Article

Bilateral Line Local Binary Patterns for Face Recognition

Hung Phuoc Truong, Minh Bao Nguyen-Khoa, Yong-Guk Kim*

Department of Computer Engineering, Sejong University, Seoul, Korea;
* Correspondence: ykim@sejong.ac.kr

Abstract: Local binary pattern is one of the visual descriptors and can be used as a powerful feature extractor for texture classification. In this paper, a novel representation for face recognition is proposed, called it Bilateral Line Local Binary Patterns (BL-LBP). This scheme is an extension of Line Local Binary Patterns descriptors in the statistical learning subspace. The present bilateral descriptors are fused with an ensemble learning of calibrated SVM models. The performance of this scheme is evaluated using 5 standard face databases. It is found that it is robust against illumination variation, diverse facial expressions and head pose variations and its recognition accuracy reaches 98 percent, running on a mobile device with a processing speed of 63 ms per face. Results suggest that our proposed method can be very useful for the vision systems that have limited resources where the computational cost is critical.

Keywords: Bilateral Line Local Binary Patterns; Facial matrix; Statistical subspace; Face recognition; Calibrated SVM model; Ensemble learning

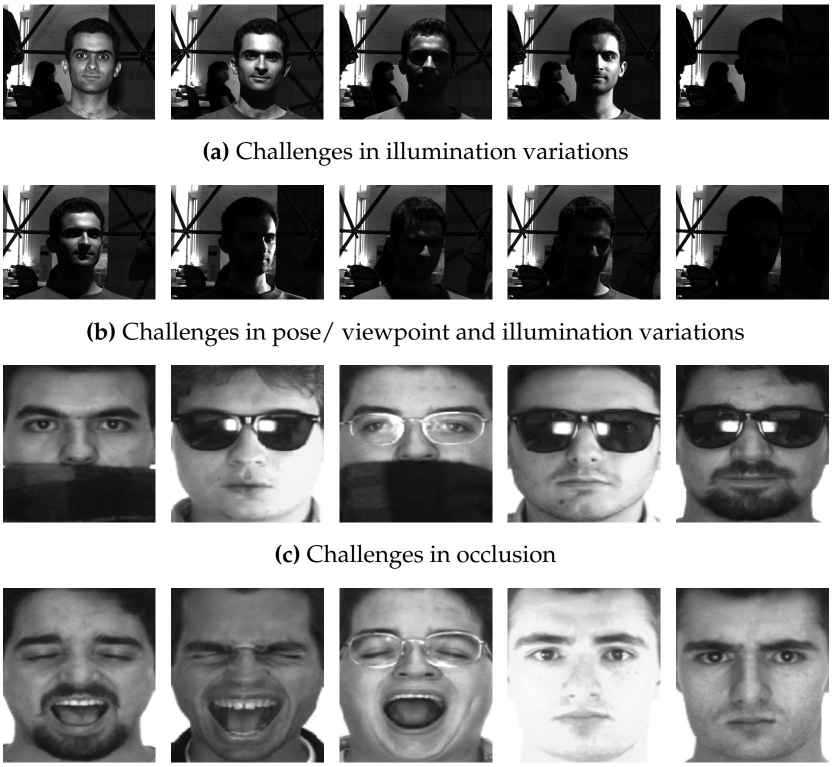
1. Introduction

Biometrics utilizes unique patterns of human body parts or its behaviors such as face, eye, hand, fingerprint, voice, signing for an identification purpose and it has been an important research area. Most of these biometric attributes maintain their characteristic over the lifetime. The question is how to pick them up safely and reliably because some of their setups are intrusive and the data acquisition is often difficult. In contrast, acquisition of facial data is relatively easy due to several factors such as non-intrusiveness, availability of diverse tools and low cost, and then face recognition becomes a viable technology in our modern life. Some interesting face recognition applications are introduced in public environments (driving license, military application, sporting event, airport, etc.) and in the private one (online service, banking, embedded application, mobile device security, etc) [1].

The standard face recognition system generally consists of three main stages: face detection, facial feature representation, and classification. The focus of the present study is on the facial representation stage. In fact, the key to successful face recognition should be how one can extract valuable and crucial facial features. The face recognition system cannot obtain good results when the input data are insufficient for any feature descriptors, even if the best classifier is utilized for the classification task. The aim of facial feature representation, of course, is to reduce the dimensionality of data while preserving their discriminative information for the classification task. The chosen descriptor should perform well in both criteria such as compactness and discriminative power. Finally, the processing speed has to be fast enough especially for mobile use. Many outstanding studies have been proposed and they can be grouped into two categories: the holistic-based methods and the local-based methods. The basic idea of holistic-based methods is to utilize all pixels on the whole face image. Several subspace methods such as Eigenfaces and its improvement [2], Principal Component Analysis (PCA)

[3], Fisher Discriminant Analysis (FDA) [4,5] and Sparse Representation-based Classification (SRC) [6] belong to the holistic approach. However, this approach encounters some limitations including several unconstrained cases such as illumination, head pose or facial expression. In contrast, the local-based approach which is known as feature-based method utilizes the local facial feature for image representation. It considers mainly the geometry structure of each patch within the given image. The obtained features of this arrangement often cover more useful information than the holistic-based case. Some examples are Gabor-wavelet features [7], Local Binary Pattern (LBP) features [8–11] or the fusion of LBP and Gabor [12]. For instance, the Gabor-wavelet feature is widely applied to either the whole face or some small face regions to describe facial features with a bank of Gabor filters. Gabor algorithm requires high computational resources since a large number of Gabor filters is required to compute and store multi-scale and multi-orientation coefficients. With this similar approach, LBP is known as a non-parametric method efficiently summarizing the local structures of an image and has been popular in facial image analysis and LBP descriptors are tolerant against illumination or scale changes while keeping its computational cost low. Since the facial recognition system is basically to classify a given face image according to the facial features extracted using the above methods, it adopts several representative classifiers such as k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Random forest, etc. at its last stage. For the best face recognition, a feature descriptor has to accompany with a high-performing classifier.

A lot of excellent works have been proposed to improve the performance of the system by solving some challenges in face problems. Recent challenges remain such as variations of facial expression, pose angle, illumination as well as background complexity (Fig. 1). Other difficulties are occlusion, aging, makeups, image quality, etc. It seems to be hard to have a method that can deal well with all these obstacles. Therefore, the face recognition problem is still an interesting topic.



(a) Challenges in illumination variations

(b) Challenges in pose/ viewpoint and illumination variations

(c) Challenges in occlusion

(d) Challenges in facial expression

Figure 1. Examples of several challenges can occur in face recognition

Among those challenges, illumination variation is one of the most difficult obstacles since it can change the appearance of the face through increasing or decreasing the intensities of face images by various light source directions. It requires a robust face representation not only works well with illumination variation but works well with facial expression. Thus, this study aims to overcome the drawback of representing face images against these variations. Inspired by the success of LBP for face recognition [13,14], we proposed a novel framework which is an extension of EL-LBP [15] called it BL-LBP (Bilateral Line Local Binary Pattern) for the face image presentation and recognition. This proposed framework aims to satisfy a number of challenging criteria:

1. *Robustness and Compactness*: The algorithm should be flexible and robust against variation of illumination, facial expression, pose and image quality.
2. *Processing Speed*: The time of the face representation stage and classification should be fast enough.
3. *Discriminant Capability*: For an unseen face image, the system can extract the efficient facial features and these features should represent for that image with a discriminant capability.

