Improved local binary pattern for real scene optical character recognition

Chu-Sing Yang, Yung-Hsuan Yang*

Department of Electrical Engineering, National Cheng Kung University, No.1, University Road, Tainan City 701, Taiwan, ROC

ARTICLE INFO

Article history:
Received 10 December 2016
Available online 8 August 2017

Keywords:
Local binary pattern
Optical character recognition
Integral image
Recognition speed

ABSTRACT

A strong edge descriptor is an important topic in a wide range of applications. Local binary pattern (LBP) techniques have been applied to numerous fields and are invariant with respect to luminance and rotation. However, the performance of LBP for optical character recognition is not as good as expected. In this study, we propose a robust edge descriptor called improved LBP (ILBP), which is designed for optical character recognition. ILBP overcomes the noise problems observed in the original LBP by searching over scale space, which is implemented using an integral image with a reduced number of features to achieve recognition speed. In experiments, we evaluated ILBP’s performance on the ICDAR03, chars74K, IIIT5K, and Bib digital databases. The results show that ILBP is more robust to blur and noise than LBP.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

With the increase in the number of portable cameras, smart phones, and surveillance cameras, the details in their images have considerably increased. As a result, the information in these images cannot be analyzed by humans. Thus, efficiently and automatically analyzing this information is an important challenge faced by image processing and pattern recognition. This information includes object images, text information, and characters. Extracting and understanding text in images have numerous applications in daily life such as helping vision-impaired people understand text information, automatic document generation, or even in surveillance systems. Therefore, we have mainly conducted research on the detection and recognition of character information.

The general method for recognizing text in an image is to extract features from the image and classify the characters by feeding them into a machine learning algorithm such as a neural network (NN) [1], support vector machine (SVM) [2,3], or Adaboost [4]. Alternatively, simpler algorithms such as the K-nearest neighbors (K-NN) [5] or mean shift [6] can be used. Recently, as the bag-of-words model [7] has gained considerable attention, many methods have been proposed to divide a character into smaller image “words” such as the end of stroke, curved stroke, or cross stroke. These smaller words improve the recognition rate and are very successful. However, fundamentally, the performance of these methods depends on the robustness of the feature extraction algorithm.

A robust texture descriptor must address numerous issues such as glare, blur, low contrast, and reflective surfaces in natural scene optical character recognition (OCR). Hence, the design of a robust text descriptor has become a popular topic in recognition systems, even for local descriptors of the bag-of-words (BOW) model. There are several fully developed descriptor methods such as histograms of Gaussian (HOG) [8], local binary pattern (LBP) [9], Harr-like descriptors [10], dense [11], BRIEF [12], SIFT [13,14], and SURF [15] descriptors. These methods have different advantages depending on the scene. For example, the HOG descriptor is more sensitive to edge changes, the Harr-like descriptor is more robust in blurred regions, and LBP and BRIEF overcome low contrast problems. However, while recognizing a character image in complex and varied real environments, there are still numerous problems to overcome.

The original LBP algorithm has many benefits with respect to texture recognition, such as luminance invariance, rotation invariance, easy implementation, and low computational cost. Moreover, LBP, both with and without rotation invariance, is very successful in the field of face recognition [16]. However, in our experience, LBP’s performance for real scene OCR is worse than that of HOG. This is because there is a large amount of hardware noise, blurred edges, and other interfering factors in real-scene character images. Hence, in this study, we propose an improved LBP (ILBP) texture descriptor, which is modified LBP without rotation invariance and with concepts from HOG added. There are four main modified methods in our implementation: creation of an LBP by averaging
pixel intensity via integral image, reduction of the number of histogram bins, creation of an LBP over scale space, and using a blocks and cells concept from HOG. In the bin reduction step, we reduce the number of LBP histogram bins and add a magnitude map similar to that of HOG; thus, each pixel can express an integral magnitude and direction information, which we called edge type. The second step is to create a bin-reduced LBP over scale space, and the third step is to determine a meaningful scale in scale space. We use the concept from MLBP [17] and SIFT that uses average intensity instead of single pixel intensity comparisons in LBP and different kernels between the center and neighbor pixels to achieve scale invariance. Fig. 1 shows the flow chart of ILBP.

In Section 2, we discuss related works, which are state-of-art OCR methods, comprise the original LBP, its variants, integral images, and scale space. Our main method, including scale space LBP, bin reduction, and scale selection, is explained in Section 3. Section 4 describes our experiment and recognition rate results for four letter datasets, IIIT5k [18], ICDAR03 challenge2 [19], chars74K task 15 [20], and Bib digital dataset [21]. Section 4 concludes the study.

