

RESEARCH ARTICLE

Hypercomplex extreme learning machine with its application in multispectral palmprint recognition

Longbin Lu^{1*}, Xinman Zhang¹, Xuebin Xu²

1 MOE Key Lab for Intelligent Networks and Network Security, School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, China, **2** Guangdong Xi'an Jiaotong University Academy, Foshan, Guangdong, China

* baqihuti@stu.xjtu.edu.cn

OPEN ACCESS

Citation: Lu L, Zhang X, Xu X (2019) Hypercomplex extreme learning machine with its application in multispectral palmprint recognition. PLoS ONE 14(4): e0209083. <https://doi.org/10.1371/journal.pone.0209083>

Editor: Zhiqiang Cai, The University of Memphis, UNITED STATES

Received: July 5, 2018

Accepted: November 29, 2018

Published: April 15, 2019

Copyright: © 2019 Lu et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The source code together with datasets involved in this paper can be available for free download at <https://figshare.com/s/01aef7d48840afab9d6d>.

Funding: This work is supported by the National Natural Science Foundation of China (No. 61673316; <http://www.nsf.gov.cn/>), and the special grants from the Major Science and Technology Foundation of Guangdong Province (No. 2015B010104002; <http://zdkjzx.gdstc.gov.cn/>). XMZ is the receiver of all the mentioned funds. The funders had no role in study design, data

Abstract

An extreme learning machine (ELM) is a novel training method for single-hidden layer feed-forward neural networks (SLFNs) in which the hidden nodes are randomly assigned and fixed without iterative tuning. ELMs have earned widespread global interest due to their fast learning speed, satisfactory generalization ability and ease of implementation. In this paper, we extend this theory to hypercomplex space and attempt to simultaneously consider multi-source information using a hypercomplex representation. To illustrate the performance of the proposed hypercomplex extreme learning machine (HELM), we have applied this scheme to the task of multispectral palmprint recognition. Images from different spectral bands are utilized to construct the hypercomplex space. Extensive experiments conducted on the PolyU and CASIA multispectral databases demonstrate that the HELM scheme can achieve competitive results. The source code together with datasets involved in this paper can be available for free download at <https://figshare.com/s/01aef7d48840afab9d6d>.

1 Introduction

Nowadays, machine learning has been playing an increasingly significant role in our daily life and a variety of machine learning areas have attracted great interests from researchers. The neural network technology, as a kind of typical machine learning method, is proven to be a successful tool for artificial intelligence (AI). With the rapid development of computer hardware, deep neural network techniques also achieve a huge success in various kinds of recognition tasks [1, 2].

Recently, a novel machine learning theory called extreme learning machine (ELM) was proposed by Huang et al. [3, 4] and has aroused growing worldwide concerns. It is a machine learning method for single-hidden layer feedforward neural networks (SLFNs) that differs from the traditional back-propagation (BP) algorithm and its variants [5–7]. Unlike the BP-based learning approach, which employs an iterative process to tune the hidden nodes of an SLFN, the ELM method completes a training task without any repeated optimizing steps. An ELM randomly assigns the input weights and bias of an SLFN and analytically calculates the

collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

output weights by a simple Moore–Penrose generalized inverse. This algorithm can avoid many difficulties of conventional learning methods, such as the settings of stopping criteria, learning rates and learning epochs. ELM has been shown to be able to find a global optimal solution with excellent universal approximation ability and a very fast learning speed. The advantages of ELM—computation cost and generalization performance—render it one of the most popular machine learning methods, with extensive and successful applications in classification, regression, clustering, compression and feature learning problems [8–12].

In the past decade, a variety of ELM variants were proposed to address problems in the original theory; they have significantly enhanced the contributions of ELM to theoretical studies and engineering applications. For example, a regularized extreme learning machine [13] was investigated to solve the overfitting problem based on structural risk minimization principle and weighted least squares. In [14], a fast and accurate online sequential learning variant was developed and applied to gesture recognition and object tracking; it has shown excellent performance with regard to accuracy and computation time. In [15], the original ELM model was extended to complete classification and regression tasks with noisy or missing data. From the implementation aspect, a stacked ELM invariant [16] was designed to render it feasible for large data sets and real-time reasoning. In recent years, the ELM concept has been introduced in multilayer perception [17, 18]. As demonstrated in [17], compared with the greedy layer-wise training of deep learning, the ELM-based framework has a substantially better learning efficiency.

Although ELM has been successfully applied in an extensive range of domains, it is primarily utilized for single-source information classification. In the case of multisource features, a fusion operation must be performed either at the feature level or the matching score level [19–21] to achieve a final result. For feature-level fusion, multisource features are simply jointed. The input hidden nodes of ELM are adjusted to the dimension of the joint features, which generates a very large computational cost. Regarding matching score level fusion, features from each channel are calculated to separately obtain a matching score by ELM; these scores are fused for the final decision. This strategy merely considers the matching score information that has lost some discriminative features. Therefore, the high accuracy of the fusion at the feature level cannot be achieved. In this paper, we propose a hypercomplex extreme learning machine (HELM) from a different perspective for the classification of multisource information. A hypercomplex representation [22, 23] is introduced in the ELM theory. Multisource features are employed to construct the hypercomplex space, and hypercomplex operation rules are applied to determine the output weights of SLFNs. In addition, a fusion strategy is performed on the hypercomplex output nodes to obtain a decision.

As a typical kind of multisource information processing problem, multispectral palmprint recognition has gained widespread attentions in recent years. Some previous works tried to process the multisource information using a fusion operation. For example, Lu et al. [24] completed an illumination-invariant palmprint recognition system by fusing the multispectral images at image level, in which a FABEMD+WFC fusion framework was developed. Similarly, Xu et al. [25] fused the multispectral images using a digital shearlet transform based method and then classified the fused images with the extreme learning machine. Gumaei et al. [26] proposed a kind of Gabor-based feature extraction method and employed the optimal spectral band to determine the identities. The same authors [27] further utilized a hybrid feature extraction method named HOG-SGF instead of Gabor-based one to represent the multispectral palmprint images. Recently, Gumaei et al. [28] developed a new anti-spoof multispectral biometric cloud-based identification approach for privacy and security of cloud computing, in which a tree-complex wavelet transform was applied to complete the multispectral fusion task and Gabor features were used to represent the fusion images. Different from a fusion view, in

this paper we try to address the multispectral palmprint recognition problem by the proposed HELM framework, in which the fusion stage could be circumvented. To evaluate the performance of the proposed method, we conduct some experiments using the PolyU and CASIA multispectral palmprint databases [29–33]. Palmprint images from multispectral bands are employed to construct the hypercomplex representation.

The remainder of this paper is organized as follows: Section 2 provides a brief review of the ELM, describes the HELM theory and introduces the application of HELM in multispectral palmprint recognition. Section 3 illustrates the experimental results of the proposed method, which is tested using the PolyU and CASIA multispectral palmprint databases. Some concluding remarks are provided in the last section.

2 Related work

2.1 Extreme learning machine

The ELM is a novel learning method for SLFNs that randomly assigns the hidden layer and analytically determines the output weights of SLFNs. For N distinct training data $\{\mathbf{x}_i, \mathbf{t}_i\}$, $i = 1, 2, \dots, N$, \mathbf{x}_i is a $1 \times n$ input vector, and \mathbf{t}_i is a $1 \times m$ output vector with only one entry (correspond to the class to which \mathbf{x}_i belongs) equal to one. n is the dimension of the input data, and m is the number of classes. To train an SLFN with \tilde{N} hidden nodes, the appropriate input weight vectors $\boldsymbol{\alpha}_j$, $j = 1, 2, \dots, \tilde{N}$ and output weight vectors $\boldsymbol{\beta}_j$, $j = 1, 2, \dots, \tilde{N}$ are required, such that

$$f_{\tilde{N}}(\mathbf{x}_i) = \sum_{j=1}^{\tilde{N}} g(\mathbf{x}_i^e \boldsymbol{\alpha}_j) \boldsymbol{\beta}_j = \mathbf{t}_i, \quad i = 1, 2, \dots, N, \tag{1}$$

where $\boldsymbol{\alpha}_j$ is a $(n+1) \times 1$ vector that connects the input nodes to the j th hidden node, $\boldsymbol{\beta}_j$ is a $1 \times m$ vector that connects the j th hidden node to the output nodes, \mathbf{x}_i^e is the augmenting vector of \mathbf{x}_i with the format of $[\mathbf{x}_i, 1] \in R^{n+1}$, and $g(x)$ is the activation function.

