IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 7, 2019, accepted May 24, 2019, date of publication May 29, 2019, date of current version June 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919806

Realization of a Hybrid Locally Connected Extreme Learning Machine With DeepID for Face Verification

SHEN YUONG WONG[1], KEEM SIAH YAP[2], QINGWEI ZHAI[1], AND XIAOCHAO LI[1,3]

[1]Department of Electrical and Electronics Engineering, Xiamen University Malaysia, Sepang 43900, Malaysia
[2]Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Kajang 43000, Malaysia
[3]Department of Electronic Engineering, Xiamen University, Xiamen 361005, China

Corresponding author: Xiaochao Li (leexcjeffrey@xmu.edu.cn)

This work was supported by the Xiamen University Malaysia under Grant IECE/0001.

ABSTRACT Most existing state-of-the-art deep learning algorithms discover sophisticated representations in huge datasets using convolutional neural networks (CNNs) that mainly adopt backpropagation (BP) algorithm as the backbone for training the face recognition problems. However, since decades ago, BP has been debated for causing trivial issues such as iterative gradient-descent operation, slow convergence rate, local minima, intensive human intervention, exhaustive computation, time-consuming, and so on. On the other hand, a competitive machine learning algorithm called extreme learning machine (ELM) emerged with extreme fast implementation and simple in theory has overcome the challenges faced by BP. The ELM advocates the convergence of machine learning and biological learning for pervasive learning and intelligence and has been extensively researched in widespread applications. Nonetheless, till date, none of the work of ELM has proved its competency in tackling face verification problem. Hence, in this paper, we are going to probe for the first time the feasibility of ELM-based network in handling the face verification task. We devise and propose a novel and distinguished hybrid local receptive field-based extreme learning machine with DeepID (hereinafter denoted as H-ELM-LRF-DeepID), to discriminate face pairs. The experimental results on the YouTube face database, labeled faces in the wild (LFW), and CelebFaces datasets have shed light upon the feasibility and usefulness of the H-ELM-LRF-DeepID in the face verification task.

INDEX TERMS DeepID, extreme learning machine, face verification, tuning free feature mapping.

I. INTRODUCTION

Recent years have seen the emerging advances in machine learning technology and big data analysis [1]–[5], which have raised its capabilities across a widespread of modern applications, and power the next wave of innovation in the face recognition task. Face recognition can be categorized into two tasks: face identification and face verification. Face identification is to recognize and classify the identity of a probed face given a set of face images from the database with labeled known identities. Face verification is to determine whether two human face images belong to the same person, as shown in Fig. 1 and Fig. 2. In this paper, we focus on the face verification. Face verification is useful in many applications, for example, verifying if a face matches with a valid identity, unlock a mobile device or an automated door, providing security in various means [5].

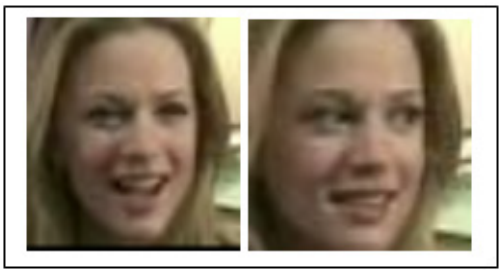

FIGURE 1. Pair of face images of same subject.

Earlier works on face verification often engaged low-level feature extractors and combined them to obtain feature representations for face images. The models that produce

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

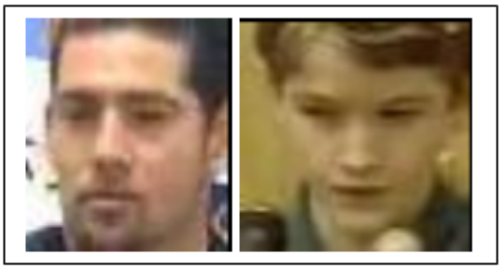

FIGURE 2. Pair of face images of different subject.

impressively good performance often employ tens of thousands of image descriptors [6]. Nowadays, it is more common to learn the features that often make use of the neural networks with deep architectures instead of engineering them [7]. Convolutional Neural Networks [8]–[11], with many levels of abstraction allow computational models to comprise several processing layers to learn representations of an image have gained much popularity. Ameur *et al.* [12] presented a deep learning network that used feature extraction based on data processing components, with Enhanced Fisher model for analysis and Heaviside step function to transform the image into binary format. Di *et al.* [13] published a metric learning method using the supervised knowledge of Joint Bayesian in the CNN architecture. It involves the face representation learning and recognition in training and fine-tuning the CNN model. Two faces are jointly modeled using a suitable prior on the deep face representation. The innovation of the article lies in the joint modeling of two human faces.

Like the aforementioned deep learning neural networks, most of the existing state-of-the-art face verification algorithms have one thing in common, they adopt backpropagation (BP) algorithm as the backbone for training that involves iterative gradient-descent steps to fine tune the weights parameters that are used to compute the representation in each layer from the representation in the previous layer [14]. In a typical deep learning system, there could be hundreds of millions of these adjustable weights, and hundreds of millions of labeled training samples to train the network. As such, all of the BP-based methods suffer from the common dilemmas, i.e., iterative and laborious gradient-descent operation, slow convergence rate, local minima, intensive human intervention, exhaustive computation, time-consuming, and so on [14], [15].

In addition to the challenge faced by BP algorithm, handling massive raw images in the face verification task is usually considered as big dataset problems. The direct approach to overcome big dataset problems often incurs intangible costs, such as the investment of hardware, parallel computing framework, and GPU computing that accelerates the processing speed, etc. The success of deep learning series is often attributed to large scale training dataset and the availability of a powerful computer or GPU computing. However, not all researchers can afford expensive hardware for the computational task that requires high processing power, speed and memory.

Therefore, it also serves as our motivation to develop a unique ELM-based learning face verification framework that can perform comparably well, if not better, with the existing state-of-the-art deep learning architectures, but with the advantage of eliminating the tedious BP training procedures that are time consuming and of high computational complexity that require the availability of the powerful computer with fast processor and ample memory to run huge dataset. Recently, a powerful and competitive machine learning algorithm called the Extreme Learning Machine (ELM) proposed by Huang *et al.* [16] has overcome the challenges faced by BP with its straightforward learning framework. ELM advocates the convergence of machine learning and biological learning for pervasive learning and intelligence, has been extensively researched in widespread applications. ELM and its variants have shown themselves highly efficient and prominent in providing solutions for various kind of problems in different practical applications, i.e., regression, two-class, multiclass classifications, given its higher scalability and less computational complexity in operation [17]–[20]. ELM also extended its work to handle the generalized multi hidden layer feed-forward networks in which a neuron could be a subnetwork consisting of other hidden neurons when handling the image classification tasks, i.e., Local Receptive Field based Extreme Learning Machine (ELM-LRF) [21]–[26].

However, none of the work of ELM has proved its competency in tackling face verification problem. Hence, we are going to probe for the first time the feasibility of ELM-based network in handling the face verification task. In this paper, we devise and propose a novel and distinguished Hybrid Local Receptive Field based Extreme Learning Machine with DeepID (hereinafter denoted as H-ELM-LRF-DeepID), to discriminate face pairs.

The main contributions of H-ELM-LRF-DeepID to the ELM variants are as follows:

(i) The research work of using ELM-based framework to determine whether a pair of face images belong to the same person or not is unprecedented.
(ii) Different from most of the state-of-the-art algorithms, H-ELM-LRF-DeepID does not implement an end to end deep CNN based framework for face verification tasks.
(iii) Elimination of BP-based algorithm for training to adjust the connection weights which requires high computational time and exhaustive training epochs to minimize loss function.
(iv) Hybridization of ELM with the renowned DeepID for meaningful multi-scale feature extractions containing both mid-level and global high-level features because meaningful features representations are essential for face verification.
(v) H-ELM-LRF-DeepID is a simplified learning algorithm which strategizes on the tuning free hidden neurons even when the output shapes and function modeling of the neurons are unknown, for handling complex face verification tasks.

By eliminating the extensive computational cost of gradient descent operation by BP, there is no qualm to admit that the proposed H-ELM-LRF-DeepID is at ease of implementation, has provided an ultimate way of handling face verification in a notable simple and lighter weight manner.

