

Sub Oriented Histograms of Local Binary Patterns for Smoke Detection and Texture Classification

Feiniu Yuan¹, Jinting Shi^{1,2}, Xue Xia¹, Yong Yang¹, Yuming Fang¹ and Rui Wang³

¹School of Information Technology, Jiangxi University of Finance and Economics
Nanchang 330032, Jiangxi, China
[e-mail: yfn@ustc.edu]

²Vocational School of Teachers and Technology, Jiangxi Agricultural University
Nanchang 330032, Jiangxi, China
[e-mail: icanflysjt@126.com]

³School of Software & Communication Engineering, Jiangxi University of Finance and Economics
Nanchang 330032, Jiangxi, China
*Corresponding author: Feiniu Yuan

*Received November 4, 2015; revised January 17, 2016; accepted February 15, 2016;
published April 30, 2016*

Abstract

Local Binary Pattern (LBP) and its variants have powerful discriminative capabilities but most of them just consider each LBP code independently. In this paper, we propose sub oriented histograms of LBP for smoke detection and image classification. We first extract LBP codes from an image, compute the gradient of LBP codes, and then calculate sub oriented histograms to capture spatial relations of LBP codes. Since an LBP code is just a label without any numerical meaning, we use Hamming distance to estimate the gradient of LBP codes instead of Euclidean distance. We propose to use two coordinates systems to compute two orientations, which are quantized into discrete bins. For each pair of the two discrete orientations, we generate a sub LBP code map from the original LBP code map, and compute sub oriented histograms for all sub LBP code maps. Finally, all the sub oriented histograms are concatenated together to form a robust feature vector, which is input into SVM for training and classifying. Experiments show that our approach not only has better performance than existing methods in smoke detection, but also has good performance in texture classification.

Keywords: Sub Oriented Histogram, Smoke detection, Image Classification, Local Binary Patterns, Support Vector Machine.

A preliminary version of this paper appeared in ISITC 2015, August 15-17, Tianjing, China. This version includes a concrete analysis, a totally different feature extraction method and extended experiments on Brodatz dataset. This work was supported by Natural Science Foundation of China (61363038, 61063034), Cultivated Talent Program for Young Scientists of Jiangxi Province (20142BCB23014) and Science Technology Application Project of Jiangxi Province (KJLD12066).

1. Introduction

Timely detection of fire is the basic guarantee of avoiding harm caused by fire, which is a serious threat to people's lives and property. Among a variety of methods for fire detection, computer vision based methods are considered promising because they are quick in responses, low in cost, and less susceptible to environmental factors, etc. It has become a more important branch of fire detection.

According to objects for detection, vision based fire detection methods can be classified into two categories: flame detection and smoke detection. Smoke often emerges before flame. In addition, early flame may not necessarily fall into the video surveillance area due to the size of flame, occlusion and so on. Smoke is a very important clue for early fire detection.

Smoke not only has some obvious features such as color, texture, and shape, etc., but also has important dynamic features such as flowing, and flicker, etc. However, it is worth noting that dynamic feature extraction requires background modeling or frame differences, which are often based on thresholds. It is very difficult for users to specify appropriate thresholds, which greatly affect experimental results [1]. The performance of smoke detection can be improved if more discriminative static features are extracted from images.

Currently, there are many methods proposed to extract static features of smoke image. Ferrari [2] proposed a method of performing wavelet transformation on images and building a Hidden Markov Tree Model to extract smoke texture. Maruta [3] proposed a method of using Co-occurrence Matrix to detect smoke. Yuan [4] proposed a method of using Local Binary Pattern (LBP) and the variance of Local Binary Pattern (LBPV) to detect smoke. Chen et al. [5] statistically extracted the feature of smoke images in RGB color space and proposed a rule of color for smoke classification. Leonardo et al. [6] applied the rule to YCbCr color space and proposed a new representing method. Genovese et al. [7] applied the rule to the YUV space. Krstinic' [8] proposed a new color model called HS'I specifically for smoke detection. Surit et al. [9] proposed a method of combining irregular smoke shapes with color features. Wang et al. [10] proposed a modified Center Symmetric Local Ternary Pattern (CS-LTP) for smoke texture descriptor. The background subtraction method was used to differentiate between smoke and non-smoke region. Progresses of face recognition, texture classification and scene analysis can be modified and used to improve classification accuracy of smoke detection. Wang et al. [11] proposed two dimensional principal components of natural images (2D-PCs) that is similar to principal components of natural images (1D-PCs), and then designed two kinds of statistical texture features (STF(1D) and STF(2D)) for multi-class facial expression recognition. Soh and Tsatsoulis [12] used gray-level co-occurrence matrices (GLCM) to quantitatively evaluate textural parameters and representations to determine which parameter values and representations are best for mapping sea ice texture. Alceu et al. [13] proposed the Segmentation-based Fractal Texture Analysis (SFTA) by decomposing the input image into a set of binary images for Fractal Analysis of Textures. Li et al. [14] proposed a contextual Bag-of-Words for visual categorization, which was extensively evaluated on video event and scene categorization. For intensive survey on scene analysis, we refer interested readers to [15].

Previous work has shown that the most outstanding methods are still one based on texture feature extraction. Ojala et al. [16] proposed Local Binary Pattern, which is one of the most prominent methods and has attracted increasing attention in the field of texture analysis due to gray scale invariance, rotational invariance, low computational complexity, etc.

Consequently, many LBP variants have been presented in recent literatures to further enhance the performance. Tan et al. proposed Local Ternary Pattern (LTP) that is more robust against noise [17]. Qian et al. proposed a local binary pattern texture descriptor with pyramid representation (PLBP) to remove some of the noise impact [18]. Guo et al. [19] proposed Completed Modeling of Local Binary Pattern (CLBP) that combines the sign and magnitude component about the center pixel. Although significant progress has been made, most LBP variants just consider each LBP code independently. There are few investigations into the relationship of LBP codes. Zeng et al. [20] proposed a Pixel-Pattern-Based Texture Feature (PPBTF) that is insensitive to variance of illumination by transforming a gray scale image into a pattern map. PPBTF captures edges and lines to characterize texture information.

In the paper, we propose a novel computationally simple but effective approach for image smoke detection. We extract LBP codes and analyze the orientation of the gradient on LBP codes in a local neighborhood to propose an oriented histograms of LBP. Although traditional LBP histograms can represent features of smoke images well, they completely discard the spatial distribution of LBP codes. To extract this spatial distribution information, it is necessary to consider the relationship between the two LBP codes. The problem is that an LBP code is just an integer value without numerical meaning. Therefore, we propose to use Hamming distance of LBP codes to compute gradients over LBP codes to capture more discriminative features. Then, we compute LBP histograms based on the gradient orientation. At last, we concatenate these LBP histograms together to extract a robust feature vector for smoke detection, and use support vector machine (SVM) that has good generalization performance for training and classification. To further evaluate the performance of our methods, we also test our method on texture data sets. Experimental results on both smoke data sets [4] and Brodatz texture data sets [21] show that our method achieves promising results.