This formula can be compactly written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \tag{2}$$

where $\mathbf{H} = \begin{bmatrix} g(\mathbf{x}_1^e \boldsymbol{\alpha}_1) & \cdots & g(\mathbf{x}_1^e \boldsymbol{\alpha}_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(\mathbf{x}_N^e \boldsymbol{\alpha}_1) & \cdots & g(\mathbf{x}_N^e \boldsymbol{\alpha}_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}$, $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times m}$ and $\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{bmatrix}_{N \times m}$. The hidden layer output matrix \mathbf{H} is compactly described as

$$\mathbf{H} = g(\mathbf{x}^e \boldsymbol{\alpha}), \tag{3}$$

where $\mathbf{x}^e = \begin{bmatrix} \mathbf{x}_1^e \\ \vdots \\ \mathbf{x}_N^e \end{bmatrix}_{N \times (n+1)}$ and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1 \cdots \boldsymbol{\alpha}_{\tilde{N}}]_{(n+1) \times \tilde{N}}$.

Generally, a typical ELM training process consists of two main steps. The first step is to calculate the hidden layer output matrix with the random map $\boldsymbol{\alpha}$ and a nonlinear piecewise continuous function, such as the following sigmoid function, sin function and atan function:

1. Sigmoid function:

$$\mathbf{H}(i, j) = \frac{1}{1 + \exp(-\mathbf{x}_i^e \boldsymbol{\alpha}_j)} \tag{4}$$

2. Sin function:

$$\mathbf{H}(i, j) = \sin(\mathbf{x}_i^e \boldsymbol{\alpha}_j) \tag{5}$$

3. Atan function:

$$\mathbf{H}(i, j) = \text{atan}(\mathbf{x}_i^e \boldsymbol{\alpha}_j) \tag{6}$$

where $\mathbf{H}(i, j)$ is the value of \mathbf{H} at the position (i, j) .

A remarkable characteristic of ELM is that the input weight matrix $\boldsymbol{\alpha}$ of the hidden nodes can be randomly generated according to any continuous probability distribution, for example, the uniform distribution on $[-1, 1]$. ELM distinctly differs from conventional feedforward neural networks. As demonstrated by Eq (2), the only parameters that need to be optimized in the

training process are the output weights $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times m}$ between the hidden nodes and the out-

put nodes. Mathematically, training an SLFN by an ELM can be transformed into solving a regularized least squares problem, as illustrated in Eq (2). Additional iterative steps are not required to tune the parameters of SLFNs, which are significantly more efficient than BP-like algorithms.

In the second step, ELM attempts to determine the output weights by minimizing the following loss function:

$$\begin{aligned} E &= \min_{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_{\tilde{N}}} \sum_{j=1}^{\tilde{N}} (g(\mathbf{x}_i^e \boldsymbol{\alpha}_j) \boldsymbol{\beta}_j - \mathbf{t}_i)^2 \\ &= \min_{\boldsymbol{\beta} \in R^{\tilde{N} \times m}} \|g(\mathbf{x}^e \boldsymbol{\alpha}) \boldsymbol{\beta} - \mathbf{T}\| \\ &= \min_{\boldsymbol{\beta} \in R^{\tilde{N} \times m}} \|\mathbf{H} \boldsymbol{\beta} - \mathbf{T}\| \end{aligned} \tag{7}$$

Huang et al. [4] proved that if the activation function g is infinitely differentiable, for $N = \tilde{N}$ arbitrary distinct samples $\{\mathbf{x}_i, \mathbf{t}_i\}, i = 1, 2, \dots, N$, for any randomly assigned $\boldsymbol{\alpha}$ according to any continuous probability distribution, the hidden layer output matrix \mathbf{H} is invertible and $\|\mathbf{H} \boldsymbol{\beta} - \mathbf{T}\| = 0$. Thus, the output weight matrix $\boldsymbol{\beta}$ can be calculated by

$$\boldsymbol{\beta} = \mathbf{H}^{-1} \mathbf{T}, \tag{8}$$

where \mathbf{H}^{-1} denotes the inverse matrix of \mathbf{H} .

In most cases, the number \tilde{N} of hidden nodes is significantly less than the number N of distinct training samples, \mathbf{H} is a non-square matrix, and an inverse matrix for \mathbf{H} does not exist. Huang has provided another method for finding the smallest norm least squares solution of

Eq (2), that is,

$$\beta = (\mathbf{H}^T\mathbf{H} + \frac{1}{C}\mathbf{I})^{-1}\mathbf{H}^T\mathbf{T}, \tag{9}$$

where \mathbf{H}^T is the transpose of \mathbf{H} , C is a penalty coefficient, and \mathbf{I} is an identity matrix with the size $\tilde{N} \times \tilde{N}$.

Here, the procedures for training an SLFN using ELM theory are as follows:

Step 1: Initialize the number \tilde{N} of hidden nodes; note that $\tilde{N} \leq N$.

Step 2: Select the suitable activation function g .

Step 3: Randomly assign the input weight matrix α .

Step 4: Construct the output matrix \mathbf{H} of the hidden layer.

Step 5: Calculate the output weight matrix β .

2.2 Hypercomplex extreme learning machine

To classify the multisource patterns, an invariant of ELM—Hypercomplex extreme learning machine—is presented. Instead of using a fusion strategy to combine the multisource information, HELM is built on hypercomplex representation, by which the model converts the multisource features into a hypercomplex space. It circumvents the process of designing fusion rules and therefore avoids the interference due to someone’s limited knowledge. In addition, HELM takes advantages of all multispectral images and learns the model parameters adaptively according to training data. Thus HELM could be more efficient and accurate than the fusion-based strategy. To elaborate on the HELM model, initially we must introduce some basic concepts of hypercomplex operation. Mathematically, a hypercomplex number is a linear combination of a real scalar and the fixed number d of imaginary units:

$$y = y^{(1)} + y^{(2)}e_1 + y^{(3)}e_2 + \dots + y^{(d+1)}e_d, \tag{10}$$

where $y^{(1)}, y^{(2)}, \dots, y^{(d+1)}$ are real numbers, and e_1, e_2, \dots, e_d are the imaginary units. They have the following relationship:

$$e_1^2 = e_2^2 = \dots = e_d^2 = e_1e_2 \dots e_d = -1, \tag{11}$$

y^* denotes the conjugate of y and is calculated by

$$y^* = y^{(1)} - y^{(2)}e_1 - y^{(3)}e_2 - \dots - y^{(d+1)}e_d. \tag{12}$$

The norm of a hypercomplex number is defined as

$$|y| = \sqrt{yy^*} = \sqrt{(y^{(1)})^2 + (y^{(2)})^2 + \dots + (y^{(d+1)})^2}. \tag{13}$$

HELM aims to extend the extreme learning theory to hypercomplex space. In the case of classification of multisource features, HELM utilizes each type of feature to construct a hypercomplex matrix. Then, the weights of an SLFN are analytically determined with the hypercomplex operation rules. Fig 1 shows the structure of the proposed HELM network, which primarily consists of four key stages: mapping the multisource features into the hidden layer using the randomly generated real input weights, constructing the hypercomplex hidden layer output matrix, calculating the hypercomplex output weight matrix, and performing a fusion strategy on the output nodes to achieve a final decision.

