

# Local binary patterns

**Local binary patterns (LBP)** is a type of visual descriptor used for classification in computer vision. LBP is the particular case of the Texture Spectrum model proposed in 1990.<sup>[1][2]</sup> LBP was first described in 1994.<sup>[3][4]</sup> It has since been found to be a powerful feature for texture classification; it has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.<sup>[5]</sup> A comparison of several improvements of the original LBP in the field of background subtraction was made in 2015 by Silva et al.<sup>[6]</sup> A full survey of the different versions of LBP can be found in Bouwmans et al.<sup>[7]</sup>

## Contents

**Concept**

**Extensions**

**Implementations**

**See also**

**References**

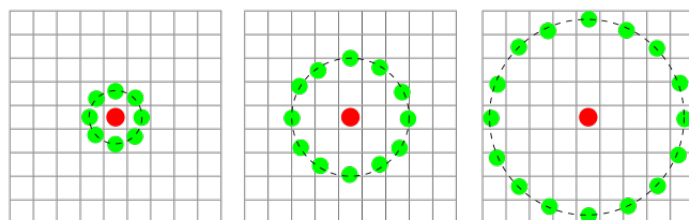
## Concept

The LBP feature vector, in its simplest form, is created in the following manner:

- Divide the examined window into cells (e.g. 16x16 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.
- Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1". This gives an 8-digit binary number (which is usually converted to decimal for convenience).
- Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center). This histogram can be seen as a 256-dimensional feature vector.
- Optionally normalize the histogram.
- Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window.

The feature vector can now be processed using the Support vector machine, extreme learning machines, or some other machine-learning algorithm to classify images. Such classifiers can be used for face recognition or texture analysis.

A useful extension to the original operator is the so-called uniform pattern<sup>[8]</sup>, which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor. This idea is motivated by the fact that some binary patterns occur more commonly in texture images than others. A local binary pattern is called uniform if the binary pattern contains at most two 0-1 or 1-0 transitions. For example, 00010000(2 transitions) is a uniform



Three neighborhood examples used to define a texture and calculate a local binary pattern (LBP)

pattern, 01010100(6 transitions) is not. In the computation of the LBP histogram, the histogram has a separate bin for every uniform pattern, and all non-uniform patterns are assigned to a single bin. Using uniform patterns, the length of the feature vector for a single cell reduces from 256 to 59. The 58 uniform binary patterns correspond to the integers 0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254 and 255.

## Extensions

---

- Over-Complete Local Binary Patterns (OCLBP):<sup>[9]</sup> OCLBP is a variant of LBP that has been shown to improve the overall performance on face verification. Unlike LBP, OCLBP adopts overlapping to adjacent blocks. Formally, the configuration of OCLBP is denoted as  $S : (a, b, v, h, p, r)$ : an image is divided into  $a \times b$  blocks with vertical overlap of  $v$  and horizontal overlap of  $h$ , and then uniform patterns  $LBP(u2, p, r)$  are extracted from all the blocks. Moreover, OCLBP is composed of several different configurations. For example, in their original paper, the authors used three configurations:  $S : (10, 10, 12, 12, 8, 1), (14, 14, 12, 12, 8, 2), (18, 18, 12, 12, 8, 3)$ . The three configurations consider three block sizes:  $10 \times 10$ ,  $14 \times 14$ ,  $18 \times 18$ , and half overlap rates along the vertical and horizontal directions. These configurations are concatenated to form a 40877 dimensional feature vector for an image of size  $150 \times 80$ .
- Transition Local Binary Patterns(tLBP):<sup>[10]</sup> binary value of transition coded LBP is composed of neighbor pixel comparisons clockwise direction for all pixels except the central.
- Direction coded Local Binary Patterns(dLBP): the dLBP encodes the intensity variation along the four basic directions through the central pixel in two bits.
- Modified Local Binary Patterns(mLBP): the mLBP compares the values of neighboring pixels to the average of the intensity values in the  $3 \times 3$  window.
- Multi-block LBP: the image is divided into many blocks, a LBP histogram is calculated for every block and concatenated as the final histogram.
- Volume Local Binary Pattern(VLBP):<sup>[11]</sup> VLBP looks at dynamic texture as a set of volumes in the  $(X, Y, T)$  space where  $X$  and  $Y$  denote the spatial coordinates and  $T$  denotes the frame index. The neighborhood of a pixel is thus defined in three dimensional space, and volume textons can be extracted into histograms.
- RGB-LBP: This operator is obtained by computing LBP over all three channels of the RGB color space independently, and then concatenating the results together.