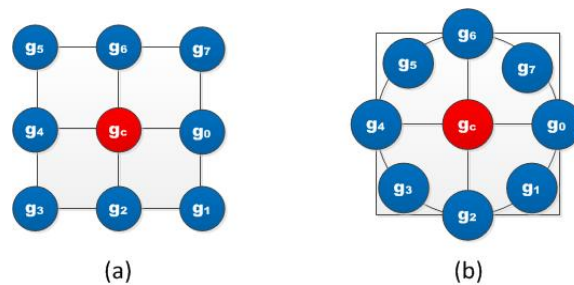


Fig. 1. Neighborhood types. (a) Rectangular neighborhood; (b) Circular neighborhood

The main contribution of this paper is to propose sub oriented histograms of LBP codes according to two discrete orientations of LBP codes. The sub oriented histogram is used to capture the spatial distribution of LBP codes. The second contribution is to use Hamming distance of LBP codes with two coordinates systems for computation of orientations. Two coordinates systems increase the number of neighboring LBP codes for computation of orientations.

The paper is organized as follows: We first review Local Binary Patterns in Section 2.1. We introduce computation of Hamming distance between two LBP codes in Section 2.2. In Section 2.3, we discuss how to calculate the oriented gradients over original LBP code map. Section 2.4 presents the sub oriented histogram of LBP codes. In Section 3, we test our method

for smoke detection and texture classification, and compare our method with the state-of-the-art methods. At last, we conclude this paper with a discussion in Section 4.

2. Local Binary Pattern with Gradient analysis

2.1 Local Binary Patterns

Traditional LBP computes in a simple way of comparing the value of a center pixel g_c with the values of its 3×3 rectangular neighborhood pixel value $g_i (i = 0, \dots, 7)$ as shown in **Fig. 1(a)**. The LBP pattern code is computed as follows:

$$LBP = \sum_{i=0}^7 s(g_i - g_c) 2^i \quad (1)$$

where

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

Ojala et al. [16] expanded rectangular neighborhood to circularly symmetric neighborhood to achieve multiresolution performance. However, P points are re-sampled in a circular neighborhood with radius R around the center pixel. There are three mapping patterns, which are defined as “Uniform”, “Rotation Invariant” and “Rotation Invariant and Uniform”, respectively. Experimental results show that the most “frequent” uniform binary patterns correspond to primitive micro features, such as edges, corners, and spots.

“Uniform” is defined as a pattern which has no more than 2 spatial transitions (bitwise 0/1 or 1/0 changes). The patterns with more than 2 spatial transitions will be considered as identical patterns. So there are $P*(P-1)+3$ different patterns. The uniform value can be computed by:

$$U(LBP_{P,R}) = \sum_{i=1}^{P-1} |s(g_{(i+1)\%P} - g_c) - s(g_i - g_c)| \quad (3)$$

where % denotes the modulo operation.

“Rotation Invariant” is defined to remove the effect of rotation as

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R,i}) | i = 0, \dots, P - 1\} \quad (4)$$

where $ROR(x,i)$ denotes a circular bit-wise right shift on the number x i times.

“Rotation Invariant and Uniform” pattern is a combination of “Uniform” pattern and “Rotation Invariant” pattern. It is defined as:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases} \quad (5)$$

If P and R is respectively set to 8 and 1, the circularly symmetric neighborhood can be simplified to a 3×3 rectangular neighborhood as shown in **Fig. 1(b)**. Although multiresolution can be applied in a circularly neighborhood, interpolation is required and leads to increasing the computation complexity (e.g. the neighbors’ values g_1, g_3, g_5 and g_7 are computed by interpolation). Therefore, we use the 3×3 rectangular neighborhood to avoid re-sampling of pixel values.

We compute original LBP codes for the whole image. These original LBP codes will be first used for analysis of the orientation of gradient to extract the relationship between the two LBP codes described in section C. According to these gradient orientations, we calculate the LBP histogram in each orientation respectively. The original LBP code is directly used to compute the gradient orientation. Because any of “Uniform”, “Rotation Invariant” or “Rotation Invariant and Uniform” mapping patterns will generate a new numeric label, which leads to loss of texture information. The details will be described in section 2.2.

2.2 Distance measure for LBP codes

According to the definition of LBP, each bit represents the binarized difference of pixel values in a certain direction. Similarly, we can encode the changes of the bits in the same direction of two LBP codes to extract these variations of the two codes. As illustrated in Fig. 2(b), the LBP code in the center is “1” and the code in the direction “7” is “0”, so both the Euclidean and Hamming distances between the two codes are 1, and the two distances are coincident. But for the direction 5, things will be so different. The Euclidean distance between the center code (“1”) and the code (“255”) in the direction 5 is 254 while the Hamming distance between the two codes is 7. The two distances are very different. As we can see, the Hamming distance metric is more concordant than the Euclidean distance metric.

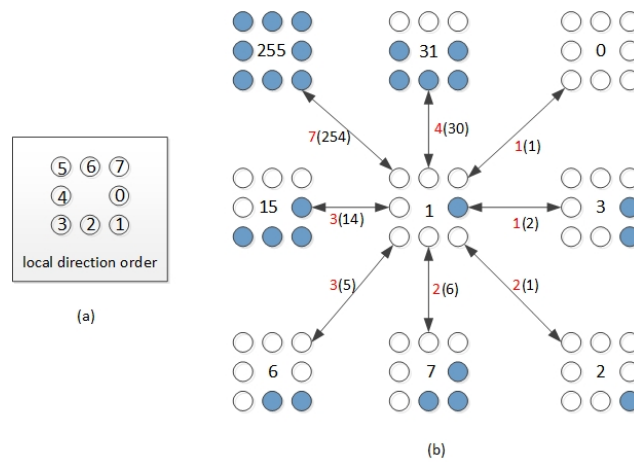


Fig. 2. Distance measures. (a) the local direction order; (b) the differences between the central LBP code and the LBP codes in its rectangular neighborhood.

Therefore, the variation degree of texture can be measured by the Hamming distance between two LBP codes along each direction. As shown in Fig. 2(b), the number in brackets following the red number, which is the Euclidean distance, is just the Hamming distance between two LBP codes. In the directions “3” and “4”, their Hamming distances are 3, i.e., there are 3 directions where the LBP bits change, but the Euclidean distances are 5 and 14, respectively. Notice that the LBP code has no numerical meaning. Thus, the difference value can't represent the number of changed directions.

