voting and *u2*-mapping we do not make use of the last 59th dimension corresponding to all non-uniform patterns. For voting, we found $\sigma = 0.6$ and $\sigma = 0.5$ to perform best for *signs* and *magns*, respectively.

The obtained results on the original CURET images, summarized in Table 1, show quite equally balanced results among the *signs* and *magns* descriptions. A conclusion, although a very careful one, is that the voting scheme used together with standard sampling performs the best. To compare voting and dense sampling as separate methods, it seems that the voting scheme performs better. The final conclusion might be that the histogram method is good enough for the estimation of the true underlying LBP distribution for image description. The added computational workload while using dense sampling and voting, together or separately, may not be worthwhile enough as the improvement in accuracy seems to be only moderate.

In the limited-sample-size scenario, in turn, we found quite interesting results, summarized in Table 2. As described, with the images size of 41×41 pixels we wanted to ensure a much lower amount of LBP codes of which the statistical description is formed, using the same parameters as in the previous experiment. Based on the results, it is evident that the standard histogram method starts suffer from the limited amount of information. In comparison to the voted descriptions, the difference in accuracy is absolutely tremendous. For example, using standard *signs*, without any mapping, together with voting yields an improvement over 20 percentage points compared to the standard *signs* histogram. Using dense sampling and voting the improvement is even better. The results clearly demonstrate that histograms can be very unstable while the amount of information is limited, whereas the KDE-based voted statistic may offer a powerful means to compensate in limited-sample-size scenarios of this kind.

Table 1. Mean accuracy on the original cropped CURET images (200×200) where (a) represents results using standard circular LBP (denoted as *signs* according to the completed modeling), (b) represents results using the magnitude component (denoted as *magns*), and (c) results using the concatenated descriptor. In this experiment, we used the (8,2)-neighborhood with raw (-), *u2* and *ri* mapped *signs*; and raw (-) and *ri* mapped *magns*. For the concatenated descriptor, we used *u2* for signs and *ri* for magns.

| (a) | | | | (b) | | |
|--------------------------------|-------|-----------|-----------|--------------------------------|-------|-----------|
| descriptor | - | <i>u2</i> | <i>ri</i> | descriptor | - | <i>ri</i> |
| <i>signs</i> | 94.38 | 93.71 | 85.68 | <i>magns</i> | 85.87 | 82.62 |
| <i>signs</i> + voting | 95.51 | 94.25 | 89.01 | <i>magns</i> + voting | 84.07 | 82.10 |
| dense(<i>signs</i>) | 94.81 | 92.74 | 88.90 | dense(<i>magns</i>) | 87.64 | 86.25 |
| dense(<i>signs</i>) + voting | 94.54 | 93.23 | 89.30 | dense(<i>magns</i>) + voting | 84.87 | 84.53 |

| (c) | |
|--------------------------------------|----------|
| descriptor | accuracy |
| <i>signs_magns</i> | 95.37 |
| <i>signs_magns</i> + voting | 95.71 |
| dense(<i>signs_magns</i>) | 95.22 |
| dense(<i>signs_magns</i>) + voting | 95.49 |

4.2 Face Verification

To evaluate performance on face verification we used the publicly available Labeled Faces in the Wild (LFW) database [8]. In our experiments, we use the LFW-a [1] version where all the original LFW images are aligned using a commercial alignment system.

The database consists of 13,233 images of 5,749 people, which are organized into 2 views. View 1 is a development set of 2,200 pairs for training and 1,000 pairs for testing, on which to build models and choose features; and View 2 is a 10-fold cross-validation set of 6,000 pairs, on which to evaluate final performance. We use View 1 for high-level model selection and report performances on each of the folds in View 2. Furthermore, there are two proposed benchmark protocols in the LFW from which we report results on the one called "image restricted training".

We used croppings of 84×74 (`img(94:177, 89:162)`) from the original 250×250 pixels size of images. Fig. 3 shows some exemplars of the cropped images. Once cropped, local descriptors are then extracted from uniformly distributed patches across the face. The face image is first divided into a set of N overlapping patches of a size 14×14 . Each patch overlaps its vertical and horizontal neighbors by four pixels. With the used image size, this results in a total of 56 patches.

