IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 April 2023, accepted 27 April 2023, date of publication 1 May 2023, date of current version 10 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271997

RESEARCH ARTICLE

Pose-Invariant Face Recognition via Facial Landmark Based Ensemble Learning

SHINFENG D. LIN, (Senior Member, IEEE), AND PAULO E. LINARES OTOYA
Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien 97401, Taiwan
Corresponding author: Shinfeng D. Lin (david@gms.ndhu.edu.tw)

ABSTRACT In recent years, pose-invariant face recognition has been mainly approached from a holistic insight. DCNNs (ArcFace, Elastic Face, FaceNet) are used to compute a face image embedding, which is used later to perform face recognition. This paper presents a novel approach to pose-invariant face recognition through the use of ensemble learning and local feature descriptors. The proposed method trains a base learner for each person's face recognition ensemble system, based on feature vectors (SIFT, GMM, LBP) extracted from image regions surrounding specific facial landmarks. Three different classification models (SVM, Naive Bayes, GMM) are exclusively used as base learners, and the training procedure for each of these models is detailed. The proposed methodology includes a novel face pose descriptor referred to as the Face Angle Vector (FAV) which is utilized by a head pose classification model to determine the pose class of a face image. This model works together with a Base Learner Selection (BLS) block, to determine a set of facial landmarks to extract local feature descriptors, and uses them as the input to their corresponding base learners. Experimental results show a better performance over state-of-the-art methods using the CMU-PIE as the testing dataset, and face poses within $\pm 90°$.

INDEX TERMS Ensemble learning, facial landmarks, local feature descriptors, pose-invariant face recognition, base learner selection.

I. INTRODUCTION

The significant role played by face recognition in real-world applications (biometric authentication, surveillance, security and law enforcement, health, education, marketing, finance, entertainment, and human-computer interaction) has been widened in the last decade, and the accuracy of the most recent works on face recognition has revealed a relevant improvement since standard face databases were established for comparison at the beginning of 1990s [1], [2]. The problem of face recognition (FR) can be approached as an identification, or a verification problem. Face identification is also referred to as the 1:N matching problem. The unknown face is compared with all the faces in the database of known identities and a decision is made as a result of all the comparisons. If the person is known to be in the database, the task is called as closed-set, otherwise, it is called as open-set. On the other hand, face verification is known as the 1:1 matching problem. The identity of the query face is either confirmed or rejected by comparing it with the face data of the claimed identity in the database [2]. Face recognition has been one of the most active research topics in computer vision for more than three decades. Over the years, significant progress has been made in automatic face recognition, with promising results being achieved in both controlled and uncontrolled environments, as well as across various types of images, including color, thermal, and multispectral images [3]. However, face recognition remains significantly affected by the wide variations of pose, illumination, and expression often encountered in real-world images [4], [5], [6].

Pose-invariant face recognition (PIFR) implies the problem of recognizing a person by analyzing face images of different poses (i.e. pose variation) [1], [7]. In other words, the identity of an unknown person is obtained by processing a non-frontal view face image of this person. However, the dramatic appearance changes (e.g. self-occlusion, nonlinear variations on the visible facial texture) in the face image caused by pose variation, make the face recognition under

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

different poses a still challenging problem (especially for pose angles beyond 45°), as argued in recent works. The most common setting for both the research and application of pose-invariant face recognition is to handle the identification problem of matching an arbitrary pose probe face with frontal gallery faces [5]. The pose problem is also usually combined with other factors, such as variations in illumination and expression, to affect the appearance of face images. In consequence, the extent of appearance change caused by pose variation is usually greater than that caused by differences in identity [5].

Many promising approaches have been proposed to tackle the pose challenge in face recognition. These methods can be broadly classified according to an approach taxonomy. Ding et al. [5] proposed a taxonomy comprising two main methodologies for PIFR. The first one is synthesis-based methods. This method can be implemented with 2D or 3D techniques. For 3D face synthesis a 3D Face Morphable Model (3DMM) must be employed. On the other hand, synthesis-free methods aim to extract pose-robust features or transform features from different poses into a shared feature subspace.

Another taxonomy is detailed in [1]. In this taxonomy, the authors classify PIFR methods into 3D, and 2D approaches. If a 3D approach is employed, it is essential to obtain 3D face data information to normalize the pose-view (i.e. generate a frontal image from a pose-view image). This can be accomplished by using the 3D face model for more accurate reconstruction, or by using a 3DMM (3D Morphable Model) to synthesize face images under new poses using the textured 3D face model. Nonetheless, if a 2D approach is utilized, multiple-view face images are required during the learning step to improve the face recognition accuracy on a neutral face image database. Another option for a method under a 2D approach is to use pose normalization within the 2D image domain. However, this technique has shown to be computationally expensive.

Zhang et al. [8] divided the methods employed to address PIFR into holistic methods and local methods. Holistic methodologies [9], [10] aim to extract discriminative feature embeddings from facial images by treating the images as a whole. These methods still cannot perfectly address the problem of facial pose variation, and the face recognition performance of these holistic methods degrade when images are captured under unrestricted environments. Conversely, the local methods [8], [11], [12] usually consider several facial regions or sets of fiducial points, from which features for classification are extracted (e.g. LBP, Gabor features).

In this paper, we present a robust pose-invariant face recognition system which exploits the concept of ensemble learning to develop face recognition models trained with data extracted from image regions surrounding a set of selected facial landmarks. The main novelty of our proposed method is the way a base learner, constituting a face recognition ensemble system (one ensemble per person), is trained according to the feature vectors extracted from image regions around an specific facial landmark. In other words, we are proposing a link between a base learner and a facial landmark. We select 3 different classification models (SVM, Naive Bayes, GMM) to be used as base learners exclusively. Furthermore, the training procedure for each of these models, according to the proposed methodology, is detailed. First, the input face image is processed by face and facial landmark detectors. Second, the locations of the facial landmarks are processed according to a novel face pose descriptor referred to as the Face Angle Vector. This descriptor is fed to a Head Pose Classification model, which returns the pose class, defined by a finite set of pose angle ranges. Third, the predicted pose class is utilized by a Base learner selection (BLS) block. Indeed, this block select a set of facial landmarks to extract local feature descriptors. Then, the descriptor of each landmark is used as the input to its corresponding base learner. We are proposing 3 feature descriptors (SIFT, HOG, LBP) to be used separately. The outputs of the utilized base learners are then combined (according to a combination rule) to compute the ensemble decision support. The mean rule, and trimmed mean rule are utilized independently to assess its impact on the overall recognition performance. At the end, the identity of a face image is obtained by choosing the ID (identity) of the ensemble system with the highest decision support.

We evaluate the performance of the proposed methods on the CMU-PIE database, and compare the results with state-of-the-art methods to show a surpassing performance, especially for face images with a large pose angle. In summary, the contributions of this paper are listed as follows:

- We propose a face pose descriptor, called FAV. This descriptor utilizes the 2D locations of a landmark group, and allows to classify the face pose angle into a finite set of pose angle ranges.
- A new base learner training, and facial landmark description procedures are introduced in this work. Besides, the potential of linking a base learner with a specific facial landmark on an ensemble-based face recognition model, is showed empirically in this work.
- We present a base learner selection algorithm. This algorithm aims to reduce the computational time during face recognition, while keeping a high recognition rate.
- The results of using 3 different feature descriptors (SVM, Naive Bayes, GMM), and 3 different base learner models (SIFT, HOG, LBP) are included in this work for a further comparison.
- The current works using the CMU-PIE database for pose-invariant face recognition employed only the Rank-1 accuracy for assessing face identification performance. We conduct additional experiments on face verification and identification. The TAR@FAR and Rank-N accuracy are used as the performance metrics.

The subsequent sections of this paper are organized as follows. In Section II, we present the related work on facial landmark description, and ensemble learning for face recognition.

In Section III, the proposed methodology is detailed. The experimental results, as well as a comparison with state-of-the-art works, are detailed in Section IV. Finally, the paper is concluded in Section V.

II. RELATED WORK

A. WORK ON FACIAL LANDMARK DESCRIPTION

Most of the works using local descriptors for face recognition, obtain and process these descriptors in a traditional way. First, a generic keypoint detector based on the image gradient is typically employed, and feature vectors are obtained from these points. In contrast to this approach, some works have focused on detecting a predefined set of facial landmarks in a face image and considering them as keypoints for further extracting feature descriptors. These vectors are then used for face recognition.

