Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9726286, 10 pages
https://doi.org/10.1155/2022/9726286

Research Article

Deep Learning Distributed Architecture Design Implementation for Computer Vision

Yizhong Zhang

Software Engineering Institute, East China Normal University, Shanghai 200062, China

Correspondence should be addressed to Yizhong Zhang; yizhong.zhang@stu.ecnu.edu.cn

Received 16 May 2022; Revised 28 June 2022; Accepted 6 July 2022; Published 20 July 2022

Academic Editor: Kuruva Lakshmanna

In the era of big data, to achieve an efficient deep learning and computer vision system for big data, developers need to build a computerized deep learning and computer vision system, and the system can simultaneously complete the tasks of deep learning and computer vision and large-scale data processing. The existing training dataset is reused, and the scene information is small, which cannot meet the needs of large-scale machine training, so it is necessary to include large-scale data, distributed computer system to complete the training. How to meet the training accuracy requirements of deep learning models and minimize the resource cost within the constrained time is a major challenge for distributed deep learning systems. Resource and batch size hyperparameter allocation are the main approaches to optimize the training accuracy and resource cost of models. Existing works have independently configured resources and batch size hyperparameters in terms of computational efficiency and training accuracy, respectively. However, the impact of the two types of configurations on model training accuracy and resource cost has complex dependencies, and it is difficult to achieve the goals of satisfying the model training accuracy requirements and minimizing the resource cost simultaneously by the existing independent configuration methods. To address these problems, this paper proposes a collaborative resource-batch size optimization configuration method for distributed deep learning systems. This method was firstly based on the monotonic function relationship between resource allocation and batch size hyperparameter allocation and model training time and training accuracy, and we select the order-preserving regression theoretical tool to build a model prediction model for single-round complete training time and final training accuracy for computer vision target classification and recognition, respectively; then, we use the abovementioned models together to solve the resource and batch size optimal allocation solutions to meet the model training accuracy requirements with the goal of minimizing resource cost. The optimal allocation of resources and batch size to meet the training accuracy requirements of the model is solved. In this paper, we evaluate the performance of the proposed method for computer vision target recognition based on the proposed distributed deep learning system.

1. Introduction

With the rise of big data, the use and rapid development of big data in the field of deep learning and computer vision has been promoted, and the deep learning and computer vision based on big data can complete machine training more effectively and accurately. Deep learning and computer vision and big data analysis are techniques that use existing computer system models to convert a large amount of data or big data, acquired by computers into useful information [1–3]. The larger the scale of the data used, the better the training effect of deep learning and computer vision, the more accurate, the more content recognition, and the reduction of overfitting and underfitting phenomena. Big data deep learning and computer vision is not just deep learning and computer vision, nor is it simply a matter of big data processing but involves the use of both deep learning and computer vision and big data processing to overcome the technical problems and integration. In this process, researchers not only need to continue to focus on deep learning and computer vision function methods and algorithms but also need to continue to research new, efficient algorithms or improve the existing imperfect deep learning and computer vision methods, to ensure the accuracy of the results in the actual operation. Building a deep learning and computer vision system based on big data involves both deep learning and computer vision and big data processing, such as algorithm models, data sets, training methods, accuracy, fitting and other issues of deep learning and computer vision, distributed storage, parallelized computing, network communication, task scheduling, fault-tolerant redundancy, and backup in big data processing. These factors affect each other, increasing the complexity of system design and the stability and accuracy of the completed system, bringing some challenges to the designers in system development and design. When designing big data deep learning and computer vision systems and studying their methods and algorithms, attention is also paid to how to combine distributed and parallelized big data processing techniques in order to complete the computation in an acceptable time [4–6].

In recent years, with the arrival of big data and the rapid development of artificial intelligence, especially deep learning, deep neural network models have made breakthroughs and been widely used in many fields, including speech recognition, image recognition, and natural language processing. As the application scenarios of deep learning and computer vision continue to be explored in depth. Research on deep learning-based commodity recognition applications is accelerating. Commodity recognition, as a technology with great commercial value, has entered the research field of research institutions and enterprises. Internet technology companies are developing and laying out deep learning-based commodity recognition methods. As this technology matures, deep learning-based merchandise recognition methods will certainly have a broad future and bring new changes and transformations to the existing retail or logistics fields. Image classification has been a popular research topic in the field of computer vision, and deep learning extracts the combined underlying features by directly inputting the lowest-level image information to form more abstract high-level features [5].

The application scenarios of deep learning and computer vision continue to be explored in depth by various research studies. Research on deep learning-based commodity recognition applications is accelerating. As a technology of great commercial value, commodity identification has entered the research field of research institutions and companies. Internet technology companies are developing and laying out deep learning-based commodity recognition methods. As this technology matures, deep learning-based product identification methods will certainly have a broad future and bring new changes and transformations to existing retail or logistics fields. Image classification has been a hot research topic in computer vision, and deep learning extracts combined low-level features by directly inputting the lowest-level image information to form more abstract high-level features. Deep learning algorithms can obtain a better representation of image features due to the increased number of layers of feature extraction. Deep learning is a new area of research in machine learning, a multilayer neural network used to simulate the human brain for analytical learning. It mimics the mechanisms of the human brain to interpret data, such as images, speech, and text. Deep learning models can be classified into different types depending on the hidden layer structural units. However, for the specific practical application of image classification, convolutional neural network models are widely used. The labeled image dataset is sent to the convolutional neural network for training, and the convolutional neural network is allowed to learn the optimal classification model automatically. Then, the trained model can be used to automatically classify the unlabeled images.

Deep learning improves itself by constantly iterating through derivations to update the model, which requires a lot of computation and is typically a computationally intensive task, making the training process of these neural networks very time-consuming.

