

Research Article

The Performance of LBP and NSVC Combination Applied to Face Classification

Mohammed Ngadi, Aouatif Amine, Bouchra Nassih, Hanaa Hachimi, and Adnane El-Attar

Systems Engineering Laboratory, National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco

Correspondence should be addressed to Mohammed Ngadi; ngadi.mohammed@univ-ibntofail.ac.ma

Received 15 October 2016; Accepted 10 November 2016

Academic Editor: Lei Zhang

Copyright © 2016 Mohammed Ngadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The growing demand in the field of security led to the development of interesting approaches in face classification. These works are interested since their beginning in extracting the invariant features of the face to build a single model easily identifiable by classification algorithms. Our goal in this article is to develop more efficient practical methods for face detection. We present a new fast and accurate approach based on local binary patterns (LBP) for the extraction of the features that is combined with the new classifier Neighboring Support Vector Classifier (NSVC) for classification. The experimental results on different natural images show that the proposed method can get very good results at a very short detection time. The best precision obtained by LBP-NSVC exceeds 99%.

1. Introduction

Researchers have shown that, to recognize a face, human uses different features such as geometry, texture, and colors of different parts of the face: eyes, mouth, nose, the front, and the cheeks. Based on this observation, several studies have been developed to verify whether it was possible to model this behavior in a computational way.

This article is devoted to the problem of computer based face classification [1], which became a popular and important research topic in recent years thanks to its many applications such as indexing and searching for image and video, security access control, video surveillance. Despite many efforts and progress that have been made during recent years, it remains an open problem and is still considered one of the most difficult problems in the community of computer vision, mainly due to the similarities between the classes and class variations such as occlusion, background clutter, perspective changes, poses scaling, and lighting. Nowadays popular detection approaches are based on descriptors and classifiers, which generally extract visual descriptors in the pictures and videos and then perform the classification using machine learning algorithms based on the extracted features.

Generally, the size of the data can be measured by two dimensions: the number of variables and the number of examples. These two dimensions can take very high values, which can be a problem with the exploration and analysis of these data. In this context, it is essential to implement some data processing tools that allow us to better understand the information contained in our dataset. Dimensionality reduction is one of the oldest approaches that answers this problem. Its objective is to select or retrieve an optimal subset of relevant characteristics according to a previously fixed criterion. This selection/extraction allows reducing the dimension of the space of the examples and making all the data more representative of the problem.

This reduction has a dual purpose, the first is to reduce redundancy, and the second allows facilitating subsequent treatments (feature extraction reduces required storage space and accordingly reduces the classification learning time and accelerates the pattern recognition process) and therefore the data interpretation.

The first step aims to select the best feature extraction [2, 3] method for the context of face classification. In this context, we found that the LBP descriptor gives the most optimal representation of the image. The principle of this

descriptor is to compare each pixel (considered as central pixel in a window of radius R and containing P points) to its neighbors and generate a binary code based on this comparison [4]. For the validation of our choice, we make a comparison with other feature extractions descriptors such as Discrete Wavelet Transform (DWT) [5] and Histogram of Oriented Gradients (HOG) [6].

The second step concerns the selection of the classification function. We have chosen to use a method placed in the context of the semisupervised classifiers, the NSVC. It is based on the combination of two classifiers belonging to two different families (nonsupervised classification: Fuzzy C-Means and supervised one: SVM). The basic idea of the Neighboring Support Vector Classifier (NSVC) is to build new vicinal kernel functions, obtained by supervised clustering in feature space. These vicinal kernel functions are then used for learning.

Finally, our experiments show that LBP-NSVC outperforms all other feature selection and classification algorithms. The main criteria used for comparison are the accuracy of the classification and the execution time, without forgetting the ability of the classifier to effectively manage practical applications where the training data may come from different environments.

The rest of the paper is organized as follows. A brief description of LBP is given in Section 2. Section 3 introduces the NSVC based on supervised partitioning of features space. Experimental results are presented in Section 4, while Section 5 concludes the article.

2. Local Binary Patterns (LBP)

The LBP method can be regarded as a unifying statistical and structural approach to texture analysis. Instead of trying to explain the formation of the texture on the pixel level, local models are formed around each pixel. Each pixel is labeled with the code of the texture that is best at the local level in his neighborhood. Thus, each LBP code can be regarded as the code that best represents the local vicinity of the pixel. The LBP distribution therefore has both structural properties: primitives of textures and the rules for placement of these primitives. For these reasons, the LBP method can be used successfully to recognize a variety of textures, in which structural and statistical methods have been traditionally applied separately.

Local binary patterns were originally proposed by Ojala et al. in 1996 [7]. The concept of the LBP is simple; it proposes assigning a binary code to a pixel based on its neighborhood. This code describing the local texture of a region is calculated by thresholding of a neighborhood with the gray of the central pixel level. In order to generate a binary pattern, all the neighbors will then take a value "1" if their value is greater than or equal to the current pixel and "0" otherwise. This binary pattern's pixels are then multiplied by weights and summoned to obtain a current pixel LBP code. We thus obtain, for any image, pixels with intensity between 0 and 255 as in an ordinary 8-bit image. Rather than describing the image by the sequence of the LBP codes,

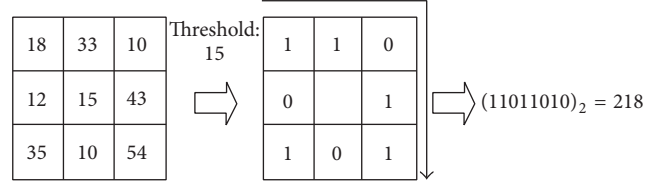


FIGURE 1: Example of LBP calculation.

one can choose as texture descriptor using a 255-dimension histogram (Figure 1).