So we use Hamming distance to measure two LBP codes [22]-[24]. Hamming distance is firstly introduced in error detection and correction code in the literatures. Hamming distance between two equal length strings is defined as the number of positions where corresponding characters are different. Consider Hamming distance between strings “smoke” and “smile”, because the third and fourth characters in the word “smoke” are different from the third and fourth characters in the word “smile” respectively, so there are two corresponding positions that have different symbols, thus the Hamming distance is 2. We can see that each bit of LBP code corresponds to a direction of local neighborhood. If we need to compare two LBP codes in each direction to acquire information of variation trends, we can compare each bit of two LBP codes with each other. In other words, we consider an LBP code as a sequence of bits. By regarding each LBP code as a binary string of length P , we calculate Hamming distance of two LBP codes x and y as follows:

$$d = \sum_{i=0}^{P-1} (x(i) \oplus y(i)) \quad (6)$$

where $x(i)$ and $y(i)$ are the i -th bits of x and y respectively for $i = 0, \dots, P-1$, \oplus represents *exclusive or* operation. In our implementation, P is set to 8.

As shown in **Fig. 2(b)**, the Hamming distance between the LBP code in the center that is “00000001” and its adjacent LBP code in direction 6 that is “00011111” is 4. The distance value just shows there are 4 directions where there are changes of bits.

2.3 Gradients of LBP codes with two coordinate systems

Edge information is one of the most fundamental characteristics in an image. The gradient of an image is usually used to approximately detect edges. There are a lot of feature extraction methods based on gradient for object detection, such as Scale-Invariant Feature Transform (SIFT) [25], Histogram of Oriented Gradients (HOG) [26] and so on. HOG achieves good performance especially in pedestrian detection and face detection.

Fig. 3 shows processing flow of HOG, where we need to estimate the gradient of an image, and compute the magnitude and orientation of the gradient. We can use Sobel operators or central differences of an image $f(x,y)$ to compute the gradient (f_x, f_y) of the image. Then we use Eq. (7) and (8) to calculate the magnitude and orientation of the gradient.

$$mag = \sqrt{f_x^2 + f_y^2} \quad (7)$$

$$\theta = \tan^{-1}(f_y/f_x) \quad (8)$$

where f_x stands for horizontal differences of the image, f_y stands for vertical differences of the image, mag in Eq.(7) represents the gradient magnitude of image, and θ denotes the gradient the orientation of the gradient. θ is quantized into several bins. In HOG, mag in Eq.(7) is accumulated as a weight into the corresponding bin defined by θ .

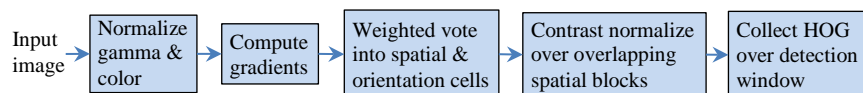


Fig. 3. Processing flow of HOG.

But when we directly use HOG for smoke detection, experiments show that the result is not as good as other detections. The reason may be that the edge of smoke is not as salient as other objects. To improve performance, we propose to extract sub oriented LBP code maps from LBP code maps and compute oriented histograms of LBPs from the oriented maps.

In order to illustrate special characteristics of smoke images, we compare the image “lena” (**Fig. 4a**) with the image “smoke” (**Fig. 4e**). As we can see, the edge of “lena” (**Fig. 4b**) is more distinct than “smoke” (**Fig. 4f**). In addition, the edges of “lena” are denser than that of “smoke”. The reason is that smoke often has low contrast, and the situation gets worse especially for mist-like smoke. Too low contrast leads to low detection rates and high false alarm rates.

LBP has powerful discriminative capabilities in feature extraction from an image. **Fig. 4(c)** and **Fig. 4(g)** are LBP code maps of “lena” and “smoke”, respectively. Histograms of LBP codes reflect frequency of each LBP code and obtain robust performance in many applications. But histograms discard spatial distributions of LBP codes in the whole image.

As we mentioned in above sections, the difference of two LBP codes is actually the Euclidean distance of the two codes, which is not coincident with our intuition. Therefore, we propose to use Hamming distance for computation of gradients over LBP codes.

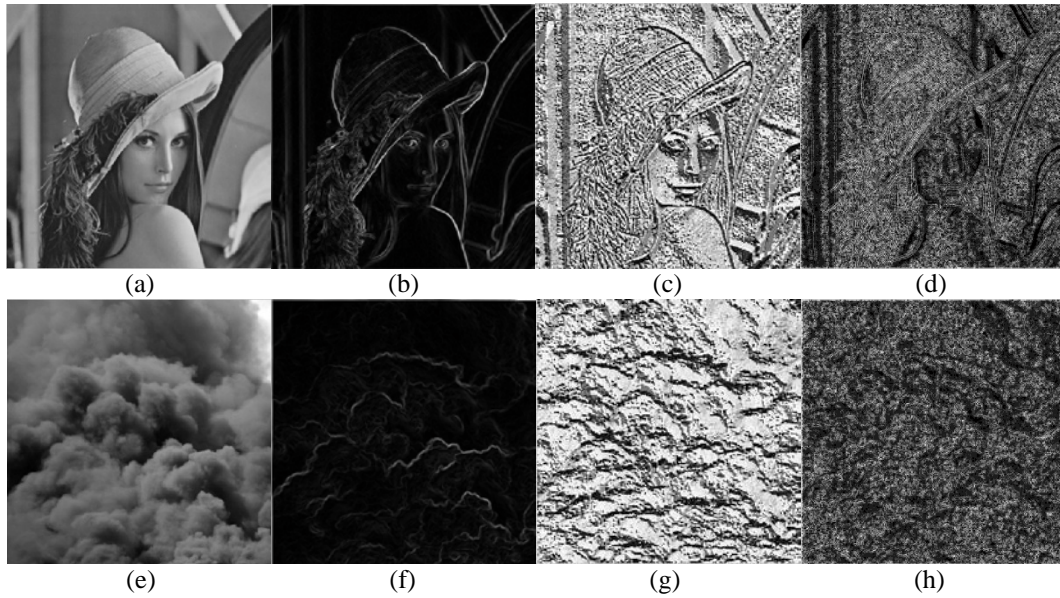


Fig. 4. Comparisons for LBP codes and gradients based on Hamming distance. (a) The image named “lena”; (b) The gradient magnitude of “lena”; (c) The LBP code map of “lena”; (d) The gradient magnitude of “lena” based on Hamming distances of LBP codes; (e) The image named “smoke”; (f) The gradient magnitude of “smoke”; (g) The LBP code map of “smoke”; (h) The gradient magnitude of “smoke” based on Hamming distances of LBP codes.