The content of the paper is organized as follows: Section II introduces preliminaries of Extreme Learning Machine, Local Receptive Fields based Extreme Learning Machine, and DeepID. And then, Section III presents the methodology of the proposed H-ELM-LRF-DeepID in a very detailed manner. Furthermore, Section IV evaluates the performance of H-ELM-LRF-DeepID against the existing state-of-the-art algorithms and discusses the results. Finally, Section V provides some concluding remarks.

II. PRELIMINARIES

A. EXTREME LEARNING MACHINES (ELM)

Extreme Learning Machine (ELM) is a fast learning algorithm proposed by Huang *et al.* [16] for training the single-hidden-layer feedforward networks (SLFNs). ELM for SLFN is independent of training data samples and it supports wide types of neural networks. Unlike the earlier work by Schmidt *et al.* [22] that only considered using random hidden neurons with only Sigmoid activation function, and a bias weight is required in the output neurons to handle the system error. When proposed, the universal approximation capability with random hidden neurons (Sigmoid) was not proved yet:

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i g_{sig}(\mathbf{a}_i \cdot \mathbf{x} + b_i) + b \tag{1}$$

where

$$g_{sig}(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x})} \tag{2}$$

On the contrary, Huang [28], [29] has clarified the differences of ELM compared to the other earlier works as aforementioned. He proved that ELM is different from the previous fully connected networks [22], as the strength of ELM lies in its randomness of network parameters in threefold :

1. All layers in ELM are fully connected, and the parameters of hidden neurons are randomly generated.
2. Not all input neurons will be connected to the hidden layers, which means the connections are also randomly generated. It is possible that only some input neurons are connected to the corresponding hidden neurons.
3. A neuron can be a subnetwork of several neurons that formed by the pooling functions, and local receptive field. And it will result in learning local features. For this reason, a single ELM may have some local parts that contain multi-hidden layers.

ELM is renowned for its universal approximation capability for a wide type of nonlinear piecewise continuous functions $G(\mathbf{a}, b, \mathbf{x})$, and bias is not needed at all in the output layer. The basic ELM for generalized SLFN with the output respectively to $\mathbf{x}_j$ will be:

$$f_L(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) \text{ for } j = 1, \ldots, N, \tag{3}$$

where $\mathbf{a}_i$ is the input weights which link the input layer and first hidden layer, b_i is bias or called learning parameter of the hidden neurons, β_i is the output weights which link the hidden layer and output layer. $G(\mathbf{a}_i, b_i, \mathbf{x}_j)$ is the i^{th} hidden neuron output correspondingly to the input vector $\mathbf{x}_i$. Eqn. (3) can also be written as:

$$\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{x})\boldsymbol{\beta} = \mathbf{T} \tag{4}$$

where

$$\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{x}) = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix} \tag{5}$$

And

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \ldots & \beta_1 \end{bmatrix}^T, \quad \mathbf{T} = \begin{bmatrix} t_1 & \ldots & t_1 \end{bmatrix}^T \tag{6}$$

where $\mathbf{H}$ is the ELM hidden layer output randomized matrix, and $\mathbf{T}$ is the training data target matrix. $\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{x})$ is named by Huang *et al.* [12], [25]. It is the hidden layer output matrix of the neural network. The i^{th} hidden neuron output is the i^{th} column of $\mathbf{H}$ and respect to inputs $x_1, x_2, \ldots x_N$. $\mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{x})$ also can be denoted as $\mathbf{H}$ for convenient. By pseudo inverse matrix of Eqn. (4), the output weights of ELM can be found as

$$\boldsymbol{\beta} = \left(\mathbf{H}^{\mathbf{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathbf{T}}\mathbf{T} \tag{7}$$

If an unlabeled sample $\boldsymbol{z}$ required classification, the ELM output function will be:

$$f_L(\mathbf{z}) = \mathbf{H}(\mathbf{a}, \mathbf{b}, \mathbf{x})\boldsymbol{\beta} \tag{8}$$

In some case like a slack variable is introduced to ELM, the output weight that given by Eqn. (7) will be updated with a new user selectable constrained parameter, C:

$$\boldsymbol{\beta} = \left(\frac{1}{C} + \mathbf{H}^{\mathbf{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathbf{T}}\mathbf{T} \tag{9}$$

B. LOCAL RECEPTIVE FIELD BASED EXTREME LEARNING MACHINE (ELM-LRF)

Inspired by ELM theories, Huang *et al.* [26] introduced the local receptive fields of ELM (ELM-LRF), that considered the local connection and local structures, with the utilization of local receptive fields for obtaining a locally connected ELM. The hidden layer of ELM-LRF refers to combinatorial neurons of a convolution layer and a pooling layer. The input connection weights between the input and convolutional layer

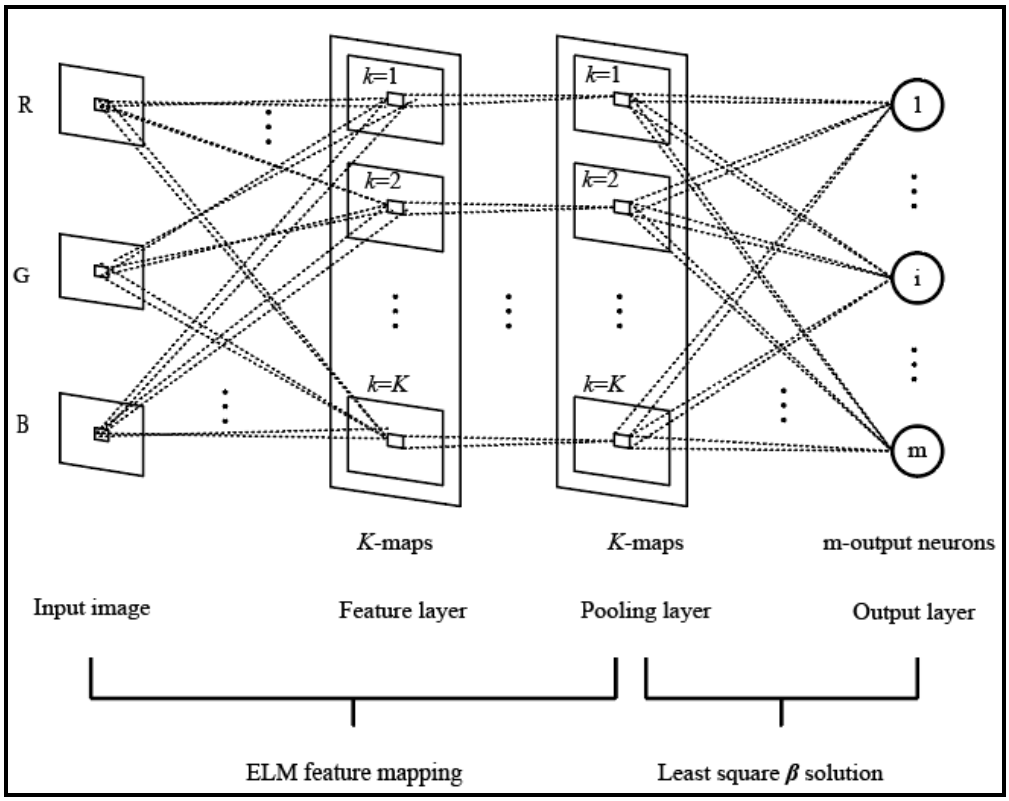

FIGURE 3. The architecture of ELM-LRF.

are randomly generated based on the continuous probability distribution. To obtain a more complete set of feature, the weights need to be orthogonalized.

In order to provide an efficient and deterministic solution, the output weights are analytically calculated using Moore-Penrose Pseudoinverse solution. To formulate the combinatorial nodes of ELM-LRF, a specific network that uses the step function is constructed to sample the local connections and the square/square-root pooling network structure.

Furthermore, the pooling layer is fully connected to the output layer. ELM theories show that different local receptive fields in ELM-LRF can be generated due to the adoption of different probability distribution used in generating random hidden neuron. Fig. 3 shows the architecture of ELM-LRF.

