ICIC Express Letters
Volume **11**, Number **8**, August **2017**

 ISSN 1881-803X
pp. **1323–1330**

AN IMPROVED DEEP LEARNING FRAMEWORK BRIEF-NET BASED ON CONVOLUTIONAL NEURAL NETWORKS

Qiang Wang[1], Xiaojie Li[1,*] and Deng Xu[2]

[1]School of Computer Science
Chengdu University of Information Technology
No. 24, Block 1, Xuefu Road, Chengdu 610225, P. R. China
[*]Corresponding author: lixj@cuit.edu.cn

[2]Chengdu No.7 Wanda High School
No. 1, Sheng'an Road, Jinniu District, Chengdu 610036, P. R. China

Received March 2017; accepted May 2017

Abstract. *This paper presents a deep-learning framework, denoted as Brief-Net, based on a convolutional neural network, and applies the framework to image classification. The Brief-Net network consists of three convolution layers and max-pooling layers followed by three fully connected layers. The softmax classifier is employed to identify image classifications. The proposed network employs a relatively small first-layer convolution kernel, overlapping pool-sampling, and eliminates the local response normalization layer for reduced training time and memory cost. In this method, we use a very efficient graphics processing unit implementation of the convolution operation to further reduce training time. Experimental results obtained for two real-world datasets demonstrate the effectiveness and efficiency of our method. Compared with two related state-of-the-art approaches CaffeNet and AlexNet, our method provides higher identification accuracy for the datasets considered.*
Keywords: Deep learning, Caffe frame, Convolution neural network, Image classification

1. **Introduction.** Fundamental research regarding artificial neural networks (ANNs) [1] began prior to the computer age. While ANNs have demonstrated a unique capacity to solve problems by extracting highly complicated patterns from complex and imprecise data, early development was severely constrained by fundamental technical issues and the lack of sufficient computer resources. However, with the parallel development of computer resources and the key contribution regarding the back-propagation (BP) algorithm [1] published by Rumelhart et al. in 1985, the development of ANNs has expanded continuously. ANN can perform different tasks such as image classification. However, ANNs suffer from some drawbacks, such as overfitting and the long training times required for networks that can include millions of parameters and the artificial selected feature set. Selection of "good" features is a crucial step in the image classification since the next stage sees only these features and acts upon them. Recently, many approaches, such as deep learning [2], have been proposed to solve such problems. Deep learning is a relatively new branch of machine learning that employs multilayered computational models to represent data with multiple levels of abstraction. Presently, numerous types of deep networks have been proposed for extracting useful information from rapidly growing volumes of digital data, such as convolutional neural networks (CNNs) [3], restricted Boltzmann machines (RBMs) [4], and stacked autoencoder (SAE) [5] networks. However, most of the top-level algorithms in image recognition are somehow based on CNNs today. CNNs have been widely used in many applications such as speech recognition [6], image classification [3], and object detection [7].

Graphics processing units (GPUs) have been widely used in deep learning to accelerate the speed of data computing and reduce the time required for the training and testing of ANNs. As such, GPUs are better suited to high-speed operations than CPUs. GPUs can effectively operate on images and graphical data far faster than conventional CPUs. For example, application of the AlexNet [3] model based on the Caffe framework [8] to a 256×256 pixel image requires only 1.17ms when processed on the NVIDIA Tesla® K40 platform. The CUDA® (NVIDIA Corp.) framework is an extensively employed hardware system that serves as the platform for some deep learning frameworks such as the convolutional architecture for fast feature embedding (Caffe) framework, Theano [9], Torch [10] and MXNet [11]. Among these, Caffe is used by the NVIDIA Deep Learning GPU Training System (DIGITS) open-source deep learning software for image classification. DIGITS is an efficient deep learning framework widely used around the world. Caffeware supports both GPU and CPU operations, and has Python and MATLAB wrappers. Currently, the Caffe library has emerged as one of the most widely used/tested libraries that implements CNNs. The Caffe framework greatly facilitates the implementation of CNNs, where implementation requires only two prototxt files containing specifications (the model definition) of the target network architectures and configurations for training and testing. In addition, the framework makes activities like fine tuning and transfer learning extremely easy.