The unique contribution of the paper includes the following:

(i) Development of a model based on self-supervised contrastive learning

(ii) This SimCLR model is implemented in association with fine tuning technique in transfer learning

(iii) The model provides enhanced accuracy and generalization ability on small commodity datasets

The organization of the paper is as follows: Section 2 discusses the related studies, Section 3 presents the algorithm design followed by the experimental results in Section 4, and the conclusion is in Section 5.

2. Related Works

2.1. Computer Vision. Computer vision technology refers to the visual process of observing and analyzing images through computers that simulate human vision. It requires computers to have the ability to use images to perceive the surrounding environment in the process of artificial intelligence, to simulate the specific process of human visual functions, and then to achieve intelligent processing of the relevant images. Computer vision technology is an artificially intelligent technology that simulates the process of human perception of the environment, so the technology incorporates a number of disciplines and technologies, including image processing, artificial intelligence, and digital technology [6–9]. This technology has a very important role in the development of computers; especially in modern society, people need computers to complete more intelligent behavior and to replace humans to solve some special environment work. In addition to the development of computer vision technology in the process of application, but also in the mechanized production has a certain application, in the future of automated production of machinery, the technology can be used to extract the image of objective things and then used in the production process of detection and control of the technology; compared to the traditional automation control, it can achieve faster, more information, and more functional control.

Computer vision technology, also known as image understanding, is the study of how to obtain the visual information required by the task from the image in a particular environment and, in more general terms, to use the relevant image processing methods to get the image information people want. The main research content and purpose of computer vision technology is threefold: first, the data analysis of the image, the use of reference objects in the image to calculate the distance between objects, so as to obtain the distance data in the image. The second is to analyze the image and learn some motion parameters of the object in motion through the data in the image. Thirdly, the calculation and analysis of the image are used to understand some physical characteristics and related parameters of the specific object in the image. Through the above three data points, it is possible to have a deeper understanding of the specific object in the image and get specific information about the object, but because the computer cannot realize the recognition of the three-dimensional image, it usually has to be converted into a two-dimensional image projection, through one or more two-dimensional image projection to achieve the analysis of the object data [10–13].

Computer vision technology involves a relatively large number of disciplines and technologies and usually requires the study of the technology from multiple perspectives to achieve the development of computer vision technology. The goal of computer vision technology is to achieve human-like recognition and processing of images to obtain intelligent data, but the current technology is not able to achieve such an image acquisition effect, which requires continuous research from multiple perspectives [14]. First, the main purpose of vision technology is to achieve the recognition and processing of images, so the first task is to achieve a technological breakthrough in the image equipment. Seek breakthroughs in optical components to ensure that the process of image acquisition can achieve high-definition or even into the 3D image acquisition technology, but also from the computer hardware to improve the relevant performance of the computer. Secondly, we need to improve the computer algorithm and data processing methods, so that the computer can analyze and process images more quickly, which requires seeking technical breakthroughs in computer software. The system related to the implementation of digital technology in image processing can effectively use the relevant theoretical knowledge of the computer to realize the data conversion and image analysis within the system.

2.2. Distributed Computing Systems. In recent years, the research of convolutional neural network has made great progress, and many excellent network models have emerged, extending the depth of neural network to hundreds of layers. Although these complex network models bring the improvement of recognition performance, they are accompanied by a huge amount of computation and a long training time. The hardware architecture of computing platform is updated and iterated, and the computing capability is rapidly improved, especially the rapid development of multicore and distributed computing platform, which provides the hardware foundation for the parallelization of deep neural network. On the other hand, the increasingly rich parallel programming framework also provides a bridge between the computing platform and the parallel training of deep neural networks. Distributed computing system has been widely used in face recognition [15], satellite communication [16], big data analysis [17], and other tasks.

There are two main ways to parallelize the training of deep neural networks using distributed clusters: model-parallel and data-parallel. Model parallelism divides the network model into different computational nodes according to certain rules, and each computational node is responsible for handling a part of the computational tasks of the model [18]. During each iteration, the intermediate results of the computation nodes need to be synchronized. Depending on the division method, different layers of the network can be divided into different computational nodes, or the same layer of the network can be partitioned and divided into different nodes. Data parallelism is to divide the training data set equally into subsets with the same number of computational nodes, and then, each computational node trains a copy of the model in the corresponding subdataset, and each computational node is independent of each other during the computation [19]. Each compute node communicates and updates gradient information through a parameter server, which is responsible for maintaining the latest parameter status of the model. The parameter server is responsible for collecting the gradient information calculated by each node, updating the model parameters on the parameter server according to the collected gradient, and sending the latest parameters to each node after the update. It is because the model parallelism requires frequent communication among computing nodes during training, and data parallelism is independent of each other. Therefore, in most cases, the communication overhead and synchronization overhead brought by model parallelism exceed data parallelism, and the acceleration effect is not as good as data parallelism. Data parallelization is superior to model parallelization in implementation difficulty fault tolerance and cluster utilization. However, model parallelism is a good choice for large models that cannot fit into the memory of a single compute node.

Hadoop is an open-source project of distributed computing framework supported by well-known foundations and is also the open-source implementation of three cloud computing papers by Google. Its emergence makes it possible for people to understand and use computing platforms [20]. Hadoop is developed in Java language. Hadoop distributed file system, MapReduce, and HBase are the implementations of Google file system, MapReduce, and Bigtable, respectively. The Hadoop distributed file system is the primary storage used in Hadoop applications. MadReduce is a programming technique that enables massive scalability of hundreds and thousands of servers in the Hadoop cluster. BigTable is a well-managed wide columned and key valued noSQL database service that helps in managing large analytical and operational workloads being part of the google portfolio. Hadoop not only has the advantage of being open source and free but also has many advantages: (1) It has strong scalability. The scalability of storage and computing is the core of Hadoop design. (2) It is economical and practical, and Hadoop's running platform has low hardware requirements. Ordinary desktop computers can meet general task computing requirements. (3) It is reliable, and the fault tolerance and backup mechanism of Hadoop distributed file system and the task monitoring mechanism in MapReduce fully guarantee the reliability of distributed computing [21–24].