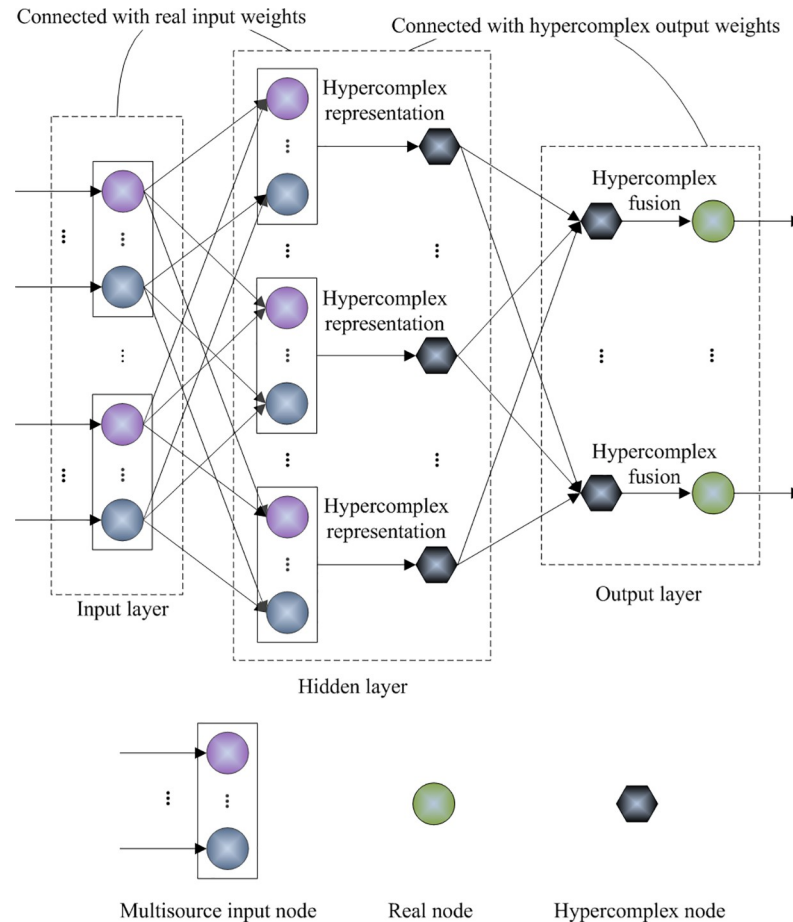


Fig 1. Structure of the proposed HELM model.

<https://doi.org/10.1371/journal.pone.0209083.g001>

For N distinct training samples with multisource features, i.e., $\{\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \mathbf{x}_i^{(3)}, \dots, \mathbf{x}_i^{(d+1)}, \mathbf{t}_i\}$, $i = 1, 2, \dots, N$, where $\mathbf{x}_i^{(j)} \in \mathbb{R}^{1 \times n}$ denotes the j th attribute of sample i , the core task for HELM is to determine the input weights and output weights of an SLFN. Similar to the settings in the ELM, we take a randomly generated map as the input weights. Each attribute of the training samples is mapped into the hidden layer as

$$\mathbf{H}^{(j)} = g(\mathbf{x}^{e(j)} \boldsymbol{\alpha}^{(j)}), \quad j = 1, 2, \dots, d + 1, \tag{14}$$

where $\mathbf{x}^{e(j)} = \begin{bmatrix} \mathbf{x}_1^{e(j)} \\ \vdots \\ \mathbf{x}_N^{e(j)} \end{bmatrix}_{N \times (n+1)}$ and $\boldsymbol{\alpha}^{(j)} = [\boldsymbol{\alpha}_1^{(j)} \dots \boldsymbol{\alpha}_N^{(j)}]_{(n+1) \times N}$. $\mathbf{H}^{(j)}$ denotes the hidden layer output matrix for the j th attribute, and $\mathbf{x}_i^{e(j)}$ is the augmenting vector of $\mathbf{x}_i^{(j)}$ with the format of $[\mathbf{x}_i^{(j)} \ 1] \in \mathbb{R}^{n+1}$. $\boldsymbol{\alpha}^{(j)}$ is the real input weight matrix for the j th attribute. The input layer and the hidden layer are connected with the set of real input weight matrices $\boldsymbol{\alpha}^{(j)}$, $j = 1, 2, \dots, d+1$.

A hypercomplex output matrix \mathcal{H} of the hidden layer is constructed using a hypercomplex representation, that is,

$$\mathcal{H} = \mathbf{H}^{(1)} + \mathbf{H}^{(2)} e_1 + \dots + \mathbf{H}^{(d+1)} e_d. \tag{15}$$

For sample i , different attributes share the same output vector \mathbf{t}_i . Thus, the hypercomplex output vector can be constructed as

$$\left. \begin{aligned} \mathbf{t}_i &= \mathbf{t}_i^{(1)} + \mathbf{t}_i^{(2)}e_1 + \dots + \mathbf{t}_i^{(d+1)}e_d \\ \mathbf{t}_i^{(1)} &= \mathbf{t}_i^{(2)} = \dots = \mathbf{t}_i^{(d+1)} = \mathbf{t}_i \end{aligned} \right\} i = 1, 2, \dots, N \tag{16}$$

This vector can be compactly described as

$$\begin{aligned} \mathcal{T} &= \mathbf{T}^{(1)} + \mathbf{T}^{(2)}e_1 + \dots + \mathbf{T}^{(j)} + \dots + \mathbf{T}^{(d+1)}e_d \\ \mathbf{T}^{(1)} &= \mathbf{T}^{(2)} = \dots = \mathbf{T}^{(j)} = \dots = \mathbf{T}^{(d+1)} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{bmatrix} \end{aligned} \tag{17}$$

where $\mathbf{T}^{(j)}$ denotes the output matrix for the j th attribute, and \mathcal{T} is the hypercomplex output matrix of an SLFN.

Having obtained \mathcal{H} and \mathcal{T} , we can solve the hypercomplex output weight matrix \mathcal{b} according to the following equations:

$$\begin{aligned} \mathcal{b} &= \mathcal{K}^{-1} \mathcal{H}^{\mathcal{T}*} \mathcal{T}, \\ \mathcal{K} &= (\mathcal{H}^{\mathcal{T}*} \mathcal{H} + \frac{1}{C} \mathcal{I}), \end{aligned} \tag{18}$$

where $\mathcal{H}^{\mathcal{T}*}$ is the hypercomplex transposition-conjugate matrix of \mathcal{H} . \mathcal{I} is the hypercomplex identity matrix with the size of $\tilde{N} \times \tilde{N}$. \mathcal{K}^{-1} is the hypercomplex matrix inversion and is calculated by a blockwise recursion process described as

$$\begin{aligned} \mathcal{K}^{-1} &= \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathcal{A}^{-1} + \mathcal{A}^{-1} \mathcal{B} \mathcal{E} \mathcal{C} \mathcal{A}^{-1} & -\mathcal{A}^{-1} \mathcal{B} \mathcal{E} \\ -\mathcal{E} \mathcal{C} \mathcal{A}^{-1} & \mathcal{E} \end{bmatrix} \\ \mathcal{E} &= (\mathcal{D} - \mathcal{C} \mathcal{A}^{-1} \mathcal{B})^{-1} \end{aligned} \tag{19}$$

where \mathcal{A} , \mathcal{B} , \mathcal{C} and \mathcal{D} are the hypercomplex matrix sub-blocks of arbitrary size. To be inverted, \mathcal{A} must be square. If the number of the rows (or columns) of \mathcal{A} exceeds one, \mathcal{A}^{-1} can be recursively solved by Eq (19). When \mathcal{A} is scalar, \mathcal{A}^{-1} is computed by

$$\mathcal{A}^{-1} = \mathcal{A}^* / |\mathcal{A}|^2 \tag{20}$$

where \mathcal{A}^* is the conjugate of \mathcal{A} , and $|\mathcal{A}|$ is the norm of \mathcal{A} . The same procedures are performed in the calculation of \mathcal{E} .

With Eq (18), the output weight matrix of the hidden layer is obtained. The hidden layer and the output layer are connected with hypercomplex weights. The input weights $\boldsymbol{\alpha}^{(j)}$, $j = 1, 2, \dots, d+1$ and the output weights \mathcal{b} of an SLFN are determined. The input weights are a series of real matrices, and the output weights are represented using a hypercomplex matrix.

Once the training process is completed, a sum rule-based fusion strategy is performed on the hypercomplex output nodes, which considers the information from multisource features. Let $\mathcal{H}\mathcal{t} = \mathbf{nt}^{(1)} + \mathbf{nt}^{(2)}e_1 + \dots + \mathbf{nt}^{(d+1)}e_d$ denote the hypercomplex output of a HELM network for a new sample with multisource features $\{\mathbf{nx}^{(1)}, \mathbf{nx}^{(2)}, \dots, \mathbf{nx}^{(d+1)}\}$. The final fusion result can be achieved by

$$\mathbf{f}(j) = \sum_{k=1}^{d+1} \left\{ \frac{\mathbf{nt}^{(k)}(j) - \min_{\mathbf{t}} \mathbf{t}^{(k)}(j)}{\max_{\mathbf{t}} \mathbf{t}^{(k)}(j) - \min_{\mathbf{t}} \mathbf{t}^{(k)}(j)} \right\} \tag{21}$$

where $\mathbf{f}(j)$ denotes the j th element of the fusion result $\mathbf{f} \in R^{1 \times m}$. $\mathbf{nt}^{(k)}(j)$ denotes the j th element of $\mathbf{nt}^{(k)}$. $\min_{\mathbf{t}} \mathbf{t}^{(k)}(j)$ and $\max_{\mathbf{t}} \mathbf{t}^{(k)}(j)$ are calculated by

$$\min_{\mathbf{t}} \mathbf{t}^{(k)}(j) = \min \{ \mathbf{t}_1^{(k)}(j), \mathbf{t}_2^{(k)}(j), \dots, \mathbf{t}_N^{(k)}(j) \} \tag{22}$$

$$\max_{\mathbf{t}} \mathbf{t}^{(k)}(j) = \max\{\mathbf{t}_1^{(k)}(j), \mathbf{t}_2^{(k)}(j), \dots, \mathbf{t}_N^{(k)}(j)\} \tag{23}$$

where $\mathbf{t}_i^{(k)}(j)$ is the j th element of the $\mathbf{t}_i^{(k)}$. $\mathbf{t}_i^{(k)}$ is the HELM network output of the training sample i for the k th attribute.