The present work adopts CNNs to develop an improved deep-learning framework denoted as Brief-Net that is based on the Caffe framework. Brief-Net consists of three convolution layers and three max-pooling layers followed by three fully connected layers. The softmax classifier [12] is employed to identify image classifications. In this method, we employ a very efficient GPU implementation of the convolution operation to greatly reduce the training time. Compared with two related state-of-the-art approaches, CaffeNet and AlexNet, our method provides higher classification accuracy for some datasets. Experimental results obtained for two real-world datasets demonstrate the effectiveness and efficiency of our method.

The remainder of this paper is organized as follows. In Section 2, we would introduce CNNs model. The Brief-Net model and motivations are proposed in Section 3. Section 4 shows the performance of the proposed model in real image classification tasks and Section 5 concludes this paper.

2. Convolutional Neural Networks.

CNNs are standard feed-forward multilayer ANNs inspired by biological processes, and employ sparse connectivity and a shared weights strategy. CNNs consist of a series of hidden convolutional and pool-sampling layers optionally followed by fully connected layers and is good at extracting useful local and global training features for classification. The standard CNN architecture is illustrated in Figure 1, which shows the arrangement of convolutional and pool-sampling sub-layers. Through the series of these hidden layers, CNNs use the BP algorithm to train weights, and, in this study, we obtain probabilities using the softmax classifier, which serves as the final output layer. For each sample, each element of the probability vector Y ($Y \in R^{c\times 1}$), employing one-hot encoding, corresponds to a class. To minimize the objective function, the network structure uses the gradient descent method [13], which adjusts the weight parameters layer by layer, and improves the network accuracy by frequent iterative training.

The primary components of the hidden layers of a CNN are described in detail as follows.

- Convolutional layer: The layer parameters consist of a set of learnable convolution kernels (filters). During the forward pass, each convolution kernel is incrementally shifted over the width and height of the image matrix, and, at each position, the dot product is computed between the entries of the convolution kernel and the local image matrix until the convolution transformation (mapping transformation) of the

original image is completed. The backward pass then applies the BP algorithm to computing the gradients.

- Activation function: This function increases the nonlinear properties of the decision function. Three common activation functions are employed, namely, the sigmoid, hyperbolic tangent (tanh), and the rectified linear unit (ReLU) functions.
- Pool-sampling layer: This layer gradually reduces the spatial size of the image representation to reduce the number of parameters and the computational load in the network. Several non-linear functions have been employed to implement pooling, such as max-pooling, mean pooling, and random pooling.
- Local response normalization (LRN) layer: The normalization operation is applied to the down-sampled image to smoothing the output characteristics.

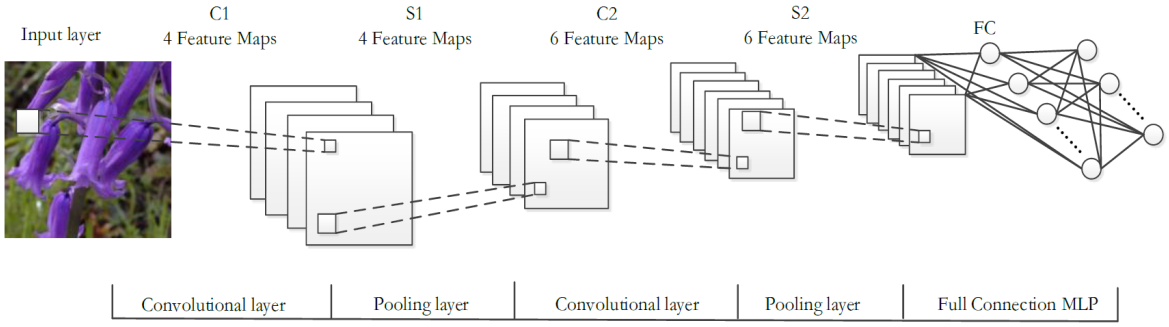

FIGURE 1. Standard convolution neural network (CNN), where C represents a convolution layer, S is a pool sampling layer, and FC is the fully connected multilayer perceptron (MLP) layer.