2. Related works

Since visual word and BOW models have been popular in the field of image object recognition and matching, many of these methods have been presented. The method in [22] addresses geometric blur in OCR applications, where the basic idea is to match characters by local descriptors, and Wang et al. [23] proposed OCR by invariant feature matching. Furthermore, Gao et al. [24] presented a model to relate local strokes in a character using spatially embedded dictionary (SED). The SED procedure associates a code-word, a feature based on a blocked HOG, with a particular location and then builds a dictionary to model the co-occurrence of several local strokes. Shi et al. [25] also presented a part-base tree structure to model each character’s visual word (i.e., the end of a stroke, line, curve, or cross) and use a tree structure model to search for each part’s score and add it to the global information in the recognition stage. Although visual word-based OCR performs well, it depends on the robustness of the texture or edge descriptor. Hence, the design of a robust edge descriptor is an important issue. In our survey of the literature, the most common descriptor is HOG because it can overcome blur with its cell histogram and low contrast by block normalization. Moreover, we note that LBP performs well on face and texture recognition. However, our experiments show that LBP cannot overcome noise in real scene OCR and loses spatial information as a result of its rotation invariance.

There are many advantages of the simple LBP algorithm with rotation invariance in the field of texture recognition. The original LBP emerged from a simple concept: a comparison between the current pixel and its neighbors within a constant radius and direction as shown below (Fig. 2):

\[
LBP_{R, K}^{u_2} = \sum_{i=0}^{P-1} f(p_i - p_c)2^i
\]

The authors also defined parameter U to be the number of transitions between 0 and 1 [26]. Moreover, this study proved that the current pixel is meaningless when \( u \geq 3 \). So, in the Eq. (1), \( u_2 \) is denoted as remove transition times, which are greater than 2. The rotation invariant LBP is denoted as \( LBP_{R, K}^{u_2} \).

However, in real scene OCR, the performance of \( LBP_{R, K}^{u_2} \) is lower than that of \( LBP_{R, K}^{u_1} \) because it loses edge information between cells after rotation invariance is implemented. Table 1 compares the real scene OCR results of the original LBP algorithm with a cell size of 8 and block size of 2 (\( LBP_{R, K}^{2} \)) (the same LBP with rotation invariance \( LBP_{R, K}^{u_2} \)), and the HOG algorithm with a cell size of 8, block size of 2, and unsigned orientation bins in 9 directions (\( HOG_{R, K}^{2} \)). These results are case-insensitive because it is difficult to distinguish “x” and “X” in real scene OCR from a single letter image. The results show that HOG’s performance is better than the original LBP in each dataset. However, there are many extended LBP methods applied in numerous fields.

In our estimation, to design a descriptor for real scene OCR, the following issues must be addressed: noise, the number of features, ability to describe edges, low contrast regions, reflective surfaces, and unbalanced edge pairs of a stroke. Noise is unavoidable; thus there are many methods that use scale space to smooth the image for more stability. The ability to describe edges is the most important issue because the entire algorithm is based on a description of edges or the difference between a light and dark area such as HOG, Haar-like, dense, and LBP methods. The most uncontrollable problems are reflection, low contrast, and unbalanced edges. These problems cannot be controlled when we capture a real scene image with a low-cost device. Therefore, each descriptor can overcome two or more issues. HOG performs well with respect to lower feature numbers, good edge description, and robustness to unbalanced stroke type because of its unsigned orientation. LBP can handle low contrast images, and Haar-like and dense features can handle blurred conditions. Many variants of these methods have been proposed to overcome various issues.
To overcome the fluctuation of luminance, low contrast, and noise, the local ternary pattern (LTP) [26] was proposed. It considers smaller intensity differences to create two state LBPs, a lower and upper pattern, and then combines them into a feature descriptor. A series of LBPs [27–29] was also proposed to overcome the noise problem. The MLBP [17] method uses the average of a 3 x 3 area instead of a single pixel to determine small intensity differences. Multi-block LBP (MBLBP) [30,31] has a similar concept, which is to combine the average of a local block instead of single pixel’s intensity. Ref. [32] presented a different approach to addressing noise by using a scale space LBP feature. In our previous study, we also proposed using average intensity and selected a meaningful edge type over scale space. The method gives LBP scale invariance, but it still cannot solve the information loss after rotation invariance. Thus, we concluded that rotation invariance is not useful in real scene OCR. Hence, in this paper, we modified LBP without rotation invariance to achieve more stability. The overcomplete LBP [33] considers the overlap of adjacent blocks with different directions and modes to carry more information between cells. However, the higher number of features in these variants of LBP lead to higher training and recognition costs. Thus, there are methods to reduce the number or LBP features. Center-symmetric LBP (CS-LBP) [34] compares pairs of neighbors to lower the number of edge types and keep the same information as the original LBP. Its extended method, XCS-LBP, [35] compares the current pixel and distance neighbors to increase stability and use fewer features. Robust LBP (RLBP) [36] uses a different way to reduce the number of bins by binary code equivalence. In general, RLBP has 30 bins, and CS-LBP has 14 bins. Lower numbers of bins will save computation cost during recognition.