A methodology considering multi-scale image patches centered at 31 landmarks (inner face), is proposed in [8]. Each of these multi-scale patches was processed using the Weber Local Descriptor (WLD) to obtain a WLD histogram for each selected landmark. The authors considered these histograms as individual histograms. They then grouped the histograms according to the facial regions they belong to (eyes, eyebrows, nose, and mouth), and proposed a feature fusion method to obtain a set of fusion histograms. Both individual and fusion histograms were used to train a KNN-based face recognition model. To perform face recognition, these histograms are fed to a group of KNN classifiers, and the final recognition result is obtained by using a majority voting rule. Experimental results were obtained on databases with small pose variation (ORL, FERET, GT), and large pose variation (LFW) databases. The method achieved accuracies between 85 - 97% on databases with low pose variability. However, it showed a high dependency on the number of training images per person. On the other hand, the accuracy dropped to 45% on the LFW database.

Some other authors differ in the way they utilize landmarks as keypoints to extract feature descriptors. Umer et al. [13], extracted image patches around a group of selected landmarks from a detected face image, and concatenated them to generate a new image. The SIFT algorithm is then applied to this new image, and the resulting feature vectors are clustered using K-means to obtain a codebook. Later, the codebook is used to obtain a new set of feature vectors using Locality-constrained Linear Coding (LLC) and Spatial Pyramid Matching (SPM). The face recognition model comprises a multiclass SVM classifier, where each class represents one subject. They conducted experiments for face identification and verification on the IITK, CASIA-V5, LIBOR, ORL, and Extended Yale B databases. According to their results, they achieved an accuracy of 69.17 - 100.00%. However, they used more than 50% of the available training images per subject.

Face recognition is not the only field facial landmark description can be used for. Indeed, a gradient-magnitude-based feature extraction method around facial landmarks were proposed for a gender recognition system [14]. This feature extraction consisted of detecting the landmark locations and computing the gradient magnitude of each color channel of a face image at those locations. Then, a face descriptor vector is built from these magnitude values. The authors employed a 68-landmark scheme (the final vector had 204 elements), and a Linear SVC as the classification model. During the evaluation of this method, the FERET database was modified to be used for gender classification, and an accuracy of $77.5 - 83\%$ was achieved.

The locations of a selected group of facial landmarks can be used to generate a feature vector as well. This approach (geometrical approach) was employed to perform emotion recognition in [15]. The methodology consisted of locating 14 landmarks (related to the eyebrows and mouth movements), on a 68-landmark scheme, and calculated the normalized distances between pairs of these landmarks. These distances comprised the elements of the feature vector. To perform emotion recognition, a Random Forest model was trained with the proposed feature vector computed over 156 images of the Extended Cohn-Kannade (CK+) database. The achieved average accuracy (tested on 225 frontal images) reached a value of 90%.

B. WORK ON ENSEMBLE LEARNING FOR FACE RECOGNITION

Yuan and Abouelenien [16] proposed a multi-class AdaBoost-based boosting method, called MultiBoost, to address face recognition with imbalanced training data. In order to recover balance among all classes (minority and majority classes), the authors proposed a resampling strategy (perturbation strategy) to diversify the training data according to the smallest class size (i.e. subject). The boosting model comprises a set of base learners trained independently on different balanced data (eigenfaces) sets. The number of employed base learners was obtained according to a weighted face recognition error function. The weights employed in this error function are updated in an iterative way, such that the error value of the currently trained base learner is employed to compute the error weights of the next base learner. Furthermore, the error value for a given base learner determined how much a learner contributes to the final decision (i.e. its decision weight). At the end, face recognition is performed by gathering the recognition results from all the base learners and combining them using a weighted majority voting rule. In order to obtain experimental results, the authors used AT&T, AR, and Yale face databases and two synthetic data sets, and simulated imbalanced training data conditions. The results showed an improved performance under highly imbalanced data scenarios.

Feng et al. [11] proposed an ensemble learning face recognition system, which utilized 3 feature descriptors (PCA, LBP, GIST), and combined the recognition results (based on these features) to get a final decision. For each face image, they created a 5-scale image pyramid (5 face images were generated by downsampling each face image), and applied the 3 descriptors to each image in the pyramid. Thus, 15 face

recognition models were trained for each subject. The effectiveness of the proposed ensemble identification algorithm was proved by conducting experiments on the ORL face database. The recognition rate of the proposed ensemble system (97.65%) was higher than that of the results obtained by performing face recognition using PCA, LBP, and GIST independently (89.21%, 93.24%, and 94.84%, respectively). Therefore, the improvement obtained by using an ensemble learning approach was shown.

Choi and Lee [17] proposed a Gabor DCNN (GDCNN) ensemble FR method which exploits various Gabor face representations as inputs during training and testing phases of a DCNN ensemble comprising several VGG-Face and Lightened CNN as base learners. Instead of using grayscale or color input representations for FR, they proposed the use of different Gabor face representations (different parameters are used for each Gabor filter) to train an ensemble of DCNNs and to execute DCNN-based ensemble FR. The study suggested that this approach can be useful for learning different and complementary DCNN models for a given FR task. The authors also proposed an effective decision rule called Confidence based Majority Voting (CMV) as a decision rule to combine multiple and complementary FR outputs obtained from the proposed GDCNN ensemble, which results in significantly enhanced FR performance on challenging face datasets (FERET, CAS-PEAL-R1, FLW, MegaFace). The proposed method achieved recognition rates of 93.6% and 80.09% on the FLW and MegaFace databases, respectively.

III. PROPOSED METHODS

A. FACE POSE CLASSIFICATION

The problem of finding the orientation of a human face in a digital image is called Head Pose Estimation (HPE). This problem involves processing a raw image containing a face and computing its three orientation angles (yaw, pitch, roll). HPE has been considered a crucial task in computer vision given its potential applications on human behavior analysis, driving safety, surveillance and VR systems [18]. In the last ten years, this problem has been addressed using different approaches (e.g. 2D appearance based methods, geometric methods, regression methods, Deep learning methods) and the performance of each proposed method was tested on publicly available datasets (e.g. Pointing04, BIWI, AFLW, VGGFace2) [18], [19], [20].

As the literature shows, HPE can be implemented on two different levels. The first one is the coarse level. This implementation level aims to identify a head pose from a finite set of orientations (i.e. head pose classification). The granular level conversely, returns continuous values for the rotation angles. Thus, the head pose classification (HPC) problem is defined as a simplified way to implement HPE. In this problem, the number of pose classes N_{pose}, and the face pose representation of a face image $\mathcal{X}(I_j)$ are defined at the beginning. The main goal is to train a model which can predict the pose class $y'_j \in \{1, 2, .., N_{pose}\}$ given $\mathcal{X}(I_j)$, such that the expression $\sum_j [y'_j = y_j]$ is maximized. Where $y_j \in \{1, 2, .., N_{pose}\}$ is the real pose label for the j^{th} face image and I_j is the face image [21]. The simplest case comprises $N_{pose} = 3$, where a face image is classified into a frontal or profile face (right and left profiles) [22].

The HPC problem implies the definition of a face pose representation. This face pose representation must convey useful data about the face pose of a person in a digital image such that pose classification can be carried out accurately. For the purposes of this work, the term face pose descriptor is employed to denote face pose representation. The methodology employed to address HPC determines the definition of the face pose descriptor [18]. Some works use 3D information, while others use 2D information to infer the pose of the head. We are employing a geometrical approach in a 2D environment. Thus, the face pose descriptor is defined as a vector containing information about the geometrical relations between a set of facial features (e.g. eyes, eyebrows, nose, lip, cheeks) on a face image.