In the dense mode, instead of the original scheme [7], we explored the functionality of the above described alternative implementation for the dense sampling combining it with our soft voting scheme. Therefore, we first upsample cropped images to 167×147 correspondingly magnifying the block window size to 27×27 and the overlapping to seven pixels. As in the standard mode we set the $R = 2$

Table 2. Mean accuracy on the CURET under the limited-sample-size scenario (41×41) where (a) represents results using *signs*, (b) represents results using *magns*, and (c) results using the concatenated descriptor. Parameters set as in the previous experiment (see Table 1).

| (a) | | | | (b) | | |
|--------------------------------|-------|-----------|-----------|--------------------------------|-------|-----------|
| descriptor | - | <i>u2</i> | <i>ri</i> | descriptor | - | <i>ri</i> |
| <i>signs</i> | 50.63 | 53.14 | 29.54 | <i>magns</i> | 44.75 | 30.76 |
| <i>signs</i> + voting | 72.76 | 66.87 | 40.82 | <i>magns</i> + voting | 57.20 | 38.02 |
| dense(<i>signs</i>) | 64.16 | 63.46 | 37.56 | dense(<i>magns</i>) | 56.99 | 38.70 |
| dense(<i>signs</i>) + voting | 76.92 | 69.89 | 44.74 | dense(<i>magns</i>) + voting | 65.45 | 64.80 |

| (c) | |
|--------------------------------------|----------|
| descriptor | accuracy |
| <i>signs_magns</i> | 64.21 |
| <i>signs_magns</i> + voting | 76.91 |
| dense(<i>signs_magns</i>) | 73.82 |
| dense(<i>signs_magns</i>) + voting | 79.80 |

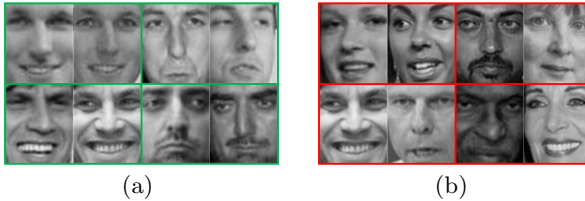


Fig. 3. Cropped exemplars from the LFW database where (a) shows pairs from same and (b) from not same individuals

for the given operator, in the dense mode the radius must be doubled yielding $R = 4$. In both modes, we use the *u2*-mapping for *signs* and the *ri*-mapping for *magns*. Again, mappings are applied only after constructing the LBP statistic in the whole LBP space. Also, using *u2*-mapping the last 59th dimension was not taken into account. In the experiments, the alternative dense scheme is referred as $\widehat{\text{dense}}(\cdot)$.

To make a decision whether two images are coming from the same person, we use Euclidean distance between the corresponding image descriptions. Before calculating the distance, we perform min-max normalization to the training and testing feature vectors in order to minimize the sensitiveness of the distance function to those dimensions that exhibit the largest range in values. Finally, to classify the resulting distance scores we make use of linear SVM. As suggested by the authors of the LFW database, we perform high-level model selection using the View 1 and then report the performance on View 2 showing mean accuracy and standard errors. Based on the model selection on View 1, we found $\sigma = 0.6$ and $\sigma = 0.5$ to perform the best for *signs* and *magns*, respectively.

The results, shown in Table 1, indicate that the voting scheme always improves the accuracy no matter using it in the standard sampling or in the dense

Table 3. Mean (\pm standard error) accuracy on the LFW-a face verification benchmark (View 1). (a) Represents results using *signs*, (b) represents results using *magns*, and (c) represents results using the concatenated descriptor. In all cases, we used the (8,2)-neighborhood.