Figure 2 shows a conceptual diagram of our method for the face recognition system. With this scheme, we have two primary contributions. The first contribution is to build a group of descriptors for the feature representation task. The fusion of LLBP descriptors by horizontal and vertical direction with Bilateral 2D-PCA subspaces produces compact and robust descriptors that effectively adapt to various angles of the light sources of the image input. The second contribution is to apply an ensemble learning strategy based on the late fusion of SVM models for the classification task which leads to higher accuracy performance. The evaluation results suggest that our method outperforms state-of-the-art LBP approaches, including a deep learning approach.

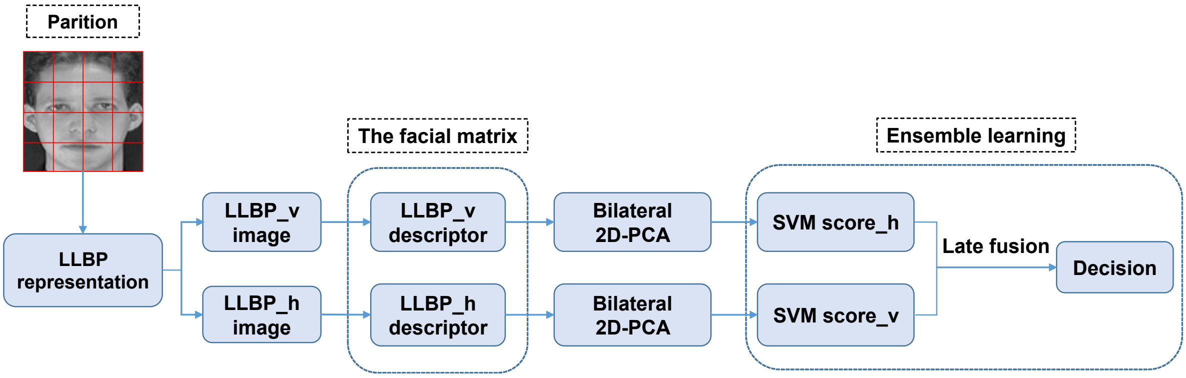


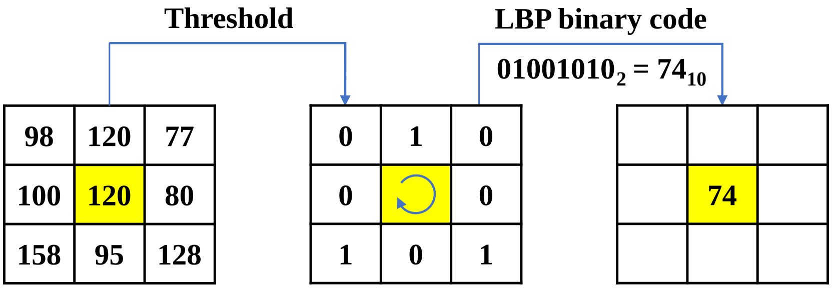
Figure 2. A conceptual diagram of our proposed BL-LBP method

The paper is organized as follows. Section 2 introduces briefly the principle of LBP and statistical subspace learning. Section 3 describes a few techniques on feature selection for the image representation since they are the main motivation of the present study. Our proposed framework will be described in Section 4. The experimental result using four face databases is presented in Section 5. In section 6, a conclusion is given.

2. Backgrounds

2.1. Local Binary Patterns (LBP)

The basic LBP operator encodes the relationship between a central pixel and its surrounding neighborhood 3×3 pixels to make a binary bit string. That bit string can be easily converted to its corresponding decimal value to produce the LBP code. The process of encoding an LBP bit string is illustrated in Fig. 3. The intensity of each pixel g_p in the neighborhood is compared to that of the central pixel $g_c(x_c, y_c)$. If g_p is either equal or greater than g_c, then it is encoded as 1 otherwise as 0. These binary codes are concatenated in the clockwise direction to build a binary number, and its corresponding decimal value is given as a label.



98	120	77
100	120	80
158	95	128

0	1	0
0		0
1	0	1

	74	

Figure 3. Process of the basic LBP operator

The original LBP operator can be extended by utilizing different sizes of a neighborhood to capture some dominant features underlying in the large scale structure [10]. The extended version of LBP can be described as Eq. (1), where (P, R) denotes the neighborhood of P sampling points on a circle of radius R at a given central pixel (x_c, y_c):

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P} 2^p s(g_p - g_c) \tag{1}$$

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

With this operator, LBP operator can work as invariant against the monotonic gray-scale transformations. Based on P pixels in the LBP neighborhood, 2^P different output values, corresponding to 2^P different binary patterns, are formed. Thus, we can construct a 2^p bins histogram of the LBP pattern as an occurrence frequency vector. However, there is a small subset of patterns, called "uniform patterns", which occur frequently and have the discriminative power. These special patterns contain at most two bitwise transitions from 0 to 1 or vice versa when the corresponding bit string is considered in a circular shape. With P = 8, there are 58 cases of these, and the counts for the remaining patterns are pooled into a single "non-uniform" bin, then LBP uniform (denoted as $LBP^{u2}_{P,R}$) obtains a 59-bins output histogram instead of a 256-bins histogram. From this definition, the number of different uniform output labels for mapping patterns of P bits in $LBP^{u2}_{P,R}$ is $P(P-1)+3$.

2.2. LBP Descriptor

For LBP-based face recognition, a face image is partitioned into m small regions $S^{(j)}$ $(j = 0..m-1$ and $m = k^2, k \in \mathbb{N})$ having the same size and non-overlapping regions where LBP descriptors can be extracted and then concatenated into a single feature vector to represent the texture of the given face image. A histogram for j^{th} local region $S^{(j)}$ is computed as follow:

$$H_i^{(j)} = \sum_{(x,y)\in S^{(j)}} T\{f_l(x,y) = i\}, i = 0, ..., L-1 \tag{2}$$

where $T\{A\} = \begin{cases} 1, & \text{if A is true} \\ 0, & \text{if A is false} \end{cases}$

in which i denotes the i^{th} LBP label in the range $[0, L-1]$, $f_l(x,y)$ is a LBP image corresponding to original image and $H_i^{(j)}$ is the number of pixels (with LBP label i) in the local region $S^{(j)}$.

With Eq. (2), the regional LBP descriptor for $S^{(j)}$ can be expressed:

$$\mathbf{H}^{(j)} = [H_0^{(j)}, H_1^{(j)}, ..., H_{L-1}^{(j)}]^T \tag{3}$$

where T stands for the matrix transpose operator and the standard column vector $\mathbf{H}^{(j)}$ belongs to L-dimensional real space. Note that in Eq. (3), $H^{(j)}$ provides regional LBP histogram information for

$S^{(j)}$. Finally, LBP histogram for all $S^{(j)}$ can be obtained by concatenating all $H^{(j)}$ into a single column vector:

$$\mathbf{H} = [(\mathbf{H}^{(1)})^T(\mathbf{H}^{(2)})^T...(\mathbf{H}^{(m-1)})^T]^T \tag{4}$$

The LBP histogram H is applied to extract LBP feature descriptor of the given face image. Further detail for this process will be described in the following subsection.

3. LBP Feature Selection

For face recognition using the LBP based method, many studies propose to boost the discriminative capability within feature vectors such as: (1) increase the number of blocks within an image; (2) increase the number of neighbors to capture more dominant textures; (3) combine LBP with other features extraction method. However, these methods lead to a significant increase in feature vector length as well as a redundant representation including unnecessary information [16]. In addition, the issues for LBP feature selection have been studied. According to a recent survey [17], LBP feature selection techniques are classified into two main categories: (1) exploitation of some feature extraction techniques to find the best discriminant feature, although such initiative could boost LBP feature capacity it requires an off-line training processing; (2) reducing the feature-length of LBP descriptors by using some rules, although this approach has limited feature selection ability and yet it is simple and has the low computational cost.