As was mentioned above, a geometrical approach is utilized in the current work. Indeed, we propose a face pose description method which processes the 2D spatial locations of a set of facial landmarks and computes a vector which will be used later for training and testing purposes. This vector is called the Face Angle Vector (FAV), and it comprises the angles between the mouth landmarks and the eye centers. In order to compute the FAV, the angle between two points $\mathbf{v_a}, \mathbf{v_b} \in \mathbb{N}^2$ in a digital image is computed by using the $\mathbf{Atan2}(\mathbf{v_a}, \mathbf{v_b})$ operation, as defined in (1). Where atan2 is the 2-argument arctangent function.

$$\mathbf{Atan2}(\mathbf{v_a}, \mathbf{v_b}) = \text{atan2}(\mathbf{v_b}[0] - \mathbf{v_a}[0], \mathbf{v_a}[1] - \mathbf{v_b}[1]) \quad (1)$$

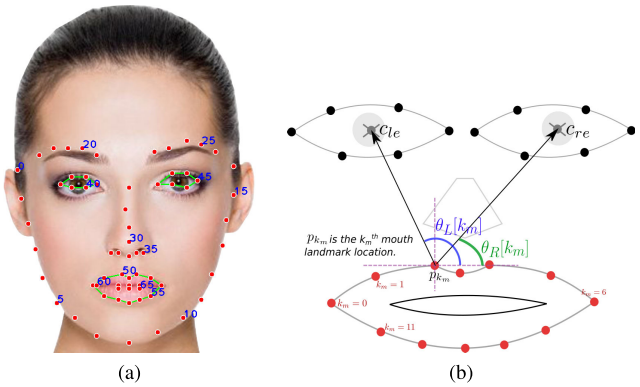
In the present study, a facial landmark detection model [23] with a 68-landmark scheme (Fig. 1a) is adopted, and $L_m = 12$ mouth facial landmarks (depicted in pink on Fig. 1b) are considered. The location of the k_m^{th} mouth landmark on a face image is denoted as $\mathbf{p}_{k_m} \in \mathbb{N}^2$. The angles between the mouth landmarks and the left eye center $c_{le} \in \mathbb{N}^2$ are contained in the vector $\theta_\mathbf{L} \in \mathbb{R}^{L_m}$. While $\theta_\mathbf{R} \in \mathbb{R}^{L_m}$ contains the angles between the mouth landmarks and the right eye center $c_{re} \in \mathbb{N}^2$. These vectors are defined in (2) and (3) respectively. A graphical representation of $\theta_\mathbf{L}[k_m]$ and $\theta_\mathbf{R}[k_m]$, for the k_m^{th} mouth landmark, is shown in Fig. 1b. At the end, the FAV $\Omega \in \mathbb{R}^{2L_m}$ is defined in (4).

$$\theta_\mathbf{L}[k_m] = \mathbf{Atan2}(\mathbf{p}_{k_m}, c_{le}); k_m \in \{0, 1, .., L_m - 1\} \quad (2)$$

$$\theta_\mathbf{R}[k_m] = \mathbf{Atan2}(\mathbf{p}_{k_m}, c_{re}); k_m \in \{0, 1, .., L_m - 1\} \quad (3)$$

$$\Omega = \begin{bmatrix} \theta_\mathbf{L} \\ \theta_\mathbf{R} \end{bmatrix} \quad (4)$$

As can be seen in (4), the vector Ω comprises $\theta_\mathbf{L}$ and $\theta_\mathbf{R}$. Then, Ω can be expressed in terms of these two vectors as defined in (5). Some face image examples and their



(a) (b)

FIGURE 1. Face landmarks used for computing the FAV: (a) The employed 68-Facial landmark scheme; (b) Geometric representation of the angles comprising the FAV.

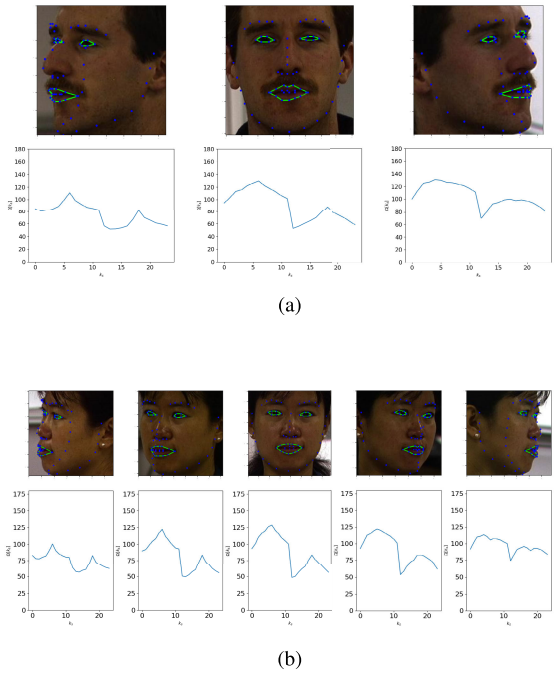
corresponding FAVs are depicted in Fig. 2.

$$\Omega[k_s] = \begin{cases} \theta_{\mathbf{L}}[k_s] & \text{if } k_s < L_m \\ \theta_{\mathbf{R}}[k_s - L_m] & \text{otherwise} \end{cases} ; \quad k_s \in \{0, 1, .., 2L_m - 1\} \tag{5}$$

In the proposed face pose classification method, after the definition of the face pose descriptor (i.e. FAV), the next step is to train a classifier which can predict the face pose class y' of an input image I, from a finite set of pose classes $\{1, 2, 3, \ldots, N_{\text{pose}}\}$, by using its FAV as input. The training process consists of defining N_{pose} and assigning an angle range to each pose class. Then, a set of face images with known yaw pose angles that vary between $\pm 90°$ is selected. The third step involves computing the FAV Ω_j for each selected image I_j and assigning a pose class value y_j according to its real pose angle value (i.e. ground truth data). The face pose classification model is trained with all the Ω_j as input data, and their y_j as the ground truth labels. A Linear Support Vector Classifier (Linear SVC) is employed as the pose classification model. The number of pose classes are set to $N_{\text{pose}} = 3$ (Fig. 2a) and $N_{\text{pose}} = 5$ (Fig. 2b) independently, to assess its effect on the overall face recognition performance. During the testing stage of the proposed face pose classifier, its multiclass confusion matrix metrics are analyzed to assess the classification performance for each N_{pose} instance.

B. FACIAL LANDMARK DESCRIPTION

The procedure carried out for describing the facial information surrounding a specific landmark on a face image is defined as Facial landmark description. This procedure is based on the description of a generic point (i.e. a pixel comprising its intensity and location) on a digital image. In this work, three local descriptors are considered for facial landmark description. Each of the local descriptors employed in this work requires the specification of a set of parameters (summarized in Tabl. 1), which must be tuned precisely.



(a) (b)

FIGURE 2. Sample images and their corresponding Face Angle Vectors (FAV): (a) For $N_{\text{pose}} = 3$ pose classes. ; (b) For $N_{\text{pose}} = 5$ pose classes.

Otherwise, the face recognition performance might be affected drastically.

The first local descriptor is SIFT. In this work, we do not employ the SIFT keypoint detector. But, a generic point is converted into a SIFT keypoint by specifying the point location and its diameter [24]. Then, the SIFT description algorithm is computed from this SIFT keypoint and the result is considered as the image point descriptor. This process is carried out by the function *ComputeSIFT()*, defined in Algorithm 1. The second local descriptor is HOG. The function *ComputeHOG()*, in Algorithm 1, details the steps performed to obtain the HOG descriptor vector. First, a patch $\mathbf{I}_{\text{patch}}$ is obtained from the image by using the function *getPatch()*. This patch is a small region of the main image I, centered at the point $p \in \mathbb{N}^2$ and with a size of s_p (e.g. 40×40 pixels). Then, the HOG vector is computed from this patch, according to [25], by using the *getHOGDescriptor()* function. The size of the HOG vector $|f_{\text{HOG}}|$ depends on its parameters K_{hog}, ppc_{hog}, cpb_{hog} and also on the patch size s_p, as defined in (6).

$$n_{\text{blocks}} = \lfloor \frac{s_p}{ppc_{\text{hog}}} - cpb_{\text{hog}} \rfloor + 1$$
$$|f_{\text{HOG}}| = n_{\text{blocks}}^2 cpb_{\text{hog}}^2 K_{\text{hog}} \tag{6}$$

The third local descriptor employed in this work is LBP. The process for computing this descriptor is described in the function *ComputeLBP()* in Algorithm 1. As for the HOG descriptor, the *getPatch()* function is employed to obtain a small region of interest in the image. Then, the LBP algorithm [26], expressed as the *getLBP()* function, is used to

TABLE 1. Facial landmark descriptor parameters.

