DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-388

Noise Tolerant Histogram Voting for Gender Classification Based on LBP

Sanghun Lee and Chulhee Lee, School of Electrical and Electronic Engineering, Yonsei Univ., Seoul, South Korea 03722

Abstract

In this paper, we present a noise tolerant descriptor based on a local binary pattern (LBP) method. Due to threshold-based operations, these types of LBP methods are sensitive to noise factors. The use of a robust LBP (RLBP) reduced some noise effects. However, it may lead to a loss of subtle local texture information. Instead of concatenating the LBP and RLBP features, we produced a histogram as a weighted sum of the histograms of the LBPs and the RLBP. The proposed noise tolerant LBP (NTLBP) was calculated using the LBP histogram and histogram voting results of the RLBP. Without increasing the number of features, NTLBP proved to be robust against noise effects. We conducted several gender classification experiments using the FERET database and the NTLBP outperformed both the LBP and the RLBP methods.

Introduction

Gender classification can be used in various applications such as human-robot interaction, advertising, marketing, demographic analysis, etc. Human faces are commonly used in many vision-based applications since they provide high-level information about gender, age and expression [1]. In the last two decades, several gender classification methods based on face images have been proposed [2-9].

Most of these methods use a pipeline that consists of feature extraction and classification steps. One widely used feature extraction method is the local binary pattern (LBP) method [10-12] because of its illumination invariant characteristics. A center pixel is compared with its neighboring pixels and is encoded as a binary code. However, the LBP is sensitive to noises since it uses a threshold operation to produce outputs. In other words, small additive noises can produce different LBP values (Fig. 1). To improve robustness to noises, some modifications have been proposed.

Some researchers have tried to increase the number of binary codes. Tan and Triggs extended the LBP to the local ternary pattern (LTP) [13, 14] that uses 3-valued ternary codes: 1, 0 and -1. With a user-specified value t, the threshold function in the LTP gives 0 when the difference between a center pixel and one of its neighboring pixels was within $\pm t$. This method showed more robustness to noises. However, it produced two binary codes for each pixel, and the histogram concatenation doubled the number of features compared to the LBP. Nanni Brahnam et al. introduced a local quinary pattern (LQP) [15] with two threshold values and four binary patterns. More generally, the shift LBP (SLBP) [16] generates k binary patterns and k different thresholds.

On the other hand, thresholding methods can be redesigned for noise robustness. Instead of using the center pixel intensity, the median and the mean of neighboring pixels can be used in the median binary pattern (MBP) [17] and the improved LBP (ILBP) [18], respectively. In both methods, more bits are required than in the conventional LBP. Similar to the LTP, a buffer is used in the robust LBP (RLBP) [19], which encodes small differences between the center pixel and its neighboring pixels as zero. However, some facial information in homogeneous regions can be lost in the RLBP compared to the LBP.

To reduce noise sensitivity, we propose a noise tolerant LBP (NTLBP) that is based on a histogram voting method that mixes LBP and RLBP histograms without increasing the number of histogram bins. The extracted NTLBP histograms are concatenated into a single vector. The support vector machine (SVM) is used for classification. Fig. 2 shows a flowchart of the proposed method for gender classification. The proposed method was tested using the FERET database [20, 21].

Figure 1. *The LBP is sensitive to small additive noise factors. On the other hand, the RLBP is robust against such small additive noise factors.*

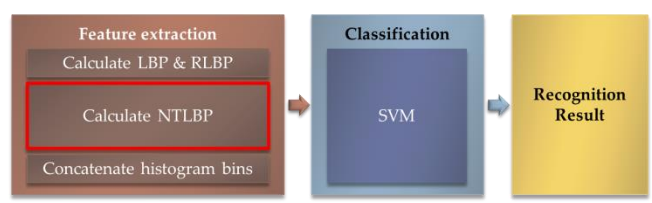

Figure 2. *Flowchart of the proposed method. First, the LBP and RLBP were computed and then the NTLBP was computed. We used a support vector machine (SVM) for classification.*

Related Works

LBP

The LBP [10-12] method is usually used to represent local textures. In this method, the relationship between a pixel (x_c, y_c)

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-388

and its N-neighboring points (x_n, y_n) on the circle with radius R are encoded as an N-bit code as follows:

$$LBP_{N,R}(x_c, y_c) = \sum_{n=0}^{N-1} s\left(I(x_n, y_n) - I(x_c, y_c)\right) 2^n \qquad (1)$$

where $s(u)$ is an indication function ($s(u) = 1$ when $n \geq 0$, $s(u) = 0$ otherwise) and $I(x, y)$ denotes the intensity of pixel (x, y) . Local texture can be simply encoded with the LBP. However, it is sensitive to small noises. If the difference between a pixel and its neighbors is small, the LBP varies with small additive noises as shown in Fig. 1. Especially, there is no appropriate binary code to represent homogeneous regions where the intensity differences between a pixel and all its neighbors are very small. Therefore, some variations of the LBP have been proposed to solve this problem.

