A Collaborative-competitive Representation-based Broad Learning System for Image Classification

Guangheng Wu
College of Information Science and Technology
Jinan University
Guangzhou, China
wuguangheng2002@gmail.com

***Abstract*—Broad Learning System (BLS) has been viewed as an alternative network paradigm in the fields of classification and regression.Compared with deep neural network (DNN), BLS can achieve a competitive performance but with faster training speed. There are two essential parts in BLS: the mapping features and the enhancement features. The mapping features are first randomly initialized and then fine-tuned by sparse autoencoder. However, sparse autoencoder just considers constructing good representations to reconstruct the original data, ignoring the relationship between features, which can not guarantee good enough representation of features. Inspired by the effectiveness of collaborative-competitive representation (CCR) mechanism in face recognition, we first propose a novel collaborative-competitive representation based autoencoder (CCRAE), and then a CCRAE-based broad learning system (CCR-BLS) is proposed to handle the disadvantage mentioned above. We evaluate the performance of our proposed model with several datasets in image classification tasks, and the experimental results shows that our proposed model can achieve superior or comparable performance in comparison with other competing methods in image classification tasks.**

***Index Terms*—broad learning system (BLS), collaborative-competitive representation (CCR), aotoencoder (AE), image classification.**

I. Introduction

THE success of machine learning methods in supervised learning tasks such as classification, is usually dependent on the choice of data representation on which they are applied. And feature representation is an efficient way of data representation that captures good characteristics of the data. The most common feature representation methods are principal components analysis (PCA), autoencoder (AE) [1].

As an unsupervised method, AE has become increasingly popular for extracting features in the past few years. AE consists of two parts: an encoder and a decoder. AE first learns the representation of the input data with its encoder, and then reconstructs the input data from the representation learned before with its decoder part. The reconstruction error must be minimized so as to guarantee the effectiveness of the learned representations, which means that the output of AE is as identical as possible to the original input data. Different flexible design of the AE's cost function leads to a variety of meaningful representations, for example, the sparse autoencoder (SAE) [2] adopts a l_1-norm penalty on the connecting weights while the extreme learning machine based autoencoder (ELMAE) [3] penalizes with l_2-norm.

In the context of representation based classification methods (RBCM), SAE can be viewed as sparse representation (SR) based method while ELMAE can named as collaborative representation (CR) based method for feature representation. CR replaces the l_1-norm in SR with l_2-norm, and it has been revealed that it is the collaborative representation mechanism that makes SR successful for classification, rather then the l_1-norm sparsity. What's more, [4] proposed novel collaborative-competitive representation (CCR) based classifier model, which incorporates a regularization constraint term into the objective function of CR.

The broad learning system designed by Chen and Liu [5] is presented as a flatted and incremental learning neural network. In BLS, the original inputs are first transformed to the mapping features, then the enhancement features are obtained by transforming the mapping features further. Compared with DNN, BLS has advantages of faster training speed, competitive performance and universal approximation ability. And BLS has been applied to many fields. For example, For example, Du et al. [6] proposed a novel recurrent BLS (Recurrent-BLS) and further expanded it to LSTM-BLS by adding corresponding gatings (Gated-BLS) for sequential data. Chu et al. [7] proposed a novel weighted BLS to deal with the problem of outliers in sample data. And Jin et al. [8] incorporated the manifold learning into the objective function of the standard BLS, called graph regularized BLS (GBLS), which is under the assumption that similar images data may share similar properties. However, we makes no guarantee that the concatenation of mapping features and the enhancement features can provide ideal feature representation in BLS. There are mainly two way attempting to address the problem. The first one is to extend BLS to deep architecture. The second one is to change the transforming ways of mapping functions in the mapping layer and the enhancement layer. For example, Tang et al. [9] designed a mapping function with the use of a label-based autoencoder with two-layer encoder structure. The mapping features and the enhancement features are transformed with this nonlinear function.