However, there are two issues to be solved. The first is that the difference of gradient can be either positive or negative, but the Hamming distance is always a non-negative number. If we use Hamming distance to compute gradients for HOG, the orientation θ is always located in the first quadrant that leads to a drop of discriminative performance. To solve this issue, we introduce a reference code c , and the difference between two LBP codes can be re-defined as the difference between the differences of the two LBP codes and the reference code c as follows:

$$d = \|x - c\| - \|y - c\| \quad (9)$$

where x and y are the two LBP codes, and $\|\cdot\|$ denotes a distance measure that can be L1 or L2 norms [24]. Since the distance defined in Eq. (9) may be zero, positive or negative, θ may be located in any quadrant. So the discriminative capability can be improved. If we set the reference code c to “00000000”, Eq. (9) is reduced to

$$d = \|x\| - \|y\| \quad (10)$$

Fig. 4(d) and **Fig. 4(h)** respectively show gradient magnitudes of “lena” and “smoke” based on Hamming distances of LBP codes, which are estimated by Eq.(10).

Second, original LBP codes reflect variation of pixel values in sample directions. We often use the distances of two LBP codes along two orthogonal directions to estimate gradients, so four LBP codes are involved, as shown in **Fig. 5(a)**. If the LBP code map is rotated 45° as shown in **Fig. 5(b)**, the other 4 LBP codes can be involved. Rotation of 45° can avoid

re-sampling of pixel intensities for computation efficiency. To include more information of LBP codes for estimation of gradients, we propose to use two coordinates systems to estimate two gradients. Therefore, there are two orientations θ_1 and θ_2 for each point.

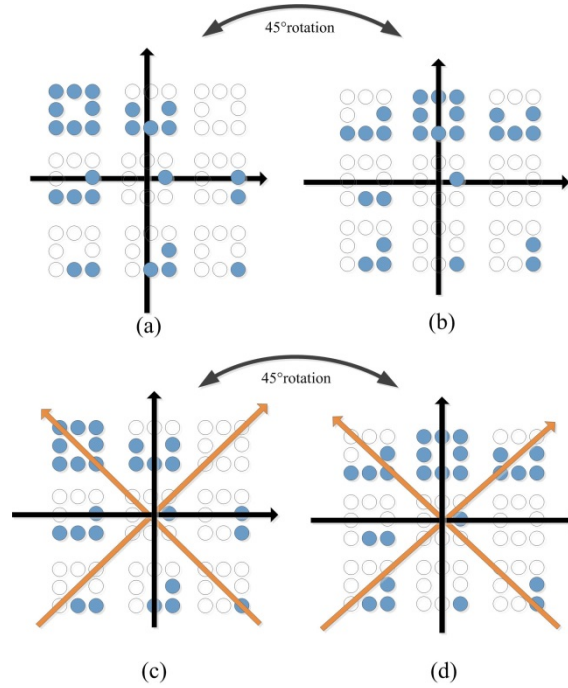


Fig. 5. Gradient computation over LBP maps. (a) coordinate system for computing gradient in traditional HOG; (b) coordinate system for computing gradient in traditional HOG after the LBP map is rotated 45° ; (c) two coordinate systems in our approach; (d) two coordinate systems in our approach after LBP map is rotated 45° .

2.4 Sub oriented histograms of LBP codes

As shown in **Fig. 6**, suppose that we quantize the two orientations into n_1 and n_2 bins, so there are $n_1 * n_2$ different combinations of the two discrete orientations (θ_1, θ_2) that are $(0, 0), (0, 1), (0, 2), \dots, (n_1 - 1, n_2 - 1)$, respectively. As for a specific pair of orientations θ_1 and θ_2 , we can generate a sub LBP code map from the original LBP code map. Hence, we obtain $n_1 * n_2$ sub LBP code maps from the original LBP code map denoted as \mathbf{M}_{lbp} . A sub code map $\mathbf{M}_{sub}^{o_1, o_2}$ for a specific pair of discrete orientations (o_1, o_2) can be generated as follows:

$$\mathbf{M}_{sub}^{o_1, o_2}(i, j) = \begin{cases} \mathbf{M}_{lbp}(i, j) & \text{if } o_1 = \theta_1(i, j) \text{ and } o_2 = \theta_2(i, j) \\ L_{new} & \text{else} \end{cases} \quad (11)$$

where L_{new} is a new label that can be specified any value out of the LBP value range.

We can generate an LBP histogram from each sub LBP map. For a given pair of orientations (o_1, o_2) , the histogram of a sub LBP map, which is also called sub oriented histogram by us, can be computed as follows:

$$\mathbf{H}_{sub}^{o_1, o_2}(k) = \frac{1}{w * h} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \delta(\mathbf{M}_{sub}^{o_1, o_2}(i, j) - k) \quad (12)$$

where $\delta(v)$ is the delta function returning 1 if $v=0$ and 0 for $v \neq 0$, w and h are the width and height of the sub code map, respectively.

Since there are $n_1 * n_2$ sub LBP code maps, we have $n_1 * n_2$ histograms. As for each histogram, we compute only the frequency of the original LBP labels except for the new label L_{new} . The reason is that the new label of a given pair stands for LBP labels whose orientations are not inside the given pair but must be inside some pairs, so it is unnecessary to count the frequency repeatedly. Finally, we concatenate all the histograms $\mathbf{H}_{sub}^{o_1, o_2}$ together to form a robust vector \mathbf{F} to describe image samples:

$$\mathbf{F} = \{\mathbf{H}_{sub}^{0,0}, \mathbf{H}_{sub}^{0,1}, \dots, \mathbf{H}_{sub}^{n_1-1, n_2-1}\} \quad (13)$$

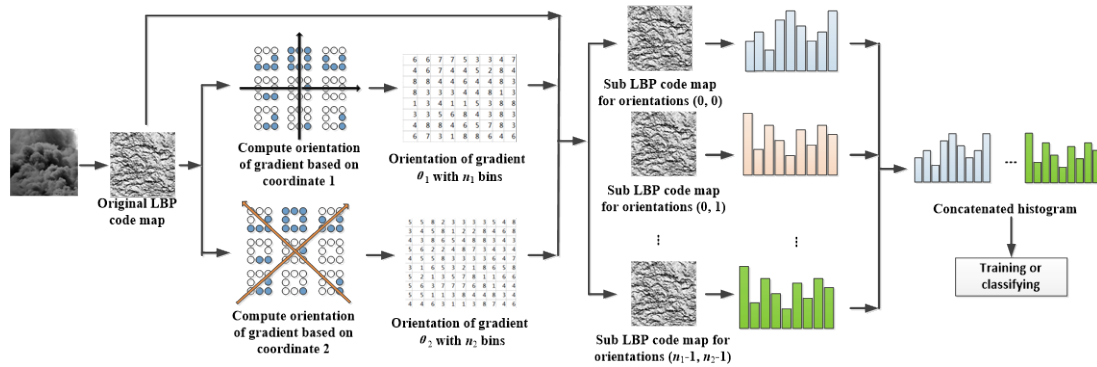


Fig. 6. Framework of feature extraction.