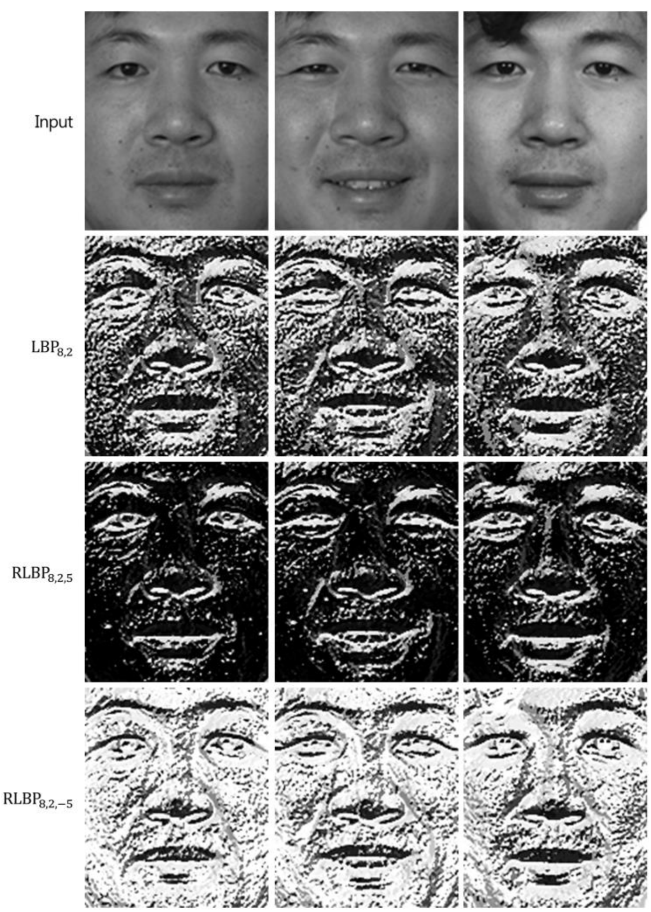

Figure 3. *Some examples of the LBP and the RLBP.*

RLBP

Heikkilä and Mietikäinen proposed a robust LBP [19] that used a buffer t to increase robustness against noises. In the RLBP, (1) can be simply rewritten with the buffer as:

$$RLBP_{N,R,t}(x_c, y_c) = \sum_{n=0}^{N-1} s\left(I(x_n, y_n) - I(x_c, y_c) - t\right) 2^n. \qquad (2)$$

Unless difference $I(x_n, y_n) - I(x_c, y_c)$ is equal to or higher than t for $n \in \{0, 1, \cdots N-1\}$, $RLBP_{N,R,t}(x_c, y_c)$ is 0. In other words, local differences smaller than t are ignored. However, some facial information in homogeneous regions can be lost when using the RLBP.

LTP

Tan and Triggs extended the LBP by using 3-valued codes with a new indication function [13, 14]:

$$s'(u,t) = \begin{cases} 1 & u \geq t \\ -1 & u \leq -t \\ 0 & otherwise \end{cases}. \qquad (3)$$

An absolute difference, $\left|I(x_n, y_n) - I(x_c, y_c)\right|$, smaller than t is quantized to zero. This made the patterns of the near-homogeneous regions less sensitive to additive noises. The 3-valued codes are usually split into two binary codes: upper pattern and lower pattern. For example, upper and lower patterns of codeword "110000(-1)(-1)" are "11000000" and "00000011." Accordingly, each pattern is treated as two LBPs. Also, the LTP can be represented as two RLBPs:

$$LTP_{N,R,t}^{upper}(x_c, y_c) = RLBP_{N,R,t}(x_c, y_c) \qquad (4)$$

$$LTP_{N,R,t}^{lower}(x_c, y_c) = 1 - RLBP_{N,R,-t}(x_c, y_c) \qquad (5)$$

Thus, local differences smaller than t are ignored in the LTP. However, the LTP doubles the feature size compared to the LBP.