Descriptor name	Parameter	Parameter description
SIFT	σ_{sift}	The sigma value of the Gaussian filter applied to the input image at the octave #0
	$\varnothing_{\text{sift}}$	Keypoint diameter
HOG	K_{hog}	Number of orientations
	ppc_{hog}	Pixels per cell
	cpb_{hog}	Cells per block
	s_p	Patch size
LBP	K_{lbp}	Number of circularly symmetric neighbor set points
	r_{lbp}	Neighborhood circle radius
	n_b	Number of bins on the histogram
	s_p	Patch size

compute the LBP image $\mathbf{I}_{\text{LBP}}$ from the image patch $\mathbf{I}_{\text{patch}}$ (with $K_{\text{lbp}} = 16$, and $r_{\text{lbp}} = 3$). Finally, the function *getHistogram()* constructs a normalized histogram, with n_b bins, from the intensity values of $\mathbf{I}_{\text{LBP}}$. This histogram is considered as the descriptor vector using LBP.

So far, the process for obtaining a descriptor vector from a generic point on an image has been explained. This process is used later to compute the landmark description matrix as can be seen in Algorithm 1 (lines 20−26). First, a list P containing the locations of a selected landmark set on a face image I is obtained by using the function *getLandmarkLocations()*. This function uses the landmark detection model [23] to find the facial landmark locations, according to the above mentioned 68-landmark scheme. Then, some landmarks are selected from the total 68 according to their indexes (e.g. 0, 1, . . . , 67). These indexes are specified by the index list i_{list}.

In the second step, the landmark description matrix M is initialized with zeros. Finally, the rows of M are filled with the descriptor vectors of each landmark in P by using the function *computeDescriptor()*, as observed in lines 23 − 25. Actually, this function is a generic notation for the three descriptor vector computing functions defined in Algorithm 1. In other words, depending on which local descriptor is utilized, *computeDescriptor()* represents *ComputeSIFT()*, *ComputeHOG()*, or *ComputeLBP()*.

Algorithm 1 Facial Landmark Descriptor Matrix Computation From a Face Image

Input: A gray scale image $\mathbf{I}$, landmark index list i_{list}
Result: Facial landmark descriptor matrix $\mathbf{M}$.
Data: Descriptor parameters:
- SIFT: σ_{sift}, $\varnothing_{\text{sift}}$
- HOG: K_{hog}, ppc_{hog}, cpb_{hog}, s_p
- LBP: K_{lbp}, r_{lbp}, n_b, s_p

1: **Function** `ComputeSIFT` $(\mathbf{I},p)$ **:**
2: $\quad$ **kp** = getKeypoint(p,$\varnothing_{\text{sift}}$)
3: $\quad$ **f** = getSIFTDescriptor($\mathbf{I}$,**kp**,σ_{sift})
4: $\quad$ **Return f** $\in \mathbb{N}^{128}$
5: **End Function**
6:
7: **Function** `ComputeHOG` $(\mathbf{I},p)$ **:**
8: $\quad$ $\mathbf{I}_{\text{patch}}$ = getPatch($\mathbf{I}$,p,s_p)
9: $\quad$ **f** = getHOGDescriptor($\mathbf{I}_{\text{patch}}$,$K_{\text{hog}}$, ppc_{hog}, cpb_{hog})
10: $\quad$ **Return f** $\in \mathbb{R}^{|f_{\text{HOG}}|}$ `// |f_HOG| is defined in (6).`
11: **End Function**
12:
13: **Function** `ComputeLBP` $(\mathbf{I},p)$ **:**
14: $\quad$ $\mathbf{I}_{\text{patch}}$ = getPatch($\mathbf{I}$,p,s_p)
15: $\quad$ $\mathbf{I}_{\text{LBP}}$ = getLBP($\mathbf{I}_{\text{patch}}$, K_{lbp}, r_{lbp})
16: $\quad$ **f** = getHistogram($\mathbf{I}_{\text{LBP}}$, n_b)
17: $\quad$ **Return f** $\in \mathbb{R}^{n_b}$
18: **End Function**
19:
20: P = getLandmarkLocations(I, i_{list})
21: $M \leftarrow 0_{|P|\times|f|}$ `// |f| is the size of the selected local descriptor`
22: **for** $(i = 0; i < |P|; i++)$ **do**
23: $\quad$ $M[i]$ = computeDescriptor(I,$P[i]$)
24: **end**
25: **Return** M

C. PROPOSED ENSEMBLE LEARNING SYSTEM FOR PIFR

On a classification environment, a generic ensemble system has three components. The first component is the set of base learners. This set comprises single classifiers which receive the input data (different classifiers can receive different portions of the input data or all the same) and output a classification decision value, which can be discrete or continuous depending on the model employed for a base learner (e.g. SVC, ANN, Naive Bayes). The second component is the base learner training method. The methodology used for training a set of base learners depends on the underlying approach used to obtain a classification result from the ensemble system (e.g. Bagging, Boosting, AdaBoost) [27], [28]. The last component is the combination rule. The combination rule (e.g. mean rule, product rule) defines how the outputs from the base learner set are processed together in order to obtain the ensemble support value. This value is employed later to classify the input data.

In the present work, the ensemble system framework depicted in Fig. 3a, is adopted to carry out face recognition. In this framework, the input face image is processed to find the face bounding box and the facial landmark locations. In the following steps the input face image and landmark locations are processed by two different blocks. The first block performs Face Pose Classification (FPC) as was explained above. According to the face pose class computed by the FPC block, the Base Learner Selection (BLS) block defines which base learners will be used in the ensemble system. Indeed, there is a set B comprising all the available base learners, and only a subset T of selected base learners is used (i.e. $T \subseteq B$). In this work, a base learner is linked exclusively to a facial landmark according to the 68-landmark

scheme. Therefore, after defining the selected base learner subset, the BLS block also specifies which landmarks should be described in the face image. Then, the second block (Facial Landmark Description) receives this information and compute the descriptor matrix from the input face image, as detailed in Algorithm 1.

For the purpose of this work, a base learner $\beta_{j,k}$ is a model expert in performing face recognition in a not very accurate way. This base learner processes the feature descriptor vector obtained from the k^{th} facial landmark of the j^{th} subject on a face database (i.e. a base learner is linked to a facial landmark) to compute its decision support $d_{j,k} \in [0, 1]$. The ensemble system receives the landmark descriptor matrix, and distributes the descriptor vectors to their corresponding base learners, selected by the BLS block, and combines the outputs $d_{j,k}$ of the selected base learners $\beta_{j,k}$ by using the combination rule to compute the ensemble decision support value D_j. The value of D_j indicates the probability that the input image matches the j^{th} subject (i.e. person) for which the j^{th} ensemble system was trained. We are employing the mean rule and trimmed mean rule to compute D_j, as defined in (7), and (8) respectively. The landmark set K, comprises all the landmarks for a given face pose class, while the trimmed set $K' \subset K$ removes the landmarks whose $d_{j,k}$ is in the top or bottom $p_{\%\text{top}}$ of all the $d_{j,k}$ values for a given j.

In order to obtain an unknown person's ID (i.e. to perform face recognition), the ensemble systems trained from all the J subjects on a face database (one ensemble system per subject) are employed as shown in Fig. 3b. Where only the images from the face database are used during the training and testing of the face recognition ensemble systems (i.e. closed-set face recognition). The ensemble decisions from all the ensemble systems are gathered and the person ID is computed by selecting the ID of the ensemble system with the highest decision value. Actually, the predicted identity is called the ensemble face recognition result J_E, and it is defined in (9). It is worth mentioning that, the ensemble face recognition result J_E is different from an ensemble decision value D_j. The first one is the predicted ID from the input image obtained by gathering the decision value from the ensemble systems of all the subjects in the database. While the second one is the probability value of the input face to match with the j^{th} subject.

$$D_j = \frac{1}{|K|} \sum_{k \in K} d_{j,k} \tag{7}$$

$$D_j = \frac{1}{|K'|} \sum_{k \in K'} d_{j,k} \tag{8}$$

$s.t.\ K'$ is the trimmed set of landmarks

$$J_E = \arg\max_{j=1}^{C} D_j \tag{9}$$

$$J_b(k) = \arg\max_{j=1}^{C} d_{j,k} \tag{10}$$

According to the definition of a base learner for this work, a model is trained for a specific landmark on a subject