Here, the procedures for training and testing an SLFN using HELM theory are as follows:

- Step 1: Initialize the number \tilde{N} of hidden nodes; note that $\tilde{N} \leq N$.
- Step 2: Select the suitable activation function g .
- Step 3: Randomly assign the input weight matrix $\alpha^{(j)}$, $j = 1, 2, \dots, d+1$.
- Step 4: Construct the hypercomplex output matrix H of the hidden layer.
- Step 5: Calculate the hypercomplex output weight matrix \mathcal{H} .
- Step 6: Obtain the fusion result using a sum rule.

2.3 Multispectral palmprint recognition using HELM

To evaluate the performance of the proposed HELM network, we have applied it to multispectral palmprint recognition. Images captured from different spectral bands are taken as the multisource features. Fig 2 demonstrates a multispectral palmprint sample.

Before using the HELM network to classify the multispectral palmprint images, the intensity normalization process illustrated in Eq (24), must be implemented on the palmprint images to remove the global intensity influence.

$$\mathbf{I}(x, y) = \frac{\mathbf{I}(x, y) - \min_v}{\max_v - \min_v} \tag{24}$$

where $\mathbf{I}(x, y)$ denotes the pixel value of image \mathbf{I} at position (x, y) . \min_v and \max_v are the minimum and maximum, respectively, of all pixels in \mathbf{I} . Fig 3 shows the palmprint images after intensity normalization operation.

As the input features of HELM network, an image must be adjusted to a row vector. Fig 4 lists typical training data for the HELM network. $\mathbf{x}_i^{(1)}$, $\mathbf{x}_i^{(2)}$, $\mathbf{x}_i^{(3)}$ and $\mathbf{x}_i^{(4)}$ are the multisource input features, i.e., the multispectral images captured at Blue, Green, Red and Near-infrared (NIR) bands. \mathbf{t}_i denotes the hypercomplex output of the HELM network. The entry with the value of $1+e_1+e_2+e_3$ denotes the class to which sample i belongs.

With all training data, we can follow the steps described in HELM theory to train an SLFN and complete the palmprint recognition task.

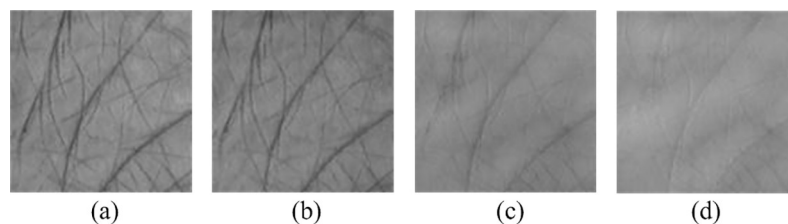


Fig 2. Multispectral palmprint sample: (a) Blue, (b) Green, (c) Red and (d) Near-infrared.

<https://doi.org/10.1371/journal.pone.0209083.g002>

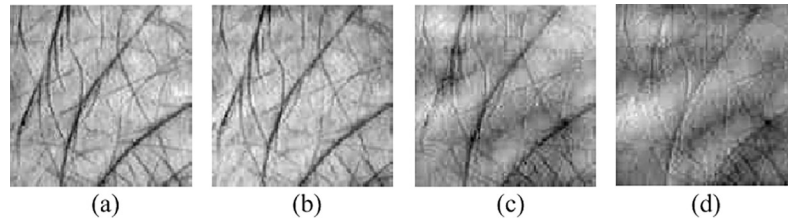


Fig 3. Multispectral palmprint sample after intensity normalization: (a) Blue, (b) Green, (c) Red and (d) Near-infrared.

<https://doi.org/10.1371/journal.pone.0209083.g003>

3 Experimental results and performance analysis

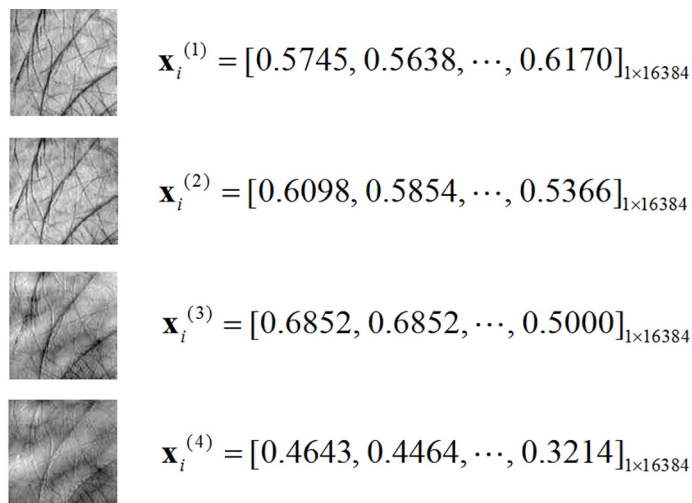
In this section, we present the experimental results and assess the performance of the proposed HELM method. All experiments have been conducted on a computer with a 2.50 GHz Intel core processor and 8 GB memory. MATLAB 2017a was utilized as the simulation software.

3.1 Database description and evaluation criteria

To demonstrate the effectiveness of the proposed method, we conducted a series of experiments using the following two public multispectral palmprint databases.

The PolyU database is [29–32] created by Hong Kong Polytechnic University. The database consists of 24000 palmprint images collected from 250 volunteers, who comprised 195 males and 55 females. The age of each volunteer ranged from 20 to 60 years old. During the acquisition process, each volunteer was sampled 12 times in two separate sessions for his/her left and right palms. The palmprint images were acquired at four spectral bands, i.e., Red, Green, Blue and NIR. For the convenience of researchers, the Hong Kong Polytechnic University provides the region of interest (ROI) images with the size 128×128. Fig 5 shows some multispectral palmprint samples in the PolyU database.

The CASIA database [33] is provided by the Chinese Academy of Sciences’ Institute of Automation. It has 7200 palmprint images in total collected from 100 volunteers. The



$$t_i = [1 + e_1 + e_2 + e_3, -1 - e_1 - e_2 - e_3, \dots, -1 - e_1 - e_2 - e_3]_{1 \times 500}$$

Fig 4. Typical input for HELM network.

<https://doi.org/10.1371/journal.pone.0209083.g004>

acquisition was performed in two separate sessions with a minimum time interval of one month. In one session, each volunteer was required to provide 3 samples for his/her left and right palm respectively. Each sample was captured at 460nm, 630nm, 700nm, 850nm, 940nm and white light (WHT) spectral bands respectively. Fig 6 shows some multispectral palmprint images in the CASIA database.

The performance of the proposed method is evaluated in terms of recognition accuracy and computational cost. In the recognition process, a certain number of multispectral palmprint images are treated as the testing samples. If the determined class label of one testing sample is the same with its actual label, it is considered as a correctly recognized sample. Otherwise, it is an incorrectly recognized one. Then the recognition accuracy is defined as

$$RA = N_c/N \quad (25)$$

where N_c is the number of the correctly recognized samples in the testing group. N is the number of the samples in the testing group.

The computational cost including the training and testing time is also used to compare the performance of the method. The training time is referred to as the time cost for constructing the HELM model using the training data. And the testing time is the time cost for determining the class labels of the testing samples using the trained HELM model.