Numerous methods exist to enhance the ability of LBP edge description. One of them is to increase the number of neighbors and variant radius to obtain more texture information. Extended-LBP (ELBP) [37,38] generates multiple LBP codes to describe the current pixel state using more information. Complete-LBP (CLBP) [39] presents a similar idea with sign and gray-value differences, called intensity differences, to improve the current pixel’s discriminative power. And in discriminative robust-LBP (DRLBP) [40] enhances the edge description ability by combining RLBP and the difference of LBP, which comprises the complete bins of RLBP. The authors of [40] also presented a new concept to combine edge type and magnitude information. Instead of per-pixel integer counts, the histogram of DRLBP uses gradient magnitude \( \omega_{x,y} = \sqrt{\partial_x^2 + \partial_y^2} \), where \( \partial_x \) and \( \partial_y \) are the first-order derivatives in the x and y directions. DRLBP increases the edge information of a single pixel and makes LBP more like HOG. Hence, it is becoming common to let a single pixel carry more information. In our proposed method, we also applied this concept by different way.

Although there are numerous ways to improve LBP’s performance, not all the methods are suitable for real-scene OCR or a local edge descriptor and some methods improve texture recognition. For example, in our experiments, LTP is not suitable for OCR. In Char57k_15, LTP with zero threshold obtains the same recognition as the original LBP, but the recognition rate decreases when the threshold is increased.

Thus, we conclude that a strong texture descriptor will have the following four characteristics: 1. Robustness to noise. 2. Tolerance to unbalanced luminance. 3. Fewer feature dimensions. 4. More information per pixel.

3. Main method

We propose ILBP to create an LBP method with the characteristics mentioned in Section 2 by searching a scale space to realize meaningful edge information, reducing feature numbers with two procedures, and obtaining more information per pixel. In brief, we use a magnitude map to carry more information; however, the magnitude calculation for the current pixel is different from that of HOG and DRLBP. In this section, we explain how to create the LBP over scale space using an integral image. We then explain how to reduce number of bins and select the scale space method.

3.1. Scale space LBP

To overcome noise problems in LBP, we use an image blurred by integration to create scale-space LBP. First, we define the integral image as \( I_{p}(p) \), where \( p \) is a pixel index, and the image was smoothed by an \( m \times m \) box filter as \( I_{p}(p) \). Because, in this paper, we only use eight neighbors and a variable radius with one other parameter, we call the LBP method that uses eight neighbors original LBP. We next define \( LBP_{p}(x, y) \) as a comparison of center pixel \((x, y)\) with its neighbor pixels using the following smoothing kernel.

\[
LBP_{p}(p) = \sum_{i=0}^{7} s(I_{p-i+1}(c) - I_{p+i-1}(c))2^{i}
\]  

(3)

Note that \( s \) is always greater than or equal to one. We then redefine its neighbor as \( p_{r} \), Index \( i \) is the position relative to the current pixel, and \( r \) means the distance from the center position. The value of \( r \) is an integer obtained by rounding the Euclidean distance. The current pixel’s intensity is decided by taking the mean of a rectangle area with size \( 2s + 1 \), and its neighbors’ intensities are also smoothed with a smaller rectangle with a size of \( 2s - 1 \). When \( s = 1 \), the intensity of \( p_{r} \) is the mean of a 3 x 3 pixel rectangle area and the neighbor pixels are a single pixel intensity, as shown in Fig. 3 (right). Fig. 3 (left) shows the case for \( s = 2 \), where \( p_{r} \) is the mean of a 5 x 5 pixel rectangle area and \( p_{0,2} \) is the average of a 3 x 3 pixel rectangle area with a radius of 2. In our experiment, variant radii and smooth sizes between the center and its neighbor have a better recognition rate. Also, we proved the same using MLBP and ILBP in each dataset. Thus parameter \( r \) starts from 1 and is proportional to parameter \( s \).