The implementation of ELM-LRF is made up of two parts:

Part 1: Tuning-free ELM feature mapping

i. *Random convolutional weights:* The convolutional weights between feature layer and input are randomly generated based on standard Gaussian distribution. The local reception field is $r \times r$ and input image is $d \times d$. Therefore, the feature map is $(d - r + 1) \times (d - r + 1)$. Each input pixel is a neuron. Thus, the neurons (i, j) in the k-th feature map, $c_{i,j,k}$ is :

$$c_{i,j,k}(\mathbf{X}) = \sum_{m=1}^{r}\sum_{n=1}^{r} x_{i+m-1,j+n-1} \cdot a_{m,n,k} \quad i,j = 1,\ldots,(d-r+1) \tag{10}$$

ii. *Square/square-root pooling*: The output map of the pooling layer, h_{p,q,k_s} is:

$$h_{p,q,k} = \sqrt{\sum_{i=p-e}^{p+e}\sum_{j=q-e}^{q+e} c_{i,j,k}^2} \tag{11}$$

$p, q = 1, \ldots, (d - r + 1) c_{i,j,k} = 0$ if (i,j) out of bound

Part 2: ELM learning based on regularized least-squares solution

Throughout the ELM learning process, only the output weight $\boldsymbol{\beta}$ needs to be analytically calculated. Consider N training samples and concatenate all pooling neurons into a row vector, the matrix $\mathbf{H} \in \mathbf{R}^{N \times (d-r+1)^2}$ of ELM-LRF is then yield:

$$\beta = \begin{cases} \mathbf{H}^T \left(\frac{1}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{T} & \text{if} \quad N \le K \cdot (d-r+1)^2 \\ \left(\frac{1}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T\mathbf{T} & \text{if} \quad N > K \cdot (d-r+1)^2 \end{cases} \tag{12}$$

C. DeepID

Deep hidden Identity features (DeepID) is introduced by Sun Y *et al.* [27] which is capable of learning high-level features from face images in an efficient way with deep Convolutional Neural Network (CNN) for face verification.

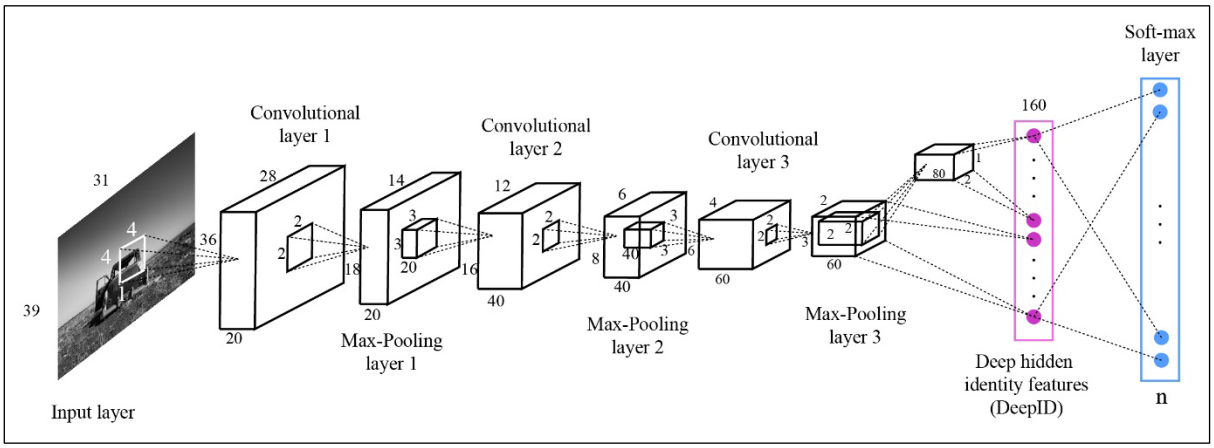

FIGURE 4. The architecture of DeepID [27].

Fig. 4 shows the structure of DeepID for learning features. It comprised an input layer, four convolutional layers each with a max-pooling layer except the last convolutional layer, and then a fully connected DeepID layer and an output layer constituted by softmax function that represents the different class labels. ReLU nonlinearity activation function is used for hidden neurons instead of a sigmoid function for better fitting competency [28].

Different from the general CNN, the last hidden layer of DeepID is fully connected to the third and fourth convolutional layer to extract both high-level and low-level features [28].

The strength of DeepID is that it extracts multi-scale features containing both mid-level and global high-level features, which can have good generalization on face verification task and therefore reduces the possible information loss. Moreover, the generalized high-level features will not overfit to small subsets of faces. DeepID represents a huge amount of class labels with a small number of hidden neurons. It is due to the fact that the use of more identities or class labels during training enables extraction of a more compact and discriminative DeepID feature, which in turn helps to increase the dimensionality of prediction and additionally, improves the performance of face verification.

Later DeepID series [11], [33], [34] further refine their work using non-parametric Joint Bayesian method, as well as joint identification-verification supervisory signal, auxiliary supervisory structure, deep architecture, and other data samples. For example, DeepID2 and DeepID3 are developed with the aim to decrease the intra-personal variations and increase the inter-personal differences. The architecture of the DeepID2 is similar to the DeepID [27], with local weight-sharing in the third convolutional layers and fourth convolutional layers. Identification and verification signals are weighted by a hyperparameter, λ, to learn the DeepID2 features. DeepID2 is a deep learning methodology that has deep architecture and good learning capability.

III. OUR PROPOSED ALGORITHM

From the current literature, none of the work of ELM has demonstrated its competency in tackling face verification problem. Therefore, we are going to probe for the first time the feasibility of ELM-based network to discriminate face pairs. In this respect, we introduce some advancement and assimilation of significant properties of the profound DeepID [27] on the Local Receptive Field based ELM (ELM-LRF) to form H-ELM-LRF-DeepID. The ELM-LRF is selected as the basis of the new proposed model because it turns out to be the closest match to the theory of conventional ELM since the two algorithms are developed by the same founder researchers in [16], [26]. The ELM is originally proposed by Huang *et al.* [16] as a fast learning algorithm that provides good approximation performance with random hidden neurons and analytically determines the output weights of single layer feedforward neural network. Later, the ELM-LRF [26] is devised to have 1 convolution layer and 1 pooling layer, which can be assimilated to the structure of DeepID. Hence, it serves as our main motivation to develop the proposed H-ELM-LRF-DeepID to enhance the operation of feature mapping of the ELM-LRF for effective learning of the meaningful features representations of the face images to tackle the unprecedented face verification task.

As shown in Fig. 5, H-ELM-LRF-DeepID accepts raw images that come with RGB component for processing. H-ELM-LRF-DeepID leverages on the superiority of DeepID in terms of feature extraction, is endowed with the ability to extract compact and discriminative multi-scale feature containing both mid-level and global high-level features that ensure good generalization on face verification task.

Following the architecture of DeepID, the map size used in every feature extraction layer, i.e., feature layer and pooling layer, of DeepID is adopted. Besides that, downsampling concept in the pooling layer of DeepID is utilized to reduce the dimension of the pooled output vector. The feature vector of the last n-th layer of convolutional neurons and the

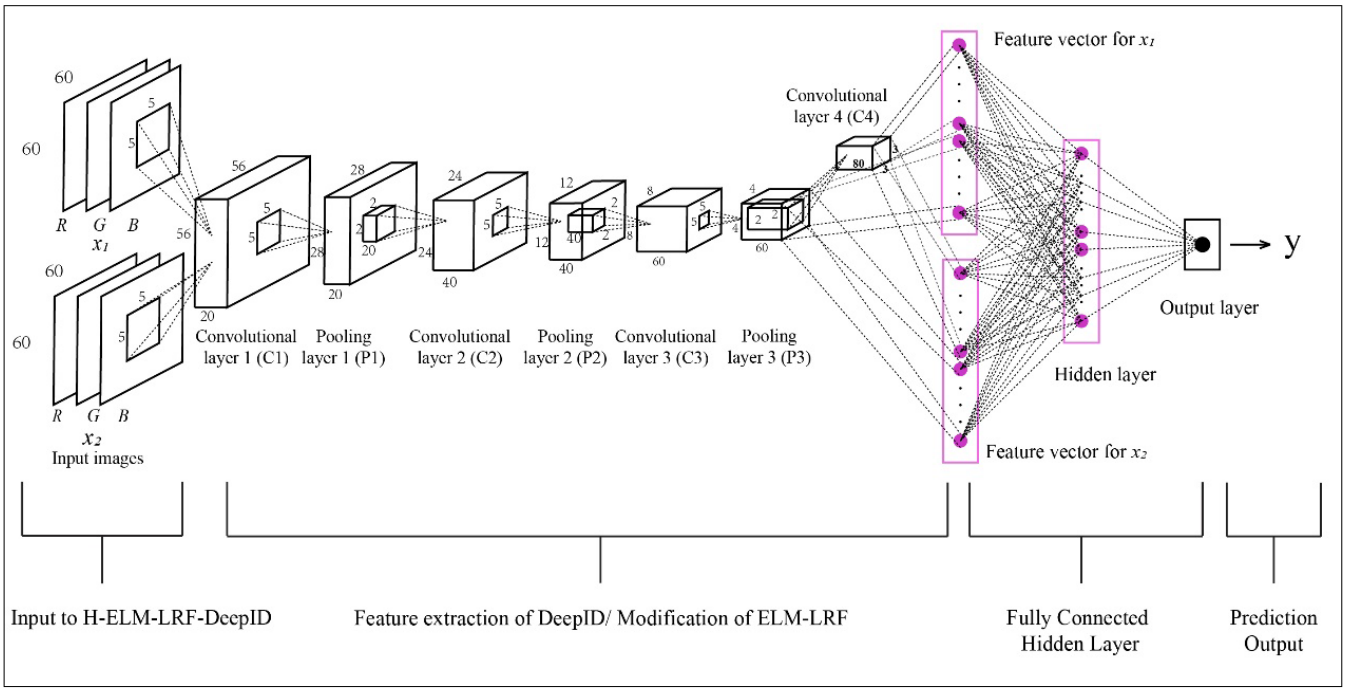

FIGURE 5. The architecture of the proposed H-ELM-LRF-DeepID for face verification.