In this article, we first propose an autoencoder based on collaborative-competitive representation mechanism, termed as CCRAE, and then a novel collaborative-competitive representation based broad learning system (CCR-BLS) is proposed by integrating CCRAE into BLS. In CCR-BLS, the initial weights can be calculated directly with the regard of the

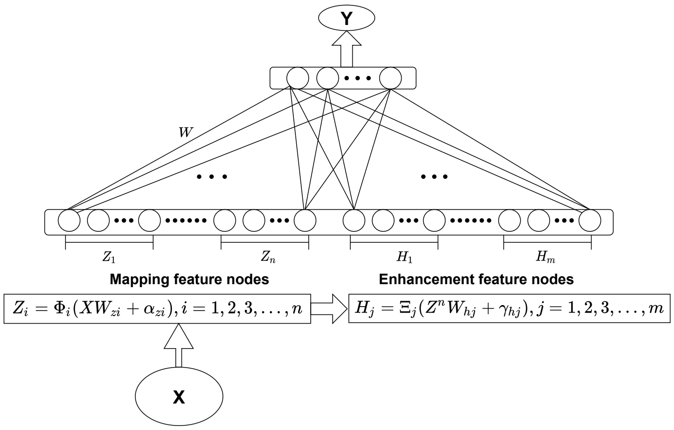

Fig. 1: Classic Broad Learning System

collaborative or competitive relationship between features. The remainder of this article is organized as follows. In Section II, we first briefly introduce SAE and ELMAE, then review the classic broad learning system. In Section III, we introduce the proposed CCR-BLS and give the details of our proposed algorithm. In Section IV, we evaluate the performance of our proposed model with a series of experiment and show the experimental results. Conclusion comes in Section V.

II. Related Works

A. Broad Learning System

Over the past few years, broad learning system (BLS) has raised a revolution in conventional artificial intelligence methods. With the advantages of simple structure and incremental learning algorithm, broad learning system can achieve faster training speed and higher accuracy, which can be extended to other research fields [10], [11]. BLS mainly consists of three important parts: 1) the mapping feature nodes; 2) the enhancement feature nodes; and 3) the output matrix. And there are two stages in the training of BLS: 1) generating the weight for the mapping feature nodes and the enhancement feature nodes randomly; 2) computing the weights. The structure of classic BLS is shown in Fig.1.

Suppose the training data of a supervised task has the form of $\{(X, Y)|X \in \mathbb{R}^{N\times M}, Y \in \mathbb{R}^{N\times K}\}$ from K classes, where N notes the sample size, M notes the feature dimension. For mapping feature node, it contains n groups, each consisting of k nodes, for enhancement feature node, it contains m groups and there are q nodes in a group. In this way, the ith group of mapping feature nodes can be expressed as

$$Z_i = \Phi_i(XW_{zi} + \alpha_{zi}), i = 1, 2, 3, ..., n, \tag{1}$$

where weighted matrix W_{zi} and deviation α_{zi} are randomly initialized. Next, collecting these mapping features together and denote it as $Z^n = [Z_1, Z_2, ..., Z_n]$. With the help of Z^n, BLS will expand the enhancement layer as the equation

$$H_j = \Xi_j(Z^n W_{hj} + \gamma_{hj}), j = 1, 2, 3, ..., m, \tag{2}$$

where $\Xi_j(\cdot)$ is the nonlinear activation function. The output of the enhancement feature nodes can be expressed as $H^m = [H_1, H_2, ..., H_m]$. Concatenating the mapping nodes and enhancement nodes together, the final form of BLS could be written as the equation

$$\begin{aligned} Y &= [Z_1, ..., Z_n|H_1, ..., H_m]W \\ &= [Z^n|H^m]W. \end{aligned} \tag{3}$$

Then in the second stage, pseudoinverse algorithm can be used to obtain the weights. In this case, the output weights of the final BLS can be computed as

$$W = [Z^n|H^m]^+Y. \tag{4}$$

However, it may be too costly to compute the generalized inverse $[Z^n|H^m]^+$ with some standard methods as the dimension may be too large in the training data. Instead, we can solve the pseudoinverse in this way:

$$arg \ \min_W \ ||BW - Y||_a^{\sigma_1} + \lambda||W||_b^{\sigma_2} \tag{5}$$

where $\sigma_1 > 0$ and $\sigma_2 > 0$, and both a and b are norm regularization. The constraint term λ is used to ensure that we can find the pseudoinverse even the generalized inverse is under ill condition. Let $B = [Z^n|H^m], \sigma_1 = \sigma_2 = a = b = 2$, the weights can be approximated as

$$W = (\lambda I + BB^T)^{-1}B^TY. \tag{6}$$

Specially, B^+ can computed as

$$B^+ = \lim_{\lambda\to 0}(\lambda I + BB^T)^{-1}B^T. \tag{7}$$

In short, BLS adopts a flat structure, which makes it easy to train and extend. And the applications of BLS are summarized in [10]. From [10], we can get that BLS can achieve a more generalized performance compared to other conventional methods.

