Using local binary patterns for object detection in images

Karel Petranek *, University of Hradec Kralove, Hradec Kralove, Czech Republic.
Pavel Janecka, University of Hradec Kralove, Hradec Kralove, Czech Republic.
Jan Vanek, Faculty of Military Health Sciences, University of Defence, Hradec Kralove, Czech Republic.

Abstract

The article discusses a texture operator called Local Binary Patterns (LBP) and its applications in image processing and object detection. We provide a description of the algorithm for computing LBP together with a rationale for using LBP as a feature for object detection and image recognition. Based on the algorithm we show that LBP features have a low computational overhead compared to more complicated image features such as the commonly used SIFT or SURF features or neural network based approaches because they exploit the use of extremely fast bitwise and integer operations of the CPU. We demonstrate that LBP is robust to changes in brightness, contrast, image rotation, image scale. We develop two enhancements for LBP that improve its resistance to camera noise and enhance the discriminative power of LBP when it is used as a feature for machine learning algorithms. We present the results on a challenging real-world object detection task.

Keywords: computer vision, object detection, local binary patterns.

*ADDRESS FOR CORRESPONDENCE: **Karel Petranek,** University of Hradec Kralove, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic. *E-mail address:* karel.petranek@uhk.cz

1. Introduction

Texture analysis is one of the basic steps of image analysis. Textures can contain structural, lighting and view independent information about the scene. This information can be used as an input for segmentation [1, 2], scene reconstruction [3, 4] or machine learning algorithms [5, 6, 7].

Traditionally, textures have been processed by transforming the 2D signal into the frequency domain using Fourier or more generally wavelet transforms and performing analysis in the frequency domain [8, 9, 10]. There is a computational disadvantage of this approach because the transform can be costly and may need to be computed twice – into the frequency domain and back, depending on the application.

Local Binary Patterns (LBP) operator [11] is a texture operator that directly manipulates the raster information of an image. The image does not need to be transformed into another domain which saves computational resources. The output of the LBP operator is an integer representing texture type at the given image pixel. This integer can be then directly used for further analysis in the subsequent higher-level algorithms. The LBP operator also allows for an easy compensation for image rotation, scale, brightness and contrast transformations [12].

In the first part of the paper we provide a summary of existing LBP approaches and describe the algorithms for computing LBP. In the second part of the paper we propose two generalizations of LBP that allow for a more fine-grained noise/information content trade-off.

2. Computing LBP

The LBP operator assigns an integer value to every pixel of the input image. The integer value is then interpreted as a texture class at the given pixel. The individual bits of the LBP integer are derived from local gradient information around every pixel as shown in
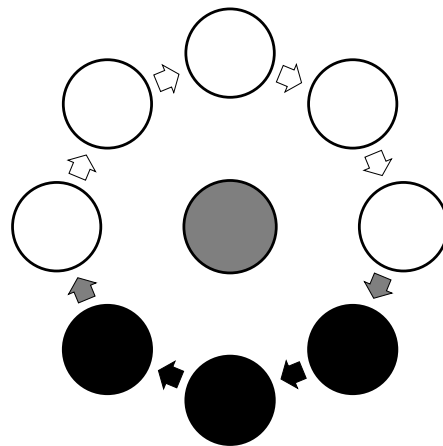

Figure 1. First, a circular neighbourhood is analysed and intensities of every point in this neighbourhood are compared to the intensity of the point in the circle's centre. Points with a lower intensity than the centre are assigned the value of 0, points with a higher intensity are assigned 1. The resulting string of zeroes and ones is interpreted as an integer that represents the texture class at the centre. More formally, the LBP value at the given image point (x, y) can be defined as:

$$LBP_{p,r}(x, y) = \sum_{i=0}^{p-1} 2^i \text{thr}\left(I(x_i, y_i) - I(x, y)\right)$$

where $LBP_{p,r}(x, y)$ is the LBP value at $[x, y]$, p the number of points in the circular neighbourhood, r radius of the neighbourhood, $I(a, b)$ image intensity at the coordinates a, b

and $\text{thr}(v)$ a threshold function with threshold value of 0. The $[x_i, y_i]$ coordinates represent points on the circle of the radius r.

The number of points p is usually set to the size of common computer registers: 8, 16 or 32 bits. Using these pattern sizes allows for a very efficient LBP manipulation.

Changing the radius r leads to a bigger or smaller neighbourhood. It is possible to reach scale invariance by computing LBP for multiple values of r. Usually, r is increased by multiples of two every step, leading to a logarithmic number of LBP calculations per pixel in proportion to the image size.