We extract the feature of testing images in the same way for training images. Support vector machine (SVM) has outstanding performance on classification, so we input the extracted feature vector \mathbf{F} into SVM for training and testing. Fig. 6. gives the overall framework of the feature extraction procedure.

3. Experiments and Results

3.1 Smoke Detection

To evaluate the performance of the proposed method, we use our publicly available smoke and non-smoke image datasets [4]. Some images of the datasets were captured by us, and others were manually collected from internet. All images of the datasets were resized to the size of $48 * 48$ for feature extraction. The datasets can be downloaded via <http://sit.jxufe.cn/yfn/vsd.html>. Table 1 shows four data sets named as **Set1**, **Set2**, **Set3** and **Set4**. **Set1** has 1383 images including 552 smoke images and 831 non-smoke images. In **Set2**, there are 1505 images consisting of 688 smoke images and 817 non-smoke images. **Set3** has 10712 images including 2201 smoke images and 8511 non-smoke images. **Set4** consists of 10617 images that have 2254 smoke images and 8363 non-smoke images. We adopted the relatively small dataset **Set1** for training, and another small dataset **Set2** and two large datasets (**Set3** and **Set4**) for classifying. Some smoke and non-smoke samples are shown in Fig. 7. All RGB images are resized to $48 * 48$ pixels and converted to grayscale images.

We used LIBSVM by Chih-Jen Lin et al. [27] for classification. Each component of feature vectors was normalized to the range $[0, 1]$ all over the images. The type of SVM is set to C-SVC. An efficient additive kernel approximation (AKA) proposed by Vedaldi et al. [32] was utilized because the kernel enables the fast training and testing possible for nonlinear kernel and is scalable to data size. The parameter *cost* in the loss function is set to 500. Since

there is unbalance between the numbers of smoke and non-smoke samples, we set the parameter *weight* to the ratio of each class sample quantity to all the sample quantity.

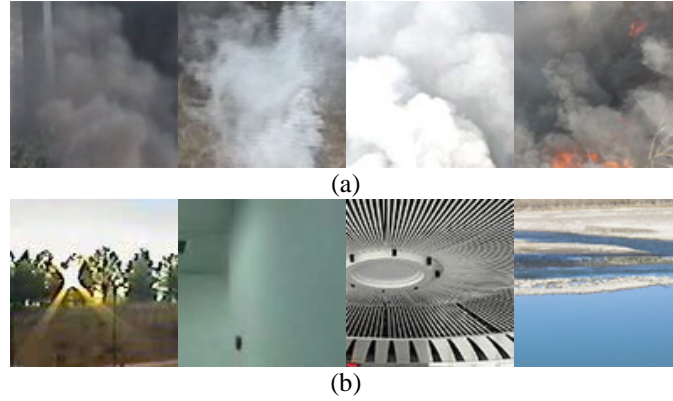


Fig. 7. Smoke samples and non-smoke samples. (a) Smoke samples. (b) Non-smoke samples.

Table 1. The image datasets for training and classifying

Data sets	Number of Smoke images	Number of non-smoke images	Total number of images	Usage
Set1	552	831	1383	Train
Set2	688	817	1505	Classify
Set3	2201	8511	10712	Classify
Set4	2254	8363	10617	Classify

Table 2. Summary of some texture detection methods

Methods	Description and references
LBP U2/RI/RIU2	The original LBP with uniform pattern, rotation invariant and rotation invariant uniform patterns [16], respectively.
PLBP U2/RI/RIU2	Local binary patterns are extended to pyramid transform domain [18] with uniform pattern, rotation invariant and rotation invariant uniform patterns, respectively.
CLBP U2/RI/RIU2	Completed local binary patterns (CLBP) [19] with uniform pattern, rotation invariant and rotation invariant uniform patterns, respectively.
HOG	Histograms of oriented gradients for human detection [26]
NRLBP	Noise-Resistant Local Binary Patterns [28]
POEM U2/RI/RIU2	The Patterns of Oriented Edge Magnitudes [29] with uniform pattern, rotation invariant and rotation invariant uniform patterns, respectively.
LTrP U2/RI/RIU2	Local Tetra Patterns presented in [30] with uniform pattern, rotation invariant and rotation invariant uniform patterns, respectively.
LDP4 U2/RI/RIU2	Local derivative pattern (LDP) [31] with uniform pattern, rotation invariant and rotation invariant uniform patterns, respectively.
PRICoLBP	Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern [32]

We implemented several methods that are listed in [Table 1](#) for comparisons. Experimental results are shown in [Table 3](#). There are three criteria for performance evaluation, i.e. Detection Rate (DR), False Alarm Rate (FAR), and Error Rate (ERR). An ideal detection method should have high DR, low FAR and ERR at the same time.

In order to intuitively analyze the experimental results of the compared methods on the three testing sets, we visualize the results of each dataset by drawing charts of DRs and FARs, as shown in [Fig. 8](#). Obviously, a good method should have high DR and low FAR, and hold the maximized distance between DR and FAR. As we can see from these figures, our method has very good classification performance on smoke detection.

Table 3. Experimental results

Methods	Set2 (%)			Set3 (%)			Set4 (%)			
	DR	FAR	ERR	DR	FAR	ERR	DR	FAR	ERR	
Our method	97.0	1.35	2.13	95.2	1.82	2.45	94.7	1.46	2.28	
LBP	U2	98.5	3.79	2.72	96.9	3.82	3.68	96.5	3.56	3.54
	RI	98.1	3.79	2.92	96.3	4.69	4.49	96.3	4.08	4.00
	RIU2	98.8	6.98	4.32	96.7	9.98	8.60	97.4	8.90	7.54
HOG	89.4	11.5	11.1	86.6	14.9	14.6	86.4	13.8	13.7	
PLBP	U2	91.0	1.35	4.85	90.5	2.19	3.69	89.6	1.66	3.51
	RI	95.5	1.59	2.92	94.7	4.66	4.80	94.6	4.02	4.30
	RIU2	97.7	2.20	2.26	96.8	5.32	4.89	97.1	4.46	4.12
CLBP	U2	97.1	2.57	2.72	94.7	2.02	2.70	94.9	1.81	2.50
	RI	98.1	1.59	1.73	96.5	2.68	2.85	97.0	2.04	2.24
	RIU2	99.0	4.28	2.79	98.0	6.59	5.65	98.5	5.72	4.82
NRLBP	81.3	18.7	18.7	84.4	24.1	22.3	84.1	23.3	21.7	
POEM	U2	67.7	16.6	23.8	63.8	13.8	18.4	66.2	12.7	17.2
	RI	74.3	27.3	26.6	70.6	29.8	29.7	70.4	28.7	28.9
	RIU2	76.0	32.7	28.7	70.6	36.0	34.6	69.2	34.6	33.8
LTrP	U2	57.3	0.12	19.6	63.4	0.12	7.62	60.4	0.06	8.45
	RI	94.9	0.36	2.53	95.1	0.78	1.62	94.1	0.96	2.02
	RIU2	97.5	1.10	1.73	96.3	3.29	3.38	95.1	3.53	3.82
LDP4	U2	77.6	0.98	10.8	80.0	1.16	5.03	79.9	0.85	4.94
	RI	97.5	2.33	2.39	96.3	2.89	3.06	96.1	2.59	2.88
	RIU2	98.1	3.33	2.39	97.1	2.82	3.23	97.1	3.34	3.24
PRICoLBP	98.2	1.84	1.79	96.9	2.84	2.90	96.81	2.45	2.61	

HOG was originally proposed for pedestrian and face detection, and demonstrated good discriminative capabilities. However, HOG does not achieve good performance on our smoke data sets. The main reason may be that smoke has no fixed shapes. Original HOG may not be suitable for smoke detection.

