Journal of Artificial Intelligence and Technology, 2021, 1, 110-120
https://doi.org/10.37965/jait.2020.0058

ISTP
RESEARCH ARTICLE

Video-based Person Re-identification Based on Distributed Cloud Computing

Chengyan Zhong,[1] Xiaoyu Jiang,[2] and Guanqiu Qi[3]

[1]College of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[2]College of Intelligence Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[3]Computer Information Systems Department, State University of New York at Buffalo State, Buffalo, NY 14222, USA

(Received 28 July 2020; Revised 17 January 2021; Accepted 09 March 2021; Published online 11 March 2021)

Abstract: Person re-identification has been a hot research issues in the field of computer vision. In recent years, with the maturity of the theory, a large number of excellent methods have been proposed. However, large-scale data sets and huge networks make training a time-consuming process. At the same time, the parameters and their values generated during the training process also take up a lot of computer resources. Therefore, we apply distributed cloud computing method to perform person re-identification task. Using distributed data storage method, pedestrian data sets and parameters are stored in cloud nodes. To speed up operational efficiency and increase fault tolerance, we add data redundancy mechanism to copy and store data blocks to different nodes, and we propose a hash loop optimization algorithm to optimize the data distribution process. Moreover, we assign different layers of the re-identification network to different nodes to complete the training in the way of model parallelism. By comparing and analyzing the accuracy and operation speed of the distributed model on the video-based dataset MARS, the results show that our distributed model has a faster training speed.

Key words: person re-identification; distributed cloud computing; data redundancy mechanism

I. INTRODUCTION

Cloud computing is a product of the development of Internet information technology and information service requirements. It is a scalable distributed parallel computing framework and a platform for storing large-scale data sets. Because of its powerful computing power, cloud computing has important applications of the Internet of Things, big data, and artificial intelligence [1]. The technology is mainly reflected on the following characteristics: First, its resource library is open and transparent. Second, there is no restriction, and it can serve every industry, and each industry can choose its own calculation model according to its actual situation. Third, it is easy to obtain data resources, which can save users a lot of time. Fourth, the service method is more flexible and can meet the actual needs of different customers to the greatest extent. The two core functions of cloud computing are distributed storage and distributed computing, which can conveniently and quickly store and process massive amounts of data [2]–[4].

Therefore, in the past research, it is popular to apply cloud computing to image-based tasks, which need to process massive images. Wang *et al*. [5] developed a cloud-based image analysis toolbox that can provide easy access to the development tools of the past decade for a wide user base. The project focuses on the integration of various software components, such as workflow management framework, various image analysis components, and interactive image visualization components. The author utilizes various frameworks of data-intensive computing and seamlessly integrates them into a cloud-based service platform to deploy various applications.

Remote sensing processing usually involves a large amount of data. It requires a distributed computing infrastructure to manage big data sets because the amount of data is not suitable for centralized storage computers [6]. Kang *et al*. [7] developed mobile applications in the field of remote sensing, using the advantages of cloud computing for satellite image processing. The result showed that using cloud computing devices in mobile devices can share geographic data among various devices, regardless of time or location. Lage-Freitas *et al.* [8] introduced a cloud computing method that automatically deploys worker in the cloud. Cloud computing functions include the deployment of workers/master distributed programming models through the use of virtualization technologies (such as virtual machines and containers). So that remote sensing data processing is achieved by assigning tasks to workers who store data. Zou [9] proposed a high-throughput cloud computing interface and integrated disaster rapid cloud platform design. The platform provides a solution for the access and integration of distributed remote sensing data, automatic and fast remote sensing data processing through effective massive data management, and distributed parallel computing, thus enabling dynamic disaster detection within a region or country.