Proposed Method

Motivation

In general, LBP histograms are extracted and concatenated for representing a given face image as a single vector in various applications such as face recognition, gender classification, etc. As we discussed in the previous section, the LBP can be easily corrupted with additive noises. Therefore, the LBP histograms computed from such noisy images are less reliable and extracted LBP features from an identical person may show less similarity. On the other hand, the RLBP offers more robustness against such noises, but may lose some local face information. Some examples of the LBP and the RLBP are shown in Fig. 3. Homogeneous regions are the most affected part with buffer t of the RLBP. Some local texture information for those regions were removed. One may concatenate both LBP and the RLBP, but it doubles the feature size and the computation time. In this paper, we used a novel noise tolerant histogram voting method that utilizes both LBP and RLBP without increasing the number of features.

Methodology

The basic idea of the proposed method is to combine the histograms of the LBP and RLBP methods. On the other hand, if the LBP and RLBP values differed substantially, it indicated that the differences between the center pixel and the neighboring pixels were small and these pixels may have been easily affected by small

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-388

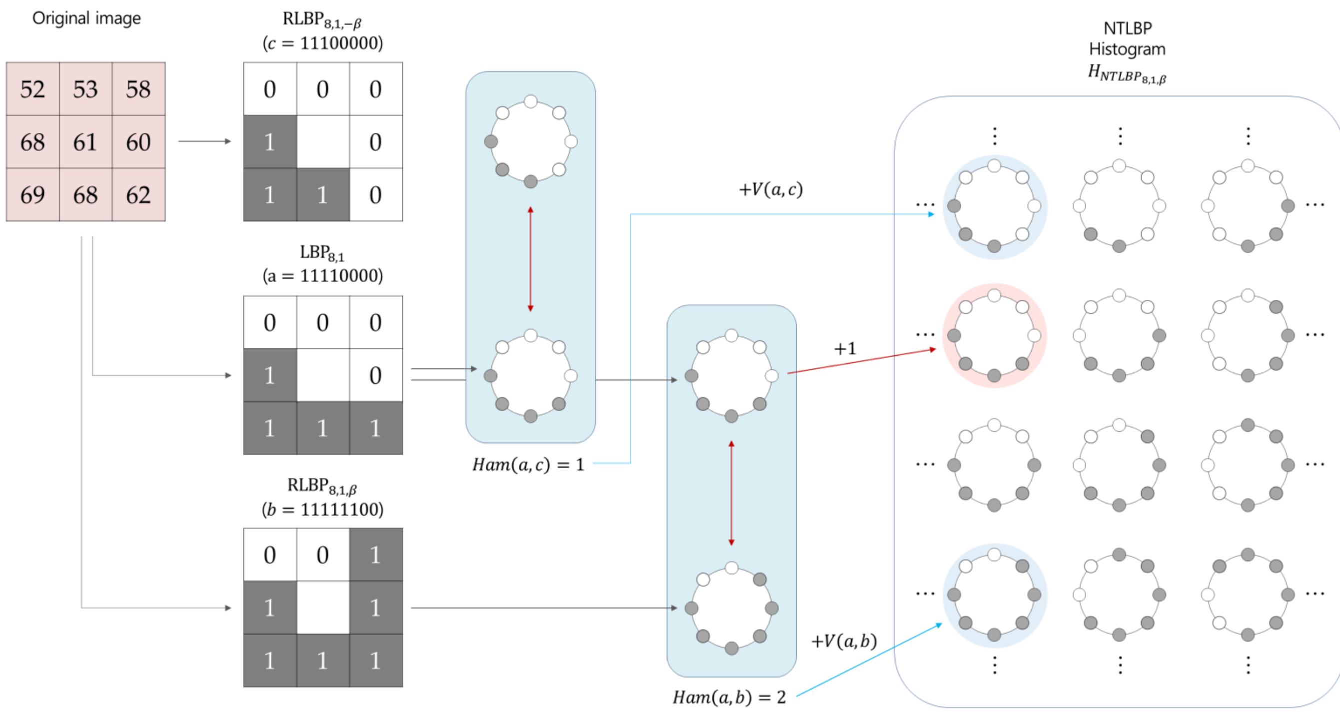

Figure 4. *Overview of the proposed NTLBP method. First, the LBP and two RLBPs were computed at every pixel. Then, the Hamming distances between the LBP and two RLBPs were calculated. Finally, the NTLBP histogram voting was done with our voting function* $V(\cdot,\cdot)$.