By definition, the LBP operator is independent of brightness and contrast as neither adding, nor multiplying the image intensity by a constant factor changes the sign of the difference between the LBP centre and the points in its circular neighbourhood.
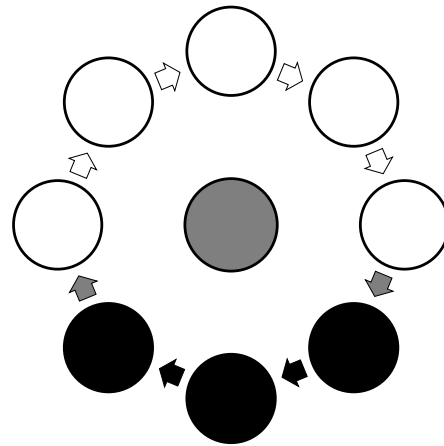

Figure 1. A circular neighbourhood representing a Local Binary Pattern. Black points have a lower intensity than the central point, white points have a higher intensity. The pattern can be represented efficiently using 8 bits.

2.1. Rotation invariance

Rotating the input image results in a shift in bits of the LBP value as the starting bit of the LBP gets rotated which in turn results in a different texture class. The aim of many computer vision applications is to reach rotation invariance – to be able to detect an object under different rotations.

Pietikäinen [13] devised a simple algorithm to make LBP values independent on the image rotation. The pattern is first rotated to its base position which is defined as a cyclic rotation of the bit pattern which has most zeroes at the start. For example, the 8bit pattern 11000010 gets normalized to 00001011. The rotation with the most zeroes at the beginning can be found very efficiently using a lookup table (for smaller bit patterns) or by rotating the pattern using a fast processor instruction and finding the integer minimum of all the possible rotations.

2.2. Uniform LBP

There are exponentially many rotationally invariant LBP values ($O(2^p)$ with p being the neighbourhood size/number of bits). This implies that the least probable patterns occur with no more than exponentially small probability. Ojala [11] observed that more than 80 % of LBP values contain at most 2 changes between 0 and 1 in the bit string. This has led to the definition

of uniform LBP which specifies rotation invariant patterns with at most two 0/1 changes in the bit string. Other patterns are grouped together into an "other" category. This gives a total of $p + 2$ categories which significantly reduces the LBP space and allows for a simpler manipulation.

When applied to texture classification, uniform LBP provides state-of-the-art results [12]. However, for many object-detection applications the 20 % values grouped to the "other" category results in classification problems as a lot of potentially useful information is lost [14].

3. Proposed extensions

3.1. Adaptive categorization

A careful analysis shows that the uniform LBP values can be generalized when described using two numbers – the number of 0/1 bit changes (t) and the number of bits set to 1 (j). Setting a limit $t \leq 2$ yields descriptor equivalent to uniform LBPs.

Because j and t are counts, they can each form a single scalar input to machine learning algorithms (an input neuron of a neural network, for instance), unlike the original LBP values where a binary input needs to be specified for each of the $p + 2$ possible classes. This allows for faster training while keeping the discriminative power of LBP. Frequencies of the individual combinations of t and j can also be grouped to a two dimensional histogram to model a probability distribution of the LBP patterns in a texture.

3.2. Noise reduction using contrast weighting

Image intensity noise can severely affect the bit string of a LBP value, especially in uniform regions where the differences between the central pixel and the points in its circular neighbourhood are small. A small change in the intensity value can result in a different sign after the subtraction, leading to a different bit string and thus a different texture class. It is possible to reduce image noise in a pre-processing step but that increases computational cost and can lead to a loss of information. It is usually better to let a subsequent machine learning algorithm to account for noise.

We propose to use LBP value contrast weighting to provide a classifier with the information of how much the given LBP value is prone to noise. The bigger the average contrast between the centre and the circular neighbourhood, the smaller is the chance of the pattern being affected by noise. We therefore assign a low weight to patterns in uniform regions and a high weight to regions with high contrast/rich texturing:

$$w = \frac{1}{p} \sum_{i=1}^{p} (c - I(x_i, y_i))^2$$

Where w is the resulting weight (ranging from 0 to 1), p the number of points in the circular neighbourhood, c the intensity of the centre and $I(x, y)$ the intensity of a point in the circular neighbourhood.

4. Results

The proposed LBP extensions were implemented in a real-world object detection system for monitoring laboratory rats (see Figure 2). The objective is to detect rat's head in a camera picture with resolution of 640×480 pixels. The images vary greatly in lighting conditions, rotation

and translation and are influenced by colour noise from the camera. The extended LBP values were binned to histograms and used for a simple histogram-based template matching to find the most probable locations of rat's head. These locations were then passed to the boosting based classifier from the OpenCV library [15].

We measured the detection accuracy against hand-annotated reference data using the F score. Our method achieved F score of 0.74 (74 %). Using the LBP implementation of object detection from the OpenCV library we achieved F score of only 0.18 (18 %).