(*n-1*)th pooling neurons is concatenated into a row vector for further use by ELM hidden layer and ELM learning process based on the regularized least squares solution [16].

It is worth highlighting that H-ELM-LRF-DeepID, is different from most of the state-of-the-art algorithms, it does not implement an end to end deep CNN based framework for face verification tasks. H-ELM-LRF-DeepID has eliminated the extensive computational cost of gradient descent operation of DeepID learning to fine tune the connection weights in many epochs by BP, instead, it uses ELM as the foundation for face verification framework which is at ease of implementation. Moreover, H-ELM-LRF-DeepID strategizes on the tuning free hidden neurons even when the output shapes and function modeling of the neurons are unknown, for handling complex face verification tasks.

It is also interesting to note the few differences between the new proposed H-ELM-LRF-DeepID and the conventional ELM-LRF, as follows:

(i) H-ELM-LRF-DeepID introduces downsampling concept of DeepID in the pooling layer of ELM-LRF, besides the Square and Squareroot operation as proposed by [16], to reduce the dimension of the pooled output vector.

(ii) H-ELM-LRF-DeepID has a fully connected hidden layer between the concatenation layer of the feature vector and output layer, in which ELM-LRF does not have.

(iii) H-ELM-LRF-DeepID flexibly extends the single-layer of convolution and pooling of ELM-LRF to a multi-layer feature mapping network to produce highly compact and predictive features.

(iv) H-ELM-LRF-DeepID concatenates two layers of combinatorial neurons (i.e., last *n*-th layer of convolutional neurons and the (*n-1*)th pooling neurons) of a feature vector into a row vector. On the contrary, ELM-LRF concatenates only the neurons of the last pooling layer into a row vector.

The methodology of our proposed H-ELM-LRF-DeepID is as follows:

A. GENERATE IMAGE TRAINING PAIRS FOR FACE VERIFICATION

Step 1: Obtain a set of training images consist of faces (one face per image) and their respective class label (i.e., real identity of the face in the image). Decide N verification pairs for training, where N is an even integer. In the training set, half of them are of same class (i.e., faces of both images belong to same identity) and the other half are of different class (i.e., faces of both images belong to two different identifies).

Step 2: Randomly select (N/2) pairs of images from training images, where in each pair, both images of human faces belong to the same identity.

Step 3: Randomly select (N/2) pairs of images from training images, where in each pair, both images of human faces are from two different identifies.

Step 4: Combine results of Step 1 and Step 2, hence the outputs of this procedures are $\mathbf{X}$ and $\mathbf{T}$ matrix are

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} \\ \vdots & \vdots \\ \mathbf{x}_{N,1} & \mathbf{x}_{N,2} \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} \tag{13}$$

where $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$ (for $i = 1, 2, \ldots, N$)) are two images of *i-th* training pair for face verification and T_i is the respective target output such as faces of the pair images belong to the same identity or two different identities (For instance, $T_i = 1$ for $i = 1, 2, \ldots, \frac{N}{2}$ for a pair of face image that share the same identity, while $T_i = -1$ for $i = \frac{N}{2} + 1, \frac{N}{2} + 2, \ldots, N$ for a pair of face image that is distinctive).

The generated $\mathbf{X}$ and $\mathbf{T}$ matrix are to be used in the training phase of H-ELM-LRF-DeepID for face verification.

B. TRAINING PHASE FOR FACE VERIFICATION

Step 1: Load the matrix $\mathbf{X}$ and $\mathbf{T}$ of N pairs that has been prepared from the aforementioned steps of generation of training image pairs for face verification.

Step 2: Referring to Fig. 3, layer 1 is the input layer of H-ELM-LRF-DeepID. Upon arrival of the first pair of image at this layer, the input image $\mathbf{x}_1$ will be presented first to undergo feature extraction, followed by input image $\mathbf{x}_2$, in sequential mode. The input image is expected to be in the square matrix, and image is resized to $q \times q$ (where $q = 60$) whenever necessarily.

Step 3: Layer 2 is Convolution Layer 1 (C1), with kernel-size = 5 (i.e., convolution mask 5×5). Following the map size setting in DeepID [27], the number of output feature maps for this layer is K_1, i.e., $K_1 = 20$. In this respect, 20 distinct input weights (also known as convolutional mask) are randomly generated with orthogonalization in place, as implemented below:

(i) Randomly initialize input weights matrix, $\overline{\mathbf{W}}^{init}$ based on continuous probability distribution. The input image size is $q \times q$ (where $q = 60$), and receptive field $r \times r$ (i.e., $r = 5$), the size of the resulted output feature map should be $(q - r + 1) \times (q - r + 1)$.

(ii) Orthogonalize the initial weight matrix $\overline{\mathbf{W}}^{init}$ using Singular Value Decomposition (SVD) method.

$$\begin{aligned} \overline{\mathbf{W}}^{init} &\in \Re^{r^2 \times K_1} \\ \overline{\mathbf{W}}_k^{init} &\in \Re^{r^2}, \quad k = 1, \ldots, K_1 \end{aligned} \tag{14}$$

In the case of $r^2 < K_1$, orthogonalization cannot be performed on $\overline{\mathbf{W}}^{init}$. Hence, the approximation method is used, i.e., $\overline{\mathbf{W}}^{init}$ is transposed, orthogonalized, and then transpose it back.

(iii) The input weights to the k-th feature map is $\mathbf{w}_k \in \Re^{r \times r}$ which corresponds to $\overline{\mathbf{W}}_k^{init} \in \Re^{r^2}$ column-wisely. Notice that the input image with dimension 60 × 60 is interfaced to the C1 layer, while the resulted output of C1 layer is having dimension $K_1 \times 56 \times 56$. The size of the image is reduced from 60 to 56 due to convolution operation. The resulted convolution at coordinate (i, j) of image in the k-th feature map, $c_{i,j,k}$ is calculated as:

$$c_{i,j,k}(\mathbf{x}) = \sum_{m=1}^{r} \sum_{n=1}^{r} \left(x_{i+m-1, j+n-1} \times w_{m,n,k} \right) \quad \text{for } i, j = 1, \ldots, (q - r + 1) \tag{15}$$

Step 4: Layer 3 is Pooling Layer 1 (P1), with pooling scale, $s = 2$. It involves two operations in this layer, namely pooling and downsampling. This concept is different from the original ELM-LRF [16], because ELM-LRF does not execute downsampling operation. Implementation of the pooling and downsampling process is as below:

(i) Modified Square and Square Root (MSSR) pooling operation is applied on the input images of size 56 × 56. The number of output map of pooling layer must follow the size of C1, due to the local "1-to-1" connection from C1 layer to P1 layer. For instance, 1st output of C1 will be served as the input to the P1 layer. The size of output in this step is $K_1 \times 55 \times 55$, and subsequently it will be going through the downsampling process, as shown in Step 4(ii).

$$p_{u,v,k} = \sqrt{\sum_{j=v}^{v+1} \sum_{i=u}^{u+1} c_{i,j,k}^2} \quad \text{for } u, v = 1, \ldots, 55, k = 1, \ldots, K_1 \tag{16}$$

(ii) Apply downsampling to the output of Step 4(i), by selecting column and row of 1, 3, 5…, 55 (ignore column 2, 4, …, 54) to form the output of pooling layer P1. Thus, the size of P1 output map is now reduced by $K_1 \times 28 \times 28$.