3. Algorithm Design

The distributed commodity classification deep learning method proposed in this paper is improved based on SimCLR model. In the recent years, various self-supervised learning methods have been predominantly used for analyzing and learning image representations. But most of their performances lag in comparison to their supervised equivalents. SimCLR is a "simple framework for contrastive learning f visual representations." This learning method is justified to be more superior to the traditional and state of the art self-supervised learning techniques. It is also found to be superior to the supervised learning techniques on ImageNet classification when the architecture is scaled up [25, 26]. The improved SimCLR model is used to perform pretraining on unlabeled commodity images, and the pretraining weights are transferred to a small number of labeled samples for fine-tuning, so as to realize the recognition of commodity categories in a small number of samples. In this paper, based on Hadoop distributed computing framework, the parallel computation of model pretraining and fine-tuning stage is implemented to reduce model training time. The proposed distributed architecture for computer vision-oriented deep learning is shown in Figure 1.

3.1. Improved SimCLR. In order to reduce the reliance on labeled samples for commodity image feature extraction and classification, an improved SimCLR contrastive learning method is designed in this paper. Its network structure is shown in Figure 2. First, the input image is preprocessed using a combination of three data enhancement methods: horizontal flip, color dithering, and grayscale, to obtain two related views. Next, the features of the input views are extracted using a convolutional neural network. Then, the model is trained by transforming the features using an asymmetric prediction operator and letting one branch of the network fit the other branch. Finally, image features are extracted using the trained convolutional neural network, and a linear classifier is trained using labeled samples to complete feature classification .

The residual network (ResNet) can reduce the gradient dissipation when the signal is propagated in the deep network. It has many advantages such as strong generalization ability and easy expansion. In this paper, we use the 18-layer residual network as the feature extraction network for the commodity classification model. For the original image x, apply data enhancement to obtain two related views $T_i(x)$ and $T_j(x)$.

The feature representation $f(\cdot)$ is obtained by encoding it using a feature extraction network $y_i = f(x) = \mathrm{ResNet}(T_i(x))$ with shared weights, where y_i is the output of the last layer of ResNet. After $f(\cdot)$, y_i is transformed by adding a fully connected layer with layers 4 and 2 to the two branches of the network, respectively. The Projector layer of the fully connected layer is a multilayer perceptron, and its output is as follows.

$$z_i = g(y_i) = \sigma^{(2)}\left(W^{(2)}\sigma^{(1)}\left(W^{(1)}y_i\right)\right). \quad (1)$$

The Predictor layer uses a 2-layer multilayer perceptron with an output dimension of 1024. The output is as follows.

$$p_i = h(z_i) = W^{(4)}\sigma^{(3)}\left(W^{(3)}z_i\right), \quad (2)$$

where σ represents the ReLU activation function and W represents the parameter of fully connected layer. Finally, negative cosine similarity is used to calculate the distance between p_i and $\varkappa_j$.

$$D(p_i, z_j) = -\frac{p_i}{\|p_i\|_2} \cdot \frac{z_j}{\|z_j\|_2}, \quad (3)$$

where $\|\cdot\|_2$ denotes the L2 regularization. In order to optimize both branches of the network simultaneously, equation (3) is transformed into a symmetric loss function as follows.

$$\mathrm{Loss} = \frac{1}{2}D(p_i, z_j) + \frac{1}{2}D\left(p_j, z_i\right). \quad (4)$$

At this point, if the loss function of equation (4) is used directly, the neural network will quickly output a degenerate solution and the loss converges to a minimum value of -1. To avoid degeneracy from occurring, the stop-gradient operator is added to equation (4) and the loss function is modified as follows.

$$\mathrm{Loss} = \frac{1}{2}D(p_i, \mathrm{stopgrad}(z_j)) + \frac{1}{2}D\left(p_j, \mathrm{stopgrad}(z_i)\right). \quad (5)$$

The network is trained by minimizing the loss value. Finally, the trained $f(\cdot)$ is removed to extract the image features and train a linear classifier, which will complete the final feature classification.

The improved SimCLR can be viewed as an algorithm based on the expectation maximization principle, which contains two sets of variables. From the perspective of alternating learning, the optimization process of the method can be transformed into a process of alternating solutions of two subproblems. The main role of projector is to reduce the loss of the output features of backbone in the calculation of the contrast loss. To simplify the derivation, the effect of g_φ on the gradient propagation in both branch networks can be ignored simultaneously. In this case, equation (3) is equivalent to the equation (6).

$$L\left(\theta^t, \theta^{t-1}\right) = E_{x,T}\left[\left\|f_{\theta^t}(T_i(x)) - f_{\theta^{\mathrm{t}-1}}\left(T_j(x)\right)\right\|_2^2\right], \quad (6)$$