The system was able to process one frame in 542 milliseconds on a dual core laptop processor.
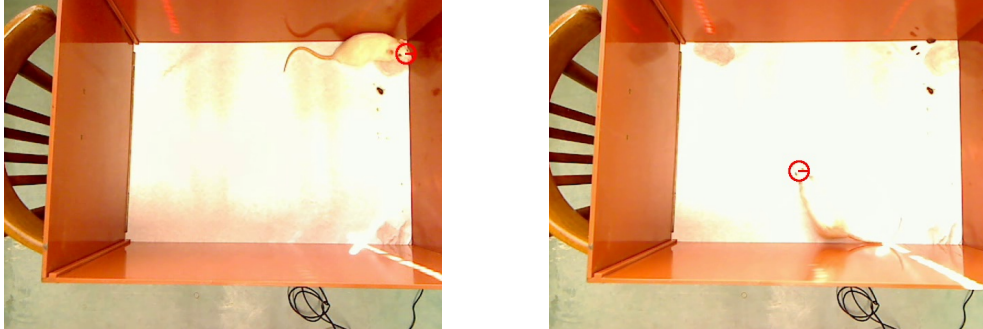

Figure 2. Detecting laboratory rat's head under difficult lighting conditions and various rotations.

4. Conclusion

We present Local Binary Patterns extensions that allow for a more efficient image processing and a simpler connection to machine learning algorithms by reducing the pattern description to two interval variables and weight. We verified LBP invariance to rotation and lighting variations in brightness and contrast on a task from practice.

The proposed weighting for noise reduction was tested in a real-world scenario with noisy cameras in a low-contrast settings (white rat heads on white background). The raw input detection results with LBP weighting were comparable to LBP without weighting on a de-noised (5×5 median filtered) input.

The extended LBP can be used for real-time object detection as there's no need to pre-process the input image, the LBP computation uses only a fast integer and bitwise operations and is easily parallelized due to inherent LBP locality. A further speed-up can be achieved by implementing the method on graphics hardware.

Acknowledgements

References

[1] Ojala, T., & Pietikäinen, M. (1999). Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, *32*(3), 477-486.
[2] Xu, Q., Yang, J., & Ding, S. (2005). Texture segmentation using LBP embedded region competition. *Electronic Letters on Computer Vision and Image Analysis*, *5*(1), 41-47.

Petranek, K., Janecka, P., & Vanek, J. (2015). Using local binary patterns for object detection in images. *Global Journal of Computer Sciences*. *5*(1), 07-12.

[3] Rara, H., Farag, A., Elhabian, S., Ali, A., Miller, W., Starr, T., & Davis, T. (2010, September). Face recognition at-a-distance using texture and sparse-stereo reconstruction. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on* (pp. 1-6). IEEE.

[4] Hadfield, S., & Bowden, R. (2012). Generalised pose estimation using depth. In*Trends and Topics in Computer Vision* (pp. 312-325). Springer Berlin Heidelberg.

[5] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*(12), 2037-2041.

[6] Wang, X., Han, T. X., & Yan, S. (2009, September). An HOG-LBP human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on* (pp. 32-39). IEEE.

[7] Trefný, J., & Matas, J. (2010). Extended set of local binary patterns for rapid object detection. In *Proceedings of the Computer Vision Winter Workshop* (Vol. 2010).

[8] Chang, T., & Kuo, C. J. (1993). Texture analysis and classification with tree-structured wavelet transform. *Image Processing, IEEE Transactions on*, *2*(4), 429-441.

[9] Livens, S., Scheunders, P., Van de Wouwer, G., & Van Dyck, D. (1997, July). Wavelets for texture analysis, an overview. In *Image Processing and Its Applications, 1997., Sixth International Conference on* (Vol. 2, pp. 581-585). IET.

[10] Hsu, T., Calway, A. D., & Wilson, R. (1993, May). Texture analysis using the multiresolution Fourier transform. In *8th Scandinavian Conference on Image Analysis*.

[11] Ojala, T., Pietikäinen, M., & Mäenpää, T. (2000). Gray scale and rotation invariant texture classification with local binary patterns. In *Computer Vision-ECCV 2000* (pp. 404-420). Springer Berlin Heidelberg.

[12] Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(7), 971-987.

[13] Pietikäinen, M., Ojala, T., & Xu, Z. (2000). Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, *33*(1), 43-52.

[14] Guo, Z., Zhang, L., Zhang, D., & Mou, X. (2010, September). Hierarchical multiscale LBP for face and palmprint recognition. In *Image Processing (ICIP), 2010 17th IEEE International Conference on* (pp. 4521-4524). IEEE.

[15] "OpenCV," *OpenCV*. [Online]. Received February 20, 2014 from: http://www.opencv.org.