The specific CNN implementation process based on the Caffe framework is illustrated in Figure 2. At first, Caffe framework convert images are stored in LevelDB databases. And then a typical network begins with a data layer that loads from disk and ends with a loss layer that computes the objective for a task such as classification or reconstruction.

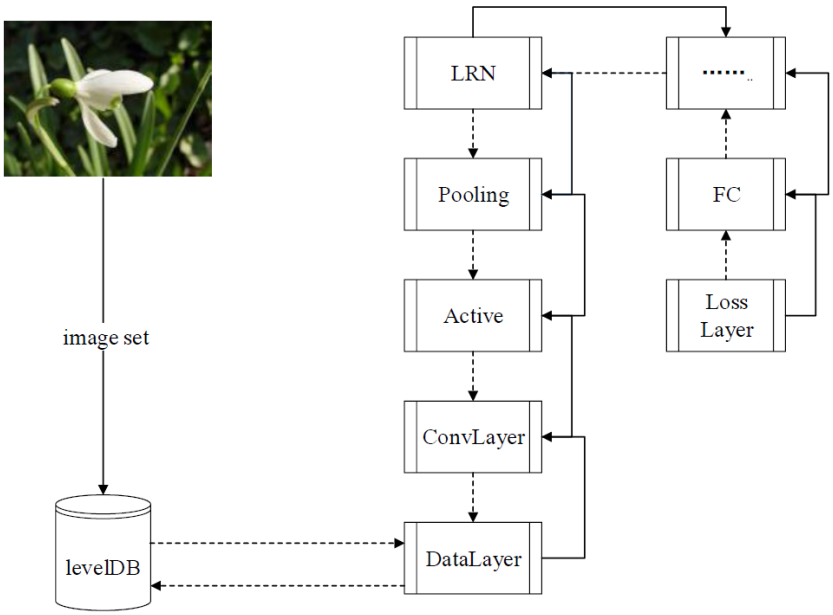

FIGURE 2. CNN implementation based on the Caffe framework, which includes a convolution layer (ConvLayer), an activation function (Active), a pool-sampling layer (Pooling), a local response normalization layer (LRN), a fully connected (FC) layer, and the Loss Layer.

3. **Proposed Brief-Net Model.** We employ two modifications that would improve the accuracy: a relatively small first-layer convolution kernel size and overlapping pool-sampling. Of the first, we used a smaller convolution kernel which can extract the changes in these details. Of the second, we employ overlapping pool-sampling that can avoid over-fitting effectively. Also, our network has more streamlined structure.

The structure of the Brief-Net network model is illustrated in Figure 3. The network model is composed of 3 convolution layers (C1-C3) employing progressively smaller kernels (9×9, 5×5, and 3×3 respectively) followed by 3 fully connected layers (FC1-FC3) with a softmax function at the end for classification. Each convolution layer applies a number of convolution kernels to its input and concatenates the resulting convolution maps. The output of each convolution layer passes through a ReLU activation function, which then serves as the input of a max-pooling layer (S1-S3). In addition, the network also uses a dropout regularization method to avoid over-fitting in the fully connected layers. Compared with other network models, the first convolution layer employs a smaller convolution kernel (9×9). An equal learning rate is employed for all layers, and the rate is automatically adjusted as the training progresses. The objective function, convolution kernel, the use of an overlapping pool-sampling layer, activation function selection, and the elimination of the LRN layer are discussed in detail as follows.