The LBP was extended later by using different neighborhood sizes [8–10]. In this case, a circle of radius R around the central pixel is considered. Values of P points sampled on the edge of the circle are taken and compared with the value of the central pixel. To obtain the values for P points sampled in the vicinity for any radius R , an interpolation is necessary. The notation (P, R) is adopted to define the vicinity of P points of radius R of a pixel. $LBPP, R$ is the LBP code for the radius R and the number of neighbors P . The main difference is that the pixels must be interpolated to obtain the values of the points on the circle. The important property of the LBP code is that this code is invariant to uniform illumination changes because the LBP for a pixel depends only on differences between its gray-level and that of its neighbors.

To calculate LBP code in a neighborhood of P pixels in RADIUS R , one simply counts the occurrences of pixels g_i superior or equal to the central value:

$$LBP_{m,R} = \sum_{i=0}^{m-1} u(g_i - g_c) \cdot 2^i, \quad (1)$$

where $u(\cdot)$ is the sign function and where g_i and g_c are, respectively, a nearby pixel and the central pixel grayscale

$$u(x) = \begin{cases} 1, & \text{si } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The concept of the multiscale LBP is based on the choice of the vicinity in order to calculate LBP code to process textures at different scales [11, 12]. A neighborhood for a central pixel is distributed on a circle and built from two parameters: the number of neighbors "P" on the circle and radius "R" to define a distance between a central pixel and its neighbors (Figure 2). The texture T of an image I is categorized by the combined distribution of gray values of $n + 1$ pixels (where $n > 0$): $T = t(g_c, g_0, \dots, g_{p-1})$, and g_c corresponds to the value of the central pixel and g_p , with $p = 0, \dots, P - 1$, corresponds to the level of P pixels regularly spaced on a circle of radius R . If g_c coordinates are equal to $(0, 0)$, then g_p coordinates are given by the following equation:

$$(x_p, y_p) = \left(x_c + R \cos\left(\frac{2\Pi p}{P}\right), y_c - R \sin\left(\frac{2\Pi p}{P}\right) \right). \quad (3)$$

From the definition of neighborhood, the authors define, first, a local binary pattern that is invariant to any monotonic transformation of grayscale, $LBPP, R$. For each pixel

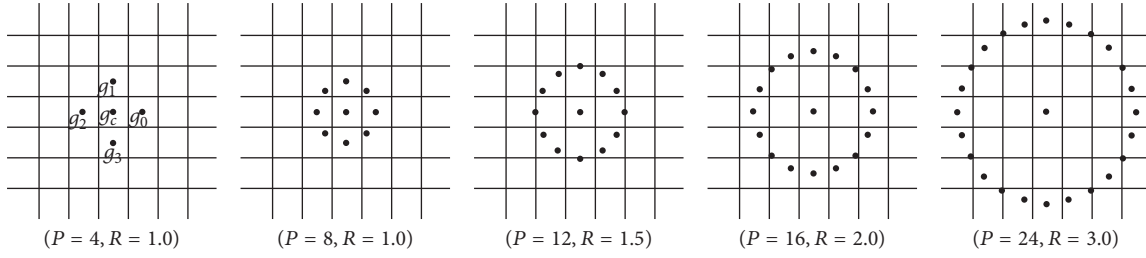
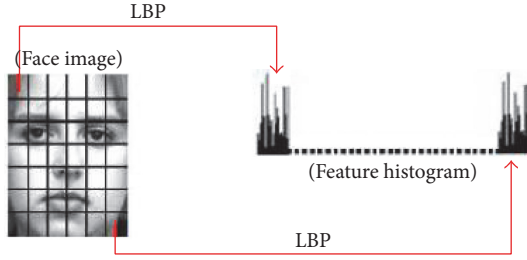

 FIGURE 2: Multiscale LBP. Examples of neighborhoods obtained for various values of (P, R) .


FIGURE 3: LBP-based facial representation.

(x, y) ($g_c = g(x, y)$), the central pixel is not used for the characterization of textures. Indeed, regardless of g_p vicinity, that pixel only describes a light intensity which is not necessarily useful [11]. Subsequently, g_c is used as a threshold in the following manner:

$$T = t(u(g_0 - g_c), \dots, u(g_{p-1} - g_c)). \quad (4)$$

Accordingly, the calculation of the LBP code can be obtained in the same way as the basic LBP (see (1)).

LBP-based face representation: each face image can be considered to be a composition of micropatterns which can be effectively detected by the LBP operator. Hadid et al. [13] introduced LBP-based face representation for facial recognition. To examine the face shape information, they divided the images of face to M small nonoverlapping areas R_0, R_1, \dots, R_M (as shown in Figure 3).