With the explosive growth of medical multimedia data onto hospital information systems, an efficient access is needed to support large-scale medical multimedia data access. The application of cloud computing to medical image retrieval is a popular trend in recent years. Zhuang *et al*. [10] proposed an effective and powerful content-based large-scale medical image retrieval method in mobile cloud computing environment. The entire query process

Corresponding author: Guanqiu Qi (e-mail: qig@buffalostate.edu).

includes, when clinical users submit query images, parallel image set reduction processing is performed at the master node, and then the candidate images are transferred to the slave nodes for refinement processing to obtain an answer set, and the answer set is finally transferred to the query node. The proposed method including the robust image blocks transmission scheme based on priority is specifically designed to solve the instability and heterogeneity in the mobile cloud environment. Compared with medical images detection tasks, medical image registration tasks are more complicated. Combining the traditional multiobjective optimization algorithm with big data analysis, a multiobjective artificial bee colony multiobjective optimization algorithm based on clustering calculation is proposed, which can accelerate the speed of solving complex problems on the Spark platform [11].

Due to the diversity of food and the influence of color, light, and perspective on food images. Food image recognition is inherently challenging, which is a computationally intensive task. Wang *et al*. [12] proposed a MapReduce programming model for food feature matching algorithm. Applying cloud computing to food image recognition tasks, this paper first extracts scale-invariant feature transform (SIFT) and Gabor descriptors of food images from food training images, and then uses K-means training to obtain the set of cluster centers as word packets. To realize recognition of a smartphone, the system is deployed using the Hadoop architecture. The Hadoop clusters consists of three computers, one of which is the master computer, and the other two are slave computers. When the amount of data reaches a certain level, Hadoop data processing efficiency can bring huge benefits. This method solves the bottleneck of processing a large number of concurrent images on mobile devices.

The above work is to use cloud-based technology to solve the problems of huge data volume and high real-time requirements in image-based recognition tasks in various fields. Inspired by above work, this paper considers the characteristics of cloud computing technology, such as strong processing power, rapid transmission, and large storage space [13] [14]. Using cloud computing technology can establish a good platform for pedestrian re-identification (Re-ID) tasks, effectively make up for the shortcomings of the management and processing of a large number of pedestrian image data in the real environment.

In this paper, different from the previous person Re-ID algorithm, which only focuses on recognition accuracy, we pay more attention to data storage and processing of large amounts of data from multiple cameras in actual situations. Based on the research of cloud computing distributed storage and distributed processing, combining with the actual application background in person Re-ID, we propose a new distributed cloud computing video person Re-ID algorithm. It is used to solve the problem of pedestrian multicamera big data collection, storage, and scalability in real environment. By analyzing the demand and actual situation of the real video surveillance system, the video surveillance system under the entire cloud architecture adopts a distributed structure to support the application of multilevel subcontrol points, and the network adopts a modular design. First, through the unified video surveillance system platform, the video surveillance signals in the actual environment are connected to the same network, and the video data are transmitted to the data nodes in the cloud database in the form of data blocks. Subsequently, the Re-ID network is hierarchically divided in a model-parallel manner, and then deployed to different cloud nodes. During Re-ID model training, a node deployed with a network model initiates a request to a data node, and then updates the generated parameters to the data node. Pedestrian data blocks and parameters are transmitted and interacted between nodes in the cloud. The distributed cloud computing method significantly improves the training speed of the network, allowing rapid training of huge models and data sets. This paper contains the following three contributions:

(1) To the best of our knowledge, in person Re-ID tasks, little attention is paid to the distributed computing of the network. The distributed cloud computing method that we proposed significantly improves the training speed and increases the error tolerance rate of the data.

(2) Considering the large amount of pedestrian data and many parameters, a distributed storage method in the cloud is proposed. We introduce a data redundancy mechanism to copy and store data, and adopt a hash cycle to optimize the storage path to achieve rapid data transmission.

(3) Re-ID networks mostly show the characteristics of huge models and deep network layers, which makes it impossible for a single computer to calculate or training takes a long time. We use the model parallel method to divide the network. Through parallel calculation, the training speed of the network is greatly improved.

II. RELATED WORK

As one of the most challenging problems in the field of surveillance video analysis, the research on person Re-ID has achieved advanced results and excellent recognition accuracy. The core issue of person Re-ID is to find the occurrence of a query person (probe) from a set of candidate persons (gallery), where the probe and the gallery are captured from different nonoverlapping camera views. However, due to huge camera network and pedestrian image data set in the real open environment, the deployment of the camera in the real environment is very cumbersome, and the cameras are not connected together, and multiple subcontrol rooms are required to store massive data. Therefore, it is a challenging task to directly use the network model of pedestrian Re-ID for pedestrian Re-ID tasks with huge data volume.