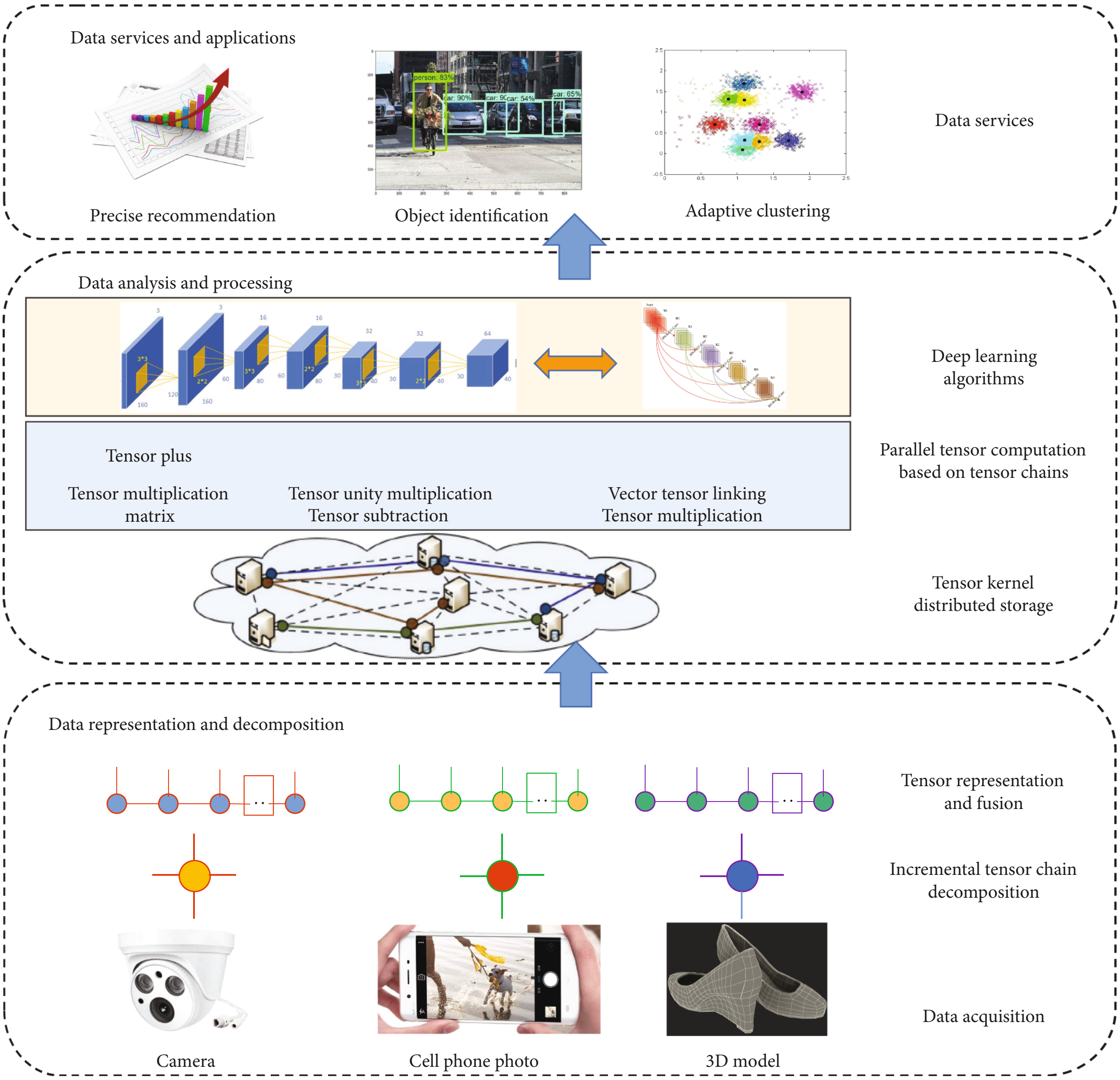

Data services and applications
Precise recommendation
Object identification
Adaptive clustering
Data services
Data analysis and processing
Deep learning algorithms
Tensor plus
Tensor multiplication matrix
Tensor unity multiplication
Tensor subtraction
Vector tensor linking
Tensor multiplication
Parallel tensor computation based on tensor chains
Tensor kernel distributed storage
Data representation and decomposition
Tensor representation and fusion
Incremental tensor chain decomposition
Data acquisition
Camera
Cell phone photo
3D model

Figure 1: Overall architecture of the proposed model.

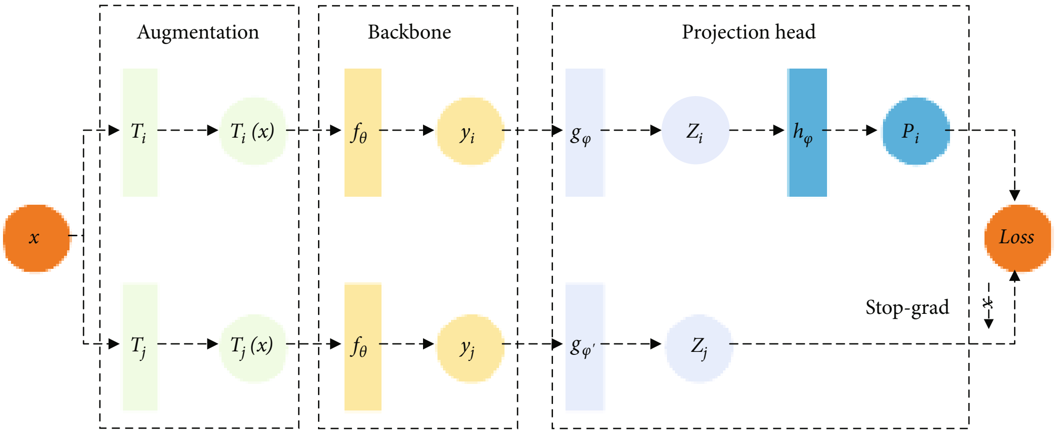


Figure 2: Structure of the improved SimCLR model.