For the three testing sets, DRs of our method are higher than PLBP U2 and PLBP RI, and its FARs and ERRs are also lower than PLBP U2 and PLBP RI. PLBP RIU2 has slightly higher DRs than our method, but FARs and ERRs of our method are obviously lower than PLBP RIU2.

Our method has lower detection rates but lower false alarm rates than CLBP with U2, RI and RIU2. In other words, it is hard to say which method is better. We achieved the same results when we compare our method with PRICoLBP. POEM achieved low DRs, and high FARs and ERRs, so POEM is not suitable for smoke detection. Our method obviously outperforms POEM with respect to DR, FAR and ERRS.

Although LTrP with “U2” has lower FAR than our method on datasets Set2, Set3 and Set4, its DR is far lower than our method. LDP4 with “U2” has very low FAR and ERR but also has very low DR. We also find that DRs of our method are slightly lower than LBP, CLBP and LDP4 with some patterns. It is worth noting that when LTrP with “RI” pattern has very low FAR and a proper DR. PRICoLBP also achieves good performance on DR, FAR and ERR. Although DRs of our method are slightly lower than those of PRICoLBP, our FARs and ERRs are obviously lower than PRICoLBP’s. When detection rates are high enough (say, nearly close to 100%), we’d better pay more attentions to false alarm rates. Since PRICoLBP used multi-scale and multi-orientation, our method is more computationally efficient than PRICoLBP.

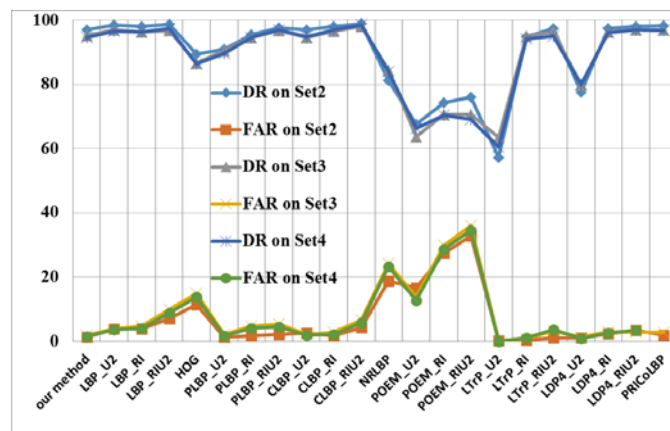


Fig. 8. Visualization of experimental performance.

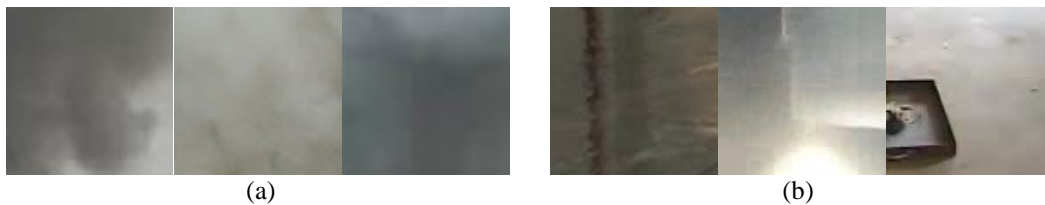


Fig. 9. Falsely classified samples of smoke (a) and non-smoke (b).

We also present some samples that were falsely classified by our method, as shown in Fig. 9. Smoke images have very low quality and contrast, and highly blurred texture. Smoke samples in Fig. 9a were falsely classified as non-smoke. On the other hand, there are a lot of non-smoke images that look like smoke in appearance. Fig. 9b shows that three non-smoke images are misclassified as smoke. From these misclassified images, we can see that it is very difficult to accurately distinguish smoke and non-smoke images due to large variations of smoke color, texture and shape.

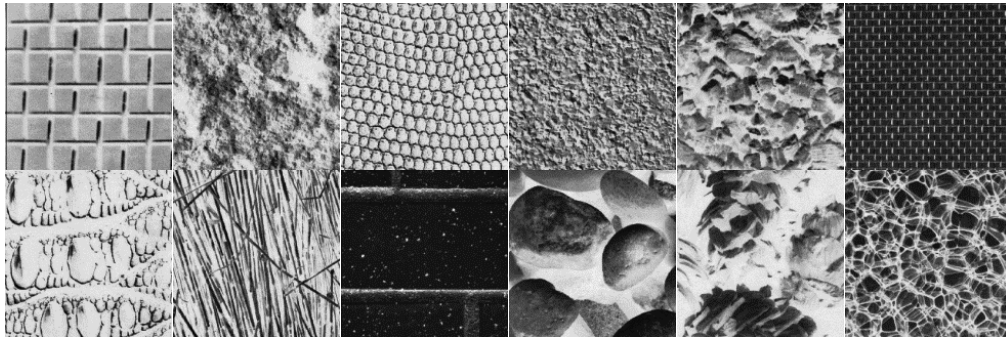


Fig. 10. Some samples of Brodatz album.

3.2 Texture Classification

We also tested our method on texture data sets to demonstrate the discriminative performance for texture classification. The Brodatz album [21] is a well-known texture classification benchmark dataset. It contains 111 texture classes with 9 images in each class. Some samples are shown in Fig. 10. We randomly select 3 images from each class to be used for training and the rest 6 images are used for testing, just in the same way as PRICoLBP [32]. We repeated experiments 100 times. At last, we compute the average accuracy of the 100 experiments.

Several methods are listed in Table 4 for comparisons. For CLBP [16], LBPV [33], LBPHF_S [34], three scales were used with codes on author's websites. CoALBP was implemented by [35] and PRICoLBP was implemented by [32]. Experimental results are shown in Table 5. Since the image size of brodatz is large and texture images are with more distinct edge information than smoke images, we use two scales of LBP, i.e., we set radius and neighbor number to (1, 8) and (2, 8), respectively. In our implementation, the orientation of gradient is quantized into 4 bins. We set the parameters of SVM in the same as smoke detection.