Step 5: Repeat Step-3 and Step-4 for another two pairs of Convolution and Pooling layer, until layer 7, that allows the feature extraction and mapping to complete in the unified H-ELM-LRF-DeepID. Note that the number of output feature maps for next layer, K_{i+1} is set as $K_i + K_1$. As for size of image qxq is updated by the formula $q - r + 1$. Feature mapping of ELM is highly advocated in [26] that explains the theoretical relationship between the local receptive fields and random hidden neurons. ELM has proved that it can flexibly apply different types of local receptive fields as long they are randomly generated based on any continuous probability distribution.

Step 6: Referring to Fig. 5, the output vector of layer-7 (i.e., P3) will be sent to layer 8. Layer 8 is the last feature extraction layer, comprising only the Convolution layer (C4), with kernel size = 3. Here in this layer, $K_4 = 80$. It means that all 80 different input weights are fully connected to the output map of layer 7 (i.e., P3). The input to C4 layer is of size $K_4 \times 4 \times 4$, with convolution operation, image size is reduced to $K_4 \times 2 \times 2$.

Step 7: The output vectors of both the P3 and C4 layer (layer-7 and 8), are concatenated into a row vector. The resulted concatenation is referred to as feature vector $\mathbf{V}$ of

input image, and then sent to the hidden layer of H-ELM-LRF-DeepID.

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_{1,1} & \mathbf{v}_{1,2} \\ \mathbf{v}_{2,1} & \mathbf{v}_{2,2} \\ \vdots & \vdots \\ \mathbf{v}_{N,1} & \mathbf{v}_{N,2} \end{bmatrix} \tag{17}$$

Note: $\mathbf{v}_{1,1}$ is denoted as feature vector of input image $\mathbf{x}_{1,1}$ of the 1^{st} training pair, and $\mathbf{v}_{1,2}$ is denoted as feature vector of input image $\mathbf{x}_{1,2}$ of the 1^{st} training pair.

Step 8: Some initialization of hidden layer parameters, as follows:

(i) Select the number of hidden neurons (L) of fully connected hidden layer. According to Huang *et al.* [25], good generalization performance can be obtained when the size of hidden neuron is large enough, i.e., 2000.

(ii) Randomly generate weights and bias of Sigmoid hidden neurons, i.e., $\{(\mathbf{a}_i, b_i)\}_{i=1}^{L}$.

(iii) Based on the tuning strategy as suggested by Huang *et al.* [16], the user-specified parameter (C) is chosen from the range of $C \in \{2^{-24}, 2^{-23}, \ldots, 2^{24}, 2^{25}\}$.

Step 9: Compute Hidden layer output matrix $\mathbf{H}$ of H-ELM-LRF-DeepID, same procedure as what native ELM does to yield $\mathbf{H}$.

$$\mathbf{H} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{v}_1) & \ldots & G(\mathbf{a}_L, b_L, \mathbf{v}_1) \\ \vdots & & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{v}_N) & \ldots & G(\mathbf{a}_L, b_L, \mathbf{v}_N) \end{bmatrix}_{NxL} \tag{18}$$

Note $G(\mathbf{a}_i, b_i, \mathbf{v}_j)$ is the output of the i-th Sigmoid hidden neuron respectively to the j-th feature vector $\mathbf{v}_j$ as equation below:

$$G(\mathbf{a}_i, b_i, \mathbf{v}_j) = \frac{1}{1 + \exp\{-(\mathbf{a}_i \cdot \mathbf{v}_j + b_i)\}} \tag{19}$$

where $\mathbf{a}_i$ and b_i are the input weights (linking the input layer to the first hidden layer) and bias (learning parameters) of the hidden neurons.

Step 10: Note the original learning equation of ELM is developed based on the following equations.

$$\mathbf{T} = \mathbf{H}\boldsymbol{\beta} \tag{20}$$

The output weights, $\boldsymbol{\beta}$ are computed using Moore-Penrose generalized inverse.

$$\boldsymbol{\beta} = (\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{T} \tag{21}$$

Note in order to improve the stability of learning, $\frac{\mathbf{I}}{C}$ is introduced in the Eqn (19) following the approach in [25]. $\mathbf{I}$ is the identity matrix of same size with $\mathbf{H}^T\mathbf{H}$.

As described in [25], there are many different methods can be used to calculate the Moore-Penrose generalized inverse of a matrix, such as orthogonal projection method, orthogonalization method, iterative method, and singular value decomposition (SVD). In this paper, we implemented Moore-Penrose generalized inverse of a matrix in Eqn. (21) using orthogonal projection method.

Step 11: Save $\mathbf{a}$, $\mathbf{b}$, and $\boldsymbol{\beta}$ for further use in testing phase.

C. TESTING PHASE FOR FACE VERIFICATION

Step 1:

(i) Obtain a pair of new and unseen testing images $\mathbf{Z} = [\mathbf{z}_1 \ \mathbf{z}_2]$ for testing face verification. Here $\mathbf{z}_1$ and $\mathbf{z}_2$ is presented to the feature extraction layers one after another, in sequential mode.

(ii) Meanwhile, load all matrix that has been saved in Step 12 of training phase, i.e., $\mathbf{a}$, $\mathbf{b}$, and $\boldsymbol{\beta}$.

Step 2: Use Step-2 to Step-7 from the training phase to complete the feature extraction procedures for $\mathbf{Z}$ to obtain feature vectors of $\mathbf{z}_1$ and $\mathbf{z}_2$, which are $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2\end{bmatrix}$.

Step 3: Compute hidden layer output matrix $\mathbf{h}$, using the feature vector $\mathbf{V}$ and the saved weights $\mathbf{a}$, $\mathbf{b}$ from the training phase.

$$\mathbf{h} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{V}) & \ldots & G(\mathbf{a}_L, b_L, \mathbf{V}) \end{bmatrix}_{1xL} \tag{22}$$

Note is the output of the i-th Sigmoid hidden neuron to the feature vector $\mathbf{V}$.

Step 4: Calculate the prediction output.

$$y = \mathbf{h}\boldsymbol{\beta} \tag{23}$$

Step 5: If the prediction output y is larger or equal to zero, it is categorized as class 1 (in other words, verification output indicates both images of this pair is sharing the same identity), else it is categorized as class -1 (both images are of different identities).

$$Verification\ \ Output = \begin{cases} +1, & if\ y \geq 0 \\ else & \\ -1, & if\ y < 0 \end{cases} \tag{24}$$

Table 1 and 2 show the operations of each step for the training and testing phase in brief.

IV. EXPERIMENTS AND RESULTS

In this section, we evaluate the efficacy and feasibility of H-ELM-LRF-DeepID using two face verification datasets, i.e., YouTube Faces dataset and Labeled Faces in the Wild (LFW) dataset. We run our proposed framework of H-ELM-LRF-DeepID in a MATLAB 2016a environment on the Intel Xeon CPU, 3.40GHz with 8GB RAM, against numerous current state-of-the-art methods as described in Section A and B below. The two hyperparameters, i.e., (i) the number of feature maps for the first pair of convolution-pooling layer (K_1) is set as 20 that follows the design in [27], and (ii) the number of hidden neurons (L) is defined as 2000 for huge face verification datasets. According to Huang *et al.* [25], good generalization performance can be obtained when the size of hidden neuron is large enough.

TABLE 1. Training phase of face verification of H-ELM-LRF-DeepID.