3.1. LBP subspace learning

Subspace learning aims to reduce the dimension of face images by projecting the original data into a new subspace wherein one can obtain an optimal and lower-dimensional representation [18,19]. It tries to boost LBP features by combining many individual LBP features with other holistic/ local features to take advantage of each feature type without caring about the dimension. Then, several well-known dimension reduction methods such as PCA, FDA and Gabor wavelet are applied to obtain a low-dimensional compact representation.

For instance, Linear Discriminant Analysis (LDA) [20] was utilized to convert a high dimensional space using multi-scale LBP features into a discriminant space, often yielding very promising results. Although a combination of the Gabor wavelet and LBP feature could lead to a high dimensional space, it was successfully performed by Tan and Triggs [21] by using PCA to obtain an optimal space. The Kernel Discriminative Common Vectors (KDCV) was applied to extract discriminant nonlinear compact features for face recognition. Zhao et al. [22] introduced a fusion of Laplacian PCA (LPCA) and LBP for feature selection by proving that LPCA outperformed PCA and KPCA.

3.2. Rule-based strategy

The rule-based strategy aims to reduce the dimension of feature vectors within the algorithm of LBP using a certain rule. For instance, the number of LBP labels such as Local Line Binary Patterns (LLBP) [14] or Center-Symmetric Local Binary Patterns (CS-LBP) [23] can be reduced for a compact histogram while considering the large neighborhood. Or, selecting a subset of available bins instead of using the whole set of patterns is another way.

Lahdenoja et al [24] introduced a new concept of symmetry for dimensionality reduction as an LBP-based face descriptor. Here, patterns were assigned different levels of symmetry based on the number of ones and zeros they contain. A high level of symmetry patterns proved to occur more frequently in facial features having the best discriminative efficiency. These patterns seem to have a certain invariance property against environmental factors such as illumination or rotation. In [25], the authors showed that a properly selected subset of patterns encoded in LBP forms an efficient and robust texture description which can achieve better classification rates in comparison with the whole LBP histogram. They introduced a method based on a beam search. The classification of the optimization set within each iteration was made using a reduced LBP histogram that contained one bin for each pattern chosen so far. The contributions from all remaining patterns (out of 256) were combined into a single bin. In their experiments, the method based on feature selection by beam search performed better than the whole set of patterns or the uniform patterns when classifying tilted textures and using non-tilted samples for training. Smith et al. [26] proposed a fusion method by using LBP to generate a large number of candidate features and then selecting from them using fast correlation-based filtering (FCBF). FCBF operates by repeatedly choosing the feature that is most correlated with class, excluding those features already chosen or rejected, and rejecting any features that are more correlated with it than with the class. As a measure of correlation, the information-theoretic concept of symmetric uncertainty is used. When applied to the LBP features, FCBF reduced their number from 107,000 down to 120. Guo at el. [27] presented a learning framework which is formulated into a three-layered model. By a simultaneous consideration of the robustness, discriminant power and representation capability of features, this model can produce the optimal pattern subset of interest. With this improvement, this learning model can be integrated with existing LBP variants to derive new image features for texture representation and classification.

Our present approach is a fusion of LBP-based rule and subspace learning to boost the discriminant of these compact descriptors.

4. The proposed approach

4.1. Line Local Binary Patterns (LLBP)

The algorithm of LLBP considers only the pixel neighbors of a given central pixel in the horizontal or the vertical direction instead of the grid shape of the neighborhood. The basic idea is illustrated in Fig. 4 and it is similar to the original LBP but there are some differences:

1. The shape of neighborhood is a straight line either in the horizontal or the vertical direction.
2. For a given central pixel, the number of neighbors to the horizontal direction should be identical to that in the vertical direction. If the number of one direction is less than that of the other direction, a 0-padding mask is inserted to balance the neighbor to make the total number, called it N, to be even.
3. A bit string comes by comparing all neighbors to the central value: the binary bit is to be 1 if the neighbor pixel is greater or equal to the current element, otherwise it is to be 0. Each element in the bit string is multiplied by weight according to its position (described in Fig. 4, Eq. (5) and Eq. (6)) and the current pixel is replaced by the sum of all values.

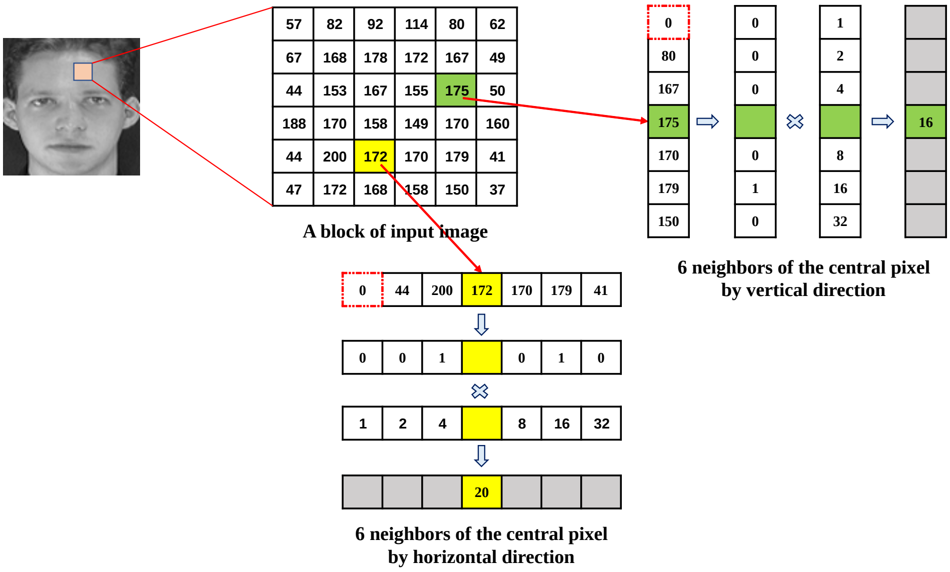
$$LLBP_h(N,c) = \sum_{n=0}^{N-1} 2^n s(h_n - g_c) \tag{5}$$

$$LLBP_v(N,c) = \sum_{n=0}^{N-1} 2^n s(v_n - g_c) \tag{6}$$

4. The range of values in LLBP representation (A_{LLBP}) must be in a range from 0 to 255 by Eq. (7)

$$\begin{aligned} \mathcal{A}_{LLBP}^{short} &= f(a_{i,j} | \forall a_{i,j} \in A_{LLBP}) \\ &= \begin{cases} a_{i,j} \bmod 256, a_{i,j} > 255 \\ a_{i,j}, \quad 0 \leq a_{i,j} \leq 255 \end{cases} \end{aligned} \tag{7}$$

where N is the number of neighbors, c is a given central pixel, g_c is the value of center c, h_n is neighborhood pixel on the horizontal line and v_c is the neighborhood pixel on the vertical line, and s function is same as in LBP. Then, this descriptor is defined by the histogram of LLBP patterns and the number of bins depends on the size of the neighborhood. For instance, 8 neighbors yield 256 bins in

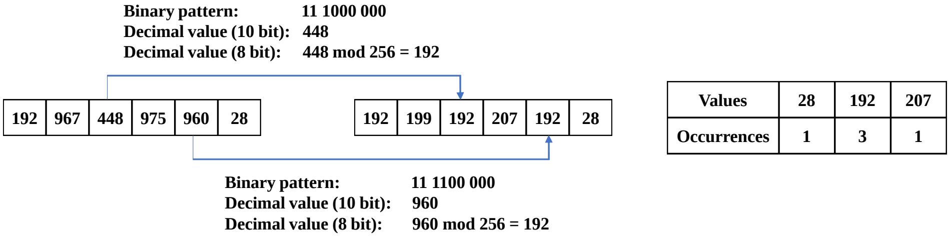


57	82	92	114	80	62
67	168	178	172	167	49
44	153	167	155	175	50
188	170	158	149	170	160
44	200	172	170	179	41
47	172	168	158	150	37