B. Sparse Autoencoder

Since BLS generates the mapping features by randomly initialized the connecting weights, in order to overcome the randomness nature, a sparse autoencoder is adopted to fine-tune the random features and give a sparse representation.

As we can see, the random features Z are generated as the equation: $Z = XW$, where W is randomly initialized. And the loss function of SAE can be defined as

$$Loss_{SAE} = ||Z\hat{W} - X||_2^2 + \lambda||\hat{W}||_1, \tag{8}$$

where $\hat{W}$ is the sparse autoencoder solution, $\lambda > 0$ is a predefined parameter. We can minimize the loss of SAE by dozens of ways, such as orthogonal matching pursuit [], alternating direction method of multipliers (ADMM) [], and fast iterative shrinkage-thresholding algorithm (FISTA) []. What's more, by replacing the penalty with l_2-norm we can derive the loss function of ELMAE, which can be denoted as

$$Loss_{ELMAE} = ||Z\hat{W} - X||_2^2 + \lambda||\hat{W}||_2^2. \tag{9}$$

It is much easier to compute $\bar{W}$ just by taking the derivative operation, and the solution is termed as

$$\hat{W} = (Z^TZ + \lambda I)^{-1}Z^TX, \tag{10}$$

where the matrix $(Z^TZ + \lambda I)$ is generally nonsingular.

III. Proposed Method

The key points of BLS's mapping features is: the randomly generated weights W_{zi} is fine-tuned by sparse autoencoder [2], which is a linear transformation with l_1-norm. But sparse autoencoder just considers constructing good representations to reconstruct the original data, ignoring the relationship between features, which can not guarantee good enough representation of features. Furthermore, [12] uncovered that it is the CR mechanism, but not the l_1-norm sparsity that makes SR successful for classification. According to the discussion above, we can find that the original BLS may not be able to sufficiently learn the useful representations of original data. In this section, we first propose a CCRAE algorithm to better generate the features. Then, we propose CCR-BLS algorithm based on CCRAE.

A. Collaborative-competitive Representation based Autoencoder (CCRAE)

Inspired by the effectiveness of collaborative-competitive representation (CCR) mechanism, in this paper, we propose a CCR based autoencoder to explore the relationship among features and expect a better feature representation. Assumed that the origin data has a form of $X \in R^{m \times n}$ and $Z \in R^{m \times k}$ denotes the randomly generated feature. As the following, the objective function of CCRAE is formulated as:

$$\min_{\hat{W}} \ ||Z\hat{W} - X||_F^2 + \lambda_1 ||\hat{W}||_F^2 + \lambda_2 \sum_{i=1}^{k} ||X - \bar{Z}_i \hat{W}||_F^2 \quad (11)$$

where $|| \cdot ||_F$ represents the Frobenius norm, and $\bar{Z}_i = [0, ..., 0, Z_i, ..., 0] \in R^{m \times k}$, where the location of Z_i in $\bar{Z}_i$ is the same as Z. The first term $||Z\hat{W} - X||_F^2$ is to use all features to collaboratively represent the origin data, and the last term $\sum_{i=1}^{k} ||X - \bar{Z}_i \hat{W}||_F^2$ is to promote each type of feature to competitively reconstruct the origin data. λ_2 is a regularization parameter to balance the collaborative and competitive representation. If λ_2 is set to 0, CCRAE reduces to ELMAE, which is the collaborative representation. As the increasing of the parameter λ_2, it steadily enhances the effect of competitive representation.

We can get a closed-form solution of Eq.11 through computing the partial derivative about $\hat{W}$, which is formulated as:

$$\hat{W} = PX \quad (12)$$

where $P = (1 + \lambda_2)(Z^T Z + \lambda_1 I + \lambda_2 M)^{-1} Z^T$, where M is defined as:

$$M = \begin{bmatrix} Z_1^T Z & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Z_k^T Z \end{bmatrix} \quad (13)$$

After obtaining $\hat{W}$, the refined feature representation is decided as:

$$Z = X\hat{W}^T \quad (14)$$

Note that the proposed CCRAE is different from ELMAE. ELMAE utilizes the collaborative ability of all features, but CCRAE also utilizes the competitive ability of each feature, which promotes features to competitively reconstruct the origin data. And this leads to a great extraction of features. The proposed GBEAE algorithm is summarized as Algorithm 1.

