IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON GIGAPIXEL PANORAMIC VIDEO WITH VIRTUAL REALITY

Received July 10, 2020, accepted July 19, 2020, date of publication July 22, 2020, date of current version July 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011127

Big Data Image Classification Based on Distributed Deep Representation Learning Model

MINJUN ZHU AND QINGHUA CHEN
College of Mathematics and Information, Fujian Normal University, Fuzhou 350117, China
Corresponding author: Minjun Zhu (zmjmath@163.com)

ABSTRACT Traditional image classification technology has become increasingly unable to meet the changing needs of the era of big data. With the open source use of a large number of marked databases and the development and promotion of computers with high performance, deep learning has moved from theory to practice and has been widely used in image classification. This paper takes big data image classification as the research object, selects distributed deep learning tools based on Spark cluster platform, and studies the image classification algorithm based on distributed deep learning. Aiming at the problems that the Labeled Structural Deep Network Embedding (LSDNE) model is applied to the attribute network and generates a large number of hyperparameters and the model complexity is too high, inspired by the Locally Linear Embedding (LLE) algorithm, this paper proposes a semi-supervised network based on the neighbor structure learning model. This model will add the neighbor information of the node at the same time when learning the network representation. Through the node vector reconstruction, the node itself and the neighbor node together constitute the next layer of representation. On the basis of Structural Labeled Locally Deep Nonlinear Embedding (SLLDNE), the node attribute is further added to propose Structural Informed Locally Distributed Deep Nonlinear Embedding (SILDDNE), and how the model combines the structural characteristics of the node with the attribute characteristics is explained in detail. The SVM classifier classifies the known labels, and SILDDNE fuses the network structure, labels, and node attributes into the deep neural network. The experimental results on the CIFAR-10 and CIFAR-100 datasets for image classification standard recognition tasks show that the proposed network achieves good classification performance and has a high generalization ability. Experiments on the CIFAR-10 data set show that the 34-layer SLLDNE pruned 40-layer Dense Net compresses about 50% of the parameter amount, increases the computational complexity efficiency by about 8 times, and reduces the classification error rate by 30%. Experiments on the CIFAR-100 data set show that the 34-layer SLLDNE parameter volume is compressed by about 16 times compared to the 19-layer VGG parameter volume, the computational complexity efficiency is increased by about 6 times, and the classification error rate is reduced by 14%.

INDEX TERMS Image classification, distributed network representation learning, deep learning, neighbor reconstruction.

I. INTRODUCTION

Big data is one of the main topics in the current information age, which determines the trend of economic and social ideology and cutting-edge technology research and development in recent years [1]–[3]. Under the guidance of a large amount of information appears in all aspects of our living environment, people live in a sea of information. Among the wide variety of information that people receive, the most important position is the image information received through vision [4].

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

More than 70% of the information received by humans is received through vision [5]. The image is the reproduction of the information that people perceive visually. Compared with sound and text, it contains a large amount of information, the transmission method is more direct, and the reception method is simpler [6]. Therefore, the recognition and processing of image information has attracted more and more research and attention, and the related technologies and methods have been increasingly improved, which has become the mainstream of the current development of artificial intelligence [7], [8].

In today's society, with the advent of the cloud era, big data has attracted more and more attention. As an important

strategic resource for modern enterprises and society, big data has also become a new focus of attention. As the main form of data information, images have become an important means for people to obtain information because of their rich content and intuitive reflection. The number is rapidly increasing at an alarming rate. However, the problem of disordered image information has become increasingly prominent with the growth of image data. How to use artificial intelligence technology to automatically identify, retrieve and classify massive image data has become the current research focus in the field of computer vision recognition.

Image classification is one of the main applications of computer vision [9]. After Turing's artificial intelligence research frenzy, it has become a popular subject and appeared in the research of major universities in the world [10]–[12]. The research direction of image classification at home and abroad can be systematically divided into three steps to develop, namely image pre-processing, feature extraction of images to obtain information elements and optimization of classification methods [13], [14]. Among them, feature extraction is the most important and the most concerned step for researchers [15], [16]. The quality of feature extraction methods often represents the advantages and disadvantages of image classification methods. Before the emergence of machine learning, the industry mainly used Scale-Invariant Feature Transform (SIFT) method for feature extraction, and then used unsupervised learning algorithms to generate feature description symbols, and then used supervised learning algorithms to train training data sets with special labels [17]–[19]. The features are learned and analyzed to obtain the corresponding feature classifier, so as to realize the classification of the image. With the introduction of machine learning, artificial intelligence is becoming formal, neural network algorithms are booming, and the emergence of classification models such as Back Propagation (BP) has made image classification technology enter a high-speed development track [20], [21]. However, due to its structural characteristics, BP algorithm is difficult to achieve large-scale training sample, and it is even more weak on the problem of multi-classified images, which makes the industry lose confidence in neural networks and makes the research of neural networks in image classification fall into a trough [22]–[24]. The new development of neural networks and the rise of deep learning quickly replaced the status of traditional classification methods, in which Convolutional Neural Network (CNN) in the field of machine vision research and application shines brightly, and is used by major companies [25], [26]. Related scholars have proposed 2 Graphic Processing Unit (GPU) based accelerated convolutional neural network training methods [27]. One is an image combination method for accelerating the forward calculation of convolutional neural networks, and the other is a method for training convolutional neural networks based on GPUs with low memory consumption. This method can train arbitrarily large networks. Scholars perform two-dimensional discrete fast Fourier transform on the input feature maps of the nodes of the convolution layer, then do convolution operations in the frequency domain, and use GPU programming to achieve [28], [29]. Because the same feature map Fourier transform can be used repeatedly, it can speed up the training of convolutional neural networks [30]. Experts have proposed two parallel training methods for convolutional neural networks [31]. One is a parallel training method based on multi-core processors, and the other is a parallel training method based on many-core processors. The former can achieve the acceleration effect of almost the same physical processor core, while the latter can achieve 90 times the acceleration ratio. Related scholars have proposed an accelerated training method for deep convolutional neural networks based on response reconstruction [32]. This method accelerates the training of deep convolutional networks by solving a nonlinear optimization problem with low rank constraints, and designs a generalized singular value decomposition method to solve the optimization problem.

By increasing the depth and width of the deep convolutional network, the classification accuracy can be effectively improved. However, as the scale of the network structure increases, a large number of parameters are generated in the network, which increases the redundancy of the network. In response to the above problems, this paper proposes a deep convolutional network model based on interlaced fusion packets. Specifically, the technical contributions of this article can be summarized as follows:

First: For the problem of network redundancy caused by deeper convolutional networks by increasing the structure scale to improve the classification accuracy, this paper proposes a Structural Labeled Locally Deep Nonlinear Embedding (SLLDNE) for extracting the network structure and the node's own attributes model. On the basis of this model, a Structural Informed Locally Distributed Deep Nonlinear Embedding (SILDDNE) model is proposed, which explains in detail how the model combines the structural characteristics of nodes with attribute characteristics.

Second: The experimental results on the CIFAR-10 and CIFAR-100 datasets show that the proposed deep network has more efficient parameter utilization and lower computational complexity, and shows a smaller model size. Compared with the network structure, it has obvious performance advantages.

The rest of this article is organized as follows. Section 2 analyzes the related methods of deep learning image classification. Section 3 constructs a semi-supervised distributed deep attribute network representation learning model based on neighbor structure. Section 4 carried out simulation experiments of big data image classification. Section 5 summarizes the full text.

II. ANALYSIS OF RELATED METHODS OF DEEP LEARNING IMAGE CLASSIFICATION

A. CONSTRUCTION METHOD OF DEEP NETWORK

In the initial machine learning, feature extraction was performed by manually setting rules, and then the data was