where $f_{\theta^t}(T_i(x))$ and $f_{\theta^{t-1}}(T_j(x))$ represent the output of the two branch networks when g_φ is not added, respectively. t represents the index number of the training step. θ^t and θ^{t-1} represent the parameters of $f_\theta(\cdot)$ in the two branch networks at step t and step $t-1$. T represents the data enhancement method. Since the degree of data enhancement is random, different data enhancement results of the input image X can be obtained by applying T_i and T_j. $\|\cdot\|_2^2$ denotes the mean square error, which is equivalent to the cosine distance in the case of using L2 regularization. The expectation $E_{x,T}[\cdot]$ reflects the overall distribution of the image x at . The solution to equation (4) can be viewed as the optimization of $\theta^t$and θ^{t-1} . The process can be expressed in the following form.

$$\min L\left(\theta^t, \theta^{t-1}\right). \tag{7}$$

Equation (7) is similar to an online clustering problem [17], using an alternating solution approach that divides its optimization process into two stages, first optimizing θ^{t-1}, fixing θ^{t-1}, and then optimizing θ^t.

Since SimCLR calculates the contrastive loss directly for the feature output from the projection head, no nonlinear activation function is used in the last layer of the projection head. Prediction head is a multilayer perceptron with bottleneck structure, which reduces the feature dimensionality to 1/4 of the input layer in the hidden layer and restores the dimensionality to the input dimension without using the BN and ReLU activation functions in the output layer. In this paper, the original prediction head is changed to the fully connected layer structure shown in Figure 3. In order to make the gradient propagation process more stable and reduce the gradient dissipation, the MLP layer with the same structure is used in the modified prediction head layer, and the final ReLU activation function is retained .

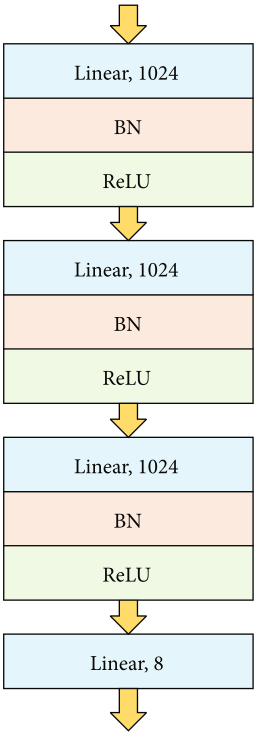


Figure 3: Improving the prediction head in SimCLR.

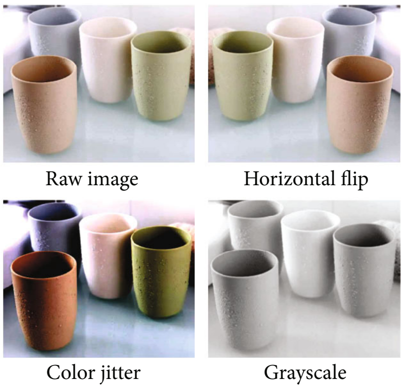
3.2. Data Augmentation. From the above analysis, it can be seen that although the improved SimCLR does not use negative samples, it can also model feature invariance and complete model training by relying on the special solution design of the twin network. In supervised learning, data augmentation can expand training samples, make the model learn more invariable features, and improve generalization ability. Compared with supervised learning, contrast learning requires stronger data augmentation strategies to improve the quality of feature extraction. The current research on data augmentation in self-supervised learning is mainly conducted in natural images. Due to the difference in imaging methods, the features of remote sensing images contain more variability than ordinary natural images. There are many differences in size, location, and shape of the same class of scenes, making it difficult for a single data enhancement method to allow the model to learn feature representations with sufficient robustness. Color features reflect the response to light, and compared with other features, color has a certain stability and will not change significantly with changes in imaging angle and spatial scale. Contour features reflect the basic shape and can provide an important basis for the judgment of the target type. In complex scenes, contour features can be used as a powerful supplement to color features. The spatial structure is a structural description of the image, and modeling the spatial structure is beneficial to improve the model's ability to perceive complex images. In contrast learning, comparing feature representations of the same image under different colors enables the model to learn the color features in the image; inputting grayscale maps helps the model discover the contour information in the image; inputting the flipped image into the network enables the model to learn the spatial invariance in the image. In this paper, a data enhancement method using a combination of horizontal flip, color Jitter, and grayscale is designed to allow the model to model the color features, contour features, and spatial structure of the image simultaneously. The augmentation effect is shown in Figure 4.

3.3. Hadoop-Based Distributed Implementation

3.3.1. Overall Architecture. The Hadoop-based commodity classification system is also a distributed image classification system. According to the characteristics of Hadoop distributed framework and the basic requirements of image classification, the overall architecture of the system can refer to the current popular Model View Controller (MVC) architecture, which is divided into the presentation layer, business logic layer controller, and data processing layer. The overall diagram is shown in Figure 5.

The role of each layer in the overall distributed image classification system is as follows:

Presentation layer: the user submits the local image to be classified or the path of the file where the image is located on


Raw image

Horizontal flip

Color jitter

Grayscale

Figure 4: Augmentation effect of commodity images.

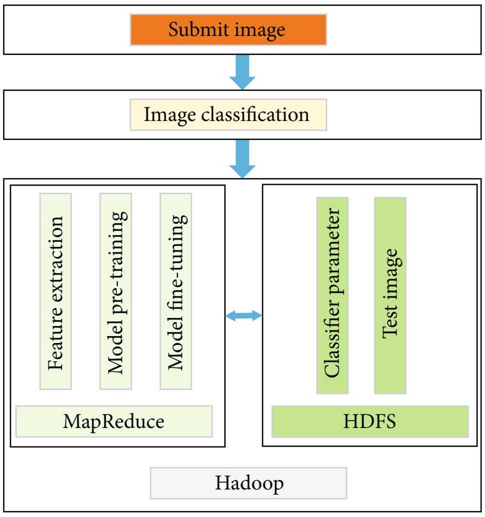


Figure 5: Overall diagram of a Hadoop-based distributed system.

the client side. The images to be classified are uploaded to HDSF via name-node.