The NSVC is a classifier adaptive to different datasets. It is based, on one hand, on a non-supervised approach such as K -means or FCM and, on the other hand, on a supervised approach: SVM.

3. Neighboring Support Vector Classifier (NSVC)

Support Vector Machines, first introduced by Vapnik and colleagues for the problems of classification and regression, can be seen as a new training technique based on traditional polynomial and radial basis function (RBF). As discussed before, SVMs have attracted considerable attention because of their high generalization ability and higher classification performance relative to other pattern recognition algorithms.

However, the assumption that the training data are identically generated from unknown probability distributions may

limit the application of SVM to the problems of everyday life [14].

To relax the assumption of identical distribution, the NSVC [15–17] uses a set of vicinal cores functions built based on supervised clustering in the feature space induced by the kernel. The basic idea of the NSVC is to build new vicinal core functions obtained by supervised clustering in the feature space. These vicinal core functions are then used to SVM training.

This approach consists of two steps:

- (i) Supervised clustering step based on SKDA algorithm (for supervised kernel-based deterministic annealing, used to partition the training data in different vicinal areas).
- (ii) A training step where the SVM technique is used to minimize the vicinal risk function (VRM) under the constraints defined in clustering step based on SKDA.

Consider the following input output data together:

$$(x_i, y_i)_{i=1}^l, \quad x_i \in R^n, \quad y_i \in \{-1, 1\}, \quad (5)$$

where l is the number of input data points and n is the dimension of the input space.

The vicinity functions $v(x_i)$ of x_i data points are built if test data points satisfy two assumptions:

- (i) The unknown density function is smooth in the neighborhood of each point x_i .
- (ii) The function minimizing the functional risk is also smooth and symmetric in the neighborhood of each point x_i .

The optimization problem based on the principle of VRM named vicinal linear SVM [18, 19] can then be formulated as

$$\begin{aligned} \text{minimize: } & \phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \\ \text{subject to: } & y_i \int_{V(x_i)} ([\langle x, w \rangle + b] p(x | V(x_i))) dx \\ & \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \dots, l, \end{aligned} \quad (6)$$

where w is a weight, C is a punishment constant for ξ_i , b is the offset, $v(x_i)$ is the vicinity associated with the test point x_i , and

$p(x | V(x_i))$ is the conditional probability of the respective vicinity in the input space.

The following theorem for the vicinal SVM solution is true (see [18] for a proof):

$$f(x) = \sum_{i=1}^l y_i \beta_i L(x, x_i) + b, \quad (7)$$

where to define the coefficients β_i one has to maximize

$$W(\beta) = \sum_{i=1}^l \beta_i - \frac{1}{2} \sum_{i,j=1}^l \beta_i \beta_j y_i y_j M(x_i, x_j)$$

$$\text{subject to } \sum_{i=1}^l \beta_i y_i = 0 \quad (8)$$

$$\beta_i \geq 0,$$

where $L(x, x_i)$ is called the monovincinal kernel and $M(x_i, x_j)$ is the bivincinal kernel of the vicinal SVM [18].

3.1. Supervised Kernel-Based Deterministic Annealing for NSVC. The clustering of training data in the feature space is a well-documented subject [20, 21]. It consists of nonlinearly mapping the observed data of an input low-dimensional space to a high-dimensional feature space using a kernel function, which facilitates the separation of linear data, denoting a nonlinear transformation of the input space X to a high-dimensional space using a kernel function as

$$\begin{aligned} \Phi : \mathcal{R}^n &\longrightarrow F \\ x_i &\longrightarrow \Phi(x_i), \\ j &= 1, \dots, l, \end{aligned} \quad (9)$$

where $\Phi(x_i)$ is the transformed point x_i .

All training data points are distributed in c vicinities/clusters in the feature space, where $\phi_k(z)$ is the center of mass of the k th vicinity residing in F . This is a similar representation to clustering based on the characteristic space of k -means:

$$\phi_k = \sum_{i=1}^l \alpha_{ki} z_i, \quad k = 1, 2, \dots, c, \quad (10)$$

where c is the number of clusters, α_{ki} are the parameters to be defined by the clustering technique (SKDA), and $z_i = y_i \phi(x_i)$ denotes the data points labeled in the feature space.

The classification problem is usually defined mathematically by a cost function to be minimized; for NSVC case, this function is the distortion function. Similar to the notation used in [22], we let $p(\phi_k | z_i)$ denote the probability of association of points z_i mapped to the cluster center ϕ_k . Using the square distance $D_k(z_i)$ [15] between the center ϕ_k and the training vector z_i , the distortion function in the function space becomes

$$J_\phi = \sum_{i=1}^l \sum_{k=1}^c p(z_i) p(\phi_k | z_i) D_k(z_i). \quad (11)$$

Since no a priori knowledge of the distribution of data is assumed, over all possible distributions which give a given value of J_ϕ we choose the one that maximizes the conditional Shannon entropy in the characteristic space:

$$H_\phi = - \sum_{i=1}^l \sum_{k=1}^c p(z_i) p(\phi_k | z_i) \log p(\phi_k | z_i). \quad (12)$$

The optimization problem can be reformulated as the minimization of the Lagrangian:

$$F_\phi = J_\phi - TH_\phi, \quad (13)$$

where T is the Lagrange multiplier.