In this section, we will briefly outline the methods of person Re-ID and introduce the related work of using spatial-temporal information for Re-ID.

A. SUPERVISED RE-ID

Supervised person Re-ID refers to the use of labeled pedestrian data sets for training and testing [15]. Existing supervised methods have achieved high accuracy in Re-ID task. Representation learning and metric learning are the most commonly used methods in supervised Re-ID. Representation learning mainly includes the appearance features of pedestrians and the features of latent semantic components (head, front, and back). For the learning of appearance features, for example, Sun *et al*. [16] proposed a local-based convolutional baseline network, which divides the input pedestrian image into six parts through a partition strategy, and then obtains six local features through deep convolution vector. Finally, the six feature vectors are connected with a fully connected layer to predict pedestrian ID. At the same time, the author considers that when the image is partitioned, different semantic information may appear because of the misalignment. Therefore, a local refinement pool is proposed, and a penalty mechanism is used to correct the deviation from the correct partition. For the potential semantic components, Su *et al*. [17] proposed the idea of using the key points of pedestrian

poses to segment the image and weight the different body blocks to enhance the detailed features. This method first uses the pose estimation algorithm to obtain the position of the joint points of the human body, then uses the joint points to locate the images of different human body parts, and then embeds the body part image input Feature Embedding SubNet to obtain the normalized body part features. Then, the complete pedestrian image and the normalized body parts image are sent to the convolutional neural network (CNN) network together to obtain global features and local features. Finally, the body features are weighted by Weighting SubNets, and then fed into Softmax loss jointly with the global features. Inspired by the posture guidance mechanism, Zhu *et al*. [18] introduced a human posture migration algorithm. Given a condition image, the author used a pose estimation algorithm to extract the existing pose and target pose in the picture. Then, the proposed progressive gesture attention transfer model is used to generate pedestrian images including target pose, so as to achieve the purpose of enlarging the sample. One of the above two methods focuses on learning the correspondence between the color and texture distribution of different person images but ignores the correspondence between semantic components. The other emphasizes the semantic component learning and ignores the corresponding color texture. Therefore, Mao *et al*. [19] proposed a network called Multi-Channel deep convolutional Pyramid Person Matching Network (MC-PPMN). It learns the corresponding representation from the semantic components and color texture distribution. The proposed framework uses a pyramid matching module of hollow convolution to solve the alignment problem. To establish the correspondence between color texture distributions, deep color texture distribution representation learning based on convolutional neural networks is introduced. Two pyramid matching modules are used to learn the relationship between the color texture distribution and semantic components of the picture, and output the corresponding representation. Finally, the two fully connected layers are used to fuse the corresponding representation, and softmax is used to predict the probability that the image pair is the same person. Different from representation learning, metric learning aims to learn the similarity of two pictures through the network. On the issue of Re-ID, it is specifically explained that the similarity of different pictures of the same pedestrian is greater than that of different pictures of different pedestrians. Finally, the loss function of the network makes the distance of the same pedestrian pictures as small as possible, and the distance of different pedestrian pictures as large as possible. Commonly used metric learning loss methods include contrastive loss [20], triplet loss [21]–[23], quadruplet loss [24], triplet hard loss with batch hard mining (TriHard loss), and margin sample mining loss.

In general, supervised person Re-ID is mainly divided into representation learning and metric learning. The effect of supervised methods can be improved by adding attention mechanism or using GAN method to expand the data set. However, when supervised model is applied to other data sets, the performance tends to be greatly reduced, and it is unrealistic to label the data in large data sets in the real environment.