Step	Operation	Input	Output
Step 1	Load data for training	Data file location	- Training Images - Target Output
Step 2	Resize data to 60 x 60 if necessary	Training Images	- Resized Training Images - Target Output
Step 3	Convolutional layer - randomly generate input weight matrix - orthogonalize the input weight matrix - convolution process	- Resized Training Images - Convolutional mask - Number of feature map	Feature maps of convolution layer
Step 4	Pooling layer -Modified Square and Square Root (MSSR) pooling and downsampling	- Feature maps - Pooling scale	Pooled and downsized maps of pooling layer
Step 5	Repeat Step-3 and Step-4 for another two pairs of Convolution and Pooling (C-P) layers.	Same as step 3 and 4	Same as step 3 and 4
Step 6	The last feature extraction layer, comprising only Convolution layer.	Pooled and downsized maps of pooling layer	Feature maps of convolution layer
Step 7	Concatenation of the last C-P layers to form a feature vector as input to the subsequent hidden layer.	Feature maps of convolution layer	Feature vector
Step 8	Initialization of hidden layer parameters.	- Number of hidden neurons - C parameter	- Hidden weights
Step 9	Compute Hidden layer output matrix that follows ELM native procedure	- Feature vector - Hidden weights	Hidden layer output matrix
Step 10	Compute output weights using Moore-Penrose generalized inverse method.	Hidden layer output matrix	Output weights
Step 11	Save weights matrices for use in testing.	- Convolutional mask - Hidden weights - Output weights	Saved data file

A. YOUTUBE FACES DATASET

The dataset under consideration is YouTube Faces dataset. YouTube Faces database is a dataset of numerous face videos designed for investigating the problem of unconstrained face recognition in the format of videos. The objective of this database is to produce a large scale collection of videos with large variations in expression, pose, illumination, age and so on, along with class labels to indicate a person's identity appearing in each video.

There are 3425 videos of 1595 different people (subjects) downloaded from YouTube. Each subject will have an average of 2.15 videos, with 48 frames for the shortest clip, and 6070 frames for longest clip duration. As such, the average length of each video clip is approximately 181 frames.

In this paper, we follow the setting as in [31] and only consider the restricted protocols. In this standard setting, we use the 5000 video pairs randomly selected in [31], half of which are from the same subjects, and the remaining half are from different subjects. These pairs are divided into ten subsets with each subset containing 250 same and 250 not same pairs for ten-fold cross validation. The pairs are precisely divided to make sure that the two categories (same/not same) remain

TABLE 2. Testing phase of face verification of H-ELM-LRF-DeepID.

Step	Operation	Input	Output
Step 1	- Obtain a pair of new, unseen testing image for testing face verification. - Load all matrix that has been saved from the training phase	- Data file of testing images - Data file of weights	- Testing image - Convolutional mask - Hidden weights - Output weights
Step 2	Complete the feature extraction procedures to obtain feature vector of the testing image.	- Input weights matrix - Feature maps of convolution layer Pooled and downsized maps of pooling layer	Feature vector
Step 3	Compute hidden layer output matrix for testing.	- Hidden weights - Feature vector	Testing hidden layer output matrix
Step 4	Calculate the prediction output	- Testing hidden layer output matrix - Output weights	Prediction output
Step 5	Decision of the face verification	Prediction output	If prediction output is positive, the faces are of same identity. Else, different identity.

subject mutually exclusive. In other words, it means that the subjects in the test set will not appear in the training set. This is to evaluate if the proposed model is efficient to learn the properties and knowledge of what determines the face similar and dissimilar. The C parameter of H-ELM-LRF-DeepID is selected as 2^5 based on the best result of ten-fold cross validation.

As shown in Table 4, the proposed H-ELM-LRF-DeepID achieves a remarkable verification accuracy of 90.32% as compared to other profound state-of-the-art face verification algorithms in the literature on Youtube Faces dataset, i.e., MBGS-LBP [31], MBGS-FLBP [31], MBGS + SVM [32], APEM-FUSION [33], STFRD + PMML [34], VSOF + OSS (Adaboost) [35], DDML [36], EigenPEP [37], LM3L [38], CNN-3DMM estimation [39] and Joint Bayesian [40], etc.

It is worth to highlight that the proposed H-ELM-LRF-DeepID has also outperformed the well-known Joint Bayesian approach in face verification published recently [40], by a considerable margin. ROC curve in Fig. 6 also shows that H-ELM-LRF-DeepID performs significantly well. The closer the ROC curve to the (0,1) point, the better the deviation from the 45-degree diagonal line. On the other hand, it can be observed that the conventional ELM achieves 66.65% in the face verification performance, while the ELM-LRF with its compelling feature mapping layer manages to elevate the accuracy further to 80.15% by learning distinctive features more effectively, in contrast to the simple 3-layer ELM.

As for training time, H-ELM-LRF-DeepID takes 3.63 seconds to complete Eqn. (21) and 346.42 seconds for the whole training process for YouTube Faces dataset. On the other hand, the ELM-LRF which consists of a pair of convolution-pooling layer for feature mapping takes around 171.79 seconds for the entire training of YouTube data, while the conventional ELM takes only 3.95 seconds to complete the training process of the same dataset in view of its simple implementation of only 3-layer architecture without any feature extraction layer.

B. LABELED FACES IN THE WILD (LFW) DATASET

We also evaluate H-ELM-LRF-DeepID on the LFW dataset, which reveals the state-of-the-art of face verification in the wild. The LFW dataset [41] consists of more than 13000 face images of 5749 subjects with huge variations in resolution, age, pose, illumination, and expression.

We follow the standard evaluation setting as in [41] and only consider LFW for training. First, 6000 pairs of images are formed, half of which are pairs of images of the same person, and half of different individuals. These pairs of images are divided into ten subsets, and each subset consists of 300 same and 300 not-same pairs. The performance is computed using ten-fold cross validation using these subsets. Here, the C parameter of the proposed H-ELM-LRF-DeepID used is 2^{10} based on best result of the ten-fold cross validation.

The subsets are distributed in a subject mutually exclusive manner, whereby if images of a subject appear in the subset, no image of that subject is included in another subset. This distribution design encourages H-ELM-LRF-DeepID to learn what makes faces similar and dissimilar, rather than learn the facial appearances and features of specific subjects.

In this experiment, we compare our proposed framework with several state-of-the-art face verification algorithms,

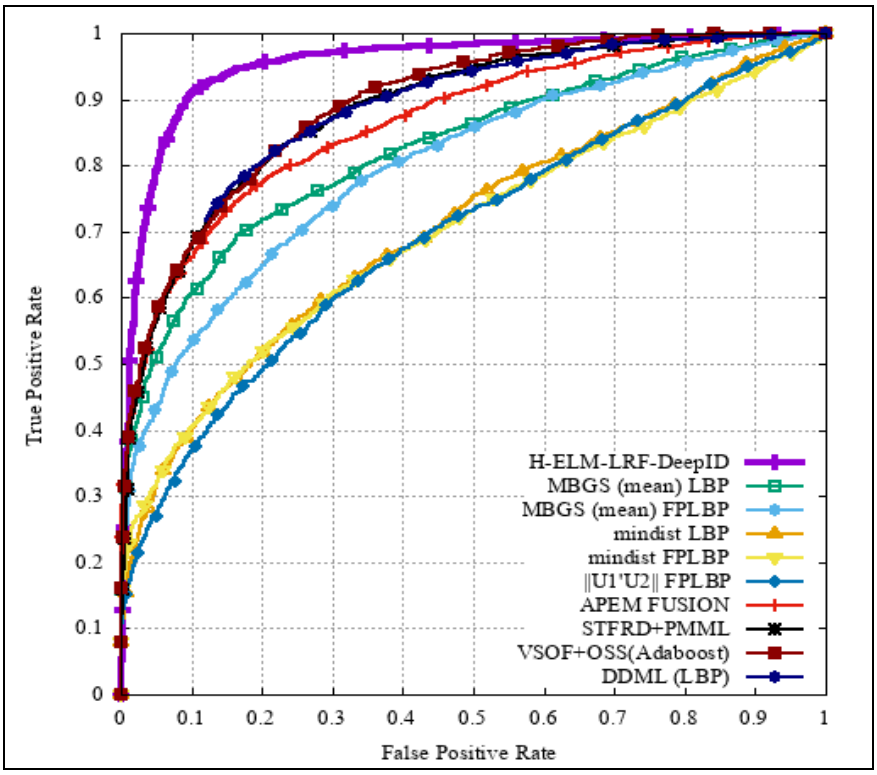

FIGURE 6. ROC curve of the proposed H-ELM-LRF-DeepID and existing state-of-the-art algorithm.

including APEM-FUSION [33], Eigen-PEP [37], Joint Bayesian [40], Conv Net-RBM [42], CMD + SLBP [43], Fisher Vector Faces [44], Tom-vs-Pete Classifiers [45], High-dim LBP [46], ConvNet-RBM [47]. From Table 5, it is obvious that the proposed H-ELM-LRF-DeepID achieves significant verification accuracy of 97.47%, which consistently outperforms other state of the art methods by 13.39% [33], 8.5% [37], 4.29% [40], 4.95% [42], 4.89% [43], 4.44% [44], 4.17% [45], 4.27% [46], and 0.39% [47]. Besides that, we also evaluate and record the verification accuracy of the conventional ELM and ELM-LRF on the LFW dataset in Table 5. H-ELM-LRF-DeepID takes 3.89 seconds to complete Eqn. (21) and 372.68 seconds for the whole training process of LFW dataset. On the other hand, the ELM-LRF with only 1 convolution layer and 1 pooling layer takes 158.98 seconds for the training of LFW dataset. Not forgetting about the conventional ELM, it tops the computational efficiency among all other methods due to the simplicity and convenience of implementation in its 3-layer neural network architecture, by showing a record of 3.72 seconds for training the LFW dataset.