To determine α_{ki} parameter, we minimize the free energy function F with respect to the likelihood of association [22], which is related to the Gibbs distribution as

$$p(\phi_k | z_i) = \frac{p(\phi_k) e^{-(D_k(z_i)/T)}}{\sum_{m=1}^c p(\phi_m) e^{-(D_m(z_i)/T)}}, \quad (14)$$

where $p(\phi_k)$ is the mass probability for k th cluster:

$$p(\phi_k) = \sum_{i=1}^l p(z_i) p(\phi_k | z_i). \quad (15)$$

And so the energy function is

$$\begin{aligned} F_\phi^* &= \min_{p(\phi_k | z_i)} (J_\phi - TH_\phi) \\ &= -T \sum_{i=1}^l p(z_i) \log \sum_{k=1}^c p(\phi_k) e^{-(D_k(z_i)/T)}. \end{aligned} \quad (16)$$

The partial derivative of F with respect to ϕ_k is

$$\frac{\partial (F_\phi^*)}{\partial (\phi_k)} = 0. \quad (17)$$

Accordingly

$$\sum_{i=1}^l p(z_i) p(\phi_k) e^{-(D_k(z_i)/T)} [z_i - \phi_k] = 0. \quad (18)$$

By dividing by the normalization factor

$$Z_{z_i} = \sum_{m=1}^c p(\phi_m) e^{-(D_m(z_i)/T)}. \quad (19)$$

And, so,

$$\begin{aligned} &\sum_{i=1}^l \frac{p(z_i) p(\phi_k) e^{-(D_k(z_i)/T)}}{Z_{z_i}} z_i \\ &= \sum_{i=1}^l \frac{p(z_i) p(\phi_k) e^{-(D_k(z_i)/T)}}{Z_{z_i}} \phi_k. \end{aligned} \quad (20)$$

Using (14) leads to

$$\sum_{i=1}^l p(z_i) p(\phi_k | z_i) z_i = \sum_{i=1}^l p(z_i) p(\phi_k | z_i) \phi_k, \quad (21)$$

$$\begin{aligned} \phi_k &= \sum_{i=1}^l \frac{p(z_i) p(\phi_k | z_i)}{\sum_{i=1}^l p(z_i) p(\phi_k | z_i)} z_i \\ &= \sum_{i=1}^l \alpha_{ki} z_i. \end{aligned} \quad (22)$$

Finally, we obtain the expression of α_{ki} that will be used to construct the vicinal kernel for NSVC functions:

$$\alpha_{ki} = \frac{p(z_i) p(\phi_k | z_i)}{\sum_{j=1}^l p(z_j) p(\phi_k | z_j)}. \quad (23)$$

3.2. NSVC with the Feature Space Partitioning. The optimization problem based on feature space partitioning is formulated as follows [18]:

$$\begin{aligned} \text{minimise: } \quad & \phi(w) = \frac{1}{2} w^T w + C \sum_{k=1}^K \xi_k \\ \text{subject to: } \quad & y_k \int_{V(\phi_k)} [(z, w) + b] p(z | \phi_k) dz \\ & \geq 1 - \xi_k, \\ & i = 1, \dots, l \\ & \xi_k \geq 0, \quad k = 1, \dots, K, \end{aligned} \quad (24)$$

where $\nu(\phi_k)$ represents the k th vicinity associated with the mass center ϕ_k in the feature space and $p(z | \phi_k)$ is the conditional probability of respective vicinity in the feature space. According to Bayes theorem, we have

$$\begin{aligned} p(z_i | \phi_k) &= \frac{p(z_i) p(\phi_k | z_i)}{p(\phi_k)} \\ &= \frac{p(z_i) p(\phi_k | z_i)}{\sum_{j=1}^l p(z_j) p(\phi_k | z_j)}. \end{aligned} \quad (25)$$

By comparing (22) and (25), we get

$$\phi_k = \sum_{i=1}^l p(z_i | \phi_k) z_i. \quad (26)$$

And the optimization constraint becomes

$$\begin{aligned} & y_k \int_{V(\phi_k)} [(z, w) + b] p(z | \phi_k) dz \\ &= y_k \left[\left\langle \int_{V(\phi_k)} p(z | \phi_k) z dz, w \right\rangle \right. \end{aligned}$$

$$\begin{aligned} & \left. + \int_{V(\phi_k)} b p(z | \phi_k) dz \right] \\ &= y_k \left[\left\langle \sum_{i=1}^l p(z_i | \phi_k) z_i, w \right\rangle + \sum_{i=1}^l b p(z_i | \phi_k) \right] \\ &= y_k [\langle \phi_k, w \rangle + b]. \end{aligned} \quad (27)$$

Let one define the mono- and bivincinal kernels as

$$\begin{aligned} L_k(x) &= \sum_{i=1}^l y_i \alpha_{ki} K(x, x_i), \quad k = 1, 2, \dots, K, \\ M_{km}(x) &= \sum_{i=1}^l \sum_{j=1}^l y_i y_j \alpha_{ki} \alpha_{mj} K(x_i, x_j), \\ & \quad k, m = 1, 2, \dots, K, \end{aligned} \quad (28)$$

where α_{ki} parameters are obtained from the SKDA clustering step. The decision boundary is

$$f(x) = \sum_{k=1}^c \beta_k y_k L_k(x) + b, \quad (29)$$

where β_k is the coefficient that maximizes the dual function:

$$\begin{aligned} \text{maximize } \quad & W(\beta) \\ &= \sum_{k=1}^c \beta_k - \frac{1}{2} \sum_{k,m=1}^c \beta_k \beta_m y_k y_m M_{km}(x) \\ \text{subject to } \quad & \sum_{k=1}^c \beta_k y_k = 0, \\ & \beta_k \geq 0. \end{aligned} \quad (30)$$

In order to obtain a sparse solution at the cost of the extra clustering procedure, a good selection of the number of clusters is required.