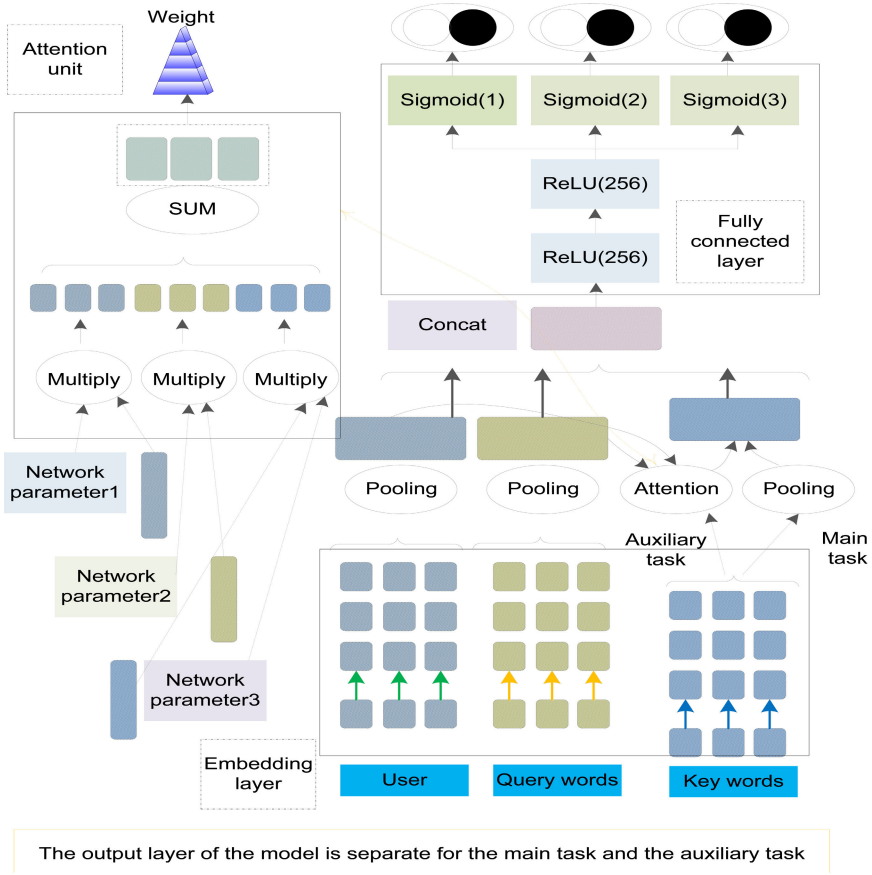


FIGURE 1. Multi-task deep learning model architecture diagram.

classified and predicted by the classification and prediction system. Most of these learning rules are enlightening and require the guidance of a lot of professional knowledge. In practical applications, especially for systems where feature construction is not so direct, the applicability and portability have certain limitations.

The artificial neural network and the back propagation algorithm have made the machine think like human beings and have made great progress. This machine learning method based on statistical models allows the computer to actively learn statistical laws from a large number of training samples to classify and predict things. This method based on statistical machine learning has many advantages over traditional methods, but from the practical analysis, this is just a shallow learning model. Models such as maximum entropy and Boosting have achieved great success in their use, but they are all shallow learning models, without in-depth feature acquisition, coupled with the difficulty of theoretical analysis and training skills, making the application not so extensive.

The deep learning method establishes a deep nonlinear structure through the network, and learns the multi-layer features of massive sample set data to improve the accuracy of classification or prediction. Therefore, deep learning is different from shallow learning in that it is essentially different from shallow learning in terms of hierarchical structure, feature expression and transformation. This kind of deep-level features that use big data to learn describe the data. Multi-task deep learning model architecture is shown in Figure 1.

If we construct such a multi-layered network and express it step by step, we will end up with a multi-level abstract expression for the original input I, similar to the process of brain-level abstract visual acquisition of information.

In the original shallow learning process, the learning process is cumbersome and complicated, and it is not easy to get a good expression network of input I, but after such a lossless expression based on information at first, you adjust the network to get a good learning effect.

B. DISTRIBUTED DEEP LEARNING

Deep learning models are complex, with many training data and large amounts of computation. On the one hand, deep learning needs to simulate the computing power of the human brain, and the human brain contains more than 10 billion nerve cells, which requires more neurons in the deep learning model, and the number of connections between neurons is also quite amazing. From a mathematical point of view, each neuron in the model contains mathematical calculations (such as Sigmoid, ReLU, or Softmax functions), and the amount of parameters to be estimated is also extremely large. In the

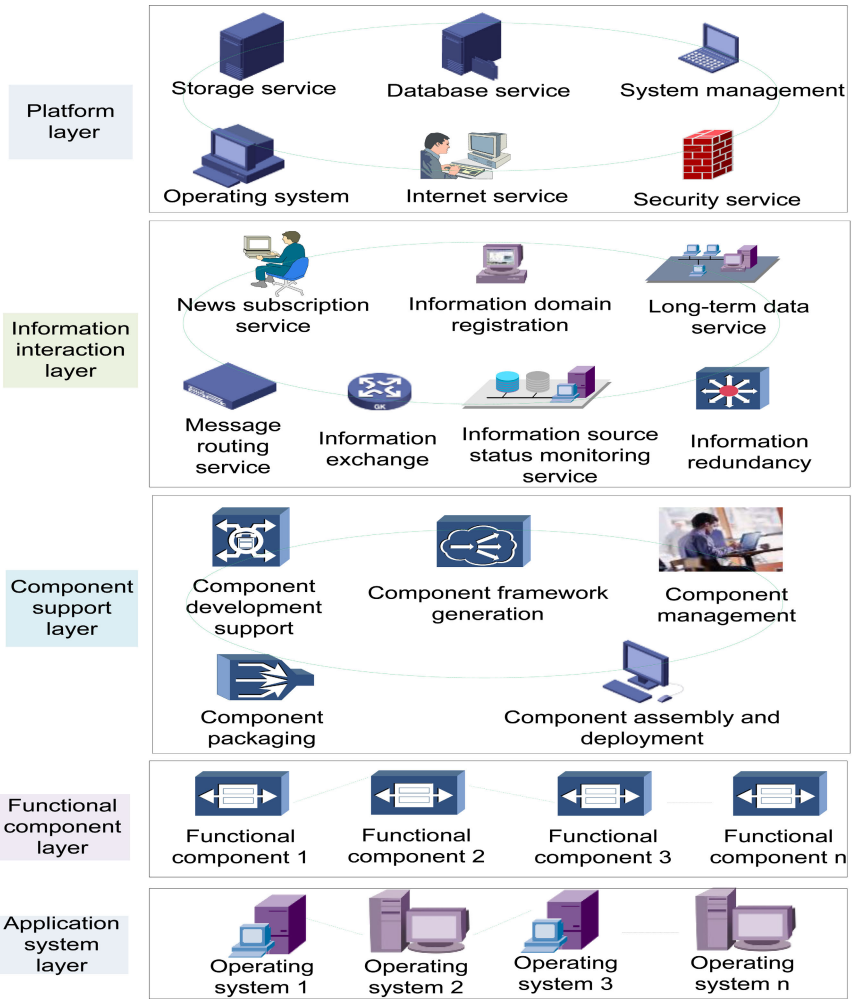


FIGURE 2. Distributed system architecture.

application of image classification, there are tens of thousands of neurons and tens of millions of parameters. The complicated model leads to a large amount of calculation. On the other hand, deep learning requires a large amount of data to train a model with high accuracy. Deep learning models have large parameters and complex models. To avoid overfitting, massive training data is required. The superposition of two factors makes the training of a model time-consuming. Taking speech recognition as an example, the sample size is usually several billions. It takes several years for a single CPU to complete a training, and it takes several weeks to complete a training with a popular GPU card.

Deep neural networks need to support large models. Taking image recognition as an example, experiments have proved that by increasing the number of filters of the convolution layer and increasing the depth of the model, a better model quality can be obtained, but the model parameters also increase. However, the introduction of the 152-layer Res Net structure and the large amount of training data in the era of big data have brought about improvements in accuracy and also brought huge training time costs. Only the current stand-alone version of deep learning is used. The tool plus GPU calculation method can no longer meet the required computing power and storage space. The distributed system architecture is shown in Figure 2.

Deep neural network training is difficult to converge and requires repeated experiments. Deep neural networks are nonlinear models, and their cost functions are non-convex functions, which easily converge to the local optimal solution. At the same time, the model structure of deep neural network, input data processing method, weight initialization scheme, parameter configuration, activation function selection, weight optimization method, etc. may have a greater impact on the final effect. In addition, the basic mathematical research of deep neural networks is slightly insufficient. Although it is possible to initialize the network model by using artificially generated modeling methods such as restricted Boltzmann machines to reduce the risk of falling into local optimality, it is still not a complete solution, and it is still necessary to use deep neural networks to solve the