Algorithm 1: Algorithm of CCRAE

Input : origin data $X \in R^{m \times n}$;
randomly generated feature $Z \in R^{m \times k}$;
the predefined parameters λ_1 and λ_2.

Output: the connecting weights $\hat{W}^T$.

Initialize the connecting weights $\hat{W}$ randomly;
Calculate M by Eq.13;
Calculate $P = (1 + \lambda_2)(Z^T Z + \lambda_1 I + \lambda_2 M)^{-1} Z^T$ with λ_1 and λ_2;
Calculate $\hat{W}$ by Eq.12;
Return the desired weights $\hat{W}^T$.

B. Collaborative-competitive Representation based Broad Learning System (CCR-BLS)

Then based on CCRAE, we proposed a new network structure termed as CCR-BLS. The structure of CCR-BLS is given in Fig.2. As shown in the figure, we first adopt CCRAE to generate the mapping features. So the mapping features in CCR-BLS are rewritten as:

$$Z_i = \Phi_i(X\hat{W}_{zi}^T), i = 1, 2, 3, ..., n, \quad (15)$$

where $\hat{W}_{zi}^T$ is obtained by CCREAE algorithm and $\Phi_i(\cdot)$ is a nonlinear function. Collecting these mapping features together and denoting it as $Z^n = [Z_1, Z_2, ..., Z_n]$, Similar to the original BLS, the jth group of enhancement nodes are formulated as:

$$H_j = \Xi_j(Z^n W_{hj}), j = 1, 2, 3, ..., m, \quad (16)$$

where W_{hj}^T is generated randomly. H_j is the high-level features of Z^n, and we collect all the enhancement nodes as $H^m = [H_1, H_2, ..., H_m]$. However, we can't ensure the quality of feature representation of H_j because the connecting weight W_{hj} is generated randomly. Therefore, we also adopt competitive representation mechanism to fine tune W_{hj}. We denote the transformed features as E, where $E = H^m W_{H^m}$. W_H can be calculated as the equation:

$$W_{H^m} = arg \ \min_{W_{H^m}} \ \sum_{i=1}^{k} ||\bar{E}_i W_{H^m} - H^m||_F^2. \quad (17)$$

where $\bar{E}_i = [0, ..., 0, E_i, ..., 0]$, where the location of E_i in $\bar{E}_i$ is the same as E. The above equation can be solved by taking the derivation operation and the solution can be written as:

$$W_{H^m} = M_{H^m}^{-1} E^T H^m \quad (18)$$

where M_H has a similar form as Eq.13:

$$M_{H^m} = \begin{bmatrix} E_1^T E & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & E_k^T E \end{bmatrix} \quad (19)$$

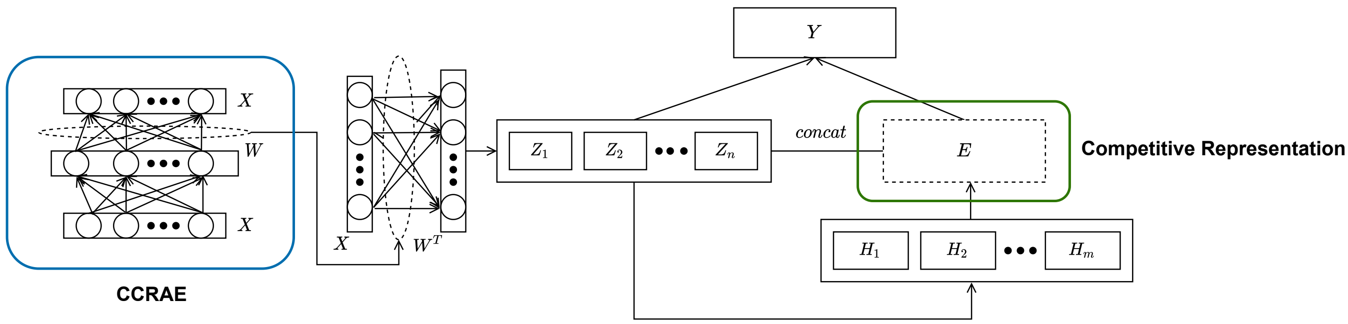

Fig. 2: The network structure of CCR-BLS

We concatenate the mapping nodes and enhancement nodes together, the final form of BLS could be written as the equation

$$Y = [Z^n | Z^m W_{H^m}] W. \tag{20}$$

Let $L = [Z^n | Z^m W_{H^m}]$, just as the origin BLS, the weights of linear model can be computed as:

$$W = (\lambda I + L L^T)^{-1} L^T Y \tag{21}$$

where λ is a hype-parameter. The algorithm of proposed CCR-BLS is summarized in Algorithm 2.