3.2 Result analysis of the proposed method

To complete the training of the HELM network and achieve excellent generalization performance, the penalty coefficient C and the number \tilde{N} of hidden nodes need to be appropriately

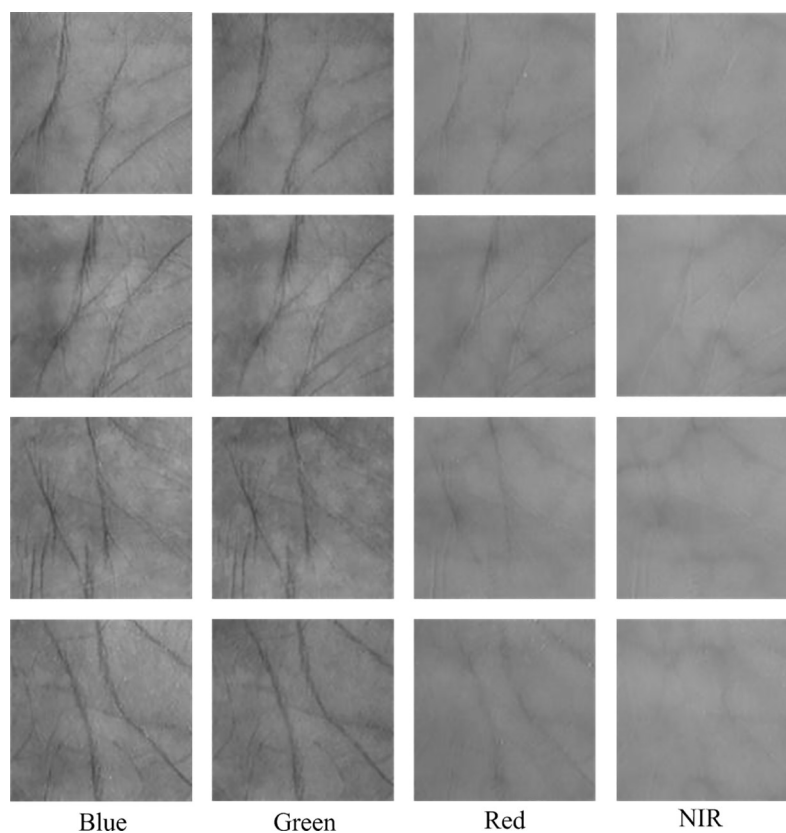


Fig 5. Multispectral palmprint samples in the PolyU database.

<https://doi.org/10.1371/journal.pone.0209083.g005>

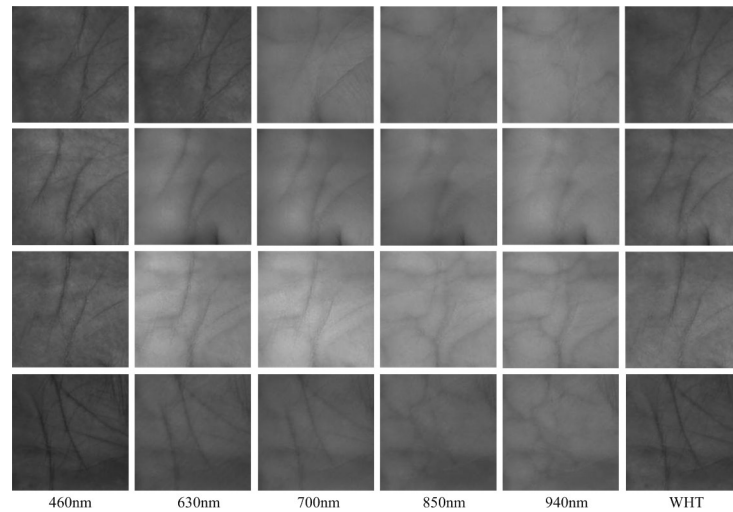


Fig 6. Multispectral palmprint samples in the CASIA database.

<https://doi.org/10.1371/journal.pone.0209083.g006>

chosen. We have employed different settings of C and \tilde{N} . The results of the experiments on the two databases are demonstrated in Fig 7. It is clear from the figure that no matter which database the experiments are conducted on, for a given penalty coefficient C the recognition accuracy has an increasing trend as the number \tilde{N} of hidden nodes progressively increases. It converges to the optimal accuracy when the value of \tilde{N} is sufficiently large. We observe that a smaller value of C yields a higher recognition accuracy. When C is zero, the best results are generated.

Considering the randomness of the HELM training method, ten repeated experiments were performed. We also tested the performance of the HELM network with different activation functions. The sigmoid, sin and atan functions were compared to determine which function can achieve the optimal result. Table 1 lists the recognition accuracies in the ten repetitions.

Based on these repeating results, Table 2 gives the statistical comparison by using the one-sided Two-sample Student T -test. It makes the hypothesis that two independent samples come

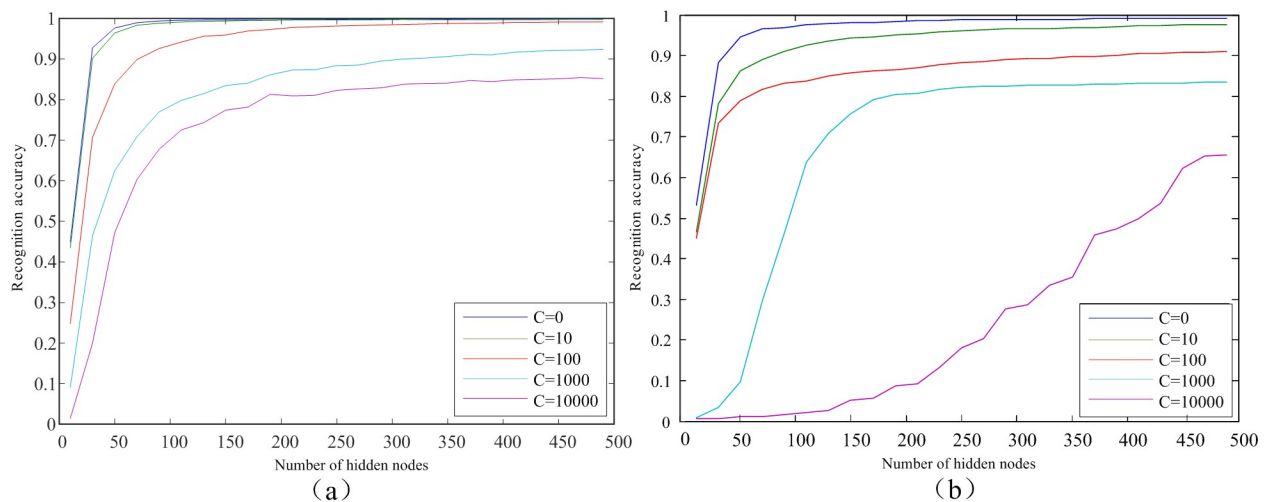


Fig 7. Performance with different settings in HELM: (a) PolyU, (b) CASIA.

<https://doi.org/10.1371/journal.pone.0209083.g007>

Table 1. Recognition accuracies of the ten repeated measurements with different activation functions.

Number of measurement	RA (%)					
	PolyU			CASIA		
	Sigmoid	Sin	Atan	Sigmoid	Sin	Atan
1	100	99.20	99.87	98.17	97.50	97.67
2	99.90	98.83	99.70	98.33	97.67	97.83
3	99.90	99.03	99.83	98.50	97.50	97.67
4	99.97	99.13	99.83	98.00	97.83	97.67
5	99.93	98.97	99.83	98.00	97.67	97.83
6	100	99.00	99.93	98.17	97.50	97.50
7	99.90	98.97	99.93	98.00	97.33	98.00
8	99.93	99.20	99.87	98.50	97.83	97.67
9	99.97	99.03	99.83	98.33	97.67	97.83
10	100	98.97	99.93	98.17	97.83	97.83

<https://doi.org/10.1371/journal.pone.0209083.t001>

from normal distributions with equal means and variances. The value of test statistic can be calculated as:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2} \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \tag{26}$$

where \bar{X}_1 and \bar{X}_2 denote the means of the two series of measurements, S_1^2 and S_2^2 denote the corresponding variances, n_1 and n_2 are the numbers of measurements in each series. The number of degrees of freedom for the Two-sample T -test is n_1+n_2-2 . To complete this statistical comparison, the built-in Matlab function of “ttest2(x, y, α , ‘right’)” is used. Here, x and y denote the two series of measurements. α is the significance level. ‘right’ denotes the right-sided test.