B. UNSUPERVISED RE-ID

To improve the effectiveness of the Re-ID algorithm on large-scale unlabeled data sets, some unsupervised Re-ID methods were proposed [25]–[29] to learn cross-view identity specific information from unlabeled data sets. However, due to the lack of information about identity labels, the performance of these unsupervised methods is usually much weaker compared with supervised methods. Yu *et al.* [29] to solve the problem of lack of pairwise label guidance in unsupervised Re-ID, compare unlabeled pedestrians with reference pedestrians in the auxiliary domain to learn a soft multilabel. The idea is to compare each unlabeled pedestrian image with a labeled reference image to obtain a soft multilabel for the unlabeled image. In addition, the author proposed soft-multilabel-guided hard negative mining that learns discriminative feature embedding for the unlabel target domain pairs through the consistency of the visual features of the unlabel target domain pairs and the soft multilabel. That is, using soft multilabel to distinguish visually similar but different unlabeled persons. Lin *et al.* [30] proposed a bottom-up clustering framework that maximizes the diversity of different pedestrians while maintaining the similarity between pedestrians of the same identity. The author used repelled loss to directly optimize the cosine distance between samples or clusters to optimize models without labels can maximize the diversity between different classes and maximize the similarity of each cluster or sample. So that diversity normalization can balance the number of clusters in each cluster, making the clustering result closer to the true distribution. UMDL [31] uses the dictionary-learning mechanism to transfer the invariant representation of the human appearance from the source-labeled data set to the unlabeled target data sets and obtain better performance. An image style migration algorithm was proposed [32], which transfers the source domain image to the target domain style in an unsupervised manner, uses SPGAN to improve performance, and keeps the ID information unchanged during the migration process. The algorithm is mainly divided into two steps. In the first step, the generator function G(*) learned by Cycle GAN is used to transfer the source domain image S to the target domain T style, and the G(S) training set is obtained. In the second step, the obtained G(S) is an image with the target domain imaging style, and the ID is consistent with the source domain, so the converted image can be used together for supervised Re-ID feature learning.

The above unsupervised Re-ID algorithm can be summarized into two ideas. One is to convert the image in the source domain to an image in the target domain style, and to transfer the image style of the pedestrian, so as to improve the recognition accuracy. The other is to use the idea of unsupervised clustering, using pseudo-tags to maximize the similarity between pedestrians of the same identity, and maximize the difference between different classes. However, there is still a large gap in accuracy between the above unsupervised algorithm and the supervised algorithm.

C. VIDEO-BASED PERSON RE-ID

In addition to image-based Re-ID methods, video-based Re-ID methods have also received more and more attention. For the first time, deep learning is used to solve the video-based Re-ID task [33]. The author proposed a new recurrent neural network architecture that uses the Siamese Network and combines a recursive and appearance data time pool to learn the feature representation of each pedestrian in the video sequence. Because the available video sequences are noisy, that is, with arbitrary sequence duration and start/end frames, each image sequence has unknown camera viewpoint changes and may have incomplete frames due to occlusion. Wang *et al*. [34] proposed a model that can automatically select more discriminative video clips. The model is formulated using a multiinstance ranking strategy for extracting more reliable spatiotemporal features from nonoverlapping cameras and learning cross-view matching through ranking. At the same time, the author

introduced a new image sequence-based human Re-ID data set, called iLIDS-VID, which is extracted from the i-LIDS multicamera tracking scene. To extract more spatio-temporal information to solve occlusion and complex background interference, You *et al*. [35] used the HOG3D descriptor to extract spatio-temporal information in the video. The HOG3D features include spatial gradient information and temporal dynamic information. For the extraction of appearance features, color histograms and local binary pattern (LBP) features are used. To obtain stable feature information, the author performed average pooling on the color histograms and LBP features of individual pedestrians in the video. And proposed top-push distance learning distance metric learning method, like many methods in recent years, is also based on Mahalanobis distance learning. This metric learning can increase the difference between interclasses and reduce the difference between intraclasses to improve matching accuracy. In recent years, some other methods [36]–[39] have been proposed to predict the quality score of local video frames. Liu *et al*. [36] proposed a simple network, that is, it adding a branch of quality score to the usual recognition network, and aggregating the multiframe information in the video into an optimal feature, which simplifies the complexity of the video recognition process. Due to the low resolution of most sequence images, the key points of human body are usually not accurate enough. By calculating the distribution of key points of the entire data set, a fixed three local regions are obtained, and then the three regions are scored, and finally the features and scores are fused by the set aggregation unit to obtain the final features [37]. To effectively extract useful information in all frames. The network introduced in [38] learns multiple spatial partial attention models and uses a diversity regularization term to ensure that multiple partial attention models focus on different parts of the body. In practical applications, each pedestrian is a sequence under the camera, and the video-based pedestrian re-recognition method can extract more abundant spatial and temporal information.