4. Experimental Results

We will now carry out a deep evaluation of the classifiers mentioned in previous sections. We start by a detailed description of the dataset and then present the classification results of all classifiers.

4.1. Dataset. Among the factors that influence or affect the performance of face detection system are scale, pose, lighting conditions, facial expression, and occlusion. For this reason we established a robust database based on diverse illumination conditions and different color and texture variations and, then, under various emotional facial expressions such as neutral expression, anger, scream, sadness, sleepiness, being surprised, wink, frontal smile, frontal smile with teeth, open or closed eyes, and facial details (glasses/no glasses, hats/no hats, and caps/no caps). Then, to use our database in the context of face classification, the different facial images were



FIGURE 4: Typical people images of database in varied environment.

TABLE 1: Accuracy of the method of extraction of features with polynomial kernel of NSVC.

	SLBP	ULBP	HOG	DWT
Accuracy %	99.99	99.47	95.73	91.90
Parameter of kernel	2	3	6	3

ULBP: Uniform Local Binary Pattern. SLBP: Simple Local Binary Pattern.

taken at different lighting conditions to make the classification model invariant to illumination. The images were adopted under divers unconstrained environment (Figure 4).

We detect people's faces in our database using the cascade detected of Viola-Jones algorithm and normalized the detected faces with a fixed size of $30 * 30$ pixels. Figure 5 presents some typical face images of database.

4.2. Results of NSVC. The basic idea of Neighboring Support Vector Classifier (NSVC) is to build new neighboring kernel functions, obtained by supervised clustering in feature space.

TABLE 2: Accuracy of the method of extraction of features with polynomial kernel of NSVC.

	SLBP + DWT	SLBP + HOG	ULBP + DWT
Accuracy %	99.53	99.50	99.57
Parameter of kernel	2	3	2

These neighboring kernel functions are then used in SVM based learning.

When using polynomial and RBF kernels, we have used cross-validation in order to compute optimal learning parameters for both kernels.

We evaluate the accuracy of each feature extraction method with NSVC. The results obtained are shown in Tables 1 and 2.

To test the performance of the proposed approach, we compare the precision of the LBP-NSVC algorithm with other combinations such as HOG-NSVC and DWT-NSVC.



FIGURE 5: Typical face images of database.

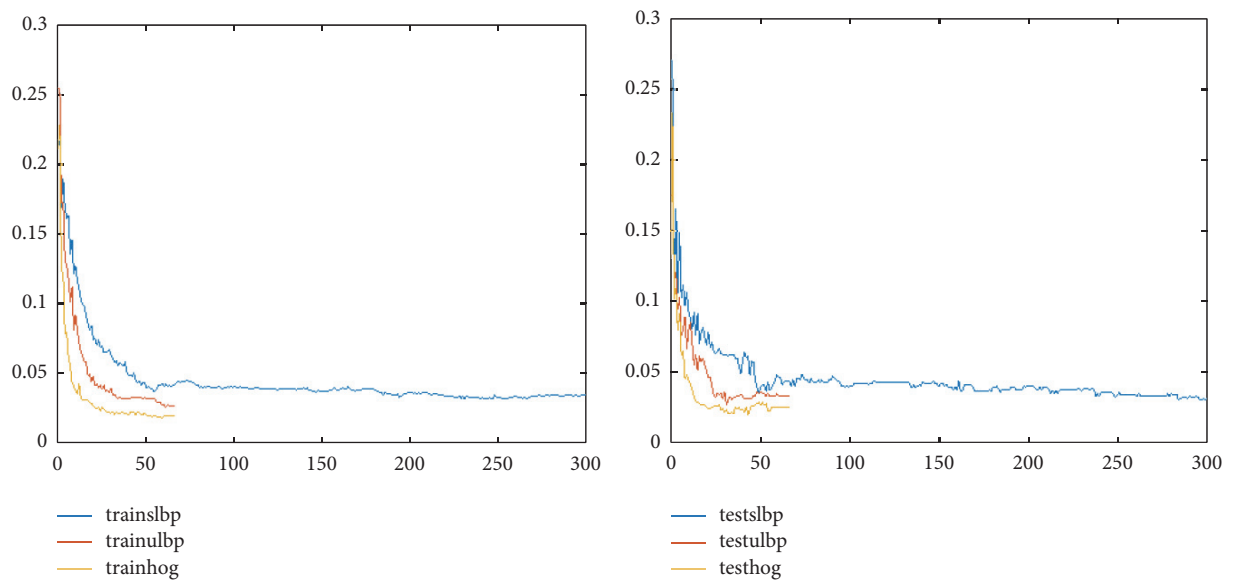


FIGURE 6: Classification error with respect to the number of weak classifiers.