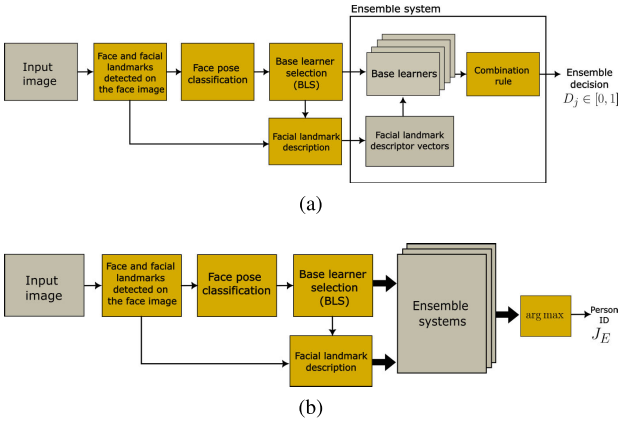


FIGURE 3. Proposed PIFR framework: (a) Proposed framework for a single subject ensemble system; (b) Proposed framework including the ensemble system from every subject in the database. The blocks in orange represent processes or operations while the blocks in gray represents data or objects.

from the database. This process is called base learner training. The steps followed during base learner training depend on the working principle of the model type used as a base learner. Three different models are employed as base learners. The first model is Support Vector Machine (SVM). A base learner using SVM is trained according to Algorithm 2. First, it obtains the ID of the subject j from the face database DB, initializes the ground truth vector $\vec{y}_j$ and descriptor matrix $\mathbf{M_{j,k}}$ with zeros, and creates a base learner object $\beta_{j,k}$. Second, it loops through all the subjects in the face database DB, to fill the elements of $\mathbf{M_{j,k}}$, and $\vec{y}_j$. In order to do this, some images $\mathscr{I}$ for the subject j' are obtained (line 9), according to a face pose angle list θ_{list} (we employ images from pose angles: 0°, 67.5°, −90°). Each of these images I are processed to detect the k^{th} landmark location $p_k \in \mathbb{N}^2$, and obtain the descriptor matrix $\mathbf{M_{j,k}}$ by stacking the feature descriptors computed from points surrounding p_k within a radius r, as depicted in Fig. 4 (lines 10 - 15). Furthermore, the elements of the ground truth vector $\vec{y}_j$ are modified when the identity id' of the subject j' is the same as id (lines 16 - 18). After training the base learner model $\beta_{j,k}$ (line 23), its decision support $d_{j,k} \in [0, 1]$ represents the probability that the landmark descriptor vector f_k obtained from an input face image corresponds to the j^{th} subject (person). The second model is Naive Bayes. The training procedure for this model is the same as for SVM. However, its decision support $d_{j,k} \in \{0, 1\}$ just indicates whether f_k match with the j^{th} subject or not.

The third model is Gaussian Mixture Model (GMM). Its training procedure, detailed in Algorithm 3, slightly differs from the one for SVM. First, the descriptor matrix $\mathbf{M_{j,k}}$ is initialized with zeros (there is no ground truth vector). Second, it obtains the descriptor matrix $\mathbf{M_{j,k}}$ by looping through all the subjects in the face database DB in the same way as for SVM, except for the ground truth vector $\vec{y}_j$. After training a base learner with a GMM model, its decision support $d_{j,k} \in (-\infty, 0]$ is the log-likelihood [29] that the landmark

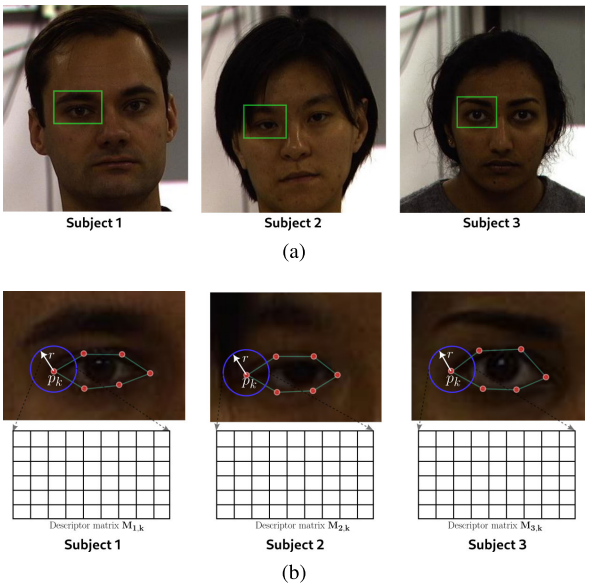


FIGURE 4. Descriptor matrix computing during base learner training for a landmark in the right eye: (a) Face images of three subjects with green rectangles depicting the regions of interest; (b) Descriptor matrices obtained from an eye landmark p_k. A circle in blue indicates the surrounding points of the p_k landmark within a radius r.

descriptor vector obtained from the k^{th} landmark of an input face image corresponds to the j^{th} subject.

After performing the base learner training for each subject, we are proposing a method to use less base learners during the testing stage for face recognition so that computational time can be reduced without considerably affecting the recognition performance. This method is called Base Learner Selection (BLS), and it aims to solve the following problem: "Given a set of available base learners and their face recognition hypothesis, develop a method to select a subset of these base learners in such a way that the recognition performance of using this subset is equal or better than the performance obtained by using all the available base learners". The BLS procedure is conducted for each pose scenario (N_{pose}), and it is shown in Algorithm 4.

The overall recognition rate a_E, the set of selected base learners T, and the individual selected base learner accuracy vector $\vec{\alpha}$ are initialized at the beginning of the BLS algorithm. Second, 8 and 4 subjects are chosen randomly from the whole face database and placed on the S_s and S_v lists, respectively. Third, the individual base learner accuracy vector $\vec{\alpha}_s$ is computed by using the computeBLAccuracy() function. This function, computes the recognition accuracy (on the images from the subjects on S_s) of each base learner in B, according to its face recognition hypothesis. The face recognition hypothesis $J_b(k)$ of the k^{th} available base learner is defined in (10). Where $d_{j,k}$ is the decision support of the k^{th} base learner for the j^{th} subject. Fourth, according to the accuracy values of the available base learners ($\vec{\alpha}_s$), the $p_{\%\text{top}}$ most accurate base learners are selected from B and placed in T_s. Fifth, the ensemble recognition accuracy values

Algorithm 2 Base Learner Training Algorithm for SVM and Naive Bayes

Input: A gray scale image $\mathbf{I}$, landmark index k, gallery face pose angle list θ_{list}
. **Result:** The set of trained base learners $\mathscr{B}$ for the landmark with index k based on the 68-landmark scheme.
Data: Landmark neighborhood radius r

$\mathscr{B} \leftarrow \emptyset$
foreach $j \in DB$ **do**
  $id = \text{getID}(j)$
  $\vec{y}_j \leftarrow \mathbf{0}$ `// initialize label vector with zeros`
  $\mathbf{X_{j,k}} \leftarrow \mathbf{0}$ `// initialize descriptor matrix with zeros`
  $\beta_{j,k} \leftarrow \text{createBaseLearnerModel}(id, k)$
  foreach $j' \in DB$ **do**
    $id' = \text{getID}(j')$
    $\mathscr{I} = \text{getImages}(s', \text{DB}, \theta_{\text{list}})$
    foreach $I \in \mathscr{I}$ **do**
      $p_k = \text{getLandmarkLocations}(I, \{k\})$
      $P' = \text{getSurroundingPoints}(p, r)$
      $i = 0$
      foreach $p' \in P'$ **do**
        $\mathbf{X_{j,k}}[i] = \text{computeDescriptor}(I, p')$
        if $id' == id$ **then**
          $\vec{y}_j[i] = 1$
        end
        $i += 1$
      end
    end
  end
  $\beta_{j,k}.\text{train}(\mathbf{X_{j,k}}, \vec{y}_j)$
  $\mathscr{B} \leftarrow \mathscr{B} \cup \{\beta_{j,k}\}$
end
Return $\mathscr{B}$

$a_{E,s}$, $a'_{E,}$ of using T_s and T as base learners respectively, are computed independently on the images from $S_s \cup S_v$. Sixth, T and a_E are updated on each iteration (lines 9 - 18). The function combineBL() uses $\vec{\alpha}_s$ and $\vec{\alpha}$ to change the elements of T_s. In case an element (i.e. base learner) of T gets a better accuracy than an element of T_s, it is exchanged. After N_{iter} iterations, the definitive set of selected base learners T is returned as the result.

IV. EXPERIMENTAL RESULTS

The proposed method is implemented in Python 3 language on an Arch Linux PC with a Core™ i7-8750H CPU and 8.00 GB of RAM. For face detection, the Google MediaPipe model is employed. Whereas the implementation of [23] is used for facial landmark detection.