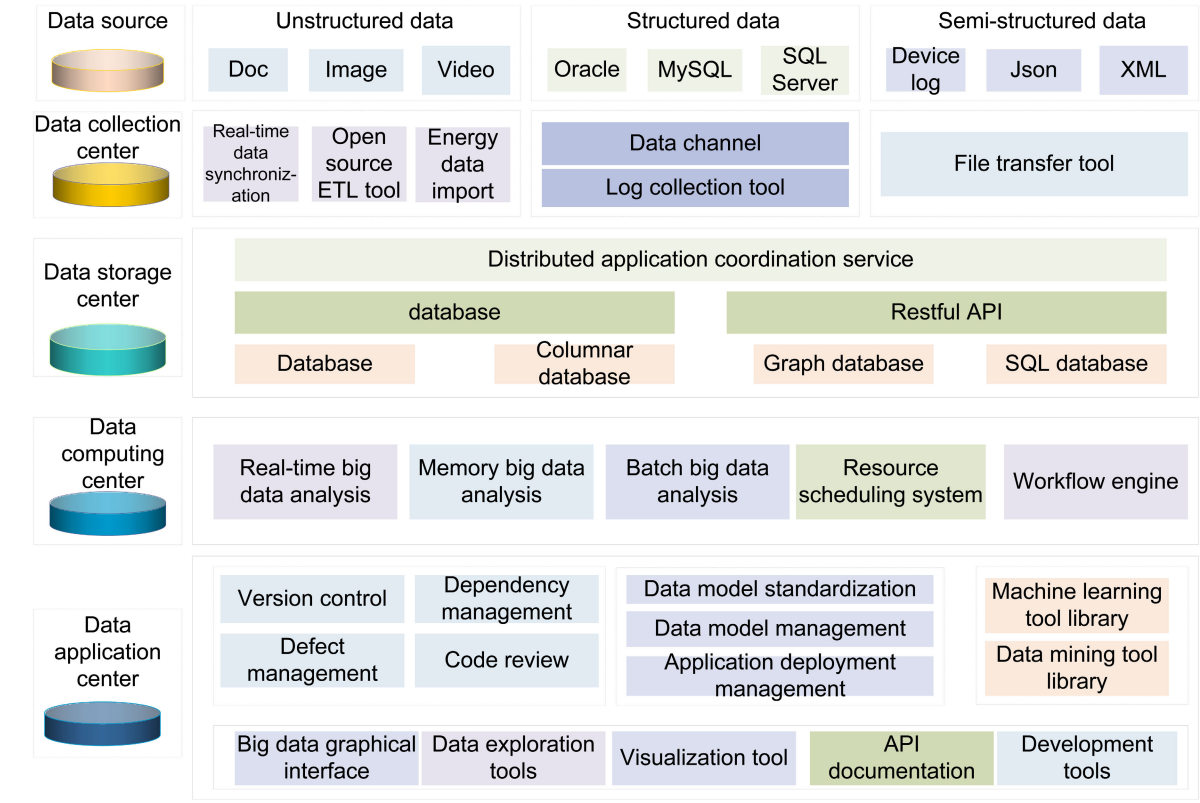


FIGURE 3. Technical architecture of big data platform.

problem. At the time, a good model was trained based on a large number of experiments.

Based on the above reasons, in order to speed up deep learning model training, research on parallel training based on distributed deep learning has received attention. According to the classification of distributed parallel structure, there are mainly the following two parallel methods.

1) DATA PARALLELISM

To complete the data parallel training, it is necessary to synchronize the parameters between different training programs, usually by a Parameter Server (PS) to help complete. In parallel training, multiple training processes are independent of each other. First, you pull the model parameter w from the parameter server to the local, then use the new model parameter w to calculate the gradient Δw of the local batch, and finally push the calculated gradient Δw to the PS. PS then uniformly update the model parameters according to the following formula.

The data parallel structure uses the Stochastic Gradient Descent (SGD) parallel algorithm. The SGD parallel algorithm has a synchronous mode and an asynchronous mode. In the synchronous SGD mode, all training programs train a batch of training data at the same time. After receiving all gradient update values, the parameter server performs an average operation, and the local model exchanges parameters with the parameter server. After the parameter exchange is completed, all training programs have a common new model parameter as a starting point, and then train the next batch. In asynchronous SGD mode, the training program completes a batch of training data and immediately exchanges parameters with the parameter server, regardless of the status of other training programs. In the asynchronous mode, the update of a training program's model parameters will not be immediately reflected in other training programs until they perform the next parameter exchange.

The parameter server is just a logical concept and may not necessarily be deployed as an independent server. Sometimes it will be attached to a training program, and sometimes the parameter server will be divided into different shards according to the model. The technical architecture of the big data platform is shown in Figure 3.

2) MODEL PARALLELISM

The model splits a single model into several parts in parallel, which are held by several training units and work together to complete the training. For large models that cannot be accommodated in a stand-alone memory, model parallelism is a good choice. When the input of a neuron comes from the output of a neuron on another training unit, communication overhead is incurred.

In most cases, the communication overhead and synchronization consumption brought by model parallelism exceed data parallelism, so the acceleration ratio is also inferior to data parallelism. Moreover, data parallelization is superior to model parallelization in terms of difficulty, fault tolerance, and cluster utilization. In practical applications, data parallelism is the preferred solution for deep learning distribution.

C. DEEP LEARNING TOOLS

1) CAFFE

Convolutional Architecture for Fast Feature Embedding (Caffe) is an efficient and practical open source deep learning framework. It mainly implements the CNN network. It has achieved outstanding achievements in image classification, target detection, face recognition and other fields. It is used for image classification. Caffe has the following characteristics:

(1) Modularity: Caffe was designed to be as modular as possible from the beginning, allowing expansion of new data formats, network layers and loss functions.

(2) Separation of representation and implementation: Caffe's model definition is written into the configuration file using Protocol Buffer language. In the form of any directed acyclic graph, Caffe supports network architecture. Caffe will correctly occupy memory according to the needs of the network.

(3) Test coverage: In Caffe, each single module corresponds to a test.

(4) Python and Matlab interfaces: they provide both Python and Matlab interfaces.

(5) Pre-training reference models: For vision projects, Caffe provides some reference models.

However, the high-speed computing claimed by Caffe in the early stage of the design simply uses GPU to accelerate the calculation, and there is no way to implement parallel training, which is obviously inconsistent with the current development trend of deep learning.

The current distributed deep learning frameworks need to build a special cluster for learning and training, which will cause unnecessary data transmission between the business cluster (Hadoop/Spark) and the deep learning cluster, wasting bandwidth.

2) CAFFE ON SPARK

In order to solve the practical problems in the production environment mentioned above, Yahoo developed Caffe On Spark based on parallel data. Caffe On Spark runs on the Spark cluster. It is a deep learning toolkit and fills the gap that Spark MLlib does not support deep learning. Caffe On Spark is a distributed version of Caffe and has all its advantages. The architecture uses a fully parallel SGD structure of data parallelism. The block diagram of data reading transmission is shown in Figure 4.

Deep learning tasks start a certain number of executors in the cluster, each executor is assigned a training data segment

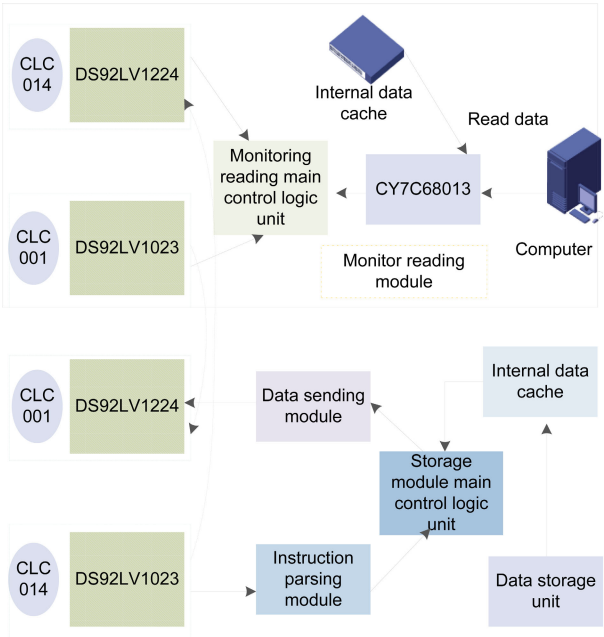


FIGURE 4. Block diagram of data reading and transmission.

stored on HDFS, and then start multiple Caffe-based training threads, each training thread is processed by a specific GPU or CPU. After processing a batch of training samples using the back propagation algorithm, these training threads exchange gradient values of model parameters. These gradient values are exchanged between multiple servers in the form of MPI Allreduce, and the deep learning model is synchronized with the RDMA protocol.

Caffe On Spark combines the strengths of Caffe and Spark. At present, Spark MLlib only supports various non-deep learning algorithms for classification, regression, clustering, etc., but lacks the key ability of deep learning, and Caffe On Spark fills this blank. Caffe On Spark API supports dataframes, easy to connect to the training data set ready to use the Spark application, and extract the predicted value of the model or the characteristics of the middle layer, used for MLLib or SQL data analysis, and can also optimize the scheduling of deep learning resources through YARN. It eliminates the traditional limitation of using a separate cluster for deep learning and the data movement that has to be done. It supports deep learning directly on the big data cluster. Direct access to big data and large-scale computing capabilities are critical to deep learning.

The Caffe On Spark structure has two layers of parameter synchronization. The first layer is to define a root GPU in each executor to replace the parameter server. After the GPU calculates a mini-batch gradient, the gradient value is immediately passed to the root GPU. The root GPU performs the average operation; the second layer defines a root executor between each executor, the root GPUs in other executors send the averaged gradient value to the root GPU in the root executor, and then perform the average operation to update

the model parameters, the updated parameters are transferred to each root GPU, and then sent to other GPUs, until all GPUs receive the updated parameters, the next round of calculation is allowed.