We also calculate the approximate response when the scale space LBP is created, initializing the prediction equation as follows.

\[
R(p_{r}) = \left| \sum_{i=0}^{7} (G(p_{r}, \sigma_{1})) - G(p_{r}, \sigma_{2}) \right|
\]  

(4)

Here, \( R(p_{r}) \) usually uses \( G(p) \), a discrete version of the Laplacian of Gaussian (LOG), calculated as follows.

\[
G(p) = \frac{1}{2\pi\sigma}e^{-\frac{(x^{2}+y^{2})}{2\sigma^{2}}}
\]  

(5)

In Lowe’s work [14], they use an approximate method to calculate the extreme to detect the fitness of a blob at comparing scale and spatial space. In our application, we use the response to find the fitness of an edge type at the current pixel by searching scale space at maximum selection stage. As mentioned before, a key point in real scene edge descriptor is robustness to noise. Thus, there are three possible conditions between \( \sigma_{1} \) and \( \sigma_{2} \) when

![Fig. 3. Schematic of LBP, (left), and LBP, with first neighbor pixel (right).](image)
the current pixel value is noise. The first condition emphasizes the noise response when $\sigma_1 > \sigma_2$ because the peak of center $G(p_c, \sigma_2)$ will have a higher value than its neighbor. The second condition ($\sigma_1 = \sigma_2$) performs like the LOG operator. However, in the third condition ($\sigma_1 < \sigma_2$), we can obtain a smoother kernel with lower peaks and wider Gaussian RMSs. Thus in our application, we use $\sigma_1 < \sigma_2$ with linear growth.

To reduce the computational cost and fit LBP based on the integral image, we use a box filter instead of a Gaussian filter. By the central limit theory, a box filter is an approximation of Gaussian blur, so we can rewrite Eq. (4) as follows when $s$ is 1, $\sigma_1 = 2s - 1$, and $\sigma_2 = 2s + 1$.

$$R_s(p_c) = \left| \sum_{i=0}^{2} (I_{2s-1}(p_c) - I_{2s+1}(p_c)) \right|$$

(6)

We select a neighbor pixel using discrete distances and keep $\sigma_1 < \sigma_2$ to overcome the noise problem when $s$ is increased. Therefore, Eq. (6) is a discrete version of LOG.

3.2. Reducing number of bins in LBP

The number of bins in RLB at a uniform pattern is 30. This reduces the number of bins calculated using Eq. (7). However, we use the HOG concept of unsigned orientation.

$$RLBP(x, y) = \min [LBPs(x, y), 2^8 - 1 - LBPs(x, y) ]$$

(7)

The characters in a natural scene are randomly brighter or darker than the background. For example, an image may have a stroke that transitions from a lighter to a darker area. Thus, the performance of unsigned HOG is better than signed HOG in each dataset in our experiments. Considering this, we regard $\{00000011\}_2$ as $\{11111000\}_2$ so we can reduce the LB feature bins by considering the two situations as one edge type. For convenience, $M$ denotes the number of ones in $\{LBPs\}_2$ and $O$ is a rotation left with repeated bits; thus, $rLBPs$ can be written as follows.

$$rLBPs = \begin{cases} 1, & \text{if } O = 0 \text{ or } O = 0 \\ 255 - LBPs, & \text{else if } (M \geq 4 \text{ and } O \geq 4) \text{ or } O > 4 \\ LBPs, & \text{otherwise} \end{cases}$$

(8)

where $rLBPs$ is the first reduction in the number of bins of LBP. Furthermore, $rLBPs$ denotes the image after the uniform pattern process. In general, the range of $LBPs$ is $1-59$, which is combined with $M = 0$, $7 \times (M \times O)$, where $M = 8$, and $U > 3$. In our proposed method, the number of bins after the first reduction is 31, which is a combination of $\{00000000\}_2$, 28 bins (as shown in Fig. 4), $\{11111111\}_2$, and $u > 2$. And Fig. 4 also shows the folds of the edge types in our proposed method.