Algorithm 3 Base Learner Training Algorithm for GMM

Input: A gray scale image $\mathbf{I}$, landmark index k, gallery face pose angle list θ_{list}
Result: The set of trained base learners $\mathscr{B}$ for the landmark with index k based on the 68-landmark scheme.
Data: Landmark neighborhood radius r

$\mathscr{B} \leftarrow \emptyset$
foreach $j \in DB$ **do**
  id = getID(j)
  $\mathbf{X_{j,k}} \leftarrow \mathbf{0}$ // initialize descriptor matrix with zeros
  $\beta_{j,k} \leftarrow$ createBaseLearnerModel(id, k)
  $\mathscr{I}$ = getImages(j,DB,θ_{list})
  foreach $I \in \mathscr{I}$ **do**
    p_k = getLandmarkLocations(I, $\{k\}$)
    P' = getSurroundingPoints(p, r)
    $i = 0$
    foreach $p' \in P'$ **do**
      $X_{j,k}[i]$ = computeDescriptor(I,p')
      $i+ = 1$
    end
  end
  $\beta_{j,k}$.train($\mathbf{X_{j,k}}$)
  $\mathscr{B} \leftarrow \mathscr{B} \cup \{\beta_{j,k}\}$
end
Return $\mathscr{B}$

Algorithm 4 Base Learner Selection (BLS) Algorithm

Input: Set of available base learners B for n_{pose} pose class
Result: The set of selected base learners T from B
Data: Face image database DB

$a_E = 0, T \leftarrow B, \vec{\alpha} \leftarrow \mathbf{0}$
for $i = 0$; $i < N_{iter}$; $i+ = 1$ **do**
  S_s = selectSubjectsDB(DB, 8)
  S_v = selectSubjectsDB(DB, 4)
  $\vec{\alpha}_s \leftarrow$ computeBLAccuracy(B, S_s)
  T_s = selectBL($B, \vec{\alpha}_s, p_{\%\text{top}}$)
  $a_{E,s}$ = computeAccuracy($T_s, S_s \cup S_v$)
  a'_E = computeAccuracy($T, S_s \cup S_v$)
  if $a_{E,s} \geq a'_E$ && $a_{E,s} \geq a_E$ **then**
    $a_E = a_{E,s}, T = T_s$
  end
  else
    T_m = combineBL($T_s, \vec{\alpha}_s, T, \vec{\alpha}$)
    $a_{E,m}$ = computeAccuracy($T_m, S_s \cup S_v$)
    if $a_{E,m} \geq a_E$ **then**
      $a_E = a_{E,m}, T = T_m$
    end
  end
  $\vec{\alpha} \leftarrow$ computeBLAccuracy($T, S_s \cup S_v$)
end
Return T

TABLE 2. Ensemble learning system parameters.

Parameter	Notation	Definition
Available base learners	B	The set comprising all the base learners available for building an ensemble.
Selected base learners	T	The set comprising the base learners selected from B after BLS.
Base learner	$\beta_{j,k}$	The k^{th} available base learner for the j^{th} subject on a face database.
Ensemble decision support	D_j	The degree of support given by an ensemble system, trained for the j^{th} subject, to an input face image.
Top BLS percentage	$p_{\%\text{top}}$	Percentage of the total number of available base learners having the highest value of accuracy according to their recognition hypothesis.

A. DATABASES

The CMU-PIE database [30] comprises over 40 000 images of 68 subjects. This database has over 600 images from 13 poses (variation in the head yaw and pitch angles), with 43 different illuminations (the authors used a ''flash system''), and with 4 expressions (neutral, talking, blinking, and smiling). In order to verify the effectiveness of the proposed pose-invariant face recognition method, only the images with ambient illumination, neutral expression and yaw angle variation (Fig. 5) are used. Thus, in this work we use 9 images per subject with a total of 612 images.

Regarding the pose class of a face image, two versions are considered for base learner training. The first version consists of training the base learner set for semifrontal images (images within a pose angle $\pm 45°$). For this version, only the images with a pose angle $0°$ are employed. The second version consists of training the base learners for profile images (images with a pose angle $\pm 67.5°$, $\pm 90°$). In this case, the images with pose angles $-90°$ (flipped) and $67.5°$ are employed. In summary, for each subject in the CMU-PIE database, 3 images are employed during base learner training.

B. FACE POSE CLASSIFICATION

As it was mentioned above, two face pose classification instances are considered in this work. The first one consists of classifying a pose-view face according to $N_{\text{pose}} = 3$ classes, while the second one comprises $N_{\text{pose}} = 5$ classes. During the training stage, 9 images of 27 subjects from the CMU-PIE database are utilized (243 images in total). The effectiveness of these classifiers are assessed according to the analysis of their confusion matrices. The results of this analysis, as well as the face pose angle range of each class, are summarized in Table 3 and Table 4 for $N_{\text{pose}} = 3$ and $N_{\text{pose}} = 5$ respectively.

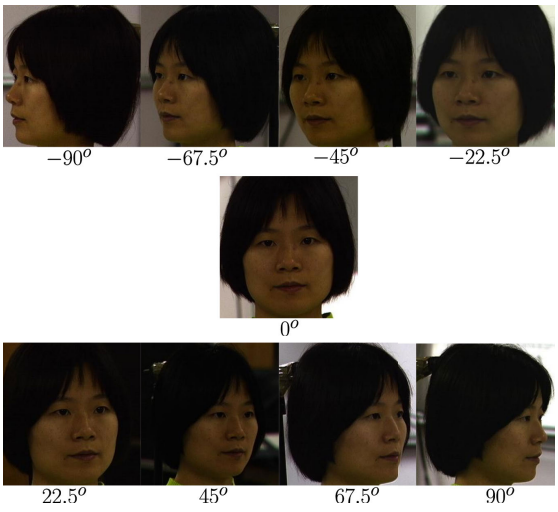


FIGURE 5. Sample images taken from the CMU-PIE database depicting pose variation.

C. PERFORMANCE ON THE CMU-PIE DATABASE

The face recognition results on the CMU-PIE database are mostly expressed in terms of the face recognition rate (also called Rank-1 accuracy). We conduct additional experiments on face verification and identification. The TAR@FAR metric is used to measure the performance on face verification, while the Rank-N accuracy is employed for face identification [31]. The TAR@FAR value is a composed metric aiming to find the similarity score threshold t^* for a given FAR (False Acceptance Rate) value k_{FAR}, as shown in (12). This threshold is then utilized to compute the TAR (True Acceptance Rate) which actually becomes the value of TAR at a given FAR, as defined in (13). Where U is the set of unmatched pairs (a face image from a subject is tested with an ensemble system trained for a different subject), S is the set of matched pairs (a face image from a subject is tested with an ensemble system trained for the same subject), and $d_s : (j, j') \rightarrow [0, 1]$ is the value of the ensemble decision D_j obtained after testing the ensemble system for the j^{th} subject, with a face image of the subject j'. On the other hand, the Rank-N accuracy is defined in (14), where $N_{\text{t-match}}(i)$ denotes the number of probe images with a true match ranked at position i or better (i.e. less than i), and N_{probe} is the total number of probe images [31].

$$\text{FAR}(t) = \frac{|\{d_s(u) \geq t \ ; \ u \in U\}|}{|U|} \tag{11}$$

$$t^* = \text{FAR}^{-1}(k_{\text{FAR}}) \tag{12}$$

$$\text{TAR@FAR}(t^*) = 1 - \frac{|\{d_s(m) < t^* \ ; \ m \in M\}|}{|M|} \tag{13}$$

$$\text{Rank-N}(r) = \sum_{i=1}^{r} \frac{N_{\text{t-match}}(i)}{N_{\text{probe}}} \tag{14}$$

1) INFLUENCE OF THE FACIAL LANDMARK DESCRIPTOR PARAMETERS

Each of the three feature descriptors mentioned in this work requires the setting of a group of parameters (Table 1). Indeed, we conduct additional experimental trials to evaluate the impact of the values assigned to these parameters on the final face recognition performance. For SIFT, σ_{sift}, and $\varnothing_{\text{sift}}$ are adjusted. The patch size s_p, and number of orientations K_{hog} are modified for HOG, while ppc_{hog} and cpb_{hog} are fixed. In the case of LBP, only s_p and n_b are modified. The experimental results of tuning the feature descriptor parameters are summarized in Tabl. 5, and the best value for each performance metric is highlighted in bold. During face verification, a considerable decline in performance can be spotted when $N_{\text{pose}} = 3$ pose classes are considered instead of 5, with SIFT as the descriptor. The effect of varying the descriptor parameters is mostly noticeable when employing LBP. However, in general, the optimal results are achieved by utilizing SIFT as the feature descriptor. By adjusting the SIFT parameters suitably, a recognition rate of 1.000 can be attained.