The synchronous SGD parallel structure adopted by Caffe On Spark, at the expense of efficiency, is exchanged for accuracy comparable to stand-alone deep learning training. The synchronous SGD parallel algorithm, when the performance of each machine in the cluster is different, will cause all nodes to wait for the slowest calculation to be completed, resulting in waiting time overhead, which we call the barrel effect problem. When ignoring the performance difference between machines, all nodes complete a mini-batch calculation at the same time, and they will send the gradient value to the parameter server together, and the nodes that have not preempted resources will be put into the waiting queue. This is called a communication conflict.

D. DEEP NETWORK LEARNING ALGORITHM

1) GREEDY ALGORITHM

The greedy algorithm is usually based on the current situation and makes the best choice according to an optimization principle, without considering all situations. In some cases, it is fast, because it saves a lot of time to find the optimal solution to think about the whole, treat a big problem as multiple small sub-problems, and after every step of greedy choice, you get the optimal solution. The greedy algorithm does not require backtracking, but it should be noted that although each step obtained by the greedy algorithm is the current best case, the resulting global solution is sometimes not necessarily the best solution. Therefore, the greedy algorithm is often used as a way to find a better initial value. In the global case, the global optimization is usually performed at the end to achieve the optimal solution in the global case.

In order to optimize the objective function, the greedy algorithm generally starts from the first step of solving the problem. Each step selects the optimal solution set for the current state. In the next step, the current optimal solution is also selected. In this way, for each step of the greedy algorithm, it is necessary to calculate whether the current solution is feasible. If feasible, the object is collected into the collection, and if it is not feasible, it is discarded. Because the greedy algorithm selects the optimal state for each step, therefore, theoretically, if the optimal solution for each step holds, then usually the first solution found by the greedy algorithm is optimal.

In our deep network training and learning algorithm, the first step is to obtain the current optimal weight for each layer by unsupervised layer-by-layer training, and assign it to the corresponding layer of the multi-layer network. In this way, the initial value of the deep network is obtained.

2) BACK PROPAGATION ALGORITHM

The basic idea of backpropagation is that during the learning and training of network parameters, the training samples are input to the network, and an output value is obtained through the network. The difference between the output value and the expected value is calculated, and the difference is used as the basis for adjusting the weight between neurons in each layer of the network. Then you carry out the propagation operation to reduce the error and obtain a new output value, calculate the difference between the value and the expected value, and then adjust the weight between the neurons of each layer according to the smaller error signal from back to forward propagation. The architecture diagram of the deep learning prototype system for image classification is shown in Figure 5.

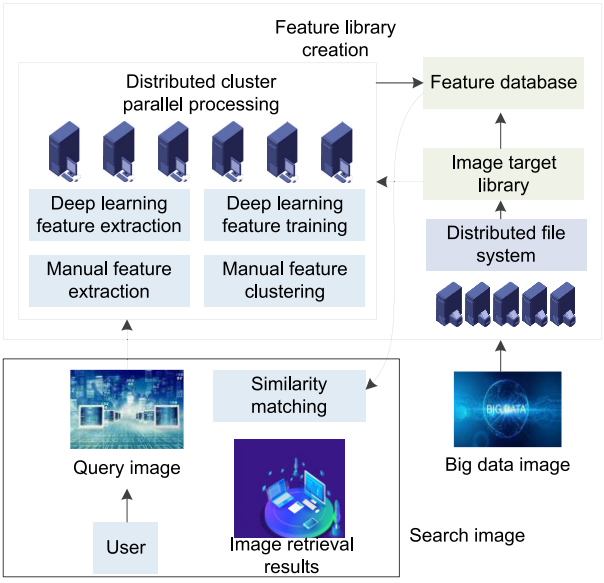


FIGURE 5. Image classification deep learning prototype system architecture diagram.

III. CONSTRUCTION OF REPRESENTATION LEARNING MODEL OF SEMI-SUPERVISED DISTRIBUTED DEEP ATTRIBUTE NETWORK BASED ON NEIGHBOR STRUCTURE

A. SEMI-SUPERVISED NETWORK REPRESENTATION LEARNING MODEL BASED ON NEIGHBOR STRUCTURE

The SLLDNE model was first proposed to solve the first problem of the Labeled Structural Deep Network Embedding (LSDNE) model: a large number of hyperparameters makes the model too complicated. LSDNE uses a compressed autoencoder, whose parameters are mainly derived from the balance parameters of first-order proximity and second-order proximity, that is, the balance between the minimum reconstruction error of the autoencoder and the Laplace feature mapping term. In addition to the parameters, there are penalty parameters for node reconstruction, normalization parameters, and classifier balance parameters. The flowchart of deep learning image target classification network framework is shown in Figure 6.

The main source of parameters comes from the encoder, so you consider replacing the autoencoder. Intuitively, they

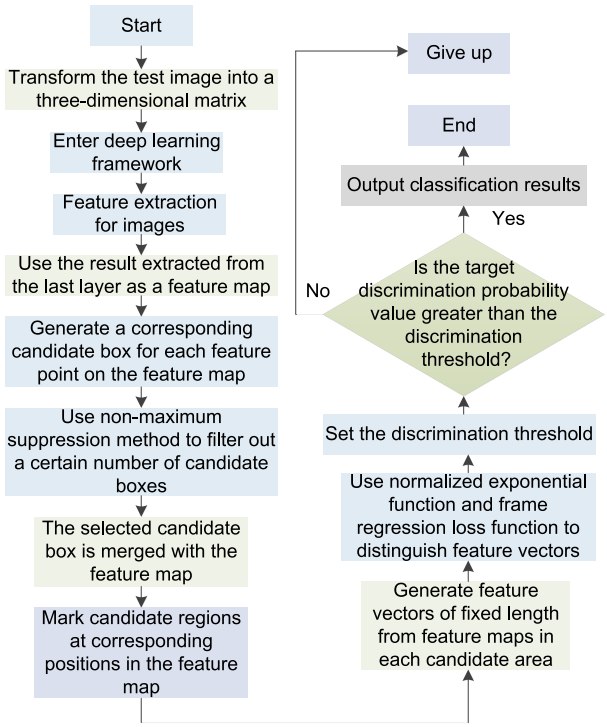


FIGURE 6. Flow chart of deep learning image target classification network framework.

replace the compressed autoencoder with a common deep neural network to reduce the reconstruction error term and Laplace feature mapping error in the autoencoder term, which can reduce the balance parameters of first-order proximity and second-order proximity. At the same time, because no auto-encoder is used, there is no need to punish the wrong items for reconstruction, so the penalty item parameters for node reconstruction are reduced.

You can use the direct push learning method to add a classification layer to the model, introduce labels while performing network representation learning, and learn both labeled and unlabeled nodes at the same time. In the training process, the sample distribution and label information will affect the final model generation, indicating that the learning process and the classification process affect each other. The final representation of the meaning of the learning results will be more targeted and more advantageous for the subsequent classification tasks.

However, it is not enough to do so. Because the ordinary deep neural network lacks the reconstruction process of the autoencoder, the model cannot obtain the characteristics that can represent the node from the input, which may cause the model to lack the generalization ability. Therefore, inspired by the Locally Linear Embedding (LLE) algorithm, the SLLDNE model will add the neighbor information of the node at the same time when learning the network representation. Through the node vector reconstruction, the node itself and the neighbor node together constitute the next layer of representation. In contrast to using only deep neural networks, the local features of each node are obtained from a single node in the upper layer into multiple nodes, which is equivalent to the model extracting more features in this way. The construction is equivalent to constraining the model parameters through neighbor nodes, which reduces the problem of insufficient generalization ability caused by a single node input.

Each node is treated as a deep neural network for feature learning, but for a single node i, the adjacency vector Yi(1) of the node at the time of its input is obtained by the weighted reconstruction of i and all its nodes. Each hidden layer after that is obtained by the weighted reconstruction of the neighbors of the previous layer, and finally the vector representation of all nodes is z(A).

B. OBJECTIVE FUNCTION

The design of SLLDNE's objective function is inspired by the LLE nonlinear dimensionality reduction algorithm. The LLE algorithm believes that each data point can be constructed from the linear weighted combination of its neighbors, and the local reconstruction weight matrix of each node is calculated by the adjacency matrix. You use the same weight matrix to construct the weighted combination of the neighbors of the node in the representation space of the node, and convert the dimensionality reduction problem into an eigenvalue decomposition problem. The purpose is to decompose the node representation to be the same as the weighted neighbor representation.