A block of input image

0
80
167
175
170
179
150
0
0
0
0
1
0
1
2
4
8
16
32
16
6 neighbors of the central pixel by vertical direction
0 44 200 172 170 179 41
0 0 1 0 1 0
1 2 4 8 16 32
20
6 neighbors of the central pixel by horizontal direction

Figure 4. An example of LLBP operator with 6 neighbors corresponding to a 6-bits string considered by the vertical and horizontal direction, respectively



Binary pattern: 11 1000 000
Decimal value (10 bit): 448
Decimal value (8 bit): 448 mod 256 = 192

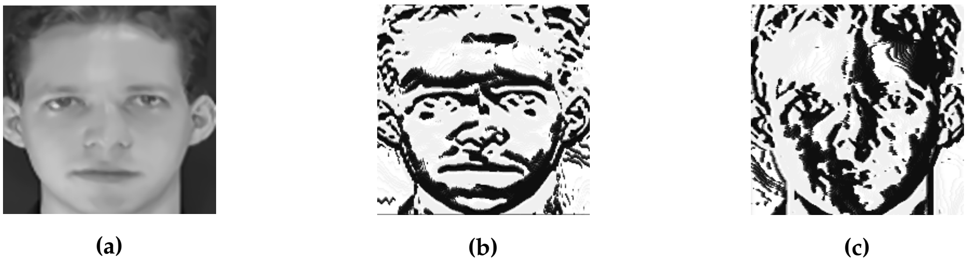
Values	28	192	207
Occurrences	1	3	1

Binary pattern: 11 1100 000
Decimal value (10 bit): 960
Decimal value (8 bit): 960 mod 256 = 192

Figure 5. An example of converting values in 10-bits string to decimal values in a new range of values by using Eq. (7)

the histogram. However, since the number of bins in this way is too long, we propose a strategy to reduce the number of bins and described its detail in the next section.

The algorithm of LLBP calculates the binary code along the horizontal and the vertical direction separately. Figure 5 shows an example of step 4 in the LLBP algorithm to convert a bit-string to decimal values in a new range of values. Thereafter, the example of LLBP images for these directions is shown in Fig. 6.



(a) (b) (c)

Figure 6. Example of face image processed by LLBP operator: (a) is an original image, (b) and (c) are LLBP images in the vertical and horizontal direction of (a), respectively.

4.2. The facial matrix (LLBP descriptor)

After applying the LLBP operation on an input image, as a result, we obtain two LLBP images by horizontal and vertical direction. Each of them is manipulated to produce an LLBP descriptor respectively, called it the facial matrix as displayed in Fig. 7.

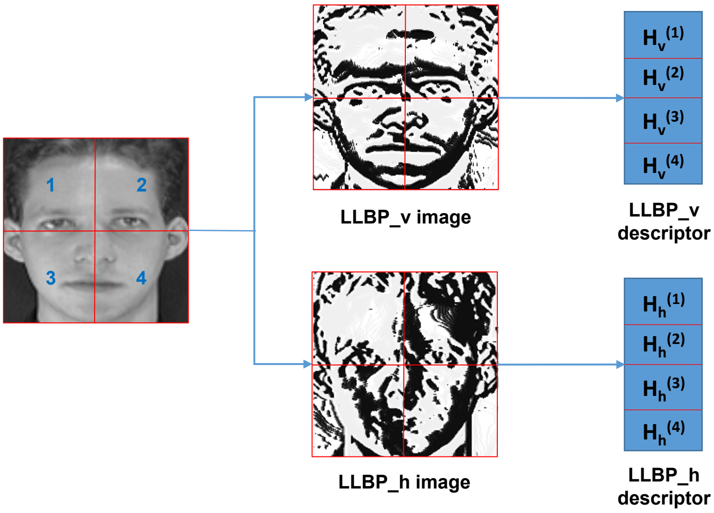


Figure 7. An example of producing LLBP descriptor through a fusion of histograms $H^{(j)}$. This fusion makes two matrices by horizontal and vertical direction called facial matrices. In this case, the partition of the image with parameter m = 4.

Firstly, we find all uniform patterns corresponding to the number of neighbors and convert all of them into decimal numbers as shown in Table 1. A local binary pattern is called 'uniform' if its uniformity measure is at most 2. Here, BIT denotes the number of bits in a bit string (the number of neighbors) and BIN indicates the number of bins that can be obtained by counting LLBP patterns. As Eq. (7) only covers LLBP values from 0 to 255, we consider the range from 2 to 8 and the neighborhood must be an even number. With this strategy, we obtain 57 bins when the number of the neighborhood is 8, whereas the traditional LBP uniform approaches produce 59 bins.

Table 1. All values of uniform patterns in different neighborhoods

BIT	The range of values between bins in the histogram	BIN
2	0, 1, 2, 3	3
4	0, 1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 15	13
6	0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 33, 35, 39, 47, 48, 49, 51, 55, 56, 57, 59, 60, 61, 62, 63	31
8	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, 255	57

Secondly, a histogram of each local region is obtained by Eq. (2), Eq. (3) and the strategy of Table 1. However, for producing facial matrix, the Eq.(4) is replaced by a new calculation as below:

$$\mathbf{H}_{m\times BIN} = [\mathbf{H}^{(1)}\mathbf{H}^{(2)}...\mathbf{H}^{(m-1)}]^T \tag{8}$$

Equation (8) produces a 2D matrix H with the size of $m \times BIN$, where m is the number of local regions of partition. This calculation preserves the spatial information of local regions instead of concatenating all histograms into a huge column vector.

4.3. Bilateral 2D-PCA Subspace Learning

2D-PCA method [28] is a well known dimension reduction method since it has two important advantages over PCA: (1) it is easier to compute the covariance matrix accurately; (2) less time is required to determine the corresponding eigenvectors. A bilateral 2D-PCA[28,29] is a variation where one encodes the spatial information embedded in both column and row directions of facial matrix.

Let $\{A_k\}_{k=1}^{M}$ be a training set, where A_k denotes k^{th} an $(m \times BIN)$ training sample and M the total number of training set. The average matrix is denoted $\overline{A} = \frac{1}{M}\sum_k A_k$. Then a non-negative definite matrix $G^{row}_{BIN\times BIN}$ by the row direction can be evaluated as follow:

$$G^{row} = \frac{1}{M}\sum_{k=1}^{M}(A_k - \overline{A})^T(A_k - \overline{A}) \tag{9}$$

Similarly, we consider the column direction of FM matrix. Let $A_k = [A_k^{(1)}A_k^{(2)}...A_k^{(m)}]$, where $A_k^{(j)}$ and $\overline{A}^{(j)}$ denote the j^{th} column vectors of A_k and $\overline{A}$, respectively. Then the covariance matrix $G^{col}_{m\times m}$ by column direction can be evaluated as follow:

$$G^{col} = \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{n}(A_k^{(j)} - \overline{A}^{(j)})(A_k^{(j)} - \overline{A}^{(j)})^T \tag{10}$$

We obtain 2 optimal projection matrices $X_{opt} = [X_1, ..., X_d]$ and $Z_{opt} = [Z_1, ..., Z_q]$ by computing the eigenvectors of Eq. (9) and Eq. (10) corresponding to the d and q largest eigenvalues, respectively. The Bilateral 2D-PCA learns two optimal matrices X and Z from a set of FM training descriptors, and then project input matrix A onto X and Z respectively, yielding coefficient matrices Y^{row} and Y^{col}

$$Y_{q\times d} = Z^T_{q\times m}A_{m\times BIN}X_{BIN\times d} \tag{11}$$

The vector $Y_{q\times d}$ in Eq. (11) represents an optimal feature of input matrix $A_{m\times BIN}$. Therefore, with two LLBP descriptors by horizontal and vertical direction, we obtain two optimal features by directions respectively:

$$Y^h_{q1\times d1} = (Z^h)^T_{q1\times m} \times (H^h)_{m\times BIN} \times (X^h)_{BIN\times d1} \tag{12}$$

$$Y^v_{q2\times d2} = (Z^v)^T_{q2\times m} \times (H^v)_{m\times BIN} \times (X^v)_{BIN\times d2} \tag{13}$$

4.4. Ensemble learning

The importance of facial features and classifiers has been emphasized above and selecting the right classifier among many plays a crucial role in achieving a good result [30,31]. Since it is known that SVM works very efficiently in the multi-classification case with the presence of various constraints, we adopt an SVM based calibrated classification model.