Different from the above methods, this paper aims to improve the representation of video sequences by mining spatio-temporal information in low- and high-level features. Therefore, we introduce the nonlocal attention mechanism into the CNN network to obtain a long-term representation. The nonlocal attention mechanism incorporates global sequence information into local features to enrich local features. By introducing the nonlocal attention mechanism into different feature levels to explore the sequence space and time diversity, and then change its feature representation.

III. METHOD

As shown in Fig. 1, it is a distributed cloud computing framework, which is a three-tier network architecture.

At the top is a remote cloud composed of cloud service providers. The cloud is used to store the pedestrian data captured by the camera, and the parameters generated during the training process. The pedestrian data increase linearly with the accumulation of time and the number of cameras, so the amount of these data will be so large that it is only suitable for storage in the cloud. At the same time, the network is hierarchically divided in a model-parallel manner and deployed to different training units to increase the training speed of the model.

The second layer is the wide area network (WAN) layer, which uses the Transmission Control Protocol (TCP) and the Internet Protocol (IP) protocol to upload the camera data stream directly to the cloud server and store it through the network interface. In addition, a computing layer can be added to improve the performance [39]. Through this layer, clients can access cloud data.

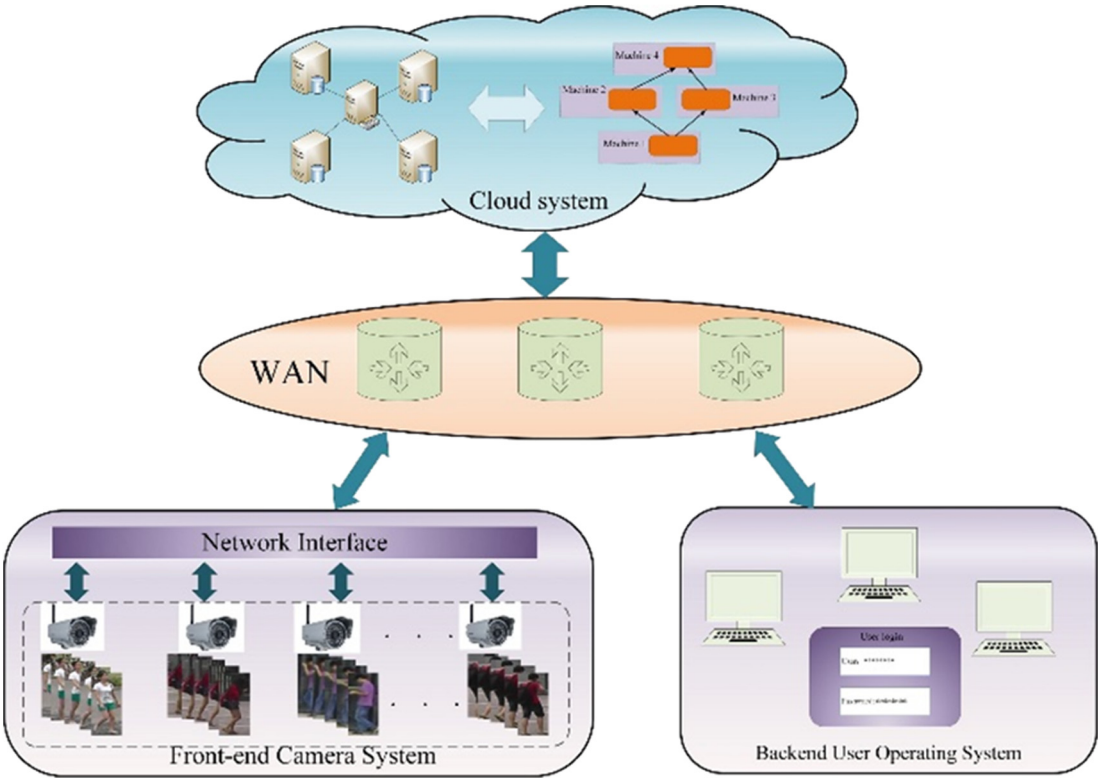

Fig. 1. Distributed cloud computing Re-ID system.