The objective function of LLE is shown in the following formula.

$$\Phi(Y) = \sum_i (\sum_j w_{ij} Y_j - Y_i)^2 \tag{1}$$

Among them, Y_i and Y_j are the representation vectors of the nodes, w_{ij} is the weighting of X_j, w_{ij} is a value between 0 and 1 when adjacent to i, and w_{ij} is 0 when i and j are not adjacent, $\sum_j w_{ij} Y_j$ represents the weighted reconstruction process of nodes.

In the SLLDNE model, to extend the weighted reconstruction of the shallow model to the deep neural network model, it is necessary to weight each hidden layer. The formula is shown below.

$$y_i^{k+1} = \sum_j w_{ij} y_j^k \tag{2}$$

where y is the hidden layer of the node, y_j is the neighbor node of y_i, w_{ij} is the weight matrix weighting the neighbor nodes, and because linear weighting is performed in the neural network, in order to avoid the hidden layer value being too large, the weighted value needs to be considered. The average of the number of neighbors of the node is equivalent to treating each data point as a linear weighted average of its neighbors, as shown in the following formula:

$$y_i^{k+1} = 1/D_i \sum_j w_{ij} y_j^k \tag{3}$$

where D_i represents the degree of node i. The next step is the process of constructing the weight matrix. In the

above formula, w_{ij} represents the reconstruction weight of the neighboring node. Intuitively thinking, the more important the neighbor node is, the greater its influence is, and w_{ij} represents the importance of the node. The matrix formed by the w_{ij} value is W_{ij}. In practical applications, a representative index for evaluating the importance of a node is the pagerank value, so here the value of the element in W_{ij} is set to the pagerank value, and the pagerank value is normalized with 1 as the average to obtain pr_norm.

Expressing the above formula as a matrix, we get:

$$Y^{k+1} = W_{ij}Y^k \otimes D^{-1}A \tag{4}$$

Among them, A represents the adjacency matrix (here, the special value $a_{ii} = 1$) and because in the neural network, the activation function needs to be introduced to perform the nonlinear transformation:

$$Y^{k+1} = \sigma(W^k) \bullet \sigma(Y^k) \tag{5}$$

Among them, W_k is the parameter matrix in the neural network, and σ represents the activation function (Relu activation function is used in this article), so it replaces Y_k in the formula to get the neural network model, as shown in the following formula:

$$Y^{k+1} = \sigma(W^k) \bullet \sigma(W_{ij}Y^k \otimes D^{-1}A) \tag{6}$$

For the output layer, the model chooses SVM as the classifier to classify the known labels. Because the model incorporates first-order and second-order into the model at the same time, the model no longer needs a decoder and calculation reconstruction error, nor does it need to design an objective function separately for the first-order, so the overall objective function of the model is the SVM classification error, linear support vector machine can be converted into an optimization problem as shown in the following formula on the problem of multi-classification:

$$L_{svm} = \sum_{k=0}^{K} \sum_{i} \max(1 - W^{s^T} z_i L_i^k, 0) + \theta \mathrm{Re}g \tag{7}$$

Among them, L_i is the real label of the node, W_s is the parameter matrix of the SVM classifier, z_i is the output of the model, that is, the representation of the node (embedding), Reg is the normalization term, and θ is the normalization parameter.

C. SEMI-SUPERVISED DISTRIBUTED DEEP ATTRIBUTE NETWORK REPRESENTATION LEARNING MODEL BASED ON NEIGHBOR STRUCTURE

The SLLDNE model can well solve the problem of using LSDNE to express too many learning parameters of the attribute network, and for node attributes, LSDNE cannot combine attributes with the second-order proximity of the network. This problem can also be solved by using SLLDNE, that is, when processing nodes with the same neighbors, the nodes can obtain the same representation through the weighted reconstruction of the neighbors. Therefore, even when processing the network node attributes, the nodes can also use the neighbor weighted reconstruction. At the same time, adjacent nodes will become one of each other's reconstructed nodes, so when dealing with network node attributes, it also incorporates first-order proximity.

In this section, a semi-supervised deep attribute network representation learning model SILDDNE based on neighbor structure is proposed. This model is based on the SLLDNE model, but the difference is that SILDDNE adds node attribute information. SILDDNE separately extracts the structural features and node attribute features of the nodes, merges them by weighting, and then uses the SVM classifier to classify the labeled nodes to classify the labels, and learns the representation of the nodes through the Adam optimization strategy.

SILDDNE is composed of SLLDNE and classifier SVM. For the network features, the input of the model is the adjacency matrix, as shown in the green dots, through the neighbor vector Vector i(A) of the node i through the deep neural network, the weighted reconstruction of the neighbors in the upper layer is obtained, and finally the structural characteristics z(A) of all nodes are obtained. For attribute features, the input of the model is an attribute matrix, as shown in the orange node, through the deep neural network through the vector i(F) of the node i's attribute vector, each hidden layer is obtained by the weighted reconstruction of the neighbors of the previous layer. You obtain the attribute characteristics z(F) of all nodes. Finally, the representation vector z of the node is obtained by weighting the structural features z(A) and attribute features z(F) of the node, as shown in the following formula:

$$z = (1 - \alpha)z(F) + z(A) \tag{8}$$

After getting z, for the nodes with known labels, you use the direct push learning method, add a SVM classification layer to the model, and learn the labeled nodes and the unlabeled nodes at the same time. As a result, the final representation of the learning results is more advantageous for the subsequent classification tasks.

Using SILDDNE for the representation learning of attribute networks, the hyperparameters are only the balance parameter α and the normalization parameter γ of the network structure characteristics and network attribute characteristics, which is greatly reduced compared with LSDNE. Specifically, after adding attribute information, there are 7 LSDNE parameters, as shown in Table 1.

SILDDNE has only 2 hyperparameters, and the model becomes more concise without reconstruction error of the autoencoder and Laplace feature map, reducing overfitting.

IV. BIG DATA IMAGE CLASSIFICATION SIMULATION EXPERIMENT

A. EXPERIMENTAL DATA SET

The CIFAR dataset contains CIFAR-10 and CIFAR-100, which is a subset of 80 million tiny images. Both datasets have 60,000 32×32 pixel images, of which 50,000 images are used for training and 10,000 images are used for testing.

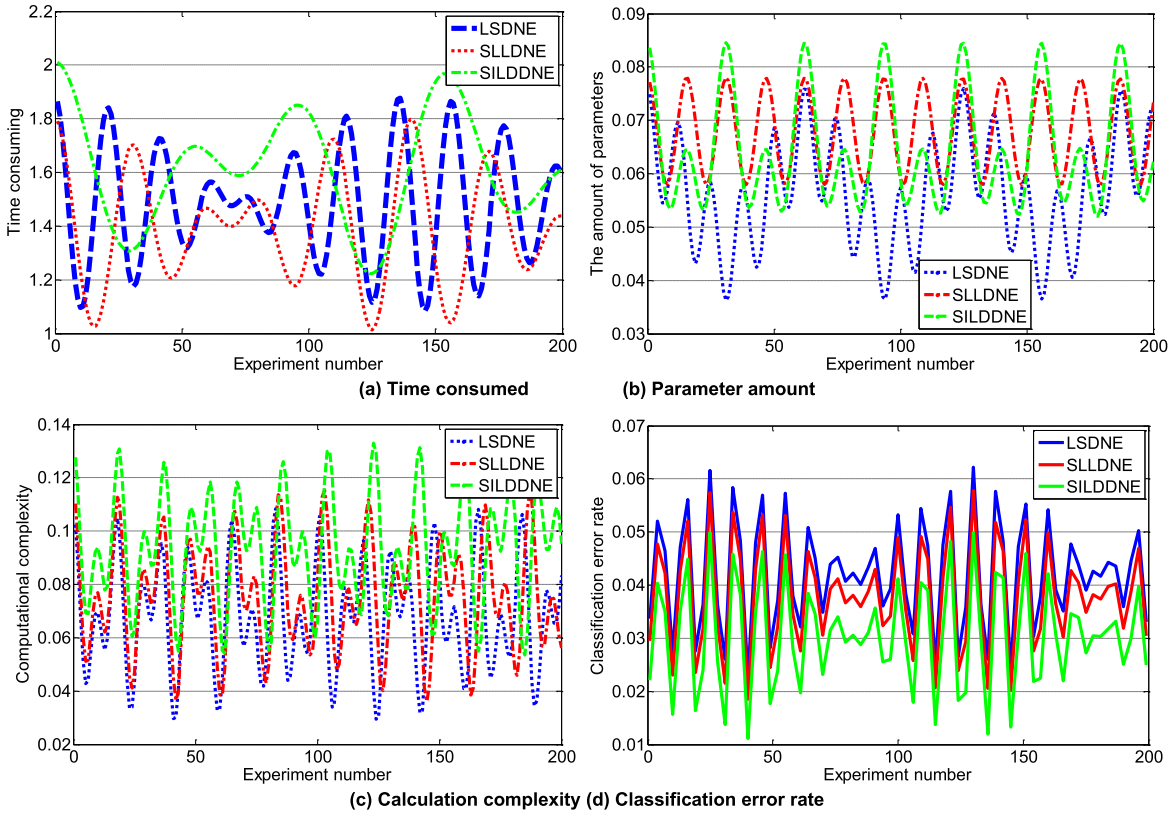


FIGURE 7. Deep network performance comparison under different template modules (CIFAR-10).