C. LABELED FACES IN THE WILD (LFW) DATASET WITH TRAINING ON THE CelebFaces DATASET

We further evaluate the face verification accuracy of the proposed H-ELM-LRF-DeepID for the LFW dataset with cross-dataset training using the outside training data, i.e., CelebFaces dataset [42]. For this purpose, we follow the setting in [42], [47], where 80% from CelebFaces are randomly chosen to train the proposed model, while the remaining 20% are used for validation purpose in order to select the best C parameter. Here, the C parameter of the proposed H-ELM-LRF-DeepID is selected as 2^7 based on the best result of the ten-fold cross validation. Note that the subjects in CelebFaces and LFW are mutually exclusive.

Table 6 shows the verification results of LFW for various state-of-the-art methods that rely on the outside training data. Here, we achieve remarkable verification accuracy of 97.37%, which is superior to the numerous alternative state-of-the-art approaches [6], [27], [40], [46]–[48]. On this assessment, the good result indicates that the verification effectiveness lies within the competence of the effective feature learning of the proposed H-ELM-LRF-DeepID which can be deemed independent of the source of the dataset. Besides that, we also record the verification accuracy of the conventional ELM and ELM-LRF for training on the CelebFaces dataset in the same table. ELM-LRF attains better face verification accuracy of 84.05% as compared to the conventional ELM due to the fact that the ELM-LRF comprises a feature mapping layer that allows effective learning and extraction of more meaningful representations of the image when dealing with computer vision and image processing tasks. As for training time, H-ELM-LRF-DeepID takes 373.55 seconds to complete the training of the CelebFaces

TABLE 3. Nomenclature.

Symbol	Description
$\mathbf{X}$	Training images
$\mathbf{T}$	Target Output
N	Total number of training images and respective output
$\boldsymbol{x_1}$	Image-1 of Training images
$\boldsymbol{x_2}$	Image-2 of Training images
q	Size of image
C1	Convolution layer 1
K_1	Number of Feature Map of Convolution layer 1
$\overline{\mathbf{W}}^{init}$	Input weights matrix
r	Receptive field
$c_{i,j,k}$	The resulted convolution at coordinate (i, j) of image in the k-th feature map
P1	Pooling layer 1
s	Pooling scale
$p_{u,v,k}$	The resulted pooling map at coordinate (u, v) of k-th feature map
K_i	Number of Feature Map of i-th Convolution layer
P3	Pooling layer 3
C4	Convolution layer 4
K_4	Number of Feature Map of Convolution layer 4
$\mathbf{V}$	The resulted concatenation feature vector
L	The number of neurons of fully connected hidden layer
$\{(\mathbf{a}_i, b_i)\}_{i=1}^{L}$	Hidden weights and bias
C	User –specific parameter, $C \in \{2^{-24}, 2^{-23}, \ldots, 2^{24}, 2^{25}\}$
$\mathbf{H}$	Hidden layer output matrix for training phase
$G(.)$	Output function of the Sigmoid hidden neuron to feature vector
$\boldsymbol{\beta}$	Output weights
$\mathbf{Z}$	Unseen testing images
$\mathbf{h}$	Hidden layer output matrix for testing phase
y	Prediction output

TABLE 4. Verification results over the YouTube faces dataset.

Method	Verification Accuracy
Min dist, FPLBP [31]	65.6
Min dist, LBP [31]	65.7
‖U1'U2‖, FPLBP [31]	64.3
‖U1'U2‖, LBP [31]	65.4
MBGS L2 mean, FPLBP [31]	72.6
MBGS L2 mean, LBP [31]	76.4
MBGS+SVM [32]	78.9
APEM-FUSION [33]	79.1
STFRD+PMML [34]	79.5
VSOF+OSS (Adaboost) [35]	79.7
DDML (LBP) [36]	81.3
DDML (combined) [36]	82.3
EigenPEP [37]	84.8
LM3L [38]	81.3
CNN-3DMM estimation [39]	88.8
Joint Bayesian [40]	84.4
ELM	66.65
ELM-LRF	80.15
H-ELM-LRF-DeepID	**90.32**

TABLE 5. Verification results over the LFW dataset (without outside training data).

Method	Verification Accuracy
APEM-FUSION [33]	84.08
Eigen-PEP [37]	88.97
Joint Bayesian [40]	93.18
ConvNet-RBM [42]	92.52
CMD+SLBP [43]	92.58
Fisher Vector Faces [44]	93.03
Tom-vs-Pete Classifiers [45]	93.30
High-dim LBP [46]	93.20
ConvNet-RBM [47]	97.08
ELM	71.13
ELM – LRF	84.22
H-ELM-LRF-DeepID	**97.47**

dataset, while the ELM-LRF with only 1 convolution layer and 1 pooling layer takes 159.90 seconds to complete the training process, and the ELM with the 3-layer structure takes only 3.86 seconds for the training.

It is worth pointing out that the proposed H-ELM-LRF-DeepID has a more straightforward and simplified architecture as compared to the other deep learning methods in Table 4, 5 and 6, on the grounds of the random generation of the weights of the masks in the convolutional layers of H-ELM-LRF-DeepID that no iterative adjustments of weights required during the training process.

In short, H-ELM-LRF-DeepID inherits the virtue of ELM [25] as a simplified and effective learning algorithm with the direct regularized least square solution. Besides that, the randomly generated connection weights of H-ELM-LRF-DeepID due to the different types of probability distributions used in applications during the initialization stage, also contributes to the ease of implementation of H-ELM-LRF-DeepID. Huang [23] mentioned that if all of the hidden neuron parameters are randomly generated according to any continuous sampling probability distribution, then the output mapping of the hidden layer is considered random feature mapping. More importantly, when the hidden layer mapping need not be tuned and is not parametric, it becomes

TABLE 6. Verification results over the LFW dataset (with training on celebFaces dataset).

Method	Verification Accuracy
High-dim LBP [46]	95.17
ConvNet-RBM [47]	97.08
TL Joint Bayesian [48]	96.33
Joint Bayesian [40]	95.17
DeepID [27]	96.05
DeepFace [6]	97.35
ELM	71.04
ELM-LRF	84.05
H-ELM-LRF-DeepID	**97.37**

independent of the training data samples and no longer sensitive to the user-defined parameters.

Based on the outstanding results, the proposed H-ELM-LRF-DeepID is now proved empirically to be an effective, practical and robust face verification algorithm for being able to distinguish the similarity and dissimilarity properties from the observations of feature vectors.

V. CONCLUSION

The ELM is an emerging approach in the field of machine learning, but it has not been investigated for solving face verification problem. It will be a new breakthrough to ELM ideology if ELM-based model can prove itself not only good at handling face or object identification, but also capable of tackling the face verification task as good as other state-of-the-art face verification algorithms. In this paper, we propose and prove a novel, distinguished and unified end-to-end face verification framework, that well integrates two stages of face verification (i.e., discriminative feature extraction and verification) into a unified locally connected ELM architecture, denoted as Hybrid Local Receptive Field based Extreme Learning Machine with DeepID (H-ELM-LRF-DeepID). The assimilation of DeepID into ELM architecture ensures discriminative multi-scale feature extraction containing both mid-level and global high-level features because meaningful features representations are essential for face verification. Different from most of the state-of-the-art algorithms, H-ELM-LRF-DeepID does not implement an end to end deep CNN based framework with BP learning for face verification. Owing to its straightforward architecture in output layer and tuning free hidden neurons that guarantees good generalization capability, H-ELM-LRF-DeepID has eliminated the need for the extensive computational cost of the iterative gradient descent operation in the complex face verification tasks. The encouraging performance of the proposed H-ELM-LRF-DeepID over the YouTube Faces dataset has well demonstrated its applicability, competitiveness, and efficacy in solving face verification task, especially taking into consideration the ease of implementation by accepting raw images that come with RGB component for processing.