In the original LBP, each pixel presents a single piece of information because one pixel is only counted once in a cell; therefore, we propose an improvement to reduce the number of bins and present more information per pixel. First, we continue to reduce the bins by their directions, which is an idea borrowed from HOG. In HOG, the performance of an unsigned orientation is better than a signed orientation for person detection, and we have also obtained the same result in character recognition on the IIIT5k and Charts74k_15 datasets. Therefore, we reduce the number of LBPs to 10, as shown in Fig. 5. The edge types in rLBPs, which means the second bin reduction, is calculated as follows.

$$v(x, y) = rLBPs^{10}(x, y) - 2$$

(9)

$$rrLBPs^{10}(x, y) = \begin{cases} 1, & \text{if } M = 0 \text{ or } M = 8 \\ 0, & \text{else if } U > 3 \\ 2 \times \text{mod}(v(x, y), 4) + (M \cap 1) \oplus 1 + 2, & \text{otherwise} \end{cases}$$

(10)

The physical meaning of the magnitude map is the length from an edge to the current pixel. Fig. 5 shows the edge types and corresponding magnitudes; the first pattern in Fig. 6 shows edge type 2 and magnitude 1, and the second pattern shows the same edge type with magnitude 2. The third and fourth patterns in Fig. 6 show the situations when $M$ is even.

3.3. Selection method over scale space

After $rrLBPs^{10}$ is created over scale space, we use two approaches called “first section” and “maximum selection” to determine the meaningful scale. In the following, we explain the physical meaning of our proposed selection method.

We first observe that there are 59 edge types in the original $LBPs^{10}$, many of which are meaningless. This number is caused by hardware noise and the wrong radius size for obtaining the best information at the current pixel.

In Fig. 7, the yellow box is at a smaller scale in $LBPs$ and blue box is at a larger scale. The edge type in the yellow box is meaningless, but when we increase the scale, the edge type in the blue box is meaningful. Hence, we conclude that selecting the best scale will improve LBP performance. In general, we use the first non-zero value as the current pixel edge type, and we called this method first selection. In the first selection method, the pixel will
have larger magnitude if the current point is very close to the edge and vice versa. Fig. 8 shows a schematic of first selection and illustrates the situation we describe. We assign the pixel as edge type 10 if the pixel is all zeros over all scales.

Thus, the equation of $FLBP^{s2}_{S}(x, y)$ can be written as follows.

$$FLBP^{s2}_{S}(x, y) = \begin{cases} 
rrLBPF^{s2}_{i}(x, y), & \text{i is first none zero in } rrLBPF^{s2}_i \\
10, & \text{if } \forall i, rrLBPF^{s2}_i(x, y) = 0
\end{cases}$$

where $S$ in $FLBP^{s2}_{S}(x, y)$ is the scale space from 1 to $s$ of $rrLBPF^{s2}(x, y)$.

The second method is called maximum selection, and it selects the maximum response from Eq. (13) in scale space. As mentioned above, the response is a discrete version of LOG. Hence, we use the extreme response to reduce the effect of noise. Thus, the maximum selection method uses the absolute maximum response in scale space, which we record when creating LBP over scale space. We still assign edge type 10 to the current pixel if there are all zeros over all scales. Therefore, we can write $MLBP^{s2}_{S}(x, y)$ as the following:

$$MLBP^{s2}_{S}(x, y) = \begin{cases} 
rrLBPF^{s2}_{i}(x, y), & \text{i = arg max}_{i}(R_i(x, y)) \text{ and } rrLBPF^{s2}_i \neq 0 \\
10, & \text{if } \forall i, rrLBPF^{s2}_i(x, y) = 0
\end{cases}$$

Parameter $s$ is the search scale space, is from 1 to $s$, and is used to find the maximum response. ILBP combines two selection methods; thus, the number of features in ILBP is 20, which is a combination of the 10 features in first selection and the 10 features in maximum selection. The final step is to create a histogram and combine it into blocks. In this step, we use the L2-norm when the block is created. Note that the histograms of first and maximum selection have already been created. In our study, the rectangle box filter performs better than or equal to the Gaussian filter in OCR. Hence, we will discuss the performance when we replace rectangle smoothing with a Gaussian filter and use the interpolated intensity value when comparing neighbor intensities in Section 4. A clipping exists in HOG’s block normalization procedure, and the purpose of this clipping is to avoid peaks in the histogram and compress it into an acceptable size. Hence, we also tested the original algorithm with clipping in HOG in FLBP and MLBP, but the results were not as effective as expected. In ILBP, the magnitude is the integer distance to the nearest edge, so the meaning of the clipping factor is different from that of HOG. First selection determines the meaningful edge type around the edges, and maximum selection ignores the noise to find the strongest edges at the current pixel. Hence, we use the clip factor after block normalization and do not renormalize the blocks. Furthermore, first and maximum selection can have different clip factors. In our experiment, the clip factor is 0.3 and 0.4 when the cell size is 8 for OCR.