TABLE 1. LSDNE parameters.

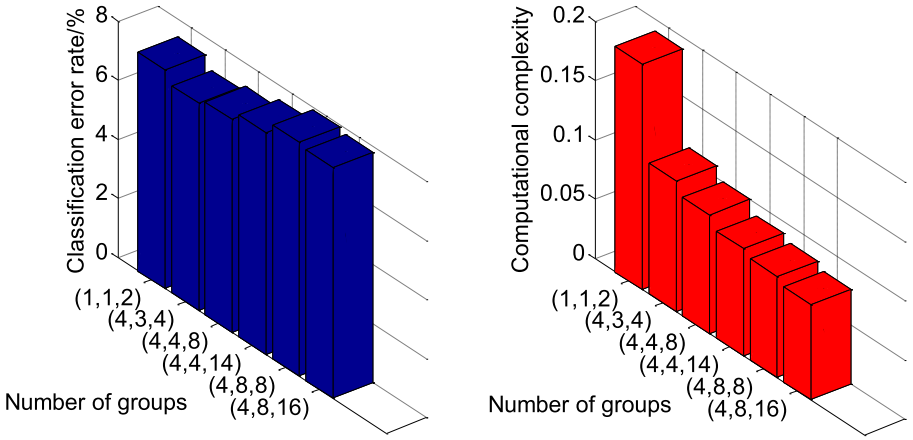
Parameter name	Parameter description
α	Balance parameters of first-order proximity and second-order proximity
β	Balance parameters of unsupervised model and classifier
$\overline{bal}$	Balance parameters of network structure characteristics and attribute characteristics
θ	Normalized parameters
$\overline{\gamma}$	Normalized parameters of attribute network compression autoencoder
$\overline{\alpha}$	Balance parameter of first-order proximity and second-order proximity of attribute network compression autoencoder
b	Node reconstruction error weighted weight parameters

The CIFAR-10 dataset has 10 classes, each class contains 6000 images, 5000 for training and 1000 for testing. The CIFAR-100 dataset has 100 classes, each of which contains 600 images, 500 for training and 100 for testing. This article follows standard data amplification strategies for the CIFAR data set. First, we fill the boundary of each image in the data set with 4 0 pixels, and expand to 40 × 40 pixels; then randomly crop the image block with a size of 18 × 18, and randomly mirror half of the image horizontally for data amplification, and finally use channel subtraction for normalized image preprocessing.

B. DEEP NETWORK PERFORMANCE ANALYSIS

1) PERFORMANCE COMPARISON OF DIFFERENT TEMPLATE MODULES

Based on the above observations and analysis, this section compares the performance of the proposed deep network composed of three different template module stacks. First, the number of branch channels L of the network is fixed to 2, and the number of layers of the network is fixed to 22 layers; then, the initial number of packets is set to (4, 3, 4). The top-1 error rate is used to evaluate network performance, and the expanded CIFAR-10 data set is used as experimental data. The performance test results are shown in Figure 7. In general, the higher the floating-point operation value, the longer it takes to train on average. It can be seen from the experimental results that SLLDNE achieves a lower classification error rate than LSDNE with the same parameter amount and floating-point operation value. This shows that the introduction of interleaved connections between branch channels promotes the fusion and transfer of feature information between different convolutions. Similarly, the number of SILDDNE parameters has been improved by almost 50% compared with floating-point arithmetic values, and has



(a) Classification error rate (b) Calculation complexity

FIGURE 8. SILDDNE performance comparison under different group numbers (CIFAR-10).

achieved the lowest classification error rate. This shows that the introduction of channel fusion strategy increases the computational complexity and parameter amount within a certain tolerance range, but it further optimizes the information flow between channels and improves the generalization ability of the model.

2) GROUP NUMBER ANALYSIS

In this study, this section explores the change of network performance with the number of packets, and uses the pooling layer as the transition layer of the network structure, and the modules between the two adjacent transition layers have the same number of packets. We use SILDDNE as the basic network for the experiment, and set the number of packets to vary from 1 to 16. In order to effectively evaluate the network performance, the number of branch channels L of the network is fixed at 2, and the number of layers of the network is fixed at 22. This means that the computational complexity of the model changes as the number of packets changes. Figure 8 shows the performance comparison of SILDDNE on the CIFAR-10 data set as the number of packets changes. It can be seen from Figure 8(a) that when the number of packets is (4, 3, 4), SILDDNE can achieve a lower classification error rate than the original model. As can be seen from Figure 8(b), as the number of packets increases, the computational complexity of the network shows a decreasing trend and compresses nearly 50% of the computation, but the classification error rate shows an upward trend. This shows that the network can reduce the amount of computing resources required by introducing packet convolution. However, for lightweight networks, too low compression calculation will also cause a proportional loss of classification accuracy. It is worth noting that when the number of packets is (4, 3, 4), the three combined modules between the transition layers have the same computational complexity. Therefore, on the premise of lossless network performance, follow-up experimental research will use (4,3,4) as the standard number of packets.

3) ANALYSIS OF THE NUMBER OF BRANCH CHANNELS

In this study, this section uses the CIFAR-10 data set as an example to demonstrate empirically how the number of branch channels affects network performance. In order to effectively evaluate the network performance, SILDDNE with the number of packets (4, 3, 4) is used as the basic network for the experiment. Figure 9 shows how the network's computational complexity and classification error rate vary with the number of channels. The performance of the network will vary with the number and depth of branch channels. On the one hand, when the network has the same depth, as the number of branch channels increases, the network computing complexity shows an increasing trend. On the other hand, when the network increases with depth, the computational complexity of all networks is increasing, but the classification error rate is decreasing.

4) DEPTH AND WIDTH ANALYSIS OF THE NETWORK

This section also includes depth and width experiments to explore performance changes as the network deepens and widens. In this study, the number of groups is fixed at (4,3,4). For the consideration of computing resources, only the {22,28,34} layer network was constructed when the dimension of SILDDNE's feature graph was expanded.

When the dimension of the network's feature graph is fixed at {4,8,16}, Figure 10 shows the performance comparison of LSDNE, SLLDNE and SILDDNE on the CIFAR-10 data set as the number of network layers deepens. It can be seen that the classification error rates of LSDNE, SLLDNE, and SILDDNE all decrease as the number of network layers deepens. While SLLDNE and SILDDNE obtain the lowest classification error rate, SILDDNE has more computational complexity than SLLDNE.

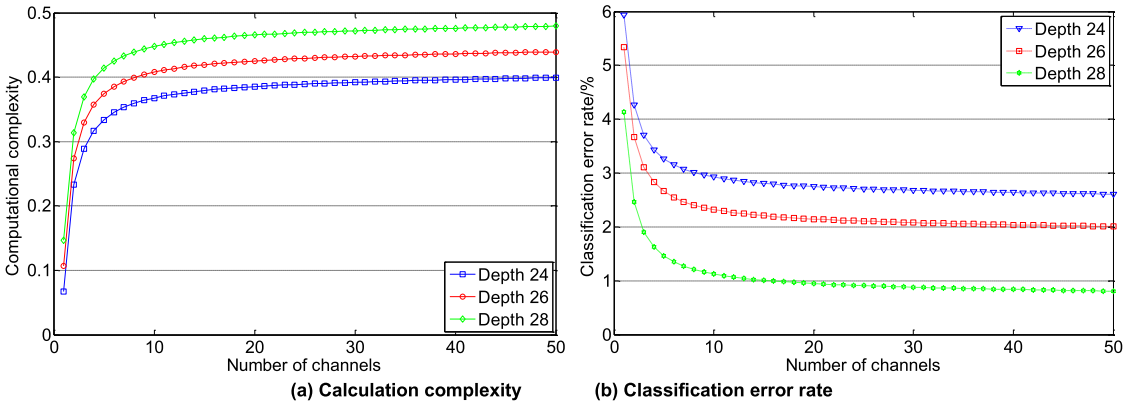


FIGURE 9. Comparison of deep network performance under different branch channel numbers (CIFAR-10).