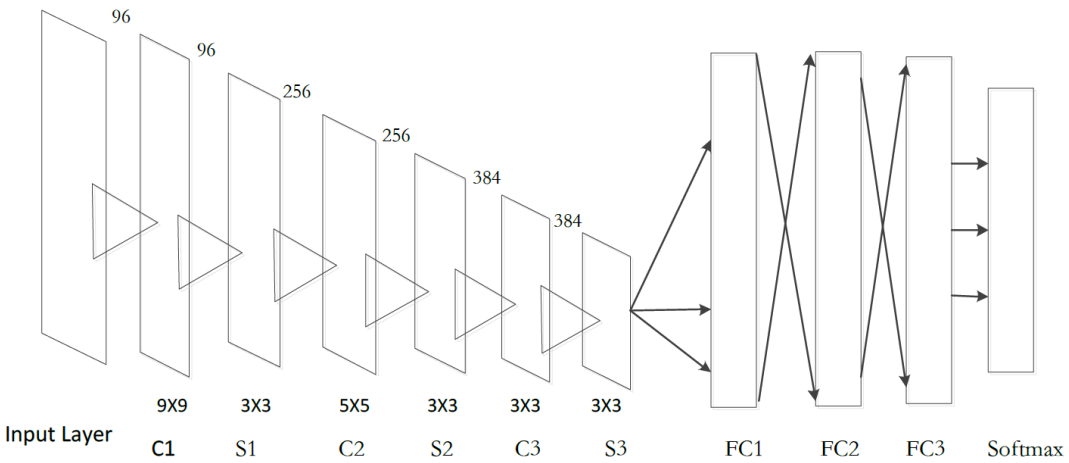

FIGURE 3. Brief-Net model structure, where C represents a convolution layer, S represents a max-pooling layer, and FC represents a fully connected layer.

3.1. **Objective function.** The Brief-Net model employs the softmax cost function as the objective function to complete image classification. The softmax function is based on softmax regression, which is a supervised learning algorithm that generalizes logistic regression to cases of multiple classes. Assuming a training set composed of m training samples $\left\{\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(i)}, y^{(i)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)\right\}$, where $x^{(i)}$ represents the i-th training sample and $y^{(i)}$ represents its corresponding class label, in a multiclass setting $y^{(i)}$ can take on k different values, i.e., $y^{(i)} \in \{1, 2, \ldots, k\}$. The softmax cost function is (1)

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] \tag{1}$$

Here, 1{.} is the indicator function that determines if $x^{(i)}$ is of class j, i.e., 1{a true statement} = 1 and 1{a false statement} = 0. The k possible values of $y^{(i)}$ are accumulated, and the probability of classifying $x^{(i)}$ as class j is (2)

$$p\left(y^{(i)}=j|x^{(i)};\theta\right)=\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \tag{2}$$

To solve for the minimum of $J(\theta)$ analytically, we employ gradient descent, which is an iterative optimization algorithm. Taking derivatives, one can show that the gradient is (3)

$$\Delta_{\theta_j} J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\left[x^i\left(1\left\{y^{(i)}=j\right\}-p\left(y^{(i)}=j|x^{(i)};\theta\right)\right)\right] \tag{3}$$

3.2. **Convolution kernel.** A convolutional layer is parameterized by the kernel sizes and the number of maps. The first-layer convolution kernel is the closest to the input layer. It is extracted from the basic features, so the parameters have the greatest influence, and the subsequent layer structure depends heavily on the output of the first convolution layer. For the image classification to extract small features, such as border, light and shade, simple stripes, a smaller convolution kernel can extract the changes in these details, so the model uses a smaller convolution kernel.

3.3. **Overlapping pool-sampling layer.** In conventional practice, the neighborhoods summarized by adjacent pooling units do not overlap. A pooling layer can be thought of as consisting of grid of pooling units spaced s pixels apart, each summarizing a neighborhood of size $z \times z$ centered at the location of the pooling unit. If we set $s = z$, we obtain the conventional local pooling commonly employed in CNNs. If we set $s < z$, we obtain overlapping pooling. In the present work, we employ $s = 2$ and $z = 3$ throughout our network. The adopted overlapping pool-sampling provides a network that is slightly less susceptible to overfitting.