Let the output of a learning method be f(x). To get the calibrated probabilities, pass the output through a sigmoid:

$$P(y = 1|f) = \frac{1}{1 + exp(Af + B)} \tag{14}$$

where the parameters A and B are fitted using the maximum likelihood estimation from a training set $(f_i; y_i)$. Gradient descent is used to find A and B by minimizing the negative log likelihood of the training data:

$$argmin_{A,B}\{-\sum_i y_i log(p_i) + (1 - y_i)log(1 - p_i)\} \tag{15}$$

where

$$p_i = \frac{1}{1 + exp(Af_i + B)} \tag{16}$$

The Platt algorithm [32,33] is used for the mapping model prediction to the posterior probabilities by an additional sigmoid function. However, to avoid overfitting to the sigmoid train set, output of the sample model is used. If there are N_+ positive examples and N_- negative examples in the training set, the Platt calibration uses the target values y_+ and y_- (instead of 1 and 0), respectively, for each training example, where

$$y_+ = \frac{N_+ + 1}{N_+ + 2}; y_- = \frac{1}{N_- + 2} \tag{17}$$

Individual Calibrated SVM models are trained with LLBP descriptors by vertical and horizontal direction respectively. Each model obtains a prediction vector $P = (p_1, p_2, p_j, ..., p_n)^T$, where p_j is the probability which belongs to j^{th}-class of model and n is the number of classes for classification. The fusion is operated as below:

$$\begin{aligned} \bar{P} = (\bar{p}_1, \bar{p}_2, \bar{p}_j, ..., \bar{p}_n)^T &= 1/2\sqrt{(P^h)^2 + (P^v)^2} \\ &= 1/2\sum_{j=1}^{n}\sqrt{(p_j^h)^2 + (p_j^v)^2} \end{aligned} \tag{18}$$

where P^h and P^v are probabilities of calibrated SVM classifiers by horizontal and vertical direction respectively. Thereafter, a majority voting is considered for the fusion vector and the final decision depends on the class with the highest probabilities.

4.5. The BL-LBP features algorithm and its characteristic

This section summarizes the proposed method as Algorithm 1 below.

Algorithm 1 The BL-LBP features algorithm

Input: A matrix of image input $\mathbf{A} \in R^{m \times n}$; the parameter of dividing m blocks $\mathbf{k}$; the number of neighbors in LLBP patterns **BIT**l; the number of principal components in 2D-PCA q and d
Output: BL-LBP feature: Y^h_{qxd} and Y^v_{qxd}

```
Partition input image A into $m$ blocks ($m = k^2$)
Initialize two empty matrices $H^h = size(m \times BIN)$ and $H^v = size(m \times BIN)$
FOR each block $i^{th}$ in the image
    Transform the gray block into new representations $(A_h)^{(i)}_{LLBP}$ and $(A_v)^{(i)}_{LLBP}$ domain according to Eq. (5) and Eq. (6), respectively.
    Convert all values of the matrix $(A_h)^{(i)}_{LLBP}$ and $(A_v)^{(i)}_{LLBP}$ into the new matrices $(A_h)^{(i)-short}_{LLBP}$ and $(A_v)^{(i)-short}_{LLBP}$ domain according to Eq. (7)
    Count the occurrence frequency using Eq. (2), Eq. (3) and BIN series of intervals $S$ which are consecutive and non-overlapping intervals of values in Table 1. Considering two directions, we obtain $H^{(i)}_h$ and $H^{(i)}_v$.
    $H^h = H^h \cup H^{(i)}_h$ by using Eq. (8)
    $H^v = H^v \cup H^{(i)}_v$ by using Eq. (8)
END FOR
Compute BL-LBP features $Y^h$ and $Y^v$ according to Eq. (12) and Eq. (13).
```

The BL-LBP feature has several advantages over traditional LBP methods. Given that BL-LBP is an extension of the LLBP method, it inherits the characteristics of LLBP and so it is robust against changes of illumination and scale, especially when a source of light lies in one part of a face (either left or right and either top or bottom). The overflow transformation of Eq. (7) and counting the occurrence frequency of the value in matrix $A_{LLBP}^{(i)-short}$ based on series of intervals would produce a descriptor with a constant size, even though the number of neighbors varies. Although dividing the original image into many smaller blocks can lead to the larger dimension, an arrangement is made to fuse sub-histograms for a facial matrix. The computation cost of the Bilateral 2D-PCA on the facial matrix is handy and it is faster than any method that uses a long vector. By using 98% Contribution Rate of Eigenvalues (CRE) in the Bilateral 2D-PCA, principal components of the row and column direction subspace are kept and yet the number of principal components is very small. Such maneuvering makes this feature descriptor efficient and fast for the classification task.

5. Experiments

5.1. Configuration for experiments

Our BL-LBP scheme was evaluated using four standard face databases such as ORL, YALE, AR, and Aberdeen as shown in Fig. 5. Each database was divided into training and testing set and each set occupied 50% of the total samples. The model ran on both a PC and a mobile device. The configurations for both cases are detailed in Table 2. The experiment for a PC was prepared to evaluate its performance, while that for a mobile device was for the real-time face recognition application.

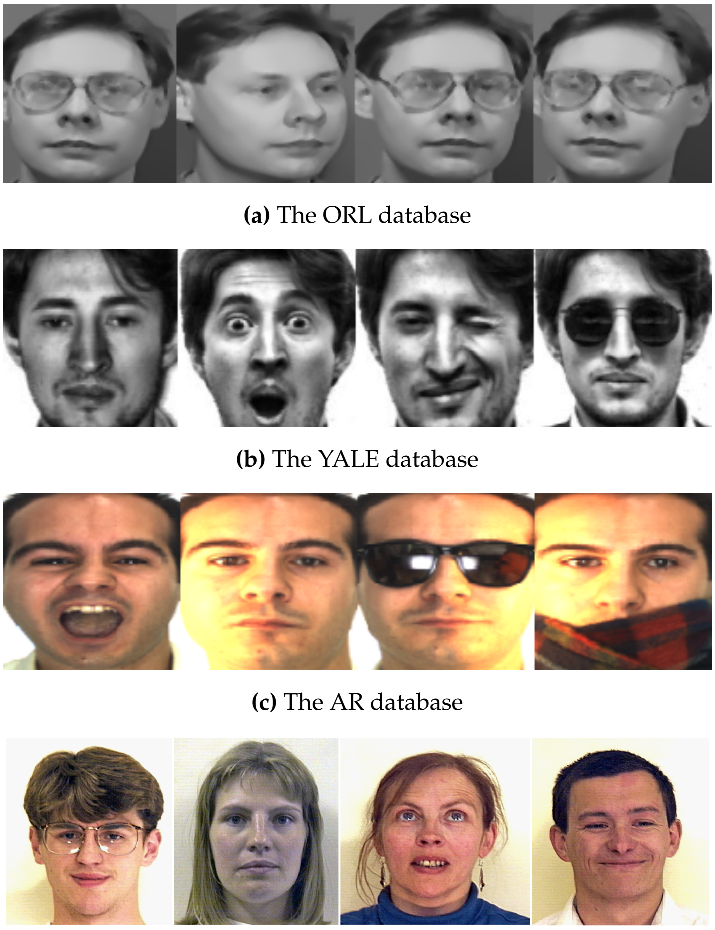
Table 2. Hardware and Software Configuration