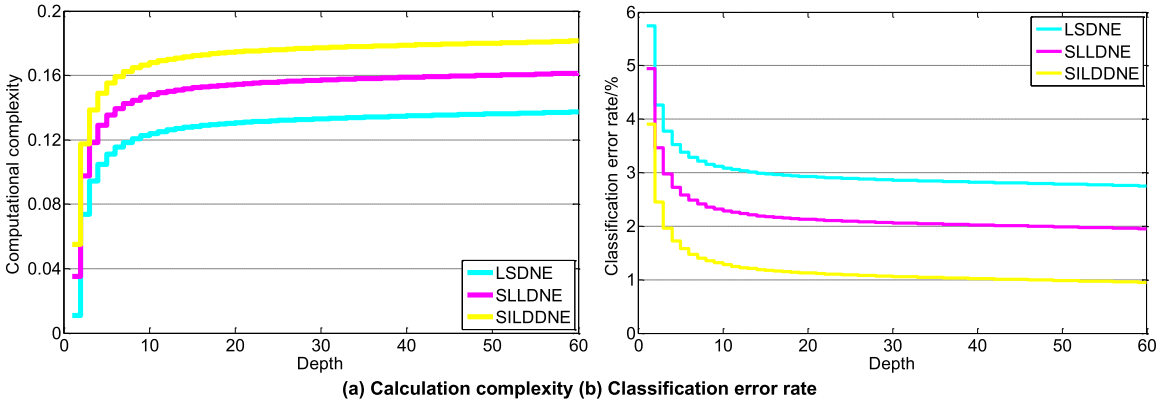


FIGURE 10. Performance comparison of deep networks with different depths and feature graphs with dimension {4,8,16} (CIFAR-10).

When the dimension of the network's feature graph expands to {8,16,32}, Figure 11 shows the performance comparison of LSDNE, SLLDNE and SILDDNE on the CIFAR-10 data set as the number of network layers deepens. It can be seen that the expansion of the dimension of the feature graph increases the width of the network, so that its performance has a significant increase. With the further deepening of the number of layers, the parameter quantity and computational complexity of the network increase, and the classification error rate also gradually increases. It is worth noting that whether you increase the depth of the network or expand the width of the network will bring about improved network performance. However, the benefit of using a relatively small depth for width expansion is greater than the benefit of increasing depth, which is consistent with the observation of conventional convolution.

C. EXPERIMENTAL RESULTS OF IMAGE CLASSIFICATION

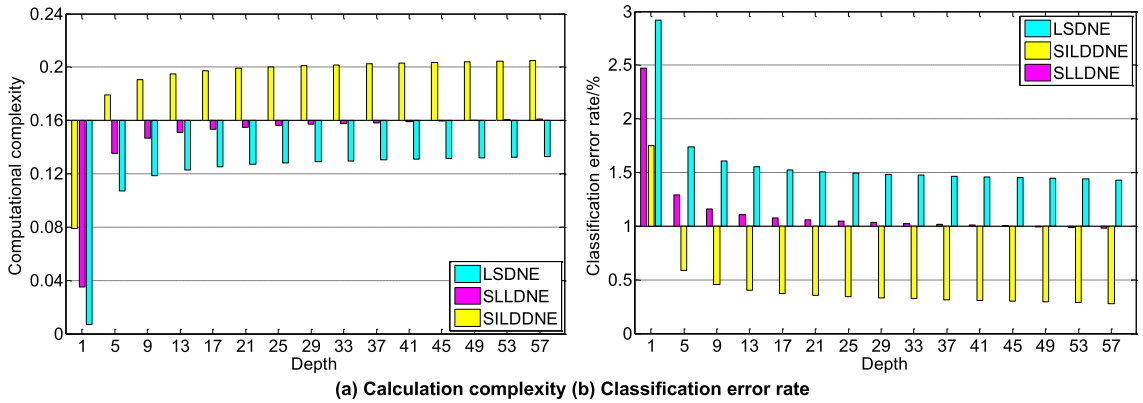
In this work, the top-1 error rate is mainly used to evaluate the deep network structure proposed in this paper.

1) COMPARED WITH EXISTING WEIGHT PRUNING TECHNOLOGY

Comparing the calculation efficiency of the deep network with the existing model obtained by weight pruning technology, the experimental results are shown in Table 2. It can be seen that on the CIFAR-10 data set, the 34-layer SLLDNE pruned 40-layer Dense Net has compressed nearly 50% in the amount of parameters, increased the computational complexity efficiency by nearly 8 times, and reduced the classification error rate nearly 30%. On the CIFAR-100 dataset, the 34-layer SLLDNE parameter volume is compressed by nearly 16 times compared to the 19-layer VGG parameter volume, the computational complexity efficiency is increased by nearly 6 times, and the classification error rate

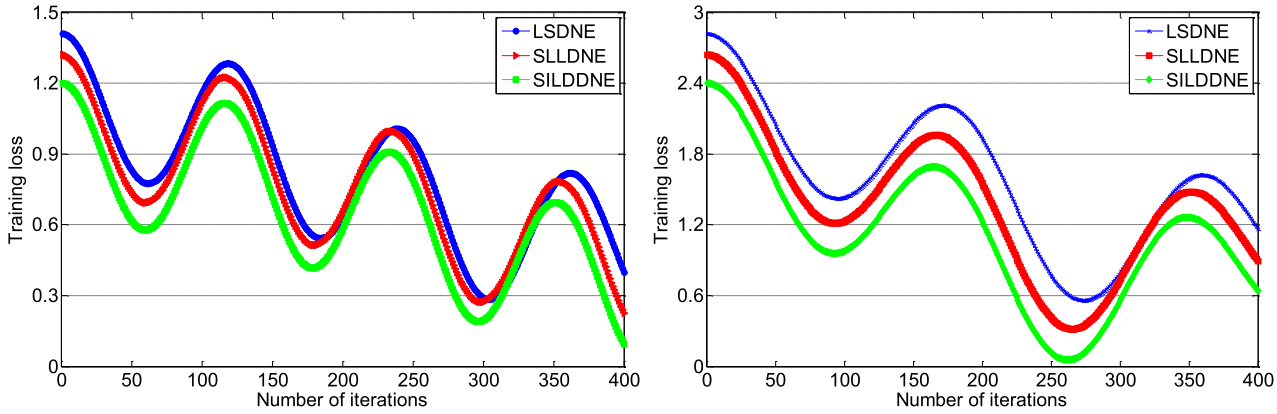
TABLE 2. Comparison of classification error rates with existing weight pruning methods (CIFAR-10/CIFAR-100).

Network type	Computational complexity	CIFAR-10 (%)	CIFAR-100 (%)
Res Net-56-pruned	1.1	12.1	28.0
Res Net-110-pruned	1.2	9.0	27.2
Res Net-164-pruned	1.4	8.1	27.3
Dense Net-40-pruned	1.1	6.0	24.1
LSDNE	0.5	5.1	19.0
SLLDNE	0.4	3.1	17.0
SILDDNE	0.8	2.0	16.2



(a) Calculation complexity (b) Classification error rate

FIGURE 11. Comparison of deep network performance at different depths and feature graph dimensions {8, 16, 32} (CIFAR-10).



(a) Loss function curve (CIFAR-10) (b) Loss function curve (CIFAR-100)

FIGURE 12. Loss function curves of LSDNE, SLLDNE and SILDDNE (CIFAR-10/CIFAR-100).

is reduced by nearly 14%. Compared with other models, the 34-layer SILDDNE and 40-layer LSDNE also have obvious advantages. Moreover, the 34-layer SILDDNE obtained the best classification results on the CIFAR-10 and CIFAR-100 datasets, respectively.

2) PARAMETER EFFICIENCY

Figure 12 shows the loss function curve of the proposed deep network on the CIFAR-10 and CIFAR-100 datasets. It can be observed that as the number of network iterations increases, the three proposed deep networks all converge to a lower training loss value. In the initial stage of network training, SILDDNE has the lowest loss function value, especially on the CIFAR-10 data set. From the first Epoch to the 400th Epoch, the loss function values of SILDDNE are lower than those of LSDNE and SLLDNE. This shows that as the amount of parameters increases, SILDDNE is easier to optimize than LSDNE and SLLDNE.

It can also be observed from the two data sets that SLLDNE and LSDNE have a large reduction in the amount of SLLDNE parameters when the loss function curve trend is similar. This shows that the interleaved network structure can further optimize the exchange of information between different convolutional layers, thereby improving the utilization of features. It is worth noting that from the 300th Epoch to the 350th Epoch, SILDDNE has only a similar and slightly smoother loss function curve compared to LSDNE and SLLDNE on the two data sets, and there is no obvious advantage. Although the introduction of fusion strategy SILDDNE further improves the network's feature expression capabilities, it also increases the network's parameter volume and training pressure. This is one of the reasons why SILDDNE has not significantly reduced its classification error rate on the CIFAR-100 dataset.