Table 4. Summary of some texture detection methods

Methods	References
LBP	The original LBP [16].
LBPV	Rotation invariant texture classification using lbp variance (lbpv) with global matching [33]
LBPHF_S	Rotation invariant image description with local binary pattern histogram fourier features [34]
CLBP	Completed local binary patterns (CLBP) [19]
CoALBP	Feature extraction based onco-occurrence of adjacent local binary patterns [35]
PRICoLBP	Pairwise rotation invariant co-occurrence local binary pattern [32]

From Table 5, we can see that our method achieved the correct classification rate of 95.3%, which is higher than ones by LBP, LBPV, LBPHF_S, CLBP and CoALBP, so our method obviously outperforms other methods. It is worth noting that we use less scales than those methods. PRICoLBP uses complex information, such multi-scale and multi-orientation, so the method achieved very good performance on Brodatz. We also find the PRICoLBP has slightly higher accuracy than our method.

Table 5. comparison with other LBP variants on Brodatz

Methods	Correct classification rates (%)
Our method	95.3
LBP	91.6
LBPV	93.8
LBPHF_S	94.6
CLBP	94.8
CoALBP	94.2
PRICoLBP	96.9

4. Conclusion

Smoke often emerges earlier than flame, so smoke detection can provide very early fire detection. LBP and its variants are the most prominent in texture extraction, but most of them just consider each LBP code independently. To capture spatial distribution of features, we propose a novel and computationally simple approach for smoke detection and texture classification. Original LBP codes are first extracted and then the orientation of gradient over LBP codes is calculated. Since an LBP code is just a value without any numerical meaning, we replace direct differences of LBP codes with Hamming distances of LBP codes for computation of the gradient. We compute sub oriented histograms of LBP codes according to specific pairs of two discrete orientations. All the sub oriented histograms are concatenated to form a robust feature vector that is then input into SVM for training and classifying. Experimental results also show that our approach has better performance than existing methods and it can decrease false alarm rates without dropping of detection rates at the same time. The method also can be useful for texture classification.

References

- [1] Sheng Luo, and Yuzheng Jiang, "State-of-art of video based smoke detection algorithms," *Journal of Image and Graphics*, Vol. 18, No. 10, 1225-1236, 2013. [Article \(CrossRef Link\)](#)
- [2] Ricardo J. Ferrari, Hong Zhang, and C. Ronald Kube, "Real-time detection of steam in video images," *Pattern Recognition*, vol. 40, No. 3, 1148-1159, 2007. [Article \(CrossRef Link\)](#)
- [3] Hidenori Maruta, Akihiro Nakamura, and Fujio Kurokawa, "A new approach for smoke detection with texture analysis and support vector machine," in *Proc. of 2010 IEEE International Symposium on Industrial Electronics (ISIE)*, IEEE, July 4-7, 1550-1555, 2010. [Article \(CrossRef Link\)](#)
- [4] Feiniu Yuan, "Video-based smoke detection with histogram sequence of LBP and LBPV pyramids," *Fire safety journal*, Vol.46, No. 3, 132-139, 2011. [Article \(CrossRef Link\)](#)
- [5] Thou-Ho Chen, Y.H. Yin, S.F. Huang, and Y.T. Ye, "The smoke detection for early fire-alarming system base on video processing," in *Proc. of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IEEE, 18-20 Dec. 2006. [Article \(CrossRef Link\)](#)
- [6] Leonardo Millan-Garcia, Gabriel Sanchez-Perez, Mariko Nakano, Karina Toscano-Medina, Hector Perez-Meana, and Luis Rojas-Cardenas, "An early fire detection algorithm using IP cameras," *Sensors*, Vol. 12, No. 5, 5670-5686, 2012. [Article \(CrossRef Link\)](#)
- [7] Angelo Genovese, Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti, "Wildfire smoke detection using computational intelligence techniques," in *Proc. of 2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMS)*, 19-21, 1-6, Sept. 2011. [Article \(CrossRef Link\)](#)

- [8] Krstinić, Damir, Darko Stipaničev, and Toni Jakovčević, "Histogram-based smoke segmentation in forest fire detection system," *Information Technology and Control*, Vol. 38, No. 3, 237-244, 2009. [Article \(CrossRef Link\)](#)
- [9] Surapong Surit, Watchara Chatwiriya, "Forest fire smoke detection in video based on digital image processing approach with static and dynamic characteristic analysis," in *Proc. of 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering (CNSI), IEEE*, Jeju Island, Korea, 23-25, 35-3, May 2011. [Article \(CrossRef Link\)](#)
- [10] Yue WANG, Teck Wee CHUA, Richard CHANG and Nam Trung Pham, "Real-Time Smoke Detection Using Texture and Color Features," in *Proc. of IEEE International Conference on Pattern Recognition (ICPR 2012)*, pp. 1727-1730, Japan, 2012. [Article \(CrossRef Link\)](#)
- [11] Dong Wang, Huchuan Lu, Xuelong Li, "Two dimensional principal components of natural images and its application," *Neurocomputing*, 74(17), pp. 2745-2753, 2011. [Article \(CrossRef Link\)](#)
- [12] L. Soh and C. Tsatsoulis, "Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-Occurrence Matrices," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, March 1999. [Article \(CrossRef Link\)](#)
- [13] Alceu Ferraz Costa, Gabriel Humpire-Mamani, and Aagma Juci Machado Traina, "An Efficient Algorithm for Fractal Analysis of Textures," in *Proc. of Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 39-46, Ouro Preto-MG, Brazil, 2012. [Article \(CrossRef Link\)](#)
- [14] Teng Li, Tao Mei, In-So Kweon, and Xian-Sheng Hua, "Contextual Bag-of-words for visual categorization," *IEEE Trans. on circuits and systems for video technology*, 21(4), pp.381-392, 2011. [Article \(CrossRef Link\)](#)
- [15] Teng Li, Huang Chang, Meng Wang, Bingbing Ni, Richang Hong, and Shuicheng Yan, "Crowded scene analysis: a survey," *IEEE Trans. on circuits and systems for video technology*, 25(3), pp.367-386, 2015. [Article \(CrossRef Link\)](#)
- [16] Timo Ojala, Matti Pietikainen, and Topi Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vo. 24, No. 7, 971-987, 2002. [Article \(CrossRef Link\)](#)
- [17] Xiaoyang Tan, and Bill Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, Vol. 19, No. 6, 1635-1650, 2010. [Article \(CrossRef Link\)](#)
- [18] Xueming Qian, Xian-Sheng Hua, Ping Chen, and Liangjun Ke, "PLBP: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, Vol. 44, No. 10, 2502-2515, 2011. [Article \(CrossRef Link\)](#)
- [19] Zhenhua Guo, Lei Zhang, and David Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, Vol. 19, No. 6, 1657-1663, 2010. [Article \(CrossRef Link\)](#)
- [20] Xiang-Yan Zeng, Yen-Wei Chen, Zensho Nakao, and Hanqing Lu, "Texture representation based on pattern map," *Signal Processing*, 84 (3), pp. 589-599, 2004. [Article \(CrossRef Link\)](#)
- [21] P. Brodatz, "Textures: A Photographic Album for Artists and Designers," New York, NY, USA: Dover, 1999. [Article \(CrossRef Link\)](#)
- [22] Sanqiang Zhao, Yongsheng Gao, "Establishing point correspondence using multidirectional binary pattern for face recognition," in *Proc. of 19th International Conference on Pattern Recognition, ICPR 2008, IEEE*, 8-11, 1-4, Dec 2008. [Article \(CrossRef Link\)](#)
- [23] Cuicui Kang, Shengcai Liao, Shiming Xiang, and Chunhong Pan, "Kernel sparse representation with local patterns for face recognition," in *Proc. of Image Processing (ICIP), 2011 18th IEEE International Conference on. IEEE*, 2011. [Article \(CrossRef Link\)](#)
- [24] Xiaopeng Hong, Guoying Zhao, Matti Pietikainen, Xilin Chen, "Combining LBP Difference and Feature Correlation for Texture Description," *IEEE Transactions on Image Processing*, 23(6), 2557-2568, 2014. [Article \(CrossRef Link\)](#)