noises. Since these RLBP values may be less reliable for classification, we updated the RLBP histogram using different increments. In conventional histogram computations, the bin counter is increased by one:

$$H(n) \leftarrow H(n) + 1 \tag{6}$$

In the proposed method, we updated the RLBP histogram using an increment function as follows:

$$H_{mod,RLBP}(n) \leftarrow H_{mod,RLBP}(n) + V(a,b) \tag{7}$$

where $V(a,b)$ is the increment function and a and b are 8-bit codewords of LBP and RLBP. As explained previously, a large difference between a and b indicates that the differences between the center pixel and neighbor pixels were small and the voting results (i.e., LBP and RLBP values) were substantially different. In this case, we increased the RLBP histogram by a smaller value. On the other hand, a small value of k indicates that LBP and RLBP values are similar and we increased the RLBP histogram by a larger value.

Thus, in the proposed method, the increment function depends on the voting reliability of the LBP and RLBP methods. We used the Hamming distance [22] between $LBP_{N,R}(x_c, y_c)$ and $RLBP_{N,R,t}(x_c, y_c)$ for the increment function. The Hamming distance is the number of differences between two codewords. For example, the Hamming distance between 00110100 and 00011000 is computed as $Ham(00110100, 00011000) = 3$. A small Hamming distance between the LBP and the RLBP may indicate that the LBP is hardly affected by noises. On the other hand, a large Hamming distance indicates that LBP values may be unreliable since the differences between the center pixels and the neighboring pixels may be small. Therefore, we used an increment function $V(a,b)$ as follows:

$$V(a,b) = \alpha\left(1 - \frac{Ham(a,b)}{8}\right) \tag{8}$$

where $a = LBP_{N,R}(x_c, y_c)$ and $b = RLBP_{N,R,\beta}(x_c, y_c)$ are 8-bit codewords and α is a weight constant with an empirically determined value. We used a XOR operation to compute the Hamming distances. Counting the number of 1s in the XOR results is equivalent to the Hamming distance. $Ham(a,b) = 0$ means that the differences between the center pixels and some neighboring pixels were large and the LBP was unlikely to be affected by additive noises. Thus in the proposed method, $H_{mod,RLBP_\beta}(n)$ was updated as follows:

$$H_{mod,RLBP_\beta}(b) \leftarrow H_{mod,RLBP_\beta}(b) + V(a,b) \quad \text{if } Ham(a,b) > 0 \tag{9}$$

We also produced $H_{mod,RLBP_{-\beta}}(n)$ with $c = RLBP_{N,R,-\beta}(x_c, y_c)$ and the proposed NTLBP which can be defined as follows:

$$H_{NTLBP}(n) = H_{LBP}(n) + H_{mod,RLBP_\beta}(n) + H_{mod,RLBP_{-\beta}}(n) \tag{10}$$

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-388

In this way, we compensated the LBP histogram using the RLBP information without increasing the number of histogram bins. Fig. 4 is an the overview of the proposed NTLBP histogram voting. We also defined the NTLBP+ histogram with only one RLBP as follows:

$$H_{NTLBP^+}(n) = H_{LBP}(n) + H_{\mathrm{mod},RLBP}(n) \tag{11}$$

The proposed method was tested using the FERET database. Also, we intentionally added additive Gaussian noises to the images to show the robustness of our method.

Table 1. Recognition rates with 10-fold cross-validation of each subset of the FERET database.

Method	*fa*	*fb*	*fc*	*dup1*	*dup2*
$LBP_{8,2}$	93.90	93.31	88.66	98.89	99.57
$RLBP_{8,2,5}$	93.48	92.72	88.14	98.20	99.57
$RLBP_{8,2,-5}$	92.81	91.30	86.60	98.62	99.57
$NTLBP^+_{8,2,5}$	94.23	93.39	88.14	98.89	99.57
$NTLBP^+_{8,2,-5}$	94.73	94.06	89.69	98.62	99.57
$NTLBP_{8,2,5}$	94.65	94.39	91.75	99.03	99.57

Table 2. Recognition rates when *fa* subset was used as a training set.