3.4. **Activation function selection.** The Brief-Net model applies the non-saturating ReLU $f(x) = \max(0, x)$ as the activation function. When the input signal is less than 0, the output is 0, and when the input signal is greater than 0, the output is equal to the input. The primary advantages of ReLU are as follows.

The convergence rate of ReLU is greater than that of other activation functions.

ReLU requires only a threshold to obtain the activation value, and the calculation complexity is low.

3.5. **Elimination of the LRN layer.** LRN is mainly the sampling layer contrast normalization operation. The image data set contains little bright information and strong contrast, the output requires no smoothing. The LRN layer is, therefore, eliminated from the Brief-Net network model to reduce training time and memory costs.

4. **Experiments and Results.** The experiments were conducted on two NVIDIA Tesla K40 GPU high-performance workstations running the Ubuntu 14.04 operating system. All color images were 256×256 pixels. We compared the number of iterations at convergence and the identification accuracy of Brief-Net with those of other network models, including AlexNet and CaffeNet, which is a replication of AlexNet with a few modifications. The three model structures are listed in Table 1 (layers 0-6) and Table 2 (layers 7-13). The convolution layers and the max-pooling layers are characterized in the tables according to the convolution kernel size and pooling layer size, respectively, step, fill boundary, and quantity. For example, the first convolution layer of AlexNet is represented in Table 1 as Conv 11-4-0-96, which indicates that the size of the convolution kernel is 11×11, the step

TABLE 1. The three network structure settings (layers 0-6)

Model	0 Input layer	1 ConvLayer	2 Max-pooling	3 ConvLayer	4 Max-pooling	5 ConvLayer	6 Max-pooling
Brief-Net	256 × 256	Conv 9-4-0-96	MP 3-2-0-96	Conv 5-0-2-256	MP 3-2-0-256	Conv 3-0-1-384	MP 3-2-0-384
CaffeNet	256 × 256	Conv 11-4-0-96	MP 3-2-0-96	Conv 5-0-2-256	MP 3-2-0-256	Conv 3-0-1-384	MP 3-2-0-384
AlexNet	256 × 256	Conv 11-4-0-96	MP 3-2-0-96	Conv 5-0-2-256	MP 3-2-0-256	Conv 3-0-1-384	No

TABLE 2. The three network structure settings (layers 7-13)

Model	7 ConvLaye	8 ConvLayer	9 Max-pooling	10FC	11 FC	12FC	13 Classifier
Brief-Net	No	No	No	2048	2048	5	Softmax
CaffeNet	Conv 3-1-1-384	Conv 3-1-1-256	MP 3-2-0-256	4096	4096	5	Softmax
AlexNet	Conv 3-1-1-384	Conv 3-1-1-256	MP 3-2-0-256	4096	4096	5	Softmax

is 4, the filling boundary is 0, and the number of convolution kernels is 96. As FC (fully connected) layer, the numbers in the FC columns representation number of connection parameters. Each convolution layer employs ReLU activation. The initial learning rate (base_lr) was set to 0.001 and the maximum number of iterations (max_iter) was set to 500. Verification was conducted every 50 iterations during training. In all cases, training was conducted using the cross validation method [14].

4.1. **RO-5 dataset.** The RO-5 image dataset from real object contained 5 classes including images of buses, dinosaurs, elephants, flowers, and horses. We selected 200 images from each category as the training set and 40 as the test set, for a total of 1200 images.