V. CONCLUSION

This paper introduces the data parallel and model parallel structure in the distributed deep learning model in detail, and analyzes its advantages and disadvantages. The two tools of deep learning, Caffe and Caffe On Spark, are introduced, analyzed and compared, and their advantages and disadvantages are summarized, which lays the foundation for the distributed deep learning image classification studied in this article. In a real-world network, in addition to the network structure and node labels, the nodes themselves will contain various attributes, which can be integrated into the learning process of the network representation as side information. When LSDNE is applied to attribute networks, a large number of hyperparameters will be generated, which will greatly

increase the complexity of the model and reduce the efficiency of model training. In order to make better use of the network node's own attributes and reduce model parameters and complexity, this paper proposes a semi-supervised network representation learning model based on neighbor structure. On the basis of SLLDNE, node attributes are further added, and SILDDNE is proposed. The SVM classifier classifies the known labels, and SILDDNE fuses the network structure, labels, and node attributes into the deep neural network. Experimental comparisons are made on the CIFAR-10 and CIFAR-100 datasets. The proposed SILDDNE model achieves higher image classification accuracy while reducing the number of parameters and improving computational efficiency. Although various existing algorithms have used the structural characteristics of the network to varying degrees, in the actual network, there are still many special structural characteristics that have not been used, such as community structure and network mode. In the network, most nodes are only connected with a small number of edges, so it is difficult to learn effective vector representations with limited information. This problem will be solved in the next research.

REFERENCES

[1] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. Atli Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[2] Y. Ariji, M. Fukuda, Y. Kise, M. Nozawa, Y. Yanashita, H. Fujita, A. Katsumata, and E. Ariji, "Contrast-enhanced computed tomography image assessment of cervical lymph node metastasis in patients with oral cancer by using a deep learning system of artificial intelligence," *Oral Surg., Oral Med., Oral Pathol. Oral Radiol.*, vol. 127, no. 5, pp. 458–463, May 2019.

[3] Y. Chen, K. Zhu, L. Zhu, X. He, P. Ghamisi, and J. A. Benediktsson, "Automatic design of convolutional neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7048–7066, Sep. 2019.

[4] P. R. Jeyaraj and E. R. Samuel Nadar, "Computer-assisted medical image classification for early diagnosis of oral cancer employing deep learning algorithm," *J. Cancer Res. Clin. Oncol.*, vol. 145, no. 4, pp. 829–837, Jan. 2019.

[5] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, Jul. 2019.

[6] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[7] X. Lv, D. Ming, Y. Chen, and M. Wang, "Very high resolution remote sensing image classification with SEEDS-CNN and scale effect analysis for superpixel CNN classification," *Int. J. Remote Sens.*, vol. 40, no. 2, pp. 506–531, Jan. 2019.

[8] S. Zhou, Z. Xue, and P. Du, "Semisupervised stacked autoencoder with cotraining for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3813–3826, Jun. 2019.

[9] X. He and Y. Chen, "Optimized input for CNN-based hyperspectral image classification using spatial transformer network," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 12, pp. 1884–1888, Dec. 2019.

[10] F. Özyurt, T. Tuncer, E. Avci, M. Koç, and İ. Serhatlioğlu, "A novel liver image classification method using perceptual hash-based convolutional neural network," *Arabian J. for Sci. Eng.*, vol. 44, no. 4, pp. 3173–3182, Apr. 2019.

[11] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A CNN with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019.

[12] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.

[13] S. Pang, A. Du, M. A. Orgun, and Z. Yu, "A novel fused convolutional neural network for biomedical image classification," *Med. Biol. Eng. Comput.*, vol. 57, no. 1, pp. 107–121, Jan. 2019.

[14] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach, "Comparison of deep learning approaches for multi-label chest X-ray classification," *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, Apr. 2019.

[15] R. R. Saritha, V. Paul, and P. G. Kumar, "Content based image retrieval using deep learning process," *Cluster Comput.*, vol. 22, no. S2, pp. 4187–4200, Mar. 2019.

[16] M. Chen, S. Lu, and Q. Liu, "Uniform regularity for a Keller–Segel–Navier–Stokes system," *Appl. Math. Lett.*, vol. 107, Sep. 2020, Art. no. 106476.

[17] C. F. Higham and D. J. Higham, "Deep learning: An introduction for applied mathematicians," *SIAM Rev.*, vol. 61, no. 3, pp. 860–891, Nov. 2019.

[18] T. S. Borkar and L. J. Karam, "DeepCorrect: Correcting DNN models against image distortions," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6022–6034, Dec. 2019.

[19] W. Wu, H. Li, X. Li, H. Guo, and L. Zhang, "PolSAR image semantic segmentation based on deep transfer learning–realizing smooth classification with small training sets," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 977–981, Jun. 2019.

[20] Y. Qin, L. Bruzzone, B. Li, and Y. Ye, "Cross-domain collaborative learning via cluster canonical correlation analysis and random walker for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3952–3966, Jun. 2019.

[21] S. Law, C. I. Seresinhe, Y. Shen, and M. Gutierrez-Roig, "Street-frontage-net: Urban image classification using deep convolutional neural networks," *Int. J. Geographical Inf. Sci.*, vol. 34, no. 4, pp. 681–707, Apr. 2020.

[22] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.

[23] X. Liu, R. Zhang, Z. Meng, R. Hong, and G. Liu, "On fusing the latent deep CNN feature for image classification," *World Wide Web*, vol. 22, no. 2, pp. 423–436, Mar. 2019.

[24] J. R. Hagerty, R. J. Stanley, H. A. Almubarak, N. Lama, R. Kasmi, P. Guo, R. J. Drugge, H. S. Rabinovitz, M. Oliviero, and W. V. Stoecker, "Deep learning and handcrafted method fusion: Higher diagnostic accuracy for melanoma dermoscopy images," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 4, pp. 1385–1391, Jul. 2019.

[25] S. Mahdizadehaghdam, A. Panahi, H. Krim, and L. Dai, "Deep dictionary learning: A PARametric NETwork approach," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4790–4802, Oct. 2019.

[26] M. Andrews, M. Paulini, S. Gleyzer, and B. Poczos, "End-to-end physics event classification with CMS open data: Applying image-based deep learning to detector data for the direct classification of collision events at the LHC," *Comput. Softw. Big Sci.*, vol. 4, no. 1, pp. 1–14, Mar. 2020.

[27] S. Antholzer, M. Haltmeier, and J. Schwab, "Deep learning for photoacoustic tomography from sparse data," *Inverse Problems Sci. Eng.*, vol. 27, no. 7, pp. 987–1005, Jul. 2019.

[28] X. Zhu, Z. Li, X.-Y. Zhang, P. Li, Z. Xue, and L. Wang, "Deep convolutional representations and kernel extreme learning machines for image classification," *Multimedia Tools Appl.*, vol. 78, no. 20, pp. 29271–29290, Oct. 2019.

[29] K. Z. Haider, K. R. Malik, S. Khalid, T. Nawaz, and S. Jabbar, "Deepgender: Real-time gender classification using deep learning for smartphones," *J. Real-Time Image Process.*, vol. 16, no. 1, pp. 15–29, Feb. 2019.

[30] J. Zhou, L. Luo, Q. Dou, H. Chen, C. Chen, G. Li, Z. Jiang, and P. Heng, "Weakly supervised 3D deep learning for breast cancer classification and localization of the lesions in MR images," *J. Magn. Reson. Imag.*, vol. 50, no. 4, pp. 1144–1151, Mar. 2019.

[31] X. Wang, W. Zhang, X. Wu, L. Xiao, Y. Qian, and Z. Fang, "Real-time vehicle type classification with deep convolutional neural networks," *J. Real-Time Image Process.*, vol. 16, no. 1, pp. 5–14, Feb. 2019.

[32] Y. Ariji, Y. Yanashita, S. Kutsuna, C. Muramatsu, M. Fukuda, Y. Kise, M. Nozawa, C. Kuwada, H. Fujita, A. Katsumata, and E. Ariji, "Automatic detection and classification of radiolucent lesions in the mandible on panoramic radiographs using a deep learning object detection technique," *Oral Surg., Oral Med., Oral Pathol. Oral Radiol.*, vol. 128, no. 4, pp. 424–430, Oct. 2019.

MINJUN ZHU was born in Fujian, China, in 1982. She received the bachelor's and M.Ed. degrees from Fujian Normal University, in 2003 and 2015, respectively, where she is currently pursuing the Ph.D. degree in mathematics education with the College of Mathematics and Information. She worked as a Mathematics Teacher at senior high school, from 2003 to 2018. She has published seven articles, and her research directions are mathematics education and testing research.

QINGHUA CHEN was born in Fujian, China, in 1962. He received the bachelor's and master's degrees from Fujian Normal University, in 1984 and 1989, respectively, and the Ph.D. degree in mathematics from Xiamen University, in 2004. He worked as a Teacher at Fujian Normal University. He is currently a Professor with Fujian Normal University, and the Deputy Dean of the College of Mathematics and Information, Fujian Normal University. He has published more than 100 articles, and his research directions are homological algebra, algebraic representation theory, and testing research.