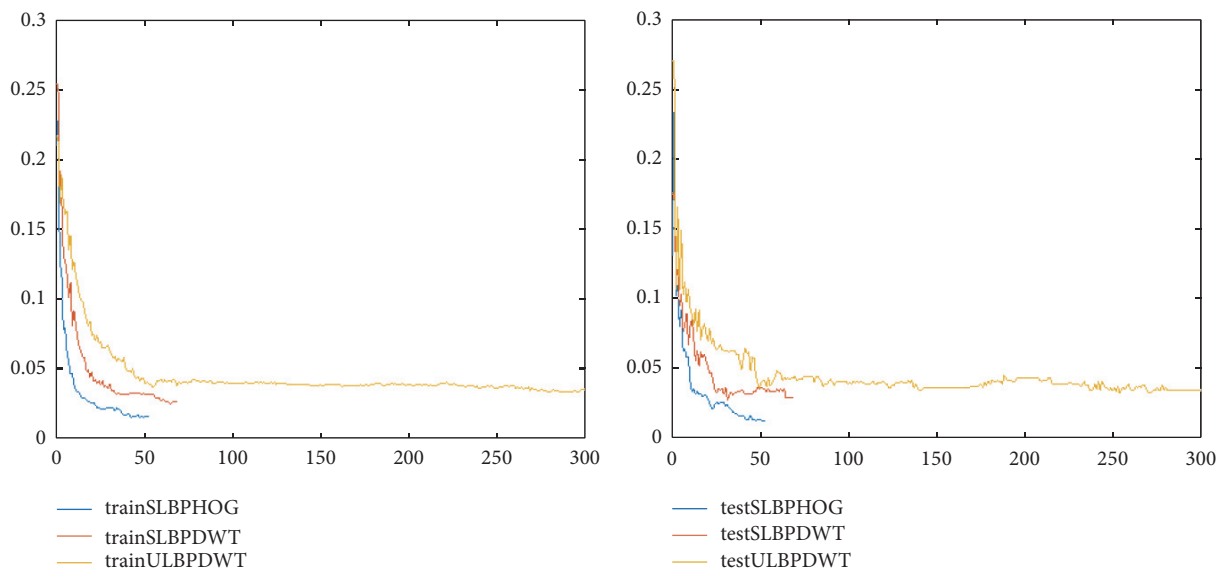


FIGURE 7: Classification error with respect to the number of weak classifiers.

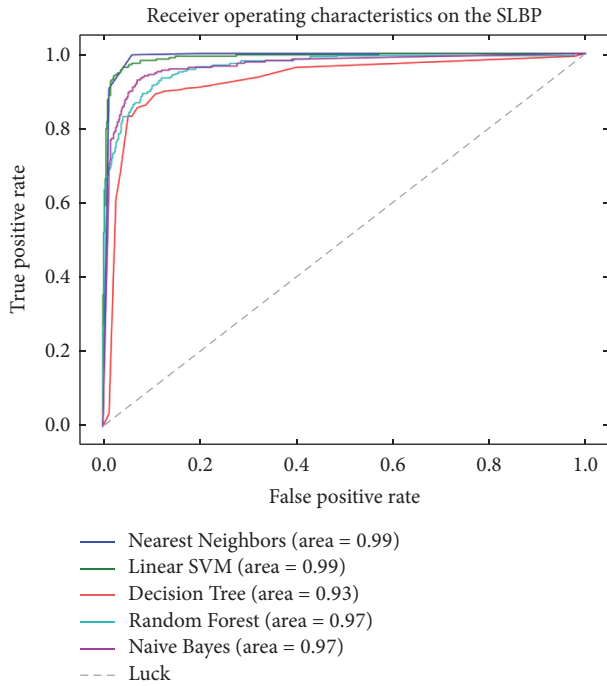


FIGURE 8: Comparison results of different classifiers methods on SLBP.

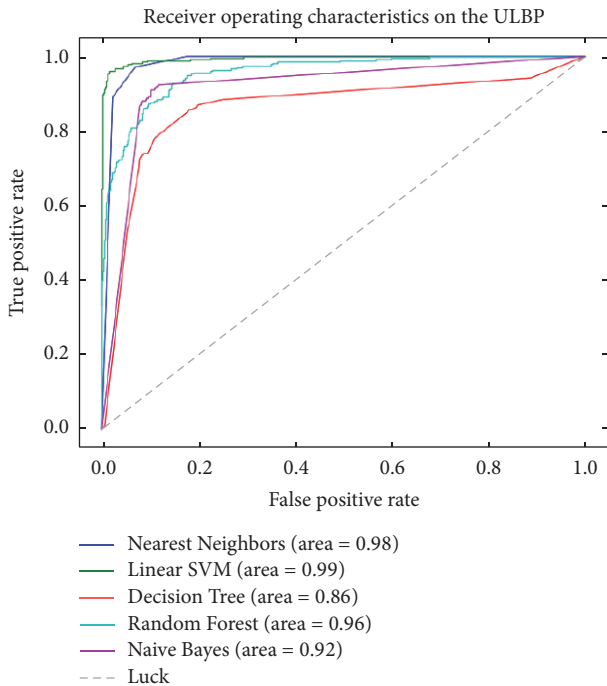


FIGURE 9: Comparison results of different classifiers methods on ULBP.