The identification accuracies of the three network models with respect to the number of iterations are shown in Figure 4. The comparison results indicate that Brief-Net provides the best performance. The Brief-Net model begins to converge after 50 iterations, which is better than the convergence times of CaffeNet (100 iterations) and AlexNet (150 iterations). Moreover, the identification accuracy of Brief-Net is superior, with an accuracy of 97%, which is considerably better than that of AlexNet (84%). Similarly, Brief-Net provides an improvement relative to CaffeNet, which has an accuracy of 94%.

4.2. **Flower dataset.** We also employed images obtained from a dataset collected by the image recognition group of Maria-Elena Nilsback and Andrew Zisserman for flower species identification [15]. The dataset images are assembled into 17 categories, and the

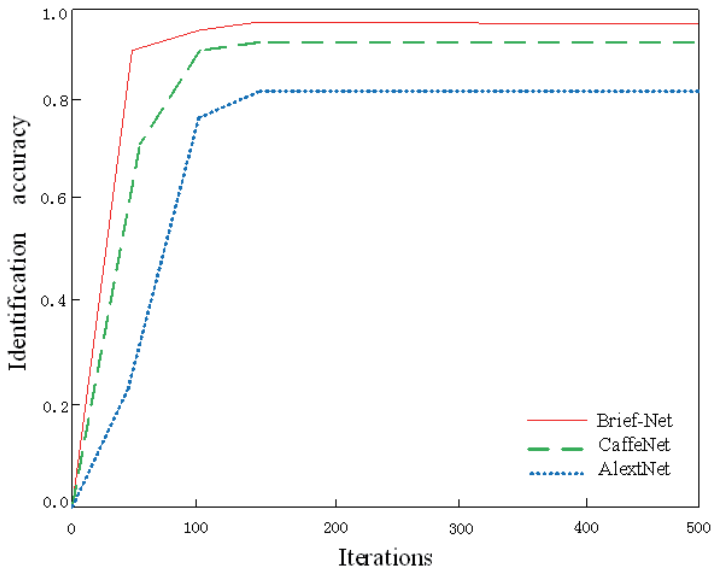

FIGURE 4. The accuracy of the three network models with respect to the number of iterations for the RO-5 dataset

images include a large degree of similarity. We selected 70 images from each category as the training set and 10 as the test set, for a total of 1360 images.

The identification accuracy of the three network models with respect to the number of iterations is shown in Figure 5. The comparison results indicate that Brief-Net provides the best performance. The Brief-Net model converges after about 150 iterations, which is better than both CaffeNet (200 iterations) and AlexNet (300 iterations). In terms of identification accuracy, Brief-Net is again superior, with an accuracy of 70%, which is significantly greater than those of CaffeNet (54%) and AlexNet (39%). These results demonstrate that the Brief-Net model has better identification accuracy with more rapid convergence for small differences image datasets.

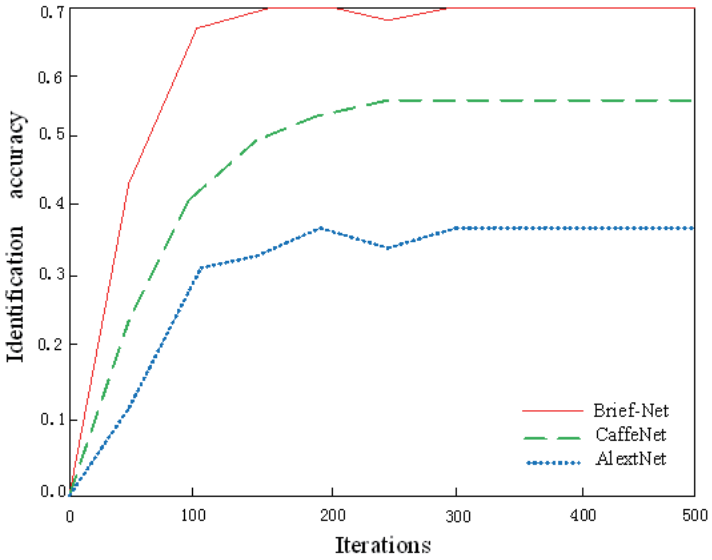

FIGURE 5. The accuracy of the three network models with respect to the number of iterations for the flower dataset

4.3. **Influence of the first-layer convolution kernel size.** The influence of the first-layer convolution kernel size employed in the Brief-Net model was examined by comparing the number of iterations at convergence and the identification accuracy obtained for the RE-5 and flower datasets with different kernel sizes. Accordingly, the first-layer convolution kernel size was decreased from 11 to 5, and the results are listed in Table 3.