Business logic layer: the image classification tasks submitted by users will be processed in this layer, and the images to be classified will be processed in the corresponding business logic.

Data processing layer: this layer is the core part of the distributed image classification system, which stores the images to be classified in HDFS and then extracts the features of the input images and saves the feature vectors in the form of files in HDFS for subsequent classification tasks. These data include the feature vector file and the category of the image obtained through the MapReduce process. After obtaining the relevant data, we use the sample image data to write a MapReduce program to train one or more classifiers, and the results of the specific parameters of the classifiers will be saved in the form of files in HDFS. Since the image features to be classified already exist in HDFS, we can use the trained classifier to predict the image classification by MapReduce program. After the prediction is done, the classification results are also saved as files in HDFS for users to view.

Table 1: Distribution of commodity categories in the dataset.

Category	Training set	Test set
Menswear	1087	436
Womenswear	989	317
Food	1023	391
Cosmetics	1264	476
Jewelry	862	280
Cell phone	975	217
Computer	1174	349
Drink	1346	328
Total	8720	2794

3.3.2. Design of MapReduce Module. The MapReduce module is used in this image classification system to parallelize the feature extraction and the feature distance between images. According to the main functions of the MapReduce module, the Map function can be used to perform feature extraction on images. The process is divided into several MapReduce jobs based on specific algorithms.

The whole workflow of map task can be summarized into 5 phases, which are read phase, map phase, collect phase, spill phase, and combine phase. Read phase: before map task enters map phase, map task parses out one key/value pair (key/value) from the input InputSplit through the RecordReader we wrote, which should be the image file path and the image file name as the value of the key/value pair. Map phase: the main task in the map phase is to hand over the key-value pairs parsed from the read phase to our own map() function for processing. Collect phase: when the data has been processed in the map phase using the map() function we wrote, we can call it directly. Spill phase: when the amount of data in the memory of a physical node reaches a certain upper limit, MapReduce writes the spilled data to the local disk based on relevant rules. Combine phase: after all data has been processed in the map phase and spill phase, the map task performs a merge of all temporary files.

Based on the above system design, the execution steps of image classification task in Hadoop distributed computing framework can be divided into five steps. When a user submits an image classification request, the job client requests a new image classification job ID from the job tracker of Hadoop. Initialization and task assignment of image classification jobs: job tracker initializes a job submitted by job client and places the job in an internal task queue. Map phase of the image classification task: job tracker after assigning tasks to task tracker, task tracker automatically obtains the JAR files and required data from HDFS to the local file disk. Reduce stage of image classification task: after obtaining intermediate temporary key-value pairs of image feature vectors calculated by map task, MapReduce framework will classify these feature vectors according to their corresponding key values. Completion of image classification task: when the reduce stage is completed, the job tracker will consider the job completed and mark the task as successful and display various parameters of the job operation to the user.

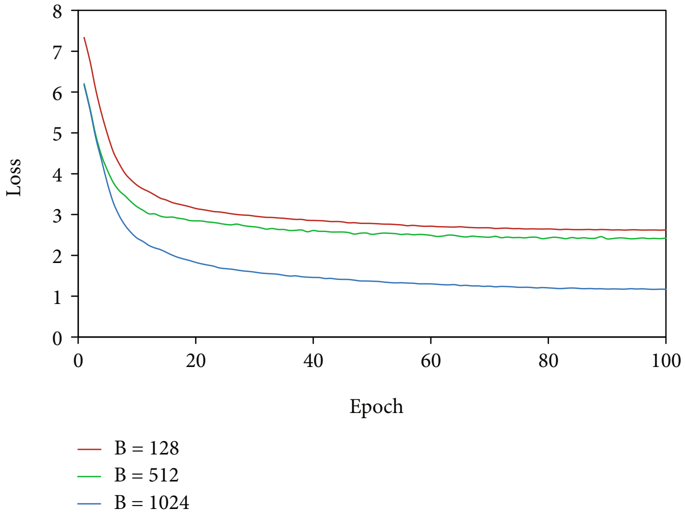


Figure 6: Loss curves with different *Batchsizes*.

4. Experiments

4.1. Experiment Preparation. The commodity image dataset used in this paper is a sample image collected on the web, containing a labeled training set and a test set. It contains 8720 training samples in its training folder and 2794 test samples in its test folder. There are 10 categories in the training and test sets. Table 1 shows the number of samples in each category, and it can be observed that the sample distribution is very unbalanced. In addition, there is an unlabeled dataset with a total of 68504 samples. The development tools used for the implementation of the algorithms in this paper are Microsoft Visual Studio 2012, the computer vision library OpenCV 3 for image processing, the database SQL Server 2008 R2 for data storage, and the deep learning framework Tensorflow 1.2.

4.2. Self-Supervised Contrast Training. During model training, the optimizer uses stochastic gradient descent method with learning rate set to lr × Batchsize/256. The base learning rate lr is set to 0.01 and uses cosine decay strategy with momentum value of 0.9 and improved projection head dimension set to 1024. The *Batchsize* is set to 1024, and epoch is set to 100. Figure 6 shows the loss curves of pretraining on unlabeled data using the improved SimCLR method at different *Batchsize* (B). It can be observed that the larger the *Batchsize* setting for self-supervised contrast learning, the better the model training effect. This is because a large *Batchsize* can contain more negative samples.