In Fig. 9, the first row depicts the histogram of each method. Both the maximum and first selection approaches obtain edge types without including any meaningless edge types. In the second row, we show the histogram results for a blurred version of the discrete edge image (blurred by a Gaussian filter with $\sigma = 2$). First selection, original LBP, and DRLBP obtain results that are similar to their results for the discrete image. Maximum selection is more affected by blur. We then add salt-and-pepper noise into the blurred image. The result of maximum selection for this image is similar to that of the discrete image. In contrast, first selection, HOG, the original LBP, and DRLBP are easily affected by noise. Therefore, we combine first selection and maximum selection to reduce the effect of both blur and noise, and the result shows that the combined method is more accurate on each dataset.

4. Experiment and results

In this section, we describe the experimental platform, then evaluate the relationship between scale and cell size, and finally present the results for OCR performance and the computation cost of ILBP.

We implemented the original LBP, LTP, RLBP, CS-LBP, DRLBP, multi-scale LDP [41], HOG, HOG-LBP [42], Gabor-wavelet [43], daisy [44], Haar-like, local phase quantization (LPQ) [45], Brief, and ILBP methods in a Windows 10 OS with i7-4720HQ on a 16 GB memory laptop. For the software, we used MATLAB for the data presentation and the core of each method was implemented using the OpenCV library and C++ code for performance. Each method’s performance was carefully verified, and we cut off the pixels on the borders to avoid edge interference. For example, our HOG’s performance is better than that of MATLAB and OpenCV (76.5% than 75.6% with a block size of 1 and cell size of 8). Moreover, we used the same histogram procedure to include the cell’s histogram and block the normalization function to guarantee that each method’s performance was compared fairly. For the classifier, we used the K-NN method to classify each character. This method can use multi-classifiers, but the recognition speed depends on the feature extraction method and number of features. K-NN and multi-classifiers (NN, SVM, or AdaBoost) have both advantages and disadvantages, but we choose K-NN as our classifier because it is an effective way to observe the robustness of each edge descriptor. More importantly, it is a standard baseline for testing the performance of each method. In general, CS-LBP will have lowest number of feature dimensions (4), HOG has 9 features per cell, ILBP has 20, RLBP has 30 features per cell, original LBP without rotation invariance has 59 features, and DRLBP has 60 features per cell. However, ILBP’s basic performance is affected by scale $S$, which determines how many LBPs we should build and scan over the scale space. In the same architecture, ILBP’s performance will be 5 times that of LBP, but by using advance CUDA architecture, the computation time can be reduced to an acceptable time.

The most controversial parameter is $S$, which is the scale space’s size. This parameter will affect accuracy and the computational cost because we have to scan all the scale space. In our experiments, an $S$ that is too big or too small will affect the maximum selection. Thus, the best $S$ depends on what we want to describe and the characteristics of the dataset. Similar to HOG, cell size also affects the accuracy and feature dimensions. Larger cell sizes will degrade accuracy but reduce the number of features and increase recognition speed. Table 2 lists the case-insensitive recognition rate with variable $S$ and cell sizes in the IIT5K dataset. Each
Table 2
ILBP recognition rate for variable scale and cell sizes on the IIT5K.

<table>
<thead>
<tr>
<th>Cell Size</th>
<th>Scale (IIT5K)</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>79.3%</td>
<td>80.4%</td>
<td>80.7%</td>
<td>80.6%</td>
<td>80.6%</td>
<td>80.6%</td>
<td>80.6%</td>
<td>80.6%</td>
</tr>
<tr>
<td>6</td>
<td>80.4%</td>
<td>82.1%</td>
<td>81.8%</td>
<td>81.7%</td>
<td>81.7%</td>
<td>81.7%</td>
<td>81.7%</td>
<td>81.7%</td>
</tr>
<tr>
<td>8</td>
<td>81.0%</td>
<td>82.1%</td>
<td>82.3%</td>
<td>82.2%</td>
<td>82.0%</td>
<td>82.0%</td>
<td>82.0%</td>
<td>81.9%</td>
</tr>
<tr>
<td>10</td>
<td>79.4%</td>
<td>81.5%</td>
<td>81.9%</td>
<td>82.0%</td>
<td>80.6%</td>
<td>80.6%</td>
<td>81.6%</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>78.2%</td>
<td>80.2%</td>
<td>80.6%</td>
<td>80.5%</td>
<td>80.5%</td>
<td>80.5%</td>
<td>80.5%</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>76.1%</td>
<td>75.9%</td>
<td>78.4%</td>
<td>78.5%</td>
<td>78.5%</td>
<td>78.5%</td>
<td>78.5%</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>74.0%</td>
<td>76.2%</td>
<td>76.8%</td>
<td>77.0%</td>
<td>76.9%</td>
<td>76.9%</td>
<td>77.0%</td>
<td></td>
</tr>
</tbody>
</table>

Table 3
ILBP recognition rate for variable scale and cell sizes on Chars74k Task 15.