So, every time we use LBP-NSVC in our experiments, we must consider polynomial kernel to obtain more accurate results. The following sections show a comparison of the accuracies achieved with our experiences and other classifiers.

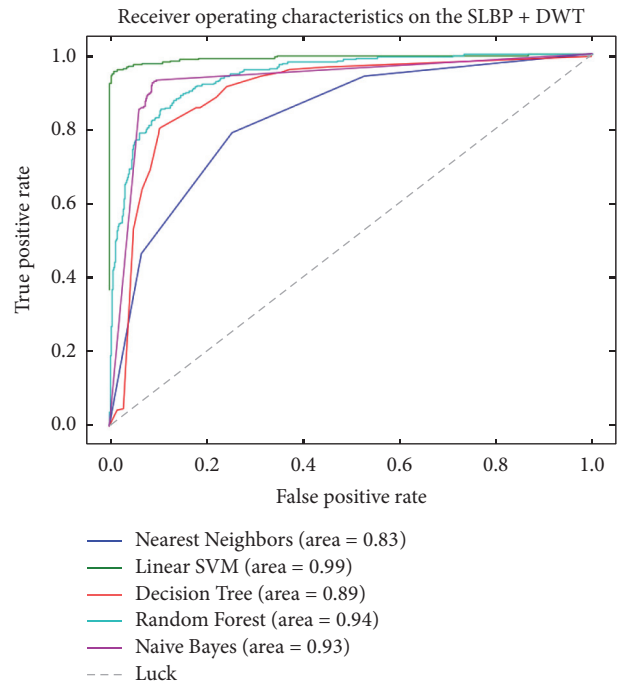


FIGURE 10: Comparison results of different classifiers methods on SLBP + DWT.

4.3. Results of Adaboost. After classifying our database using Adaboost, the method of boosting is particularly interesting because we can choose the number of classifiers in order to achieve the desired error rates on samples examples. Moreover, we observe that the error rate decreases exponentially with the number of used weak classifiers (Figures 6 and 7).

Figure 6 shows the classification error with respect to the number of weak classifiers for SLBP, ULBP and HOG with Adaboost, and SLBP + DWT, SLBP + HOG, and ULBP + DWT for Figure 7.

4.4. Classifiers Comparison. LBP-NSVC gave the best results for dataset. We seek to demonstrate the performance of this method in comparison with other classification methods.

Classifiers used for our comparison experiments are the following: Naive Bayes, Decision Tree, K Nearest Neighbors (KNN), linear Support Vector Machines (linear SVM), and Random Forest. We compare the classification results of these five algorithms together with our proposed NSVC on the SLBP (Figure 8), ULBP (Figure 9), SLBP + DWT (Figure 10), SLBP + HOG (Figure 11), and ULBP + DWT (Figure 12).

Figures 8–12 show the percentage of classification accuracy of different matching algorithms. It clearly shows that the classification accuracy is best for the majority of algorithms.

It is clear that the approach of the proposed LBP-NSVC produced the best or equal classification accuracy compared to other methods.

In addition to its high performance, the NSVC is a new theoretical method of classification which combines two methods of classification belonging to two different families (unsupervised method: Fuzzy C-Means and supervised method: SVM).

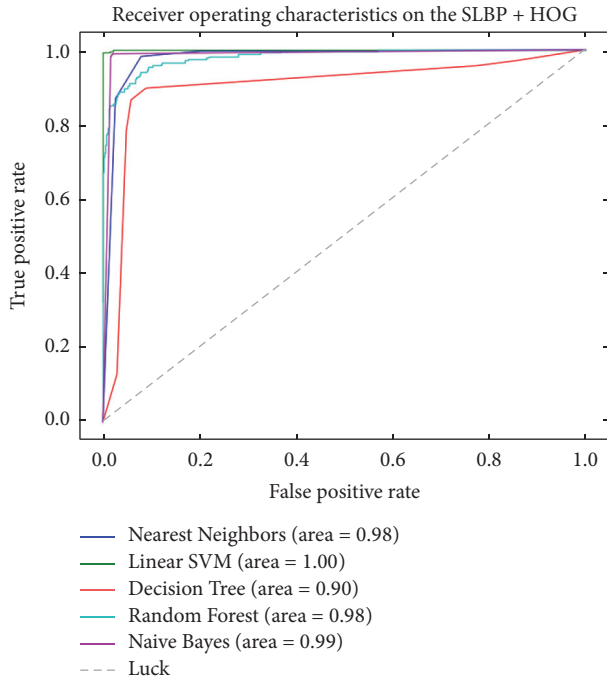


FIGURE 11: Comparison results of different classifiers methods on SLBP + HOG.