4.3. Supervised Fine-Tuning of Few-Shot Samples. To demonstrate the effectiveness of the proposed method, different proportions of samples from the training dataset are selected as the new training set, and the test set is kept unchanged to verify the performance of the small-sample supervised fine-tuning classification. The improved method is compared with three other existing methods: (1) ResNet-18 from scratch, (2) ImageNet pretrained ResNet-18 model, and (3) SimCLR before improvement.

The results of the classification experiments in Figure 7 show that the improved method performs better on the commodity dataset compared with the original contrast learning model SimCLR. It indicates that the structure proposed in this paper can reduce information loss and enhance the ability of feature extraction of the model, which achieves the expected effect of the design. The improved method significantly outperforms the ImageNet pretrained model and the ResNet18 model trained from scratch on the commodity dataset when the sample size is small. The improved method achieves an accuracy of 75.2% using only 5% of the labeled data, which is 10.9% higher than that of the ImageNet pretrained model. The reason for this is that there are large interdomain differences between commodity and natural images, and the feature distribution of the target dataset cannot be adequately fitted either by scratch training or by using the ResNet model pretrained on ImageNet when the sample size is small. The training process of the ImageNet pretraining model relies on the signals provided by the data labels and thus features unrelated to the label signals are lost during the training. The improved method does not rely on the label signal, which provides richer and more abstract supervised information compared to supervised learning, maximizes the retention of image features, and allows the linear classifier to easily classify the features extracted by the improved method.

4.4. Distributed Computing Experiments. We first performed an image classification time test. The configuration in this test is consistent with the image feature extraction test, and the image classification time test mainly compares the time of the standalone classification test with the time of the Hadoop cluster classification test. We save the features from the image library as files in the local file system and extract features from a total of 1000 images using the feature extraction module in the modified SimCLR model. We write a program for image classification and matching on a standalone machine using the relevant interface of the OpenCV image processing library and use the generated feature files

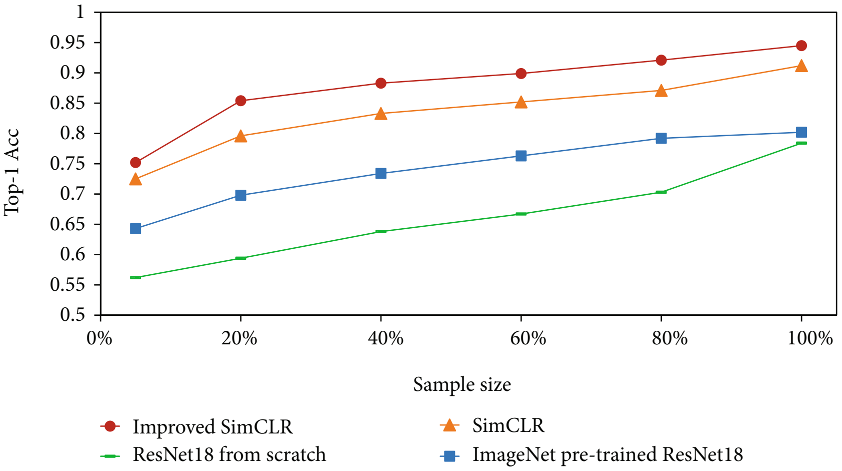


Figure 7: Few-shot classification experiments.

Table 2: Classification time of commodity images with different map and reduce quantities (s).

Number of map tasks	Number of reduce tasks	Map task time	Reduce task time	Total time
8	4	110.5	320.5	460
16	4	33.5	213.9	288
32	4	15.6	198.3	246
64	4	9.6	247.6	288
8	8	111.4	165.7	432
16	8	32.4	85.6	192
32	8	16.5	92.6	213
64	8	8.7	106.9	238

for image classification in an average time of 328 s and 0.328 s per image. We test image classification on a Hadoop cluster in an average time of 447 s, reaching an average time of 0.447 s per image.

In the Hadoop performance test, the system is mainly tested in terms of map and reduce quantity, and for the map and reduce quantity performance test, 1000 image samples of size 224 × 224 in HDFS are selected and feature extraction is performed. Table 2 shows the image feature extraction time consumption of the system with different map and number and other parameters.

The time of reduce also decreases accordingly, but it can be seen that the decrease in time is significantly smaller than the decrease in map tasks. When the number of map tasks reaches 64, the time of the reduce task does not drop as much as it did originally. The reason for this is that the increase in the number of map tasks causes the intermediate data to be generated too quickly, while the cluster is running with only 4 or 8 reduce tasks, which makes the data between different nodes not read directly into memory but is temporarily stored in a waiting queue. When the number of reduces is increased to 8, all the reduce tasks take significantly less time than when there are 4. From the table, we can see that the total time for image classification is minimal when the number of reduce and map is 8 and 16, respectively.

5. Conclusions

Commodity recognition, as a technology of great commercial value, has also entered the research horizon of research institutions and various corporate companies. To address the problem of many unlabeled samples and scarce labeled samples in the process of automated commodity classification, this paper proposes a commodity image classification method based on self-supervised contrast learning to achieve efficient learning of image features by improving the projection head layer in the SimCLR model. In addition, to reduce the time for model pretraining and supervised fine-tuning and classification, this paper provides a distributed implementation of the above algorithm based on Hadoop. The experimental results on commodity image datasets show that the method in this paper can learn feature representations from a small amount of unlabeled data and obtain better classification results. The distributed architecture effectively reduces the model training and classification time. The improved SimCLR model yields promising results but the computational cost of the same in case of large-scale implementation is yet to be explored in the study. This would act as a guidance for future research work.

Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declares that he has no conflict of interest.