<table>
<thead>
<tr>
<th>Cell Size</th>
<th>Scale (Chars74k_15)</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>62.3%</td>
<td>66.1%</td>
<td>65.7%</td>
<td>64.8%</td>
<td>64.7%</td>
<td>64.6%</td>
<td>64.6%</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>66.9%</td>
<td>68.9%</td>
<td>69.2%</td>
<td>67.6%</td>
<td>67.7%</td>
<td>67.8%</td>
<td>67.7%</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>68.6%</td>
<td>72.6%</td>
<td>71.8%</td>
<td>71.0%</td>
<td>70.5%</td>
<td>70.6%</td>
<td>68.9%</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>70.0%</td>
<td>71.7%</td>
<td>70.3%</td>
<td>69.2%</td>
<td>68.9%</td>
<td>68.7%</td>
<td>68.7%</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>67.7%</td>
<td>68.7%</td>
<td>71.4%</td>
<td>70.8%</td>
<td>70.5%</td>
<td>70.5%</td>
<td>70.5%</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>68.2%</td>
<td>69.7%</td>
<td>69.7%</td>
<td>69.4%</td>
<td>69.0%</td>
<td>68.9%</td>
<td>68.9%</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>63.6%</td>
<td>66.8%</td>
<td>67.2%</td>
<td>66.9%</td>
<td>66.8%</td>
<td>66.7%</td>
<td>66.7%</td>
<td></td>
</tr>
</tbody>
</table>

Table 4
Recognition rates for each method with a cell size of 8 and block size of 1.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Datasets</th>
<th>IIT5k</th>
<th>ICDAR03 Challenge 2</th>
<th>chars74K task 15</th>
<th>Bib digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILBP₅</td>
<td>79.9%</td>
<td>75.3%</td>
<td>63.9%</td>
<td>59.4%</td>
<td></td>
</tr>
<tr>
<td>LBP₅₅</td>
<td>65.1%</td>
<td>64.6%</td>
<td>39.1%</td>
<td>57.1%</td>
<td></td>
</tr>
<tr>
<td>LTP</td>
<td>66.3%</td>
<td>63.4%</td>
<td>38.5%</td>
<td>55.4%</td>
<td></td>
</tr>
<tr>
<td>RLB</td>
<td>67.5%</td>
<td>63.3%</td>
<td>43.4%</td>
<td>51.1%</td>
<td></td>
</tr>
<tr>
<td>CS – LBP</td>
<td>67.0%</td>
<td>62.8%</td>
<td>41.0%</td>
<td>54.37%</td>
<td></td>
</tr>
<tr>
<td>LDP</td>
<td>66.2%</td>
<td>61.1%</td>
<td>44.4%</td>
<td>64.1%</td>
<td></td>
</tr>
<tr>
<td>DRLBP</td>
<td>72.8%</td>
<td>70.6%</td>
<td>50.1%</td>
<td>59.0%</td>
<td></td>
</tr>
<tr>
<td>HOG</td>
<td>76.5%</td>
<td>72.8%</td>
<td>60.5%</td>
<td>58.3%</td>
<td></td>
</tr>
<tr>
<td>HOG-LBP</td>
<td>75.8%</td>
<td>72.1%</td>
<td>59.9%</td>
<td>59.8%</td>
<td></td>
</tr>
</tbody>
</table>

In the OCR performance test, we used four datasets, IIT5k [18], Chars74k_15 [19], ICDAR03_ch2 [20], and Bib digits [21] to test the each method's performance. The performance of the texture descriptor can easily be observed without a block overlap. Furthermore, block overlap and normalization are methods to improve an edge descriptor's robustness. Hence, we first tested a cell size of 8, normalized image size of 60 × 48 pixels with no block overlap and no case sensitivity.