In Table 2, t denotes the value of the test statistic, p denotes the probability of observing the given result if the null hypothesis H_0 is true and T -test denotes the test result. The significance level α is set to be $\alpha = 0.05$. By making the comparison between the two activation functions of sigmoid and sin, it can be found that the p values for PolyU and CASIA databases are much less than the significance level α . Thus the null hypothesis H_0 is rejected and the alternative hypothesis H_1 is accepted, meaning that the sigmoid function can produce higher recognition accuracy than the sin function. Similarly, the inference that the sigmoid function outperforms the atan function can be obtained by making the T -test between the two functions of sigmoid and atan. As for the comparison between sin and atan functions, two different test results are achieved. The performance of the atan function is not significantly better than sin function for the CASIA database. It can be concluded that for the two databases, the sigmoid function

Table 2. Student T -test of the ten repeated measurements with different activation functions.

Hypothesis	PolyU			CASIA		
	t	p	T -test	t	p	T -test
$H_0: RA_{\text{Sigmoid}} \leq RA_{\text{Sin}}$ $H_1: RA_{\text{Sigmoid}} > RA_{\text{Sin}}$	23.59	2.740E-15	H_1	7.158	5.746E-7	H_1
$H_0: RA_{\text{Sigmoid}} \leq RA_{\text{Atan}}$ $H_1: RA_{\text{Sigmoid}} > RA_{\text{Atan}}$	3.666	8.845E-4	H_1	6.205	3.714E-6	H_1
$H_0: RA_{\text{Atan}} \leq RA_{\text{Sin}}$ $H_1: RA_{\text{Atan}} > RA_{\text{Sin}}$	19.30	8.902E-14	H_1	1.670	0.0561	H_0

<https://doi.org/10.1371/journal.pone.0209083.t002>

Table 3. Recognition accuracies for different combinations of spectral bands on the PolyU database.

Input spectral bands	RA (%)		
	Sigmoid	Sin	Atan
Blue	94.79±0.18	93.64±0.14	97.13±0.16
Green	93.55±0.19	92.23±0.16	96.54±0.19
Red	94.28±0.20	92.67±0.13	96.68±0.13
NIR	94.65±0.15	91.98±0.19	97.03±0.19
Blue Green	98.63±0.14	97.24±0.12	98.37±0.13
Blue Red	99.42±0.14	98.35±0.11	99.31±0.19
Blue NIR	99.66±0.10	98.94±0.12	99.58±0.10
Green Red	99.31±0.16	98.58±0.13	99.15±0.15
Green NIR	99.74±0.10	98.62±0.14	99.44±0.12
Red NIR	99.25±0.13	97.76±0.15	99.03±0.10
Blue Green Red	99.68±0.08	98.59±0.11	99.41±0.13
Blue Green NIR	99.87±0.07	99.27±0.09	99.76±0.08
Green Red NIR	99.86±0.04	98.70±0.07	99.76±0.05

<https://doi.org/10.1371/journal.pone.0209083.t003>

consistently produces the highest recognition accuracy among the three types of activation functions. In the ten repeated measurements for the sigmoid function, the best recognition accuracies for the PolyU database and CASIA database are 100% and 98.50% respectively.

To evaluate the performance of the HELM network in the case of different combinations of input spectral bands, a series of experiments were conducted using different hypercomplex representations, i.e., $y = y^{(1)} + y^{(2)}e_1$, $y = y^{(1)} + y^{(2)}e_1 + y^{(3)}e_2$, $y = y^{(1)} + y^{(2)}e_1 + y^{(3)}e_2 + y^{(4)}e_3$ or $y = y^{(1)} + y^{(2)}e_1 + y^{(3)}e_2 + y^{(4)}e_3 + y^{(5)}e_4$. In the situation with a single input spectral band, the HELM degraded into a traditional ELM network. Considering the randomness in the training of the HELM network, ten repeated runs were performed. The average accuracy and the corresponding standard deviation were employed as the assessment. We also applied three activation functions in the network. Tables 3 and 4 illustrate the experimental results of different spectral combinations when testing the HELM method on the PolyU and CASIA multispectral databases. As shown in the Table 3, the recognition accuracies of the experiments with more than

Table 4. Recognition accuracies for different combinations of spectral bands on the CASIA database.

Input spectral bands	RA (%)		
	Sigmoid	Sin	Atan
460nm	90.55±0.46	90.38±0.51	91.17±0.30
630nm	91.85±0.48	91.81±0.36	92.13±0.33
700nm	90.53±0.42	90.37±0.49	90.87±0.30
850nm	92.25±0.35	92.07±0.48	92.36±0.28
940nm	93.23±0.42	93.15±0.37	93.50±0.22
WHT	92.50±0.37	92.22±0.38	92.75±0.21
460nm 940nm	96.86±0.17	96.38±0.11	96.53±0.21
940nm WHT	95.40±0.16	95.11±0.25	94.93±0.22
460nm 850nm WHT	97.12±0.24	96.95±0.24	96.97±0.22
460nm 940nm WHT	97.58±0.23	97.52±0.12	97.55±0.21
460nm 700nm 850nm 940nm	97.88±0.14	97.33±0.21	97.38±0.22
460nm 850nm 940nm WHT	97.93±0.16	97.68±0.16	97.73±0.22
460nm 630nm 700nm 850nm 940nm	97.90±0.11	97.30±0.11	97.68±0.17
460nm 630nm 850nm 940nm WHT	98.03±0.12	97.89±0.14	97.93±0.15

<https://doi.org/10.1371/journal.pone.0209083.t004>

one input spectral band are higher than the recognition accuracies of the experiments with a single spectral band. The recognition results based on the HELM network are satisfied when the number of input spectral bands is two or three, which has proven that the proposed HELM network is applicable for any spectral band combination. We also observe that the sigmoid function achieves the optimal results among the three activation functions when the number of input spectral bands is two or three. Regarding the case of a single spectral band, the Atan function obtains the best results for the ELM model. Similarly, we can also obtain these conclusions from Table 4. For the CASIA database, using the proposed HELM model with multispectral palmprint images can obviously improve the performance of palmprint recognition. The sigmoid activation function provides the optimal recognition accuracies for these experiments with more than one input spectral band.

HELM employs a hypercomplex representation to complete the classification task of multi-source features. To verify its effectiveness, a comparison was performed with two different strategies for ELM to process the multisource features, i.e., fusing the multisource features either at a feature level or a matching score level. These methods were compared in terms of computational cost and recognition accuracy. Similarly, ten repeated runs were performed. The time and accuracy were employed in the assessment. As reported in Table 5, we conclude that for either benchmark database, the hypercomplex representation-based method obtains the highest recognition accuracy and maintains a distinct advantage over the other two strategies. Regarding computation time, although the hypercomplex representation based-method cannot compete with the feature level fusion based method in terms of testing time, it requires the lowest training time and provides the highest recognition accuracy. The hypercomplex representation strategy outperforms both of the comparison methods.

A comparison was made with some state-of-art multispectral palmprint recognition methods, including two image level fusion methods, two matching score level fusion methods, a QPCA+QDWT method and two improved ELM-based methods. In addition, we also investigated the performance of the HELM model when using different features as the input. A dimensionality reduction method and a texture feature extraction method—PCA [34] and LBP [35]—were employed to extract the palmprint features. The experiments were conducted on the pure PolyU and CASIA databases as well as the corresponding manually generated ones by introducing different kinds of noises. Fig 8 demonstrates the manually generated palmprint samples used in these experiments. The Gaussian white noise with mean 0 and standard deviation 36, the Salt & Pepper noise with 10% noise density and the Speckle noise with variance 0.05 were utilized respectively to generate the noisy palmprint images. Table 6 lists the recognition accuracies of the comparison methods. We can discover that the HELM-based and PCA+HELM-based multispectral palmprint recognition methods consistently outperform the fusion-related methods and the QPCA+QDWT method on the testing databases. Although the two improved ELM-based methods could achieve quite satisfactory results on the PolyU database, the performance degrades when they are tested on the CASIA database. The LBP+HELM method could produce the highest recognition accuracies on the pure PolyU and

Table 5. Comparison with different strategies of processing the multisource features of ELM.