Algorithm 2: Algorithm of CCR-BLS

Input : training data X, Y;
number of groups of mapping nodes n;
number of groups of enhancement nodes m;
the predefined parameters λ, λ_1 and λ_2.
Output: the predicted label Y.

```
for i ← 1; i ≤ n do
    Calculate Ŵ_zi^T according to Algorithm1 with
      parameters λ, λ1 and λ2;
    Calculate Z_i with Eq.15;
Get the mapping groups Z^n = [Z_1, Z_2, ..., Z_n];
for j ← 1; j ≤ m do
    Calculate H_j according to Eq.16;
Get the enhancement groups H^m = [H_1, H_2, ..., H_m];
Calculate W_{H^m} according to Eq.17;
Set all features as L = [Z^n | Z^m W_{H^m}];
Calculate W according to Eq.21;
Return the predicted label Y = XW.
```

IV. Experiments

In this section, we will evaluate the effectiveness of our proposed CCR-BLS with extensive experiments. We will use several public datasets to verify our proposed model, As demonstrated in Section II, CCR-BLS is constructed progressively based on BLS and ELMAE-BLS (BLS with ELMAE). Therefore, related experiments are conducted to test the superiorities of CCR-BLS in the experiments. And we also try to apply CCR-BLS to a more specific image classification tasks: face recognition. The results are also reported in this paper.

A. Datasets

The adopted datasets are given in Table I, where 'attribute', 'instance' and 'class' represent the numbers of attributes, instances and classes respectively. All these datasets can be found in the UCI machine learning repository (https://archive.ics.uci.edu/ml/index.php).

Dataset	attribute	instance	class
messidor_features (MF)	19	1151	2
movement_libras(ML)	90	360	15
Breast Cancer Coimbra (BCC)	9	116	2
hayes-roth	4	160	3
banknote_authentication(BA)	4	1372	2
Segmentation	19	2310	7
PID	8	768	2
Statlog	36	6435	6
ionosphere	33	351	2
Australia	14	690	2
optdigits	64	5620	10

TABLE I: Details of datasets in the experiments.

B. Experimental Results

We randomly split each dataset into two parts: D1 and D2, which have the same number of instances. All these parts have the same class distribution. D1 and D2 are used to train each model by sequence. Then, we obtain two testing accuracies on D2 and D1. And the above average accuracy is the learning result of each model. This process will be repeated 20 times and the best result is taken as the final learning result of each model. And the parameters λ_1 and λ_2 in our proposed CCR-BLS are selected from the set$\{10^{-7}, 10^{-6}, \cdots, 1, 10, 100\}$.

Table II gives the best testing results of CCR-BLS and BLS on the adopted datasets. What's more, we select five datasets

Dataset	BLS	CCR-BLS
MF	0.7602	**0.7715**
ML	0.7750	**0.7861**
BCC	0.7759	**0.8190**
hayes-roth	**0.8317**	0.8173
BA	0.9096	**0.9096**
Segmentation	**0.9463**	0.9277
PID	**0.7135**	0.7083
Statlog	0.8333	**0.8424**
ionosphere	**0.9488**	0.9403
Australia	0.8116	**0.8116**
optdigits	0.9929	**0.9932**

TABLE II: Testing results of BLS and CCR-BLS.

Fig. 3: Example images in the Extended YaleB dataset

Method	10 train	15 train	20 train	25 train	30 train	Average
SVM	0.4779	0.6307	0.7981	0.8361	0.9388	0.7363
SRC	0.5093	0.5098	0.5985	0.6366	0.7025	0.5913
CRC	0.9086	0.929	0.951	0.9051	0.9672	0.9322
CCRC	0.8953	0.9235	0.9667	0.9658	0.9788	0.946
BLS	0.8943	**0.97**	**0.9812**	0.9795	0.989	0.9627
CCR-BLS	**0.8953**	0.968	0.9807	**0.9795**	**0.9922**	**0.9631**

TABLE IV: Comparison of recognition accuracies of different methods on Extended YaleB.

used for image classification from above datasets to test the superiorities of CCR-BLS. The results are reported in Table III.