## Implementations

---

- CMV (<http://www.cse.oulu.fi/CMV/>), includes the general LBP implementation (<http://www.cse.oulu.fi/CMV/Downloads/LBP Matlab>) and many further extensions over LBP histogram in MATLAB.
- Python mahotas (<http://luispedro.org/software/mahotas>), an open source computer vision package which includes an implementation of LBPs.
- OpenCV's Cascade Classifiers support LBPs as of version 2.
- VLFeat (<http://vlfeat.org/>), an open source computer vision library in C (with bindings to multiple languages including MATLAB) has an implementation ([http://www.vlfeat.org/api/lbp\\_8h.html](http://www.vlfeat.org/api/lbp_8h.html)).
- LBPLibrary (<https://github.com/carolinepacheco/lbplibrary>) is a collection of eleven Local Binary Patterns (LBP) algorithms developed for background subtraction problem. The algorithms were implemented in C++ based on OpenCV. A CMake file is provided and the library is compatible with Windows, Linux and Mac OS X. The library was tested successfully with OpenCV 2.4.10.
- BGSLibrary (<http://code.google.com/p/bgslibrary/>) includes the original LBP implementation for motion detection<sup>[12]</sup> as well as a new LBP operator variant combined with Markov Random Fields<sup>[13]</sup> with improved recognition rates and robustness.
- dlib (<http://dlib.net>), an open source C++ library: implementation ([https://github.com/davisking/dlib/blob/master/dlib/image\\_transforms/lbp.h](https://github.com/davisking/dlib/blob/master/dlib/image_transforms/lbp.h)).
- scikit-image (<http://scikit-image.org/>), an open source Python library. Provides a c-based python implementation ([https://github.com/scikit-image/scikit-image/blob/master/skimage/feature/\\_texture.pyx#L95](https://github.com/scikit-image/scikit-image/blob/master/skimage/feature/_texture.pyx#L95)) for LBP

## See also

---

- Local ternary patterns
- Local Binary Pattern (LBP) methodology in Scholarpedia ([http://www.scholarpedia.org/article/Local\\_Binary\\_Patterns](http://www.scholarpedia.org/article/Local_Binary_Patterns))

## References

---

1. DC. He and L. Wang (1990), "Texture Unit, Texture Spectrum, And Texture Analysis", Geoscience and Remote Sensing, IEEE Transactions on, vol. 28, pp. 509 - 512.
2. L. Wang and DC. He (1990), "Texture Classification Using Texture Spectrum", Pattern Recognition, Vol. 23, No. 8, pp. 905 - 910.
3. T. Ojala, M. Pietikäinen, and D. Harwood (1994), "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994), vol. 1, pp. 582 - 585.
4. T. Ojala, M. Pietikäinen, and D. Harwood (1996), "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", Pattern Recognition, vol. 29, pp. 51-59.
5. "An HOG-LBP Human Detector with Partial Occlusion Handling", Xiaoyu Wang, Tony X. Han, Shuicheng Yan, ICCV 2009
6. C. Silva, T. Bouwmans, C. Frelicot, "An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos", VISAPP 2015, Berlin, Germany, March 2015.
7. T. Bouwmans, C. Silva, C. Marghes, M. Zitouni, H. Bhaskar, C. Frelicot,, "On the Role and the Importance of Features for Background Modeling and Foreground Detection", <https://arxiv.org/pdf/1611.09099v1.pdf>
8. Barkan et. al "Fast High Dimensional Vector Multiplication Face Recognition." Proceedings of ICCV 2013
9. Barkan et. al "Fast High Dimensional Vector Multiplication Face Recognition." Proceedings of ICCV 2013
10. Trefný, Jirí, and Jirí Matas. "Extended set of local binary patterns for rapid object detection." Proceedings of the Computer Vision Winter Workshop. Vol. 2010. 2010.
11. Zhao, Guoying, and Matti Pietikainen. "Dynamic texture recognition using local binary patterns with an application to facial expressions." Pattern Analysis and Machine Intelligence, IEEE Transactions on 29.6 (2007): 915-928.
12. M. Heikkilä, M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects", IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4):657-662, 2006.
13. C., Kertész: Texture-Based Foreground Detection, International Journal of Signal Processing, Image Processing and Pattern Recognition (IJSIP), Vol. 4, No. 4, 2011.

---

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Local\\_binary\\_patterns&oldid=811982695](https://en.wikipedia.org/w/index.php?title=Local_binary_patterns&oldid=811982695)"

---

**This page was last edited on 25 November 2017, at 05:41.**

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.