- [25] Lowe D.G, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, 60(2): 91-110, 2004. [Article \(CrossRef Link\)](#)
- [26] Dalal N., Triggs B, “Histograms of oriented gradients for human detection.” in *Proc. of IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 1. 886-893, 2005. [Article \(CrossRef Link\)](#)
- [27] Chang C.C., Lin C.J., “LIBSVM: a library for support vector machines,” *ACM Trans. on Intelligent Systems and Technology*, 2:27:1-27:27, 2011. [Article \(CrossRef Link\)](#)
- [28] Ren, Jianfeng, X. Jiang, and J. Yuan, “Noise-Resistant Local Binary Pattern With an Embedded Error-Correction Mechanism,” *IEEE Transactions on Image Processing*, Vol. 22, No.10, 4049-4060, 2013. [Article \(CrossRef Link\)](#)
- [29] Ngoc Son Vu, A. Caplier, “Enhanced patterns of oriented edge magnitudes for face recognition and image matching,” *IEEE Transactions on Image Processing*, Vol. 21, No. 3, 1352-1365, 2012. [Article \(CrossRef Link\)](#)
- [30] S Murala, R. P. Maheshwari, and R. Balasubramanian, “Local tetra patterns: a new feature descriptor for content-based image retrieval,” *IEEE Transactions on Image Processing*, Vol. 21, No. 5, 2874 - 2886, 2012. [Article \(CrossRef Link\)](#)
- [31] Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, and Jianzhuang Liu, “Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor,” *IEEE Transactions on Image Processing*, Vol. 19, No. 2, 533-544, 2010. [Article \(CrossRef Link\)](#)
- [32] Xianbiao Qi, Rong Xiao, Chun-Guang Li, Yu Qiao, Jun Guo, and Xiaoou Tang, “Pairwise Rotation Invariant Co-occurrence Local Binary Pattern,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.36, no. 11, pp. 2199-2211, 2014. [Article \(CrossRef Link\)](#)
- [33] Zhenhua Guo, Lei Zhang, and David Zhang, “Rotation invariant texture classification using LBP variance (LBPV) with global matching,” *Pattern Recognition*, vol. 43, No. 3, 706–719, 2010. [Article \(CrossRef Link\)](#)
- [34] T. Ahonen, J. Matas, C. He, M. Pietikainen, “Rotation invariant image description with local binary pattern histogram fourier features,” in *Proc. of 16th Scandinavian Conf. Image Anal*, 61–70, 2009. [Article \(CrossRef Link\)](#)
- [35] R. Nosaka, Y. Ohkawa, and K. Fukui, “Feature extraction based on co-occurrence of adjacent local binary patterns,” in *Proc. of 5th PacificRim Conf. Adv. Image Video Technol*, 82–91, 2012. [Article \(CrossRef Link\)](#)



Feiniu Yuan received his B.Eng. and M.E. degrees in mechanical engineering from Hefei University of Technology, Hefei, China, in 1998 and 2001, respectively, and Ph.D. degree in pattern recognition and intelligence system from University of Science and Technology of China (USTC), Hefei, in 2004. From 2004 to 2006, he worked as a post-doctor with State Key Lab of Fire Science, USTC. From 2010 to 2012, he was a Senior Research Fellow with Singapore Bioimaging Consortium, Agency for Science, Technology and Research, Singapore. He is currently a professor and a PhD supervisor with Jiangxi University of Finance and Economics. His research interests include 3D modeling, image processing and pattern recognition.



Jinting Shi received her B.E. degree in computer science and technology from Jiangxi Normal University, Nanchang, China, in 2003, and her M.S. degree in computer science and technology from Jiangxi Agricultural University, Nanchang, China, in 2008. She is currently a PhD candidate with School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China. Her research interests include image processing and pattern recognition.



Xue Xia received her B.E. degree in Film & TV Arts and Technology and M.E. degree in Communication and Information Engineering from Shanghai University, Shanghai, in 2011 and 2014, respectively. She is currently a PhD candidate with School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China. Her research interests include 3D display technology, image processing and pattern recognition.



Yong Yang received his PhD degree in biomedical engineering from Xi'an Jiaotong University, China, in 2005. He is currently a full Professor with School of Information Technology, Jiangxi University of Finance and Economics, China. From 2009 to 2010, he was a postdoctoral research fellow at Chonbuk National University, Republic of Korea. He has been awarded the title of Jiangxi Province Young Scientist since 2012. His current research interests include image and signal processing, medical image processing and analysis, and pattern recognition. He is a member of IEEE and CCF.



Yuming Fang is currently an Associate Professor in the School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China. He received the Ph.D. degree in Computer Engineering from Nanyang Technological University, Singapore. Previously, he obtained B.E. and M.S. from Sichuan University and Beijing University of Technology, China, respectively. He was also a (visiting) Postdoc Research Fellow in IRCCyN lab, PolyTech' Nantes&Univ. Nantes, Nantes, France, University of Waterloo, Waterloo, Canada and Nanyang Technological University, Singapore. His research interests include visual attention modeling, visual quality assessment, image retargeting, computer vision, 3D image/video processing, etc. He was a special session organizer in VCIP 2013, QoMEX 2014, etc.



Rui Wang graduated in computer science and technology from Jiujiang University, Jiujiang, China, in 2003. He is currently a Master Degree Candidate with School of Software & Communication Engineering, Jiangxi University of Finance and Economics, Nanchang, China. His research interests include pattern recognition and information communication technology.