Method	PolyU			CASIA		
	Training Time (s)	Testing Time (s)	RA (%)	Training Time (s)	Testing Time (s)	RA (%)
Feature level fusion	3.95±0.11	0.65±0.09	97.13±0.14	2.14±0.11	0.21±0.04	95.40±0.17
Matching score level fusion	7.52±0.21	1.63±0.08	92.74±0.12	3.26±0.12	0.31±0.08	90.56±0.45
Hypercomplex representation	3.42±0.15	1.68±0.09	99.95±0.04	1.34±0.11	0.19±0.05	98.22±0.19

<https://doi.org/10.1371/journal.pone.0209083.t005>

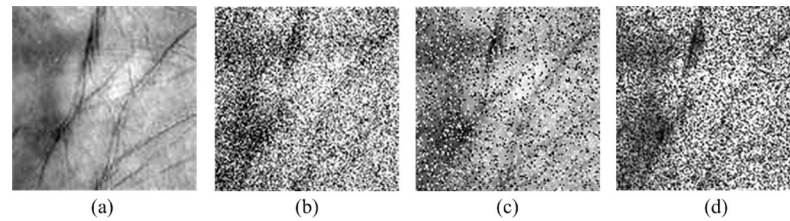


Fig 8. Pure and noisy palmprint samples: (a) Pure image, (b) Image with Gaussian white noise, (c) Image with Salt & Pepper noise, (d) Image with Speckle noise.

<https://doi.org/10.1371/journal.pone.0209083.g008>

CASIA databases (100% and 99.83). However, the recognition accuracies decrease seriously when the images are corrupted with noises. This is because the LBP features depend on the local structure of images and are very sensitive to the variation of pixel value.

Table 7 gives the statistical comparison of the ten methods in Table 6 by using the one-sided Two-sample Student *T*-test. The significance level is set to be $\alpha = 0.05$. Here, the meanings of *t*, *p* and *T*-test are same with those in Table 2 and the value of *t* is calculated as shown in Eq (26). IPCA denotes the method of “Image level fusion by PCA”. MPCA denotes the method of “Matching score level fusion by PCA”. IDWT denotes the method of “Image level fusion by DWT”. MDWT denotes the method of “Matching score level fusion by DWT”. QPCA denotes the method of “QPCA+QDWT”. By making the comparison between HELM method (Or PCA+HELM method) with the fusion-related methods, the QPCA+QDWT method and the improved ELM-based methods, we can find that the values of *p* are obviously less than the significance level α . Therefore the alternative hypothesis H_1 is accepted. That is to say, the HELM method (Or PCA+HELM method) could achieve higher recognition accuracies than the comparison methods from a statistical viewpoint. In addition, it is observed that the LBP+HELM method is not significantly better than the comparison methods due to the noise effect. As for the Student *T*-test between the three HELM-related methods, the test results show that the PCA+HELM method can produce the highest recognition accuracies.

4 Conclusions

In this paper, we have proposed HELM, which is a novel learning method for SLFNs. HELM introduces the hypercomplex representation concept into ELM theory. In contrast to the

Table 6. Recognition accuracy comparison with different multispectral palmprint recognition methods.

Method	RA (%)							
	PolyU				CASIA			
	Pure	Gaussian	Salt & Pepper	Speckle	Pure	Gaussian	Salt & Pepper	Speckle
Image level fusion by PCA [36]	95.53	94.03	94.40	94.23	93.67	92.33	93.33	92.17
Matching score level fusion by PCA [37]	97.97	96.83	97.07	96.67	96.50	96.16	96.33	95.83
Image level fusion by DWT [32]	96.60	71.20	61.07	71.87	94.00	65.67	58.50	65.33
Matching score level fusion by DWT [38]	98.00	97.46	97.37	96.97	96.67	96.16	95.83	95.50
QPCA+QDWT [39]	98.83	98.60	97.90	97.90	96.67	96.33	96.17	95.67
MPELM [25]	99.53	98.80	98.87	98.63	90.00	84.67	88.00	85.00
TELM [24]	99.95	99.00	99.17	99.10	96.83	94.83	96.00	93.50
HELM	100	99.80	99.70	99.67	98.50	98.00	97.83	97.67
PCA+HELM	100	99.93	99.83	99.90	98.50	98.17	98.00	97.83
LBP+HELM	100	93.70	91.90	90.43	99.83	93.17	90.83	89.50

<https://doi.org/10.1371/journal.pone.0209083.t006>

Table 7. Student *T*-test between different multispectral palmprint recognition methods.

Hypothesis		<i>t</i>	<i>p</i>	<i>T</i> -test
$H_0: RA_{HELM} \leq RA_{IPCA}$	$H_1: RA_{HELM} > RA_{IPCA}$	9.867	5.512E-8	H_1
$H_0: RA_{HELM} \leq RA_{MPCA}$	$H_1: RA_{HELM} > RA_{MPCA}$	5.304	5.570E-5	H_1
$H_0: RA_{HELM} \leq RA_{IDWT}$	$H_1: RA_{HELM} > RA_{IDWT}$	5.041	9.016E-5	H_1
$H_0: RA_{HELM} \leq RA_{MDWT}$	$H_1: RA_{HELM} > RA_{MDWT}$	4.620	1.986E-4	H_1
$H_0: RA_{HELM} \leq RA_{QPCA}$	$H_1: RA_{HELM} > RA_{QPCA}$	2.982	4.952E-3	H_1
$H_0: RA_{HELM} \leq RA_{MPELM}$	$H_1: RA_{HELM} > RA_{MPELM}$	2.506	1.259E-2	H_1
$H_0: RA_{HELM} \leq RA_{TELM}$	$H_1: RA_{HELM} > RA_{TELM}$	1.764	4.974E-2	H_1
$H_0: RA_{PCA+HELM} \leq RA_{IPCA}$	$H_1: RA_{PCA+HELM} > RA_{IPCA}$	10.17	3.777E-8	H_1
$H_0: RA_{PCA+HELM} \leq RA_{MPCA}$	$H_1: RA_{PCA+HELM} > RA_{MPCA}$	5.660	2.944E-5	H_1
$H_0: RA_{PCA+HELM} \leq RA_{IDWT}$	$H_1: RA_{PCA+HELM} > RA_{IDWT}$	5.065	8.618E-5	H_1
$H_0: RA_{PCA+HELM} \leq RA_{MDWT}$	$H_1: RA_{PCA+HELM} > RA_{MDWT}$	4.929	1.110E-4	H_1
$H_0: RA_{PCA+HELM} \leq RA_{QPCA}$	$H_1: RA_{PCA+HELM} > RA_{QPCA}$	3.227	3.040E-3	H_1
$H_0: RA_{PCA+HELM} \leq RA_{MPELM}$	$H_1: RA_{PCA+HELM} > RA_{MPELM}$	2.559	1.137E-2	H_1
$H_0: RA_{PCA+HELM} \leq RA_{TELM}$	$H_1: RA_{PCA+HELM} > RA_{TELM}$	1.905	3.874E-2	H_1
$H_0: RA_{LBP+HELM} \leq RA_{IPCA}$	$H_1: RA_{LBP+HELM} > RA_{IPCA}$	-0.028	5.118E-1	H_0
$H_0: RA_{LBP+HELM} \leq RA_{MPCA}$	$H_1: RA_{LBP+HELM} > RA_{MPCA}$	-2.047	9.700E-1	H_0
$H_0: RA_{LBP+HELM} \leq RA_{IDWT}$	$H_1: RA_{LBP+HELM} > RA_{IDWT}$	3.880	8.336E-4	H_1
$H_0: RA_{LBP+HELM} \leq RA_{MDWT}$	$H_1: RA_{LBP+HELM} > RA_{MDWT}$	-2.078	9.717E-1	H_0
$H_0: RA_{LBP+HELM} \leq RA_{QPCA}$	$H_1: RA_{LBP+HELM} > RA_{QPCA}$	-2.380	9.840E-1	H_0
$H_0: RA_{LBP+HELM} \leq RA_{MPELM}$	$H_1: RA_{LBP+HELM} > RA_{MPELM}$	0.265	3.974E-1	H_0
$H_0: RA_{LBP+HELM} \leq RA_{TELM}$	$H_1: RA_{LBP+HELM} > RA_{TELM}$	-2.170	9.762E-1	H_0
$H_0: RA_{HELM} \leq RA_{PCA+HELM}$	$H_1: RA_{HELM} > RA_{PCA+HELM}$	-2.516	5.975E-1	H_0
$H_0: RA_{PCA+HELM} \leq RA_{LBP+HELM}$	$H_1: RA_{PCA+HELM} > RA_{LBP+HELM}$	3.595	1.464E-3	H_1
$H_0: RA_{HELM} \leq RA_{LBP+HELM}$	$H_1: RA_{HELM} > RA_{LBP+HELM}$	3.509	1.737E-3	H_1

<https://doi.org/10.1371/journal.pone.0209083.t007>

conventional ELM model, the proposed method maintains all merits of ELM, such as fast learning speed, excellent generalization ability and ease of implementation. HELM can easily complete the classification task of multisource features by benefitting from the hypercomplex representation. We have applied this method to the task of multispectral palmprint recognition to verify the actual performance. Comprehensive experiments carried out on the PolyU and CASIA multispectral palmprint databases have demonstrated that the proposed HELM network can obtain favorable results compared with several state-of-the-art multispectral palmprint recognition methods.