TABLE 3. Confusion matrix analysis results for face pose classification with $N_{\text{pose}} = 3$ classes.

Class	Angle range	Accuracy	Recall	Precision	F-1
1	$[-90^o, -67.5^o]$	0.992	0.978	0.985	0.982
2	$[-45^o, +45^o]$	0.990	0.994	0.988	0.991
3	$[+67.5^o, +90^o]$	0.998	0.993	1.000	0.996

TABLE 4. Confusion matrix analysis results for face pose classification with $N_{\text{pose}} = 5$ classes.

Class	Angle range	Accuracy	Recall	Precision	F-1
1	$\{-90^o\}$	0.993	0.978	0.993	0.985
2	$\{-67.5^o\}$	0.992	0.985	0.978	0.982
3	$[-45^o, +45^o]$	0.995	1.000	0.958	0.978
4	$\{+67.5^o\}$	0.997	0.993	0.993	0.993
5	$\{+90^o\}$	0.997	0.985	1.000	0.993

2) INFLUENCE OF THE BASE LEARNER SELECTION AND COMBINATION RULE

As mentioned above, three types of base learners and two combination rules are employed independently in the current work for conducting experimental trials. Furthermore, we consider the impact of the BLS algorithm on the face recognition results. These experimental results are shown in Tabl. 6. The main parameter controlling the BLS algorithm is $p_{\%\text{top}}$. In Table 6 we include the face recognition results for $p_{\%\text{top}} = 80$ and $p_{\%\text{top}} = 100$, where the best value for each performance metric is highlighted in bold. As can be seen, the best performance on both face verification and identification are obtained by using SVM as the base learner, mean rule as the combination rule, and BLS with $p_{\%\text{top}} = 100$. However, when SVM is used as base learner, changes in the combination rule, or the $p_{\%\text{top}}$ value do not generate a performance drop during identification. Furthermore, the performance for face verification just experienced a small setback. In general, the use of different combination rules affects mostly the results during face verification. On the other hand, changes in

TABLE 5. Effect of the descriptor parameters on the face verification, and identification performance (Base learner model type: SVM, $p_{\%top}$= 80%, Combination rule: Mean rule).

Descriptor	Parameters		1:1 Verification TAR				1:N Identification			
			FAR=0.01		FAR=0.1		Rank-1		Rank-5	
	s_p	n_b	$N_{pose}=3$	$N_{pose}=5$	$N_{pose}=3$	$N_{pose}=5$	$N_{pose}=3$	$N_{pose}=5$	$N_{pose}=3$	$N_{pose}=5$
LBP	36	250	0.524	0.514	0.558	0.552	0.774	0.781	0.900	0.908
		300	0.563	0.558	0.568	0.562	0.810	0.816	0.923	0.915
	40	250	0.535	0.532	0.560	0.553	0.784	0.803	0.900	0.903
		300	0.566	0.560	0.568	0.562	0.803	0.810	0.910	0.915
	s_p	K_{hog}								
HOG	36	9	0.535	0.529	0.565	0.560	0.918	0.919	0.986	0.982
		10	0.540	0.535	0.565	0.562	0.923	0.913	0.983	0.983
	40	9	0.549	0.540	0.565	0.560	0.923	0.928	0.990	0.988
		10	0.542	0.539	0.565	0.562	0.923	0.921	0.990	0.986
	σ_{sift}	$\varnothing_{sift}$								
SIFT	1.4	4	0.986	0.991	**1.000**	**1.000**	0.998	**1.000**	**1.000**	**1.000**
		6	0.991	0.995	**1.000**	**1.000**	0.996	0.996	**1.000**	**1.000**
	1.6	4	0.988	0.995	**1.000**	**1.000**	0.998	**1.000**	**1.000**	**1.000**
		6	0.996	**0.998**	**1.000**	**1.000**	0.996	0.996	**1.000**	**1.000**

TABLE 6. Effect of the base learner selection and combination rule on the face verification and identification performance for different base learner model types (N_{pose} = 5, Descriptor: SIFT).

Base learner model type	Combination rule	$p_{\%top}$	1:1 Verification TAR			1:N Identification		
			FAR=0.001	FAR=0.01	FAR=0.1	Rank-1	Rank-5	Rank-10
Naive Bayes	Mean	100	0.630	0.789	0.934	0.900	0.957	0.985
		80	0.650	0.828	0.959	0.885	0.944	0.973
	Trim-mean	100	0.455	0.650	0.854	0.859	0.892	0.913
		80	0.490	0.665	0.857	0.864	0.919	0.933
GMM	Mean	100	0.465	0.614	0.754	0.973	0.996	0.996
		80	0.500	0.643	0.753	0.977	0.996	0.996
	Trim-mean	100	0.673	0.754	0.903	0.990	0.996	0.998
		80	0.697	0.776	0.919	0.978	0.996	0.996
SVM	Mean	100	**0.973**	**0.996**	**1.000**	**1.000**	**1.000**	**1.000**
		80	0.967	0.995	**1.000**	**1.000**	**1.000**	**1.000**
	Trim-mean	100	0.965	0.987	**1.000**	**1.000**	**1.000**	**1.000**
		80	0.965	0.987	**1.000**	**1.000**	**1.000**	**1.000**

the $p_{\%top}$ value do not show a considerable negative impact on the recognition performance. This suggests that the number of base learners per ensemble can be reduced to a certain point without generating a large recognition performance drop.

3) TRAINING AND TESTING TIMES

The main purpose of including a base learner selection block is to reduce the computational time during face recognition while keeping a high recognition performance. Therefore, besides assessing the accuracy of the proposed method, the time employed during the training and testing stages are also regarded as important parameters to determine the performance of the proposed method. The training and testing time values for SVM, GMM, and Naive Bayes are depicted in Figure 6. The training time bar (blue bar) indicates the time required to train a face recognition ensemble system for one subject. The testing time is divided into two bars. The first bar (orange bar) is the face recognition testing time with $p_{\%top} = 80$. It comprises the time employed to obtain the decision values of all the ensemble systems and process them in order to predict the identity of the person on the input face image. The last bar (green bar) has the same interpretation as

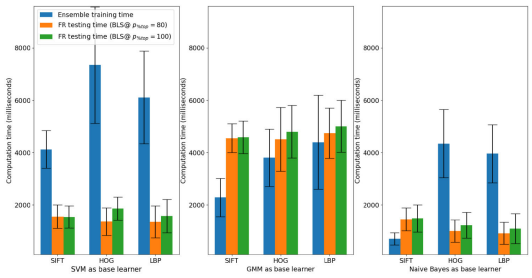


FIGURE 6. Computational times during training and testing according to the base learner model and feature descriptor.

the previous one, but using $p_{\%top} = 100$ instead. The results show that an ensemble system with SVM as base learner takes more time during training, but the testing time remains low. Furthermore, it can be seen that the use of SIFT as the feature descriptor reduces the time during training and testing. Lastly, the testing time with BLS at $p_{\%top} = 80$ is slightly lower than the one with $p_{\%top} = 100$.

4) COMPARISON WITH STATE-OF-THE-ART RESULTS

The CMU-PIE database has been employed on several state-of-the-art works to test the efficiency of their methods on

TABLE 7. Detailed performance comparison of the proposed method with state-of-the-art methods for PIFR on the CMU-PIE database.

Method	Recognition rate(%)									
	-90^o	-67.5^o	-45^o	-22.5^o	0^o	22.5^o	45^o	67.5^o	90^o	Overall
Face frontalization + Facial features† [32]	-	-	95.60	100.00	100.00	100.00	100.00	-	-	99.3
FLM + PFER-GEM [33]	88.70	100.00	100.00	100.00	100.00	100.00	100.00	98.38	91.90	98.24
Statistical classification + Local Gabor Features [34]	72.40	83.10	100.00	100.00	100.00	100.00	100.00	99.70	97.50	94.10
PBPR-MtFL [5]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.53	99.84
Divide-and-rule + LBP-Huffman† [35]	-	-	100.00	100.00	100.00	100.00	100.00	-	-	100.00
Face frontalization + LGBP† [1]	-	-	91.2	98.5	100.00	100.00	98.5	-	-	97.05
Ensemble GMM + SIFT	100.00	95.59	97.06	100.00	100.00	100.00	98.53	100.00	88.24	97.71
Ensemble SVM + SIFT	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

† Only face images with pose angles between $\pm 45^o$ were considered for testing.