| (a) | | (b) | |
|---|------------------|---|------------------|
| descriptor | accuracy | descriptor | accuracy |
| <i>signs</i> | 69.10 \pm 0.50 | <i>magns</i> | 64.90 \pm 0.72 |
| <i>signs</i> + voting | 69.37 \pm 0.49 | <i>magns</i> + voting | 67.77 \pm 0.73 |
| $\widehat{\text{dense}}(\textit{signs})$ | 69.48 \pm 0.40 | $\widehat{\text{dense}}(\textit{magns})$ | 66.75 \pm 0.63 |
| $\widehat{\text{dense}}(\textit{signs})$ + voting | 70.05 \pm 0.53 | $\widehat{\text{dense}}(\textit{magns})$ + voting | 68.03 \pm 0.64 |

| (c) | |
|--|------------------|
| descriptor | accuracy |
| <i>signs_magns</i> | 68.97 \pm 0.67 |
| <i>signs_magns</i> + voting | 70.98 \pm 0.52 |
| $\widehat{\text{dense}}(\textit{signs_magns})$ | 70.33 \pm 0.51 |
| $\widehat{\text{dense}}(\textit{signs_magns})$ + voting | 70.92 \pm 0.50 |

sampling mode. It is also visible, as hinted in [9], that the sign component is more discriminative compared to the magnitude. The most interesting fact is that the combination features *signs_magns* with voting outperforms its dense counterpart. Moreover, the improvement to the standard *signs_magns* combination is over two percentage points.

5 Conclusion

In this paper, we introduced a novel soft voting scheme for constructing robust statistics for LBP-based image appearance description. Inspired by the kernel density estimation, the method determines a set of weighted votes contributing several possible pattern types yielding a more reliable estimate of the underlying LBP distribution of the given image.

Compared to the conventional way of binning LBP occurrences or code words, our method uses a voting function to place votes for the statistic according to the similarity of the detected pattern and all the possible candidates in the $LBP_{P,R}$ space at hand. Pattern similarity is measured by the Hamming distance. As a result, the uncertainty in each LBP code is taken better into consideration making the description less sensitive to small changes in the input image. Another pleasing property of the method is that the resulting description may no longer suffer from the reduced amount of data, a common issue with LBP histograms constructed from small image patches.

In our experiments, we tested the proposed method in two major problems, namely texture classification and face verification. Based on the results, the voting scheme seems to be promising especially while using small images. In our limited-sample-size scenario, the voting based description outperformed the

standard histogram description with a huge margin. It is worth noting that the method generalizes the standard LBP histogram description and thereby takes shape as an important element for LBP image description methods.

For future directions, we plan to do more comprehensible investigation of the capabilities of the soft voting method with different kernel selections and parameter tunings. We also plan to test the voting scheme with other image description methods outputting pixelwise binary codes for histogram based image descriptions.

References

1. Wolf, L., Hassner, T., Taigman, Y.: Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE TPAMI* 33(10), 1978–1990 (2011)
2. Pietikäinen, M., Hadid, A., Zhao, G., Ahonen, T.: *Computer Vision Using Local Binary Patterns*. Springer (2011)
3. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI* 24(7), 971–987 (2002)
4. Ahonen, T., Pietikäinen, M.: Soft histograms for local binary patterns. In: *Proc. Finnish Signal Processing Symposium* (2007)
5. Ahonen, T., Pietikäinen, M.: Pixelwise local binary pattern models of faces using kernel density estimation. In: Tistarelli, M., Nixon, M.S. (eds.) *ICB 2009*. LNCS, vol. 5558, pp. 52–61. Springer, Heidelberg (2009)
6. Yang, H., Wang, Y.: A LBP-based face recognition method with hamming distance constraint. *Image and Graphics*. In: *Fourth International Conference on Image and Graphics, ICIG 2007*, pp. 645–649 (2007)
7. Ylioinas, J., Hadid, A., Guo, Y., Pietikäinen, M.: Efficient image appearance description using dense sampling based local binary patterns. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) *ACCV 2012, Part III*. LNCS, vol. 7726, pp. 375–388. Springer, Heidelberg (2013)
8. Huang, G., Ramesh, M., Berg, T., Learned-Miller, E.: *Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments*. UMass Amherst Technical Report 07-49 (October 2007)
9. Guo, Z., Zhang, L., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. *IEEE TIP* 19(6), 1657–1663 (2010)
10. Bishop, C.M.: *Pattern recognition and machine learning*. Springer (2006)
11. Aitchison, J., Aitken, C.: Multivariate binary discrimination by the kernel method. *Biometrika* 63(3), 413–420 (1976)
12. Dana, K.J., van Ginneken, B., Nayar, S.K., Koenderink, J.J.: Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics* 18, 1–34 (1999)