Method	*fb*	*fc*	*dup1*	*dup2*
$LBP_{8,2}$	95.02	93.48	96.21	97.32
$RLBP_{8,2,5}$	95.19	92.39	95.22	96.43
$RLBP_{8,2,-5}$	93.92	92.39	93.96	95.54
$NTLBP^+_{8,2,5}$	95.87	92.39	96.21	96.88
$NTLBP^+_{8,2,-5}$	94.77	92.93	96.35	98.21
$NTLBP_{8,2,5}$	95.11	94.57	96.91	98.66

Experiments

The proposed method (NTLBP) was compared with the LBP and the RLBP method. The FERET database [20, 21] was used, which contains face images of 1,196 subjects (494 females and 702 males). There were five gallery subsets (*fa*, *fb*, *fc*, *dup I*, and *dup II*). The *fa* set contained neutral and frontal face images. The *fb* and *fc* set comprised face images with alternative facial expressions and under different lighting conditions, respectively. The images in both *dup I* and *dup II* were taken later. Especially, the images of dup II were taken at least a year later. We used 3541 frontal face images (*fa* = 1,196, *fb* = 1,195, *fc* = 194, *dup I* = 722, and *dup II* = 234 images) for the experiments. All face images were normalized and cropped to 172x144 pixels.

First, the input images were represented by the LBP and the RLBP images. Then, each image was divided into 7x7 blocks and the LBP, RLBP and NTLBP histograms were calculated for every block. Then, the LBP, RLBP and NTLBP histograms were concatenated into a single vector. Note that we only used the uniform patterns proposed by Ojala et al [10]. If an 8-bit codeword showed at most two bitwise transitions, we call it a uniform pattern. For example, "00110000" showed two transitions from either 0 to 1 or vice-versa, and this represented a uniform pattern. On the other hand, "01010000" showed four transitions and this represented a non-uniform pattern. In the proposed implementation, we used 59 histogram bins for 58 uniform patterns and the other patterns (non-uniform patterns). Finally, the dimensions of the feature vectors were $59 \times 7 \times 7 = 2891$. Unless noted otherwise, we used $\beta = 5$ and $\alpha = 0.5$ for all of the following experiments.

For the classification, we used the support vector machine (SVM) with the LIBSVM library [23] (C-SVC with the linear kernel was used).

In the experiments using the FERET database, recognition rates were computed in two different ways. In the first experiment, we trained and tested each subset of the FERET database separately. For the *fc* set, we used $\alpha = 1.5$. Table 1 is a performance comparison with a 10-fold validation for each subset. In the second experiment, we used the *fa* set as a training set and the other subsets (*fb*, *fc*, *dup I*, and *dup II*) as a test set. Table 2 is a performance comparison when the *fa* set was used as a training set. In both experiments, the NTLBP and NTLBP+ showed better results than the other methods.

Table 3. Recognition rates when additive Gaussian noise factors were applied.

Method	5dB	10dB	15dB	20dB	25dB	30dB
$LBP_{8,2}$	84.36	88.63	90.80	91.89	93.23	93.31
$RLBP_{8,2,5}$	85.37	87.96	90.80	90.22	93.06	93.06
$RLBP_{8,2,-5}$	86.96	87.46	90.13	91.97	92.81	93.06
$NTLBP^+_{8,2,5}$	85.79	88.46	90.47	91.89	93.31	93.73
$NTLBP^+_{8,2,-5}$	86.62	88.96	91.14	91.89	93.65	93.48
$NTLBP_{8,2,5}$	86.12	89.46	91.14	92.47	93.23	93.65

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-388

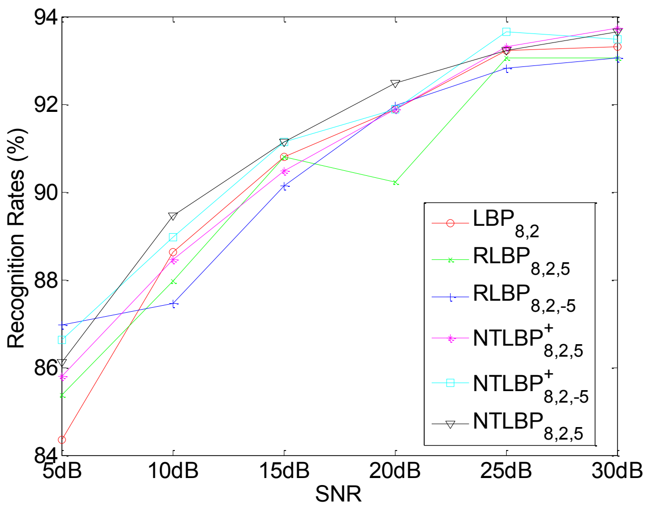

Figure 5. *Recognition rates when additive Gaussian noises with different SNRs are applied.*

To test the robustness against additive noises, we added the additive Gaussian noises (5~30dB SNR) to the images of the *fa* subset and conducted a 10-fold validation. Except for the 5dB SNR case, the NTLBP and NTLBP$^+$ showed better performance than the LBP and RLBP methods.