Item	PC	Mobile device (Galaxy S8)
Library	OpenCV 3.4.1, Scikit-Learn, Scikit-Image, Numpy	OpenCV 3.4, Numpy
Language	Python 3.6.5 version	Java (Android Studio 3.2.1)
OS	Bionic Beaver (Ubuntu 18.0 LTS)	Oreo (Android 8.0.0)
CPU	Intel Quad Core i5-7600 CPU @ 3.5 GHz (64 bits)	Octa-core (2.3 GHz Mongoose M2 Quad + 1.7 GHz Cortex-A53 Quad) (64 bits)
GPU	GeForce GTX 1060 3 GB	-
Memory	32 GB RAM	4 GB RAM

5.2. Database sets

40 subjects participated for ORL database and each subject had 10 images (92 × 112 gray image) since there were different facial expressions (open/ close eyes, smile/ not smile), several conditions of illumination, facial details (hairstyles or with/ without a beard, and with glasses/ without glasses), diverse head poses (the angle of tilting from 0^o up to 20^o), respectively. The YALE Face images were taken from 15 individuals and each person had 11 images (243 × 320 pixels) under different expressions and lighting conditions. In AR database, 3016 face images for 116 persons (63 men and 53 women) were taken and each person had 26 images in the RGB color format (768 × 576 pixels) under dramatic changes of lightning (left-light, right-light or all sidelights), expressions (happy, sad, neutral, sleepy, anger, surprised and wink), poses and occlusion including glasses and bangs. There were 625 images corresponding to 65 people's faces in the Aberdeen database which was strictly controlled in different lightning and expression conditions. Figure 8 illustrates some images on four databases.

A cross-validation method was designed for the measurement of the accuracy and processing time for each database. Half of the samples were randomly selected for training and the remaining half for testing, and this evaluation was repeated 100 times by shuffling data. Therefore, recognition



(a) The ORL database

(b) The YALE database

(c) The AR database

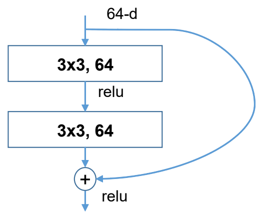
(d) The Aberdeen database

Figure 8. The four standard face databases used for the present study

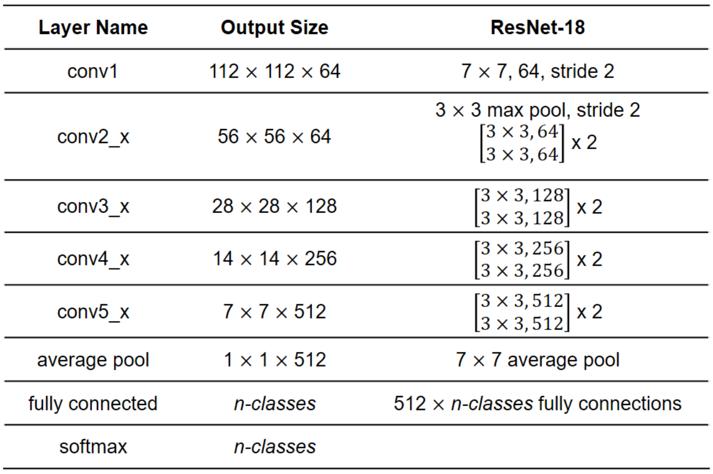
results were the average values for 100 random permutations. Moreover, each original image was preprocessed by applying denoising techniques for ORL and by applying histogram equalization for the others after converting images into gray levels, respectively.

5.3. Face recognition based on deep features

For a fair evaluation between methods, an experiment based on deep features is carried. With the advantages of pre-trained deep learning models on the general ImageNet database which has more 1.2 million images with 1000 categories, the transfer learning of deep models has been applied to many recent application regarding face images such as Race recognition [34], or smile detection [35]. Therefore, a fine-tuned strategy of pretrained model ResNet-18 [36] is considered in this experiment for comparisons. The network is a good trade-off between depth (compared to AlexNet, VGG-16, ResNet-50) and performance. The ResNet-18 architecture which is summarized in Fig. 9 consists of 18 layers. The size of feature vector of the ResNet-18 model is 512 at the average pool layer. The input of the network is an RGB image of size 224 × 224. For our classification task, we modify the network by adding average-pooling and softmax layers to the origin ResNet model. The optimization goal is to maximize the accuracy performance by a SGD optimizer with a learning rate of 0.01. The input images are also augmented such as rotation, flipping, scaling for overfitting avoidance. The results of several datasets are reported in the next section as a comparison.



(a) A residual block of ResNet-18 network



Layer Name	Output Size	ResNet-18
conv1	112 × 112 × 64	7 × 7, 64, stride 2
conv2_x	56 × 56 × 64	3 × 3 max pool, stride 2 $\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}$ x 2
conv3_x	28 × 28 × 128	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}$ x 2
conv4_x	14 × 14 × 256	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}$ x 2
conv5_x	7 × 7 × 512	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}$ x 2
average pool	1 × 1 × 512	7 × 7 average pool
fully connected	*n-classes*	512 × *n-classes* fully connections
softmax	*n-classes*	

(b) The detail of ResNet-18 network

Figure 9. ResNet-18 architecture

5.4. Experiment 1: Accuracy performance

In this section, BL-LBP scheme was evaluated and compared to the previous LBP methods as well as deep learning. The recognition rate of several methods were shown in the Table 3 where our BL-LBP showed the highest accuracy for several databases and it reached 98%.

Table 3 shows that the recognition rate of LLBP is higher than those of $LBP^{u2}_{8,1}$ and $LBP^{u2}_{16,2}$. For effective descriptors of the face recognition, each image of LBP methods was partitioned into $m(m = k^2)$ regions, thus the dimension of feature descriptor was very huge $m \times q$ (q is the number of bins in the histogram). For instance, $m = 49$, the feature descriptor for $LBP^{u2}_{8,1}$ required 2891 (= 49 × 59) bins and for $LBP^{u2}_{16,2}$ required 9555 (49 × 195) bins. The dimension of LLBP which was described in parentheses corresponding to each database could reach to 25600 (= 1024 × 5^2) bins. However, BL-LBP, which were extended by LLBP, overcame this drawback since it required very few feature descriptors since the longest feature descriptor had only 728 (= 13 × 56). Without caring the parameter k and $BINS$, BL-LBP descriptors considered two principal component directions, thus its dimension was reduced by at least 3 times compared to LLBP. Compared to the deep learning approach (ResNet-18), our BL-LBP framework produced higher accuracy performance on YALE, AR, and Arberdeen databases except for a little bit worse accuracy performance on ORL. Generally speaking, BL-LBP outperformed the others and it ran very well on an embedded machine.