As mentioned before, we tested the performance of a Gaussian filter instead of rectangle smoothing. First, we defined the image using a Gaussian filter as follows. And $G(p, \sigma)$ is defined as Eq. (5).

$$L(p, \sigma) = G(p, \sigma)*L(p)$$

(14)

Thus, the Gaussian version of ILBP₅ can be rewritten as follows.

$$ILBP₅(p) = \sum_{i=0}^{7} f(L(p, \frac{2(s - 1) + 1}{3})(p_{i+}), L(p, \frac{2s + 1}{3})p_{i})^2$$

(15)
Table 6
Recognition rate with different filters and interpolation on IIIT5K.

<table>
<thead>
<tr>
<th>Interpolation</th>
<th>Filter</th>
<th>Gaussian filter</th>
<th>Rectangle box filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>81.0%</td>
<td>82.2%</td>
<td></td>
</tr>
<tr>
<td>Not Active</td>
<td>81.1%</td>
<td>81.8%</td>
<td></td>
</tr>
</tbody>
</table>

Table 7
Recognition rate with different filters and interpolation onChars74k_15.

<table>
<thead>
<tr>
<th>Interpolation</th>
<th>Filter</th>
<th>Gaussian filter</th>
<th>Rectangle box filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>71.3%</td>
<td>71.3%</td>
<td></td>
</tr>
<tr>
<td>Not Active</td>
<td>71.3%</td>
<td>81.3%</td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 10](image) Each figure shows the recognition rate of each letter and its incorrect classified with color map. From left to right is the recognition result of ILBP and HOG on ICAR03_ch2.

Tables 6 and 7 show the result of combining the blur method and interpolation with a cell size of 8, scale size of 8, block size of 2, and block overlap of 1. The results show that ILBP with a rectangle box filter and without interpolation has an OCR recognition rate that is greater than or equal to each situation.

We analyzed the recognition rate of each letter and show the rates as confusion color maps in Fig. 10. The most errors occurred for the digit “0,” lower-case letter “o,” and upper-case letter “O.” The second most common errors involved the digit “1,” upper-case letter “I,” and lower-case letter “l.” Thus, we can observe that the error-prone characters are similar to those of HOG because ILBP is an edge-based descriptor. However, using first selection and maximum selection, ILBP is able to be more robust to noise, blur, and letters combined with dots. The left image of Fig. 10 shows the HOG recognition rate for each letter. From this figure, we can observe that the HOG and ILBP error rates of each letter are very similar. However, the final recognition rates of ILBP are similar or greater than that of HOG.

The computation cost of ILBP is S times that of original LBP for CPU-based computation, where S is the scale space size. In this study, we set our test image size to 60 × 48 pixels, and the computation times of ILBP, HOG, and the original LBP were 0.031, 0.019, and 0.014 s, respectively (on average for IIIT5K). To observe the computation time, we used 2048 × 2048 pixel images with a cell size of 8 and block overlap of 1. The average time for ILBP is 1.414 s. Furthermore, the computation time of HOG and LBP is 0.2642 s and 0.3143 s, respectively. However, when using the integral image, the computation time will be a constant when we search for a pattern in the whole image at each scale. The computation time could reduce to 0.812 s after we implement the ILBP code on CUDA architecture.

5. Conclusion and future work

In summary, we propose a robust edge descriptor based on a scale space search to find the best scale for LBP. We maintain liveness invariance and incorporate more information than LBP. ILBP is robust to blur and noise, and we have demonstrated its performance on four datasets. Clearly, the performance of the basic HOG method is successful as an edge descriptor. In addition, there have been many methods based on HOG to describe local information, such as SIFT and OCR based on bag-of-words. Thus, our future work will be to apply ILBP with a local descriptor. It will be interesting to investigate the use of SURF because both the methods are based on an integral image.

However, ILBP still has some problems to overcome. Our first problem is how to create scale space more efficiently. In the SIFT algorithm, the scales are separated by octaves and increase with a scale and not a constant value of 2. Hence, scanning scale space efficiently will be our next topic. In addition, there are many methods that use the bag-of-words model, so ILBP that is invariant to rotation is the second challenge. The original LBP achieves highly accurate results in face detection both for LBP and LBP because it detects luminance changes on the face. Hence, the third challenge is to test ILBP performance on face detection. In this study, we proved performance of the edge descriptor called ILBP, which is a new edge descriptor for pattern recognition.

Acknowledgements

This research is supported by the Ministry of Science and Technology, Taiwan, R.O.C grant no. MOST103-2221-E-006-145-MY3 and through national Academy of National Chen Kung University, Taiwan, R.O.C.

Reference