Conclusions

In this paper, we introduced a noise tolerant LBP method based on the Hamming distance. By combining the LBP and RLBP methods by considering voting confidence, the proposed method effectively handled small variations of LBP features. Experiments using the FERET database showed that the proposed NTLBP method produced better results than the LBP and the RLBP. On the other hand, the increment function of the NTLBP, which is based on the voting confidence, can be further optimized in the future with various functions such as polynomial or exponential functions.

References

[1] W. Gao and H. Ai, "Face gender classification on consumer images in a multiethnic environment," in *Advances in Biometrics*, ed: Springer, 2009, pp. 169-178.

[2] S. Baluja and H. A. Rowley, "Boosting sex identification performance," *International Journal of computer vision,* vol. 71, pp. 111-119, 2007.

[3] C. BenAbdelkader and P. Griffin, "A local region-based approach to gender classi. cation from face images," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, 2005, pp. 52-52.

[4] Y. Fang and Z. Wang, "Improving LBP features for gender classification," in *Wavelet Analysis and Pattern Recognition, 2008. ICWAPR'08. International Conference on*, 2008, pp. 373-377.

[5] A. Hadid and M. Pietikäinen, "Combining appearance and motion for face and gender recognition from videos," *Pattern Recognition,* vol. 42, pp. 2818-2827, 2009.

[6] A. Lapedriza, M. J. Marin-Jimenez, and J. Vitria, "Gender recognition in non controlled environments," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, pp. 834-837.

[7] E. Mäkinen and R. Raisamo, "Evaluation of gender classification methods with automatically detected and aligned faces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 30, pp. 541-547, 2008.

[8] B. Moghaddam and M.-H. Yang, "Learning gender with support faces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 707-711, 2002.

[9] G. Shakhnarovich, P. Viola, and B. Moghaddam, "A unified learning framework for real time face detection and classification," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, 2002, pp. 14-21.

[10] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 971-987, 2002.

[11] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer vision-eccv 2004*, ed: Springer, 2004, pp. 469-481.

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

[13] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Analysis and Modeling of Faces and Gestures*, ed: Springer, 2007, pp. 168-182.

[14] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *Image Processing, IEEE Transactions on,* vol. 19, pp. 1635-1650, 2010.

[15] L. Nanni, A. Lumini, and S. Brahnam, "Local binary patterns variants as texture descriptors for medical image analysis," *Artificial intelligence in medicine,* vol. 49, pp. 117-125, 2010.

[16] G. Kylberg and I.-M. Sintorn, "Evaluation of noise robustness for local binary pattern descriptors in texture classification," *EURASIP Journal on Image and Video Processing,* vol. 2013, pp. 1-20, 2013.

[17] A. Hafiane, G. Seetharaman, and B. Zavidovique, "Median binary pattern for textures classification," in *Image Analysis and Recognition*, ed: Springer, 2007, pp. 387-398.

[18] L. Nanni, S. Brahnam, and A. Lumini, "A local approach based on a Local Binary Patterns variant texture descriptor for classifying pain states," *Expert Systems with Applications,* vol. 37, pp. 7888-7894, 2010.

[19] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, pp. 657-662, 2006.

[20] P. J. Phillips, H. Moon, S. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, pp. 1090-1104, 2000.

[21] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and vision computing,* vol. 16, pp. 295-306, 1998.

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-388

[22] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal,* vol. 29, pp. 147-160, 1950.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 27, 2011.

Author Biography

Sanghun LEE received a B.S. degree in electrical and electronical engineering from Yonsei University, Seoul, South Korea, in 2013. He is currently working toward a Ph.D. degree in electronic engineering at Yonsei University. His current research interests include image processing and pattern recognition.

Chulhee Lee received the B.S. and M.S degrees in electronics engineering from Seoul National University in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, Indiana, in 1992. From 1986 to 1987, he was a researcher in the Acoustic Laboratory at Technical University of Denmark (DTH). From 1993 to 1996, he worked with National Institutes of Health, Bethesda, Maryland. In 1996, he joined the faculty of the Department of Electrical and Computer Engineering, Yonsei University, Seoul, Korea. His research interests include image/signal processing, pattern recognition, and neural networks.