Datasets	BLS	ELMAE-BLS	CCR-BLS
MF	0.7602	0.7602	**0.7715**
BA	0.9096	**0.9111**	0.9096
Segmentation	**0.9463**	0.9216	0.9277
Statlog	0.8333	0.8330	**0.8424**
optdigits	0.9929	0.9925	**0.9932**

TABLE III: Experimental results of BLS, ELMAE-BLS and CCR-BLS.

To test the capability of our proposed CCR-BLS, We conduct another experiment on the face recognition task.The face databases utilized is The Extended YaleB dataset. Here comes some details about the dataset.

The Extended YaleB dataset: There are 2414 frontal face images of 38 individuals. And each individual has 59–64 frontal images with different expressions and illuminations. In order to facilitate the experiments in this part, all the images have been resized to 32×32 in advance. Fig.3 shows some example images in this dataset. For a fair comparison between different algorithms, in the Extended YaleB dataset, 10, 15, 20, 25 and 30 samples from each class are randomly selected for training, and the rest for testing. The reslts are shown in the Table IV.

As the experimental results shown above, BLS behaves well in all these classification tasks, and our proposed CCR-BLS can also yield a superior or competitive performance in the same applications.

V. Conclusion

In this paper, we first discuss the disadvantages of original and then we propose a novel CCR-BLS method to address the problem. As a modified model of BLS, the learning strategies of CCR-BLS can be divided into three parts. In the first part, a collaborative-competitive representation based autoencoder (CCRAE) approach summarized in Algorithm 1 is proposed to refined the mapping features. In the second part, after getting the mapping features, we obtain the enhancement nodes based on competitive mechanism. In the last part, a linear model is used to connect the concatenated transformed features and the ground truths. The testing results of adopted datasets have verified the superiorities of CCR-BLS, in comparison with some related algorithms.

References

[1] Jun Deng, Zixing Zhang, Erik Marchi, and Björn Schuller. Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *2013 humaine association conference on affective computing and intelligent interaction*, pages 511–516. IEEE, 2013.

[2] Marc'Aurelio Ranzato, Y-Lan Boureau, Yann LeCun, et al. Sparse feature learning for deep belief networks. *Advances in neural information processing systems*, 20:1185–1192, 2007.

[3] L Chamara, H Zhou, GB Huang, and CM Vong. Representational learning with extreme learning machine for big data. *IEEE Intelligent Systems*, 28(6):31–34, 2013.

[4] Haoliang Yuan, Xuecong Li, Fangyuan Xu, Yifei Wang, Loi Lei Lai, and Yuan Yan Tang. A collaborative-competitive representation based classifier model. *Neurocomputing*, 275:627–635, 2018.

[5] CL Philip Chen and Zhulin Liu. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE transactions on neural networks and learning systems*, 29(1):10–24, 2017.

[6] Meiling Xu, Min Han, CL Philip Chen, and Tie Qiu. Recurrent broad learning systems for time series prediction. *IEEE transactions on cybernetics*, 50(4):1405–1417, 2018.

[7] Fei Chu, Tao Liang, CL Philip Chen, Xuesong Wang, and Xiaoping Ma. Weighted broad learning system and its application in nonlinear industrial process modeling. *IEEE transactions on neural networks and learning systems*, 31(8):3017–3031, 2019.

[8] Junwei Jin, Zhulin Liu, and CL Philip Chen. Discriminative graph regularized broad learning system for image recognition. *Science China Information Sciences*, 61(11):1–14, 2018.

[9] Huimin Tang, Peiwu Dong, and Yong Shi. A construction of robust representations for small data sets using broad learning system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[10] Xinrong Gong, Tong Zhang, CL Philip Chen, and Zhulin Liu. Research review for broad learning system: Algorithms, theory, and applications. *IEEE Transactions on Cybernetics*, 2021.

[11] Junwei Jin, Yanting Li, Tiejun Yang, Liang Zhao, Junwei Duan, and C.L. Philip Chen. Discriminative group-sparsity constrained broad learning system for visual recognition. *Information Sciences*, 576:800–818, 2021.

[12] Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *2011 International conference on computer vision*, pages 471–478. IEEE, 2011.