REFERENCES

[1] K. Choi, K.-A. Toh, and H. Byun, "Incremental face recognition for large-scale social network services," *Pattern Recognit.*, vol. 45, no. 8, pp. 2868–2883, Aug. 2012.

[2] B. Ammour, T. Bouden, L. Boubchir, and S. Biad, "Face identification using local and global features," in *Proc. 40th Int. Conf. Telecommun. Signal Process. (TSP)*, Barcelona, Spain, Jul. 2017, pp. 784–788.

[3] B. Moghaddam, T. Jebara, and A. Petland, "Bayesian face recognition," *Pattern Recognit.*, vol. 33, no. 11, pp. 1771–1782, Nov. 2000.

[4] D. Cui, G. Zhang, K. Hu, W. Han, and G.-B. Huang, "Face recognition using total loss function on face database with ID photos," *Opt. Laser Technol.*, vol. 110, pp. 227–233, Feb. 2019.

[5] S. Bijarnia and P. Singh, "Pyramid binary pattern for age invariant face verification," in *Proc. 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Jaipur, India, Dec. 2017, pp. 218–221.

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1701–1708.

[7] R. B. Kloss, A. Jord, and W. R. Schwartz, "Face verification: Strategies for employing deep models," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Xi'an, China, May 2018, pp. 258–262.

[8] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, K. Wang, J. Yan, C.-C. Loy, and X. Tang, "DeepID-Net: Object detection with deformable part based convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1320–1334, Jul. 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[11] H. Zhao and H. Liu, "Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition," in *Granular Computing*. Springer, 2019, pp. 1–8. [Online]. Available: https://link.springer.com/article/10.1007/s41066-019-00158-6s

[12] B. Ameur, M. Belahcene, S. Masmoudi, and A. Ben Hamida, "Weighted PCA-EFMNet: A deep learning network for face verification in the wild," in *Proc. 4th Int. Conf. Adv. Technol. Signal Image Process. (ATSIP)*, Sousse, Tunisia, Mar. 2018, pp. 1–6.

[13] D. Chen, C. Xu, J. Yang, J. Qian, Y. Zheng, and L. Shen, "Joint Bayesian guided metric learning for end-to-end face verification," *Neurocomputing*, vol. 275, pp. 560–567, Jan. 2018.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[15] Z. Bai, L. Lekamalage, C. Kasun, and G. Huang, "Generic object recognition with local receptive fields based extreme learning machine," *Procedia Comput. Sci.*, vol. 53, pp. 391–399, Aug. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915018190

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, May 2006.

[17] S. Y. Wong, K. S. Yap, and H. J. Yap, "A constrained optimization based extreme learning machine for noisy data regression," *Neurocomputing*, vol. 171, pp. 1431–1443, Jan. 2016.

[18] S. Y. Wong, K. S. Yap, H. J. Yap, and S. C. Tan, "A truly online learning algorithm using hybrid fuzzy ARTMAP and online extreme learning machine for pattern classification," *Neural Process. Lett.*, vol. 42, no. 3, pp. 585–602, Dec. 2015.

[19] S. Y. Wong, K. S. Yap, H. J. Yap, S. C. Tan, and S. W. Chang, "On equivalence of FIS and ELM for interpretable rule-based knowledge representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1417–1430, Jul. 2015.

[20] Y. Peng, W. Kong, and B. Yang, "Orthogonal extreme learning machine for image classification," *Neurocomputing*, vol. 266, pp. 458–464, Nov. 2017.

[21] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Nov. 2013.

[22] W. Schmidt, M. A. Kraaijveld, and R. P. W. Duin, "Feedforward neural networks with random weights," in *Proc. 11th IAPR Int. Conf. Pattern Recognit. Methodol. Syst.*, Hague, The Netherlands, Aug./Sep. 1992, pp. 1–4

[23] G. B. Huang, "What are extreme learning machines? Filling the gap between frank Rosenblatt's dream and John von Neumann's puzzle," *Cogn. Comput.*, vol. 7, no. 3, pp. 263–278, 2015.

[24] G.-B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cognit. Comput.*, vol. 6, no. 3, pp. 376–390, 2014.
[25] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[26] G.-B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Comput. Intell. Mag.*, vol. 10, no. 2, pp. 18–29, May 2015.
[27] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Washington, DC, USA, Jun. 2014, pp. 1891–1898.
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 1–9.
[29] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, vol. 2, 2014, pp. 1988–1996.
[30] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2892–2900.
[31] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Colorado Springs, CO, USA, Jun. 2011, pp. 529–534.
[32] L. Wolf and N. Levy, "The SVM-minus similarity score for video face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3523–3530.
[33] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3499–3506.
[34] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3554–3561.
[35] H. Méndez-Vázquez, Y. Martínez-Díaz, and Z. Chai, "Volume structured ordinal features with background similarity measure for video face recognition," in *Proc. Int. Conf. Biometrics (ICB)*, Madrid, Spain, Jun. 2013, pp. 1–6.
[36] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1875–1882.
[37] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, "Eigen-PEP for video face recognition," in *Proc. 12th Asian Conf. Comput. Vis. (ACCV)*, Singapore, 2014, pp. 17–33.
[38] H. Dong, S. Gong, C. Liu, Y. Ji, and S. Zhong, "Large margin relative distance learning for person re-identification," *IET Comput. Vis.*, vol. 11, no. 6, pp. 455–462, 2017.
[39] A. T. Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing robust and discriminative 3D morphable models with a very deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5163–5172.
[40] D. Chen, X. Cao, D. Wipf, F. Wen, and J. Sun, "An efficient joint formulation for Bayesian face verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 32–46, Jan. 2017.
[41] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Proc. Workshop Faces 'Real-Life' Images, Detection, Alignment, Recognit.*, Marseille, France, 2008, pp. 1–14.
[42] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 1489–1496.
[43] C. Huang, S. Zhu, and K. Yu, "Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval," NEC, Irving, TX, USA, Tech. Rep. TR115, 2011.
[44] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Bristol, U.K., 2013, pp. 1–12.
[45] T. Berg and P. N. Belhumeur, "Tom-vs-pete classifiers and identity-preserving alignment for face verification," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Surrey, U.K., 2012, p. 7.
[46] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 3025–3032.
[47] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1997–2009, Oct. 2016.
[48] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 3208–3215.

SHEN YUONG WONG received the bachelor's and master's degrees (Hons.) in electrical and electronic engineering and the Ph.D. degree in engineering from Universiti Tenaga National (National Energy University), Malaysia, in 2010 and 2012, respectively. She is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Xiamen University Malaysia. She is a Professional Engineer (P.Eng./Ir.) registered to the Board of Engineers Malaysia, accredited under the Washington Accord. Her research interests include computer vision and image processing, extreme learning machine, deep learning, classification, regression, and other applications of artificial intelligence. She serves as an Editorial Board Member of the *Applied Soft Computing Journal*.

KEEM SIAH YAP received the bachelor's and M.S. degrees in electrical engineering from the University of Technology Malaysia, in 1998 and 2000, respectively, and the Ph.D. degree in electronics engineering from the University of Science, Malaysia. He is currently a Full Professor with Universiti Tenaga Nasional, Malaysia. He is currently a Professional Engineer registered to the Board of Engineers Malaysia, accredited under the Washington Accord. His research interests include theory and applications of artificial intelligence, pattern recognition, regression, clustering, and extreme learning machine.

QINGWEI ZHAI is currently pursuing the bachelor's degree in electrical and electronics engineering with Xiamen University Malaysia. Through this research project, he hopes to get some exposure to research as an undergraduate.

XIAOCHAO LI received the B.Sc. degree in electronic engineering from the Beijing Institute of Technology, China, in 1992, and the M.S. degree in electrical engineering and the Ph.D. in solid-state physics from Xiamen University, China, in 1995 and 2005, respectively. He was a Postdoctoral Fellow with Xidian University, a Visiting Scholar with North Carolina State University, and a Visiting Fellow with the University of Macau. He is currently an Associate Professor of electronic engineering with Xiamen University China, the Head of Department with Xiamen University Malaysia, and the Director of the Fujian Key Laboratory of Integrated Circuit Design and Measurement, Xiamen University China. He has authored over 50 research papers, six patents of invention, seven software or integrated circuit layout copyright, and one book. His research interests include artificial intelligence, mixed signal integrated circuit design, parallel and distributed processing, and embedded systems.