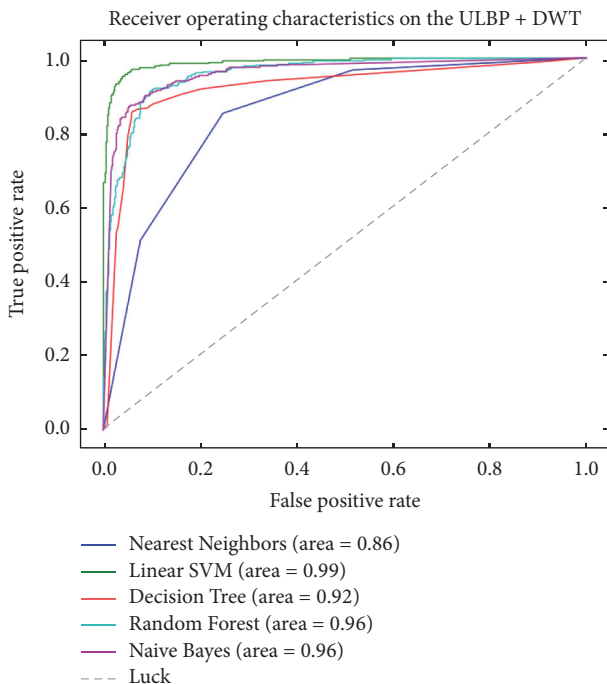


FIGURE 12: Comparison results of different classifiers methods on ULBP + DWT.

5. Conclusion

We have proposed an original method for face detection. Our system is based on the combination of two types of information: LBP descriptors and descriptors such as DWT and HOG. In order to manage these descriptors and combine

them in an optimized way, we propose using an advanced learning system the NSVC. It allows selecting the most important information through kernel weighting depending on their relevance.

The experimental results on different real images show that the proposed method can get very good results.

Our goal in the near future is to continue the study of LBP-NSVC to test it on different datasets from other research areas and try to find the best compromise between precision and execution time.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: past, present and future," *Computer Vision and Image Understanding*, vol. 138, pp. 1–24, 2015.
- [2] A. Amine, S. Ghouzali, M. Rziza, and D. Aboutajdine, "Investigation of feature dimension reduction based DCT/SVM for face recognition," in *Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC '08)*, pp. 188–203, July 2008.
- [3] A. Majid, A. Khan, and A. M. Mirza, "Gender classification using discrete cosine transformation: a comparison of different classifiers," in *Proceedings of the 7th IEEE International Multi Topic Conference (INMIC '03)*, pp. 59–64, Islamabad, Pakistan, December 2003.
- [4] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 32–39, September 2009.
- [5] O. Jemai, M. Zaied, C. Ben Amar, and M. A. Alimi, "Fast learning algorithm of wavelet network based on fast wavelet transform," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 8, pp. 1297–1319, 2011.
- [6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 886–893, San Diego, Calif, USA, June 2005.
- [7] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [8] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*, vol. 40, Springer, 2011.
- [9] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision—ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11–14, 2004. Proceedings, Part I*, vol. 3021 of *Lecture Notes in Computer Science*, pp. 469–481, Springer, Berlin, Germany, 2004.
- [10] A. Gunay and V. V. Nابیev, "Automatic age classification with LBP," in *Proceedings of the 23rd International Symposium on Computer and Information Sciences (ISCIS '08)*, pp. 1–4, Istanbul, Turkey, October 2008.
- [11] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

- [12] T. Ojala, M. Pietikäinen, and T. Mäenpää, “A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification,” in *Advances in Pattern Recognition—ICAPR 2001: Second International Conference Rio de Janeiro, Brazil, March 11–14, 2001 Proceedings*, vol. 2013 of *Lecture Notes in Computer Science*, pp. 399–408, Springer, Berlin, Germany, 2001.
- [13] A. Hadid, M. Pietikainen, and T. Ahonen, “A discriminative feature space for detecting and recognizing faces,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, Washington, DC, USA, June 2004.
- [14] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An introduction to kernel-based learning algorithms,” *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.
- [15] X. Yang, A. Cao, Q. Song, G. Schaefer, and Y. Su, “Vicinal support vector classifier using supervised kernel-based clustering,” *Artificial Intelligence in Medicine*, vol. 60, no. 3, pp. 189–196, 2014.
- [16] A. Cao, Q. Song, X. Yang, S. Liu, and C. Guo, “Mammographic mass detection by vicinal support vector machine,” in *Proceedings of the IEEE International Joint Conference Neural Networks*, vol. 3, pp. 1953–1958, Budapest, Hungary, July 2004.
- [17] M. Ngadi, A. Amine, H. Hachimi, and A. El-Attar, “A new optimal approach for Breast Cancer Diagnosis Classification,” *International Journal of Imaging and Robotics*, vol. 16, no. 4, pp. 25–36, 2016.
- [18] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 2nd edition, 2000.
- [19] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, “Vicinal risk minimization,” in *Proceedings of the 14th Annual Neural Information Processing Systems Conference (NIPS '00)*, MIT Press, December 2000.
- [20] F. Camastra and A. Verri, “A novel Kernel method for clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 801–805, 2005.
- [21] J. M. Leski, “Fuzzy c -varieties/elliptotypes clustering in reproducing kernel Hilbert space,” *Fuzzy Sets and Systems*, vol. 141, no. 2, pp. 259–280, 2004.
- [22] K. Rose, “Deterministic annealing for clustering, compression, classification, regression, and related optimization problems,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