Table 3. The recognition rate of several methods on four databases

Methods	**ORL**	**YALE**	**AR**	**Aberdeen**
$\text{LBP}^{u2}_{8,1}$	82.50	80.10	87.50	81.60
$\text{LBP}^{u2}_{16,2}$	86.10	83.40	89.20	83.93
LLBP	95.41 (16 × 5^2)	81.75 (1024 × 5^2)	92.60 (256 × 5^2)	93.48 (1024 × 5^2)
ResNet-18	**98.11**	91.46	93.82	93.75
BL-LBP	97.42 (3 × 43)	**97.41** (10 × 54)	**98.33** (13 × 56)	**98.84** (10 × 33)

Among four standard databases, BL-LBP worked the most efficiently on YALE since its recognition rate could reach 97.41% with very few feature descriptors (540 = 10 × 54) while the ResNet-18 with feature vector size of 512 could reach 91.46%. Many YALE images were obscured on eyes or noses by dark glasses and darkness (various light source directions), thus important information of images

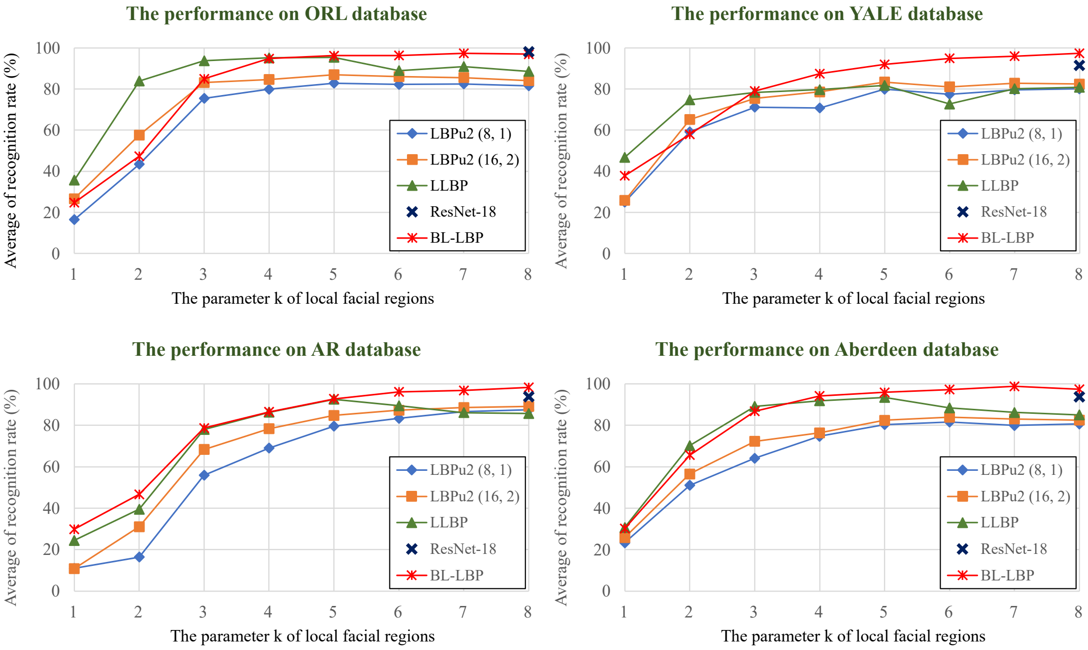


Figure 10. The results of several methods on four standard databases

was lost, making it a tough database. The larger databases with facial expressions such as AR and Aberdeen databases were actually a challenge for face recognition. Besides expression variation, Aberdeen database was made under strict lighting conditions while AR database was collected by changing head poses extremely. However, BL-LBP was on the top in term of accuracy: 98.33% on AR and 98.84% on Aberdeen, respectively. Also, it handled well the pose variation problem of the ORL database. Such result suggests that BL-LBP has some characteristics that is robust against the facial expression variations and the extreme head poses.

For the details, we explored the accuracy of various methods by considering $m = k^2$ local facial regions as shown in Fig. 10 while the input image of the deep learning approach was not partitioned. Therefore, we showed only the result of deep learning as a point to compare methods. In fact, the accuracy of each method dramatically increased when the value of parameter k was in a range of 1 to 5, and yet that of BL-LBP slightly increased. However, BL-LBP was always on the top accuracy when the value of parameter k was in the range of 6 to 8. In this case, BL-LBP outperformed LLBP and the others in terms of the recognition rate and the dimension of feature descriptors.

5.5. Experiment 2: Evaluation on an extension of YALE face database under illumination variations

The extension of YALE face database [37] is a combination of the YALE face database B with 10 subjects and the extended YALE face database B with 29 subjects. Both databases contain face images under 64 illumination conditions with 9 poses. In this experiment, we conduct only frontal face images for each subject with various illumination, thus this data contains 2,467 frontal face images of 39 subjects. The database which is divided into 5 subsets is displayed in Table 4. The goal of this split is to check the tolerance of several methods according to various angles of the light source.

The impact of illumination variations was evaluated by picking two images/subject from all subsets for the training set, 5 subsets with the remaining images were 5 test sets. Table 5 compared the results of each subset as well as the overall performance. LLBP method showed the worst result among methods since its descriptors were huge and contained redundant information and decreased the performance. Contrastly, the feature descriptors of BL-LBP and ResNet-18 were rather small and can produce higher accuracy. Our BL-LBP scheme obtained the highest and lowest accuracy on Subset 2 and Subset 5, respectively. With Subset 5, the accuracy of the deep learning approach was a little bit

Table 4. The extension of YALE face database

Image set	Illumination angle	Quantity	Description
Subset 1	$0^0 - 12^0$	270	7 face images/subject, discarded 3 corrupted images
Subset 2	$13^0 - 25^0$	468	12 face images/subject
Subset 3	$26^0 - 50^0$	466	12 face images/subject, discarded 2 corrupted images
Subset 4	$51^0 - 77^0$	538	14 face images/subject, discarded 8 corrupted images
Subset 5	$> 78^0$	725	19 face images/subject, discarded 16 corrupted images

better than BL-LBP. However, the overall performance of BL-LBP was higher. This experiment proved that our BL-LBP scheme can be tolerant of illumination variations very well and works better than the deep learning approach (ResNet-18).

Table 5. The recognition rate on extension of YALE face database

Methods	Subset 1	Subset 2	Subset 3	Subset 4	Subset 5	Overall
LLBP	78.41	93.63	74.7	76.55	70.36	78.73
ResNet-18	91.6	97.82	91.7	84.19	83.53	89.77
BL-LBP	97.65	100	93.64	87.5	82.61	**92.29**

5.6. Experiment 3: Processing Time

The processing speed for the BL-LBP scheme was measured using a PC and a mobile device (Galaxy S8). The processing time is an important factor for the real-time application. As the mobile devices have a limited computational resource, the processing time is considered as a crucial criteria. For a fair comparison, all experiments was carried out under the similar conditions for both machines. The detail information for libraries and hardware specifications is given in Table 2.

Table 6 described the total processing time for the whole process of face recognition, i.e the average time for a testing query on an image on several methods. Note that the average time for $LBP^{u2}_{16,2}$, $LLBP$ and $LBP^{u2}_{8,1}$ was 193, 153, and 106 ms, respectively, whereas that for BL-LBP was 63 ms, indicating that BL-LBP was 2.4 times faster than LLBP. The deep learning approach was faster than BL-LBP on the PC machine, mainly because ResNet-18 ran on a GPU for its manipulation while BL-LBP on a CPU. The processing time of ResNet-18 on a mobile device could not report here since this deep learning model was not fitted for a machine with limited resources. Under the CPU configuration, the processing time of our BL-LBP was the fastest one.

Table 6. The processing time (ms) of each method for testing images

Methods	$\mathbf{LBP}^{u2}_{8,1}$	$\mathbf{LBP}^{u2}_{16,2}$	LLBP	ResNet-18	BL-LBP
Time on PC	18	35	29	**8**	**12**
Time on Mobile	106	193	153	-	**63**

6. Conclusions

Local Binary Pattern is one of the favored visual descriptors in recent years. It has been used as a powerful feature extractor for diverse computer vision problems. In this paper, a novel BL-LBP scheme is proposed for face image recognition which is easily suited to a mobile application. The present encoding scheme for face recognition is extended from Line LBP descriptors where bilateral statistical learning is applied for efficient descriptors. Such descriptors are then boosted with an ensemble learning strategy of SVM models for the classification task. Performance evaluation using 5 face image databases suggests that the BL-LBP scheme is robust and flexible under various conditions such as variations of illumination, extreme head poses, scale change, and facial expressions. The proposed scheme outperforms the previous LBP-based methods as well as a deep learning approach. The major advantage of it is that it allows making bilateral feature descriptors from a raw image. Since the dimension of feature is smaller than other LBP methods, this representation scheme makes a better sense in the mobile vision system where the computational resources are critical as we demonstrate here.