References

[1] H. Sun, X. Zhang, X. Han, X. Jin, and Z. Zhao, “Commodity image classification based on improved bag-of-visual-words model,” *Complexity*, vol. 2021, 10 pages, 2021.

[2] J. Du, C. Jiang, Z. Han, H. Zhang, S. Mumtaz, and Y. Ren, “Contract mechanism and performance analysis for data transaction in mobile social networks,” *IEEE Transactions on*

Network Science and Engineering, vol. 6, no. 2, pp. 103–115, 2019.

[3] Y. Li, Y. Wang, Z. Miao, J. Wang, and R. Zhang, "Contrastive self-supervised hashing with dual pseudo agreement," *IEEE Access*, vol. 8, pp. 165034–165043, 2020.

[4] L. Zhao, W. Luo, Q. Liao, S. Chen, and J. Wu, "Hyperspectral image classification with contrastive self-supervised learning under limited labeled samples," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[5] E. A. Sim, S. Lee, J. Oh, and J. Lee, "GANs and DCGANs for generation of topology optimization validation curve through clustering analysis," *Advances in Engineering Software*, vol. 152, p. 102957, 2021.

[6] X. Li, D. Zhang, M. Ye, X. Li, Q. Dou, and Q. Lv, "Bidirectional generative transductive zero-shot learning," *Neural Computing and Applications*, vol. 33, no. 10, pp. 5313–5326, 2021.

[7] H. Hou, J. Huo, J. Wu, Y. K. Lai, and Y. Gao, "MW-GAN: multi-warping GAN for caricature generation with multi-style geometric exaggeration," *IEEE Transactions on Image Processing*, vol. 30, pp. 8644–8657, 2021.

[8] N. Barzilay, T. B. Shalev, and R. Giryes, "MISS GAN: A multi-IlluStrator style generative adversarial network for image to illustration translation," *Pattern Recognition Letters*, vol. 151, pp. 140–147, 2021.

[9] T. Guo, C. Xu, B. Shi, C. Xu, and D. Tao, "Optimizing latent distributions for non-adversarial generative networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 1–2672, 2020.

[10] T. Hoang, T. T. Do, T. V. Nguyen, and N. M. Cheung, "Multi-modal mutual information maximization: a novel approach for unsupervised deep cross-modal hashing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.

[11] C. Tang, X. Yang, J. Lv, and Z. He, "Zero-shot learning by mutual information estimation and maximization," *Knowledge-Based Systems*, vol. 194, p. 105490, 2020.

[12] J. Kang, R. Fernandez-Beltran, P. Duan, S. Liu, and A. J. Plaza, "Deep unsupervised embedding for remotely sensed images based on spatially augmented momentum contrast," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 2598–2610, 2021.

[13] P. Kumar and S. Chauhan, "*Study on temperature $$(\tau)$$(τ) variation for SimCLR-based activity recognition*," *Signal, Image and Video Processing*, pp. 1–6, 2022.

[14] Z. Wang, Z. Li, J. Wang, and D. Li, "Network intrusion detection model based on improved BYOL self-supervised learning," *Security and Communication Networks*, vol. 2021, 23 pages, 2021.

[15] J. K. Park, H. H. Park, and J. Park, "Distributed eigenfaces for massive face image data," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 25983–26000, 2017.

[16] D. Jiang, F. Wang, Z. Lv et al., "QoE-aware efficient content distribution scheme for satellite-terrestrial networks," *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[17] X. Liang and K. Wang, "Ocean big data service technology in a distributed network environment," *Journal of Coastal Research*, vol. 106, no. sp1, pp. 576–579, 2020.

[18] R. Shevchenko and A. Doroshenko, "A time cost model for distributed objects parallel computation," *Future Generation Computer Systems*, vol. 18, no. 6, pp. 807–812, 2002.

[19] A. Dubey, M. Zubair, and C. E. Grosch, "A general purpose subroutine for fast Fourier transform on a distributed memory parallel machine," *Parallel Computing*, vol. 20, no. 12, pp. 1697–1710, 1994.

[20] A. Rasooli and D. G. Down, "Guidelines for selecting hadoop schedulers based on system heterogeneity," *Journal of grid computing*, vol. 12, no. 3, pp. 499–519, 2014.

[21] L. Alarabi, M. F. Mokbel, and M. Musleh, "St-hadoop: a mapreduce framework for spatio-temporal data," *GeoInformatica*, vol. 22, no. 4, pp. 785–813, 2018.

[22] K. Lakshmanna, N. Subramani, Y. Alotaibi, S. Alghamdi, O. I. Khalafand, and A. K. Nanda, "Improved metaheuristic-driven energy-aware cluster-based routing scheme for IoT-assisted wireless sensor networks," *Sustainability*, vol. 14, no. 13, pp. 7712–7719, 2022.

[23] M. Elhoseiny, S. Huang, and A. Elgammal, "Weather classification with deep convolutional neural networks," in *2015 IEEE international conference on image processing (ICIP)*, pp. 3349–3353, Quebec City, QC, Canada, 2015.

[24] C. Zhang, X. Wang, F. Li, Q. He, and M. Huang, "Deep learning–based network application classification for SDN," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 5, article e3302, 2018.

[25] B. Li and Y. He, "An improved ResNet based on the adjustable shortcut connections," *IEEE Access*, vol. 6, pp. 18967–18974, 2018.

[26] G. T. Reddy, A. Srivatsava, K. Lakshmanna, R. Kaluri, S. Karnam, and G. Nagaraja, "Risk prediction to examine health status with real and synthetic datasets," *Biomedical and Pharmacology Journal*, vol. 10, no. 4, pp. 1897–1903, 2017.