PIFR. We gather the experimental results of these works and compare them with our results in Tabl. 7. As can be seen, some works [1], [32], [35] just considered a pose angle range of ±45° during their experimental trials. In this work, we employ pose angles between ±90°, and achieve a recognition rate of 100% for every pose angle. Furthermore, it can be seen that other works considering ±90° pose-view images, experimented a severe performance decline in their results for images beyond the range of ±45°.

V. CONCLUSION

In this paper, we address the pose-invariant face recognition problem from an ensemble learning approach. One ensemble system is trained exclusively for one person. A base learner, constituting an ensemble system, is trained for a specific facial landmark, according to a 68-landmark scheme. Therefore, we demonstrate the potential of using local methods for face recognition, with facial landmarks as the keypoints. On the other hand, we also propose a simple, yet effective face pose classification method (a simplified version of HPE), which aims to increase the face recognition performance by deciding which facial landmarks should be considered for applying local feature description. Besides, a base learner selection (BLS) algorithm works conjointly with the pose classification model to reduce the number of base learners while keeping a high recognition rate. Experimental results are obtained on the CMU-PIE face database, and show a recognition rate (Rank-1 accuracy) of 100% on any pose-view face image within a range of ±90°. Therefore, the proposed method surpasses the performance of state-of-the-arts methods, considering the CMU-PIE as the testing database, in terms of accuracy and pose angle range.

REFERENCES

[1] C. Petpairote, S. Madarasmi, and K. Chamnongthai, "2D pose-invariant face recognition using single frontal-view face database," *Wireless Pers. Commun.*, vol. 118, no. 3, pp. 2015–2031, Jun. 2021.
[2] M. Taskiran, N. Kahraman, and C. E. Erdem, "Face recognition: Past, present and future (a review)," *Digit. Signal Process.*, vol. 106, Nov. 2020, Art. no. 102809.
[3] F. Wu, X.-Y. Jing, X. Dong, R. Hu, D. Yue, and L. Wang, "Intraspectrum discrimination and interspectrum correlation analysis deep network for multispectral face recognition," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1009–1022, Mar. 2020.
[4] C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," 2014, *arXiv: 1401.5311*.
[5] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 980–993, Mar. 2015.
[6] M. Gunther, "The 2013 face recognition evaluation in mobile environment," in *Proc. Int. Conf. Biometrics (ICB)*, 2013, pp. 1–7.
[7] T. Q. Chung, H. C. Huyen, and D. V. Sang, "A novel generative model to synthesize face images for pose-invariant face recognition," in *Proc. Int. Conf. Multimedia Anal. Pattern Recognit. (MAPR)*, Oct. 2020, pp. 1–6.
[8] Z. Zhang, L. Wang, S.-K. Chen, Y. Chen, and Q. Zhu, "Pose-invariant face recognition using facial landmarks and Weber local descriptor," *Knowl.-Based Syst.*, vol. 84, pp. 78–88, Aug. 2015.
[9] Z. An, W. Deng, J. Hu, Y. Zhong, and Y. Zhao, "APA: Adaptive pose alignment for pose-invariant face recognition," *IEEE Access*, vol. 7, pp. 14653–14670, 2019.
[10] D. B. Giap, T. N. Le, J.-W. Wang, and C.-N. Wang, "Adaptive multiple layer Retinex-enabled color face enhancement for deep learning-based recognition," *IEEE Access*, vol. 9, pp. 168216–168235, 2021.
[11] Y. Feng, X. An, and S. Li, "Research on face recognition based on ensemble learning," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 9078–9082.
[12] S. D. Lin and P. L. Otoya, "Large pose detection and facial landmark description for pose-invariant face recognition," in *Proc. IEEE 5th Int. Conf. Knowl. Innov. Invention (ICKII)*, Jul. 2022, pp. 143–148.
[13] S. Umer, B. C. Dhara, and B. Chanda, "Biometric recognition system for challenging faces," in *Proc. 5th Nat. Conf. Comput. Vis., Pattern Recognit., Image Process. Graph. (NCVPRIPG)*, Dec. 2015, pp. 1–4.
[14] G. Azzopardi, A. Greco, A. Saggese, and M. Vento, "Fast gender recognition in videos using a novel descriptor based on the gradient magnitudes of facial landmarks," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.
[15] M. I. N. P. Munasinghe, "Facial expression recognition using facial landmarks and random forest classifier," in *Proc. IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2018, pp. 423–427.
[16] X. Yuan and M. Abouelenien, "A boosting method for learning from uneven data for improved face recognition," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, vol. 2, Dec. 2012, pp. 119–122.
[17] J. Y. Choi and B. Lee, "Ensemble of deep convolutional neural networks with Gabor face representations for face recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 3270–3281, 2020.
[18] K. Khan, R. U. Khan, R. Leonardi, P. Migliorati, and S. Benini, "Head pose estimation: A survey of the last ten years," *Signal Process., Image Commun.*, vol. 99, Nov. 2021, Art. no. 116479.
[19] A. F. Abate, P. Barra, C. Bisogni, M. Nappi, and S. Ricciardi, "Near real-time three axis head pose estimation without training," *IEEE Access*, vol. 7, pp. 64256–64265, 2019.

[20] P. Barra, S. Barra, C. Bisogni, M. De Marsico, and M. Nappi, "Web-shaped model for head pose estimation: An approach for best exemplar selection," *IEEE Trans. Image Process.*, vol. 29, pp. 5457–5468, 2020.
[21] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, "Head pose estimation: Classification or regression?" in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
[22] S. Li, L. Sun, X. Ning, Y. Shi, and X. Dong, "Head pose classification based on line portrait," in *Proc. Int. Conf. High Perform. Big Data Intell. Syst.*, May 2019, pp. 186–189.
[23] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial Landmarks)," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1021–1030.
[24] Intel. (Nov. 2022). *Open Source Computer Vision (OpenCV) Documentation*. [Online]. Available: https://docs.opencv.org/4.5.5/
[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.
[26] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
[27] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, Sep. 2006.
[28] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, 2nd ed. Hoboken, NJ, USA: Wiley, 2014, ch. 3.
[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.
[30] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.
[31] Z. Cheng, X. Zhu, and S. Gong, "Surveillance face recognition challenge," 2018, *arXiv:1804.09691*.
[32] E. A. Mostafa and A. A. Farag, "Dynamic weighting of facial features for automatic pose-invariant face recognition," in *Proc. 9th Conf. Comput. Robot Vis.*, May 2012, pp. 411–416.
[33] A. Moeini and H. Moeini, "Real-world and rapid face recognition toward pose and expression variations via feature library matrix," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 5, pp. 969–984, May 2015.
[34] L. A. Cament, F. J. Galdames, K. W. Bowyer, and C. A. Perez, "Face recognition under pose variation with local Gabor features enhanced by active shape and statistical models," *Pattern Recognit.*, vol. 48, no. 11, pp. 3371–3384, 2015.
[35] L.-F. Zhou, Y.-W. Du, W.-S. Li, J.-X. Mi, and X. Luan, "Pose-robust face recognition with Huffman-LBP enhanced by divide-and-rule strategy," *Pattern Recognit.*, vol. 78, pp. 43–55, Jun. 2018.

SHINFENG D. LIN (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Mississippi State University, in 1991. He is currently a Professor with the Department of Computer Science and Information Engineering and the Vice President of National Dong Hwa University, Taiwan. He was the Director of the Bureau of Education, Hualien, Taiwan, from January 2002 to September 2003. He has published over 150 journals and conference papers. His research interests include signal/image processing, machine learning, pattern recognition, and information security. He is a fellow of IET. He won the Gold Medal Award at the 2005 International Trade Fair "Ideas-Inventions-New Products" (IENA), Nuremberg, Germany.

PAULO E. LINARES OTOYA was born in Trujillo, La Libertad, Peru, in 1993. He received the B.S. degree in electronic engineering from Universidad Privada Antenor Orrego (UPAO), Trujillo, in 2019. He is currently pursuing the M.Sc. degree with the School of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. Between 2019 and 2021, he was a Research Assistant with the UPAO Multidisciplinary Research Laboratory (LABINM-UPAO), contributing actively to research projects that applied computer vision techniques to precision agriculture. His research interests include computer vision, artificial intelligence, face recognition, and head pose estimation.