TABLE 3. Influence of the first-layer convolution kernel size

	First-layer convolution kernel size	Identification accuracy (%)	Number of iterations at convergence
RO-5 dataset	11	96	140
	10	96	140
	9	97	80
	8	96	180
	5	95	220
Flower dataset	11	67	150
	10	68	100
	9	70	100
	8	69	140
	5	69	140

The experimental results show that the number of iterations at convergence and the identification accuracy are affected by the size of the first-layer convolution kernel. For the Brief-Net model, we find that the optimal first-layer convolution kernel size is 9×9.

5. **Conclusion.** We proposed a CNN model, denoted as Brief-Net, based on the Caffe framework, which employs a relatively small first-layer convolution kernel size, overlapping pool-sampling, and omits the LRN layer. The proposed network model demonstrated better performance for the RO-5 and flower datasets than other existing network models, including AlexNet and CaffeNet. Meanwhile, experimental results demonstrated that the optimal size of the first-layer convolution kernel of Brief-Net is 9×9, which provides the lowest number of iterations at convergence and the highest identification accuracy. Future work will research other ways to retrain the deep CNNs.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (Grant No. 61602066) and by the project supported by the Scientific Research Foundation (KYTZ201608) of CUIT, partially supported by the Scientific Research Foundation of the Education Department of Sichuan Province, China (Grant No. 17ZA0063).

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning representations by back-propagating errors, *Nature*, vol.323, pp.533-536, 1986.
[2] Y. LeCun, Y. Bengio and G. E. Hinton, Deep learning, *Nature*, vol.521, no.7553, pp.436-444, 2015.
[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, pp.1097-1105, 2012.
[4] H. Larochelle and Y. Bengio, Classification using discriminative restricted Boltzmann machines, *Proc. of the 25th International Conference on Machine Learning*, pp.536-543, 2008.
[5] P. Vincent, H. Larochelle, Y. Bengio and P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, *ICML*, pp.1096-1103, 2008.
[6] G. Hinton, L. Deng, D. Yu et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *IEEE Signal Processing Magazine*, vol.29, no.6, pp.82-97, 2012.
[7] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems*, pp.91-99, 2015.
[8] Y. Jia, E. Shelhamer, J. Donahue et al., Caffe: Convolutional architecture for fast feature embedding, *Proc. of the 22nd ACM International Conference on Multimedia*, pp.675-678, 2014.
[9] F. Bastien, P. Lamblin, R. Pascanu et al., Theano: New features and speed improvements, *Computer Science*, 2012.
[10] S. Bahrampour, N. Ramakrishnan, L. Schott and M. Shah, Comparative study of Caffe, Neon, Theano, and Torch for deep learning, *Computer Science*, 2015.
[11] T. Chen, M. Li, Y. Li et al., MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems, *Statistics*, 2015.
[12] Y. LeCun, B. Boser, J. S. Denker et al., Backpropagation applied to handwritten zip code recognition, *Neural Computation*, vol.1, no.4, pp.541-551, 1989.
[13] R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization, *Computer Journal*, vol.6, no.2, pp.163-168, 1963.
[14] S. Yadav and S. Shukla, Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification, *IEEE International Conference on Advanced Computing*, pp.78-83, 2016.
[15] M. E. Nilsback and A. Zisserman, Automated flower classification over a large number of classes, *Indian Conference on Computer Vision, Graphics & Image Processing*, pp.722-729, 2008.