Author Contributions: Conceptualization, H.P. Truong and Y.G. Kim; methodology, H.P. Truong and Y.G. Kim; writing—original draft preparation, H.P. Truong, B.M. Nguyen-Khoa and Y.G. Kim; software, H.P. Truong and M.B. Nguyen-Khoa; writing—review, editing, and funding Y.G.Kim. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00731).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chihaoui, M.; Elkefi, A.; Bellil, W.; Ben Amar, C. A Survey of 2D Face Recognition Techniques. *Computers* **2016**, *5*.
2. M.uge Carikci and Figen Ozen. A Face Recognition System Based on Eigenfaces Method. *Procedia Technology* **2012**, *1*, 118 – 123. First World Conference on Innovation and Computer Sciences (INSODE 2011).
3. Le, D.T.; Truong, H.P.; Le, T.H. Facial expression recognition using statistical subspace. 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 5981–5985.
4. Huang, P.; Gao, G.; Qian, C.; Yang, G.; Yang, Z. Fuzzy Linear Regression Discriminant Projection for Face Recognition. *IEEE Access* **2017**, *5*, 4340–4349.
5. Mi, J.; Liu, T. Multi-step linear representation-based classification for face recognition. *IET Computer Vision* **2016**, *10*, 836–841.
6. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2009**, *31*, 210–227.
7. Abhishree, T.; Latha, J.; Manikantan, K.; Ramachandran, S. Face Recognition Using Gabor Filter Based Feature Extraction with Anisotropic Diffusion as a Pre-processing Technique. *Procedia Computer Science* **2015**, *45*, 312 – 321. International Conference on Advanced Computing Technologies and Applications (ICACTA).
8. Ahonen, T.; Hadid, A.; Pietikainen, M. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2006**, *28*, 2037–2041.
9. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004; Pajdla, T.; Matas, J., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2004; pp. 469–481.
10. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2002**, *24*, 971–987.
11. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition* **1996**, *29*, 51 – 59.
12. Bankar, P.V.; Pise, A.C. Face recognition by using Gabor and LBP. 2015 International Conference on Communications and Signal Processing (ICCSP), 2015, pp. 0045–0048.
13. Benzaoui, A.; Boukrouche, A.; Doghmane, H.; Bourouba, H. Face recognition using 1DLBP, DWT and SVM. 2015 3rd International Conference on Control, Engineering Information Technology (CEIT), 2015, pp. 1–6.
14. Petpon, A.; Srisuk, S. Face Recognition with Local Line Binary Pattern. 2009 Fifth International Conference on Image and Graphics, 2009, pp. 533–539.
15. Truong, H.P.; Kim, Y.G. Enhanced Line Local Binary Patterns (EL-LBP): An Efficient Image Representation for Face Recognition. Advanced Concepts for Intelligent Vision Systems. Springer International Publishing, 2018, pp. 285–296.

16. Zhang, G.; Huang, X.; Li, S.Z.; Wang, Y.; Wu, X. Boosting Local Binary Pattern (LBP)-Based Face Recognition. Advances in Biometric Person Authentication; Li, S.Z.; Lai, J.; Tan, T.; Feng, G.; Wang, Y., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2005; pp. 179–186.
17. Huang, D.; Shan, C.; Ardabilian, M.; Wang, Y.; Chen, L. Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **2011**, *41*, 765–781.
18. Chan, C.H.; Kittler, J.; Messer, K. Multi-scale Local Binary Pattern Histograms for Face Recognition. Advances in Biometrics; Lee, S.W.; Li, S.Z., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007; pp. 809–818.
19. Tan, X.; Triggs, B. Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition. Analysis and Modeling of Faces and Gestures; Zhou, S.K.; Zhao, W.; Tang, X.; Gong, S., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007; pp. 235–249.
20. Chan, C.H.; Tahir, M.A.; Kittler, J.; Pietikäinen, M. Multiscale Local Phase Quantization for Robust Component-Based Face Recognition Using Kernel Fusion of Multiple Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *35*, 1164–1177.
21. Tan, X.; Triggs, B. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE Transactions on Image Processing* **2010**, *19*, 1635–1650.
22. Zhao, D.; Lin, Z.; Tang, X. Laplacian PCA and Its Applications. 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.
23. Heikkilä, M.; Pietikäinen, M.; Schmid, C. Description of interest regions with local binary patterns. *Pattern Recognition* **2009**, *42*, 425 – 436.
24. Lahdenoja, O.; Laiho, M.; Paasio, A. Reducing the feature vector length in local binary pattern based face recognition. IEEE International Conference on Image Processing 2005, 2005, Vol. 2, pp. II–914.
25. Topi, M.; Timo, O.; Matti, P.; Maricor, S. Robust texture classification by subsets of local binary patterns. Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, 2000, Vol. 3, pp. 935–938 vol.3.
26. Raymond S. Smith and Terry Windeatt. Facial Expression Detection using Filtered Local Binary Pattern Features with ECOC Classifiers and Platt Scaling. Proceedings of the First Workshop on Applications of Pattern Analysis; Diethe, T.; Cristianini, N.; Shawe-Taylor, J., Eds.; PMLR: Cumberland Lodge, Windsor, UK, 2010; Vol. 11, *Proceedings of Machine Learning Research*, pp. 111–118.
27. Guo, Y.; Zhao, G.; Pietikäinen, M. Discriminative features for texture description. *Pattern Recognition* **2012**, *45*, 3834 – 3843.
28. Jian Yang and D. Zhang and A. F. Frangi and Jing-yu Yang. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2004**, *26*, 131–137. doi:10.1109/TPAMI.2004.1261097.
29. Daoqiang Zhang and Zhi-Hua Zhou. (2D)2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing* **2005**, *69*, 224 – 231. Neural Networks in Signal Processing, doi:https://doi.org/10.1016/j.neucom.2005.06.004.
30. Ngoc-Son Vu and Hannah M. Dee and Alice Caplier. Face recognition using the POEM descriptor. *Pattern Recognition* **2012**, *45*, 2478 – 2488.
31. Jia, Qi and Gao, Xinkai and Guo, He and Luo, Zhongxuan and Wang, Yi. Multi-Layer Sparse Representation for Weighted LBP-Patches Based Facial Expression Recognition. *Sensors* **2015**, *15*, 6719–6739.
32. Platt, J.C. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. ADVANCES IN LARGE MARGIN CLASSIFIERS. MIT Press, 1999, pp. 61–74.
33. Niculescu-Mizil, Alexandru and Caruana, Rich. Predicting Good Probabilities with Supervised Learning. Proceedings of the 22Nd International Conference on Machine Learning; ACM: New York, NY, USA, 2005; ICML '05, pp. 625–632.
34. Vo, T.; Nguyen, T.; Le, C.T. Race Recognition Using Deep Convolutional Neural Networks. *Symmetry* **2018**, *10*. doi:10.3390/sym10110564.
35. Vo, T.; Nguyen, T.; Le, C.T. A hybrid framework for smile detection in class imbalance scenarios. *Neural Computing and Applications* **2019**, *31*, 8583–8592. doi:10.1007/s00521-019-04089-w.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

37. Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2001**, *23*, 643–660.