Acknowledgments

The authors would like to thank the Hong Kong Polytechnic University and the Chinese Academy of Sciences' Institute of Automation for sharing the multispectral palmprint databases.

Author Contributions

Formal analysis: Xuebin Xu.

Funding acquisition: Xinman Zhang.

Investigation: Xinman Zhang.

Methodology: Longbin Lu.

Project administration: Xinman Zhang.

Software: Longbin Lu.

Validation: Xinman Zhang, Xuebin Xu.

Writing – original draft: Longbin Lu.

Writing – review & editing: Xuebin Xu.

References

1. Zhou Y, Zhang L, Yi Z. Predicting movie box-office revenues using deep neural networks. *Neural Computing & Applications*. 2017; (3):1–11.
2. Meraoumia A., Kadri F., Bendjenna H., Chitroub S., Bouridane A. *Improving Biometric Identification Performance Using PCANet Deep Learning and Multispectral Palmprint*. Biometric Security and Privacy. 2017, Springer International Publishing.
3. Huang GB. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*. 2003; 14(2): 274–281. <https://doi.org/10.1109/TNN.2003.809401> PMID: 18238011
4. Huang GB, Zhu QY, Siew CK. Extreme learning machine: Theory and applications. *Neurocomputing*. 2006; 70(1):489–501.
5. Widrow B, Lehr MA. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*. 2002; 78(9):1415–1442.
6. Hecht-Nielsen R. *Theory of the backpropagation neural network*. *Neural networks for perception*. Harcourt Brace & Co.; 1992.
7. Van Ooyen A, Nienhuis B. Improving the convergence of the back-propagation algorithm. *Neural Networks*. 1992; 5(3):465–471.
8. Huang GB, Ding X, Zhou H. Optimization method based extreme learning machine for classification. *Neurocomputing*. 2010; 74(1–3):155–163.
9. Yang Y, Wang Y, Yuan X. Bidirectional extreme learning machine for regression problem and its learning effectiveness. *IEEE Transactions on Neural Networks and Learning Systems*. 2012; 23(9):1498–1505. <https://doi.org/10.1109/TNNLS.2012.2202289> PMID: 24807932
10. He Q, Jin X, Du C, Zhuang F, Shi Z. Clustering in extreme learning machine feature space. *Neurocomputing*. 2014; 128(5):88–95.
11. Lam D, Wunsch D. Unsupervised feature learning classification with radial basis function extreme learning machine using graphic processors. *IEEE Transactions on Cybernetics*. 2017; 47(1):224. <https://doi.org/10.1109/TCYB.2015.2511149> PMID: 26742156
12. Huang GB, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics*. 2012; 42(2):513–529.
13. Deng WY, Zheng QH, Chen L. Regularized Extreme Learning Machine. In *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Nashville, USA, March 30-April 2, 2009, pp.389–395.
14. Liang NY, Huang GB, Saratchandran P, Sundararajan N. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*. 2006; 17(6):1411–1423. <https://doi.org/10.1109/TNN.2006.880583> PMID: 17131657
15. Man Z, Lee K, Wang D, Cao Z, Khoo S. Robust single-hidden layer feedforward network-based pattern classifier. *IEEE Transactions on Neural Networks & Learning Systems*. 2012; 23(12):1974.
16. Zhou H, Huang GB, Lin Z, Wang H, Soh YC. Stacked extreme learning machines. *IEEE Transactions on Cybernetics*. 2015; 45(9):2013. <https://doi.org/10.1109/TCYB.2014.2363492> PMID: 25361517
17. Tang J, Deng C, Huang GB. Extreme learning machine for multilayer perceptron. *IEEE Transactions on Neural Networks & Learning Systems*. 2016; 27(4):809.
18. Tang J, Deng C, Huang GB, Hou J. A fast learning algorithm for multi-layer extreme learning machine. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Quebec, Canada, 27–30 September 2015, pp. 175–178.
19. Song JL, Hu W, Zhang R. Automated detection of epileptic EEGs using a novel fusion feature and extreme learning machine. *Neurocomputing*. 2016; 175(PA):383–391.
20. Chen Y, Yang J, Wang C, Liu N. Multimodal biometrics recognition based on local fusion visual features and variational bayesian extreme learning machine. *Expert Systems with Applications*. 2016; 64(C):93–103.

21. Li W, Chen C, Su H, Du Q. Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*. 2015; 53(7):3681–3693.
22. Fister I, Yang XS, Brest J, Fister I Jr. Modified firefly algorithm using quaternion representation. *Expert Systems with Applications*. 2013; 40(18):7220–7230.
23. Ell TA, Sangwine SJ. Hypercomplex Fourier transforms of color images. *IEEE Transactions on image processing*. 2007; 16(1):22–35. PMID: [17283762](#)
24. Lu LB, Zhang XM, Xu XB, Shang DP. Multispectral image fusion for illumination-invariant palmprint recognition. *PLOS ONE*. 2017, 12(5):e0178432. <https://doi.org/10.1371/journal.pone.0178432> PMID: [28558064](#)
25. Xu XB, Lu LB, Zhang XM, Lu HM, Deng WY. Multispectral palmprint recognition using multiclass projection extreme learning machine and digital shearlet transform. *Neural Computing and Applications*. 2016, 27(1):143–153.
26. Gumaie A, Sammouda R, Al-Salman AM, Alsanad A. An improved multispectral palmprint recognition system using autoencoder with regularized extreme learning machine. *Computational Intelligence and Neuroscience*. 2018.
27. Gumaie A, Sammouda R, Al-Salman AM, Alsanad A. An effective palmprint recognition approach for visible and multispectral sensor images. *Sensors*. 2018, 18(5):1575.
28. Gumaie A, Sammouda R, Al-Salman AM, Alsanad A. Anti-spoofing cloud-based multi-spectral biometric identification system for enterprise security and privacy-preservation. *Journal of Parallel and Distributed Computing*. 2018. <https://doi.org/10.1016/j.jpdc.2018.07.011>
29. Zhang D, Guo Z, Gong Y. An online system of multispectral palmprint verification. *IEEE Transactions on Instrumentation and Measurement*. 2010; 59(2):480–490.
30. Zhang D, Guo Z, Lu G, Zhang L, Liu Y, Zuo W, et al. Online joint palmprint and palmvein verification. *Expert Systems with Applications*. 2011; 38(3):2621–2631.
31. Zhang D, Guo Z, Gong Y. Empirical study of light source selection for palmprint recognition. *Pattern Recognition Letters*. 2010; 32(2):120–126.
32. Han D, Guo Z, Zhang D. Multispectral palmprint recognition using wavelet-based image fusion. In *Proceedings of the IEEE International Conference on Signal Processing (ICSP)*, Beijing, China, 26–29 October 2008, pp. 2074–2077.
33. CASIA-MS-PalmprintV1, <http://biometrics.idealtest.org/>.
34. Abdi H, Williams LJ. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*. 2010; 2(4), 433–459.
35. Zhang B, Gao Y, Zhao S, Liu J. Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE transactions on image processing*. 2010; 19(2), 533–544. <https://doi.org/10.1109/TIP.2009.2035882> PMID: [19887313](#)
36. Sun JF, Jiang YJ, Zeng SY. A study of PCA image fusion techniques on remote sensing. *Proceedings of SPIE*. 2005; 5985:739–744.
37. Zhu Q, Xu Y. Multi-directional two-dimensional pca with matching score level fusion for face recognition. *Neural Computing & Applications*. 2013; 23(1):169–174.
38. Koike S, Nakanishi I, Itoh Y. Score Level Fusion in DWT Domain On-Line Signature Verification. In *Proceedings of the International Technical Conference on Circuits Systems, Computers and Communications (ITC-CSCC)*, Shimonoseki City, Japan, 6–9 July 2008, pp.395–398.
39. Xu X, Guo Z, Song C, Li Y. Multispectral palmprint recognition using a quaternion matrix. *Sensors*. 2012; 12(4):4633–47. <https://doi.org/10.3390/s120404633> PMID: [22666049](#)