The third layer is the terminal device layer, which is divided into a front-end camera system and a back-end user operating system. The front-end camera system compresses and decodes the captured pedestrian video and transmits it to the wide area network through the network interface. The back-end user operating system operates on the data stored in the cloud.

A. DISTRIBUTED DATA STORAGE

The video-based Re-ID network has excellent performance in Re-ID accuracy because it can fully excavate pedestrian's spatio-temporal information. However, in a real environment, the high amount of data and calculations it brings will cause a long training time and consume a lot of computer resources. The two cores of cloud computing are distributed storage and distributed computing. Faced with the need for batch processing of massive pedestrian video data, we use distributed storage to store pedestrian data and client access information in nodes in the cloud.

We cut the input pedestrian data set into different small blocks and store them on different machine nodes, so that we can break through the upper limit of the storage capacity of a single machine. Cloud nodes are divided into master nodes and slave nodes. The master node is also called the name node, which is equivalent to the data directory and is responsible for the metadata storage of the entire storage system. The slave node is the data node, which is responsible for storing data. When a new data node is added to the cloud cluster, the data node will directly report to the name node, which data blocks are saved in its own node. As the name node of the housekeeper node, it will automatically construct a list to record the distribution of data blocks.

1) DATA REDUNDANCY PRESERVATION MECHANISM. In the cloud server, each data block will be saved redundantly. We set the redundancy factor to 3, and setting it too large will cause excessive server overhead. The use of data redundancy storage can speed up the data transmission speed because the use of data redundancy mechanism can allow the same data block to be called at the same time, thereby avoiding the congestion during the data block access process. In addition, the data redundancy mechanism allows easy checking of data errors because if they are backed up to each other, they can be cross-referenced. When an error occurs in one copy, you can refer to it by checking the other copy. Finally, the data redundancy mechanism can ensure the reliability of the data. When a copy has an error, there are other copies that can be used. When the cloud detects an error in a copy, it will automatically copy the copy, so that the number of copies in the cloud will be restored to the set value. In other words, once the redundant copy is lower than the user set value, once detected, it will automatically copy to generate a new copy until it reaches the set number.

2) DATA DISTRIBUTED STORAGE AND READ–WRITE STRATEGY. The data storage is shown in Fig. 2(a).

When a block enters the cloud, the first copy of the block is placed on the data node where the user uploaded the file. If a node outside the cloud cluster initiates a write data request, the first copy will randomly select a node with a disk that is not full, and the CPU is not too busy to place it. The second copy is placed on a node in a different rack than the first copy. The third copy is placed on another node in the same rack as the first copy. If there are more copies, a random algorithm is used.

For data reading, a basic principle is to read nearby because the network overhead of reading nearby is the smallest. An application programming interface (API) is provided in the cloud server. This API can know the rack ID to which a data node belongs. After the rack ID is calculated, if the ID is the same, it means that the data node is in the same rack. Because the internal bandwidth of the rack is very high, the communication is fast, and the cost is small, it can be transmitted nearby. The client can also call the API to obtain the rack ID to which it belongs. When the client reads the data, it obtains a list of storage locations for different copies from the name node. The list shows the data node where the copy is located. Then, call the API to determine the rack ID to which the client and these data nodes belong. When it is found that the rack ID corresponding to a copy of a data block is the same as the rack ID corresponding to the client, the copy is first selected to read the data. If it is not found, a copy is randomly selected for reading.

3) DATA STORAGE OPTIMIZATION ALGORITHM. From the above data distributed storage and near reading strategy, we can know how to optimize the data distribution to ensure that it can achieve near reading, which has a direct and huge impact on the data call. Therefore, high-performance, low-maintenance hashing algorithms can be used to optimize data distributed storage strategies, map and store related data to the same rack, and reduce communication costs. And the data access information is summarized to facilitate subsequent query and analysis of the data access information.

The schematic diagram of the hash algorithm is shown in Fig. 2(b), and it is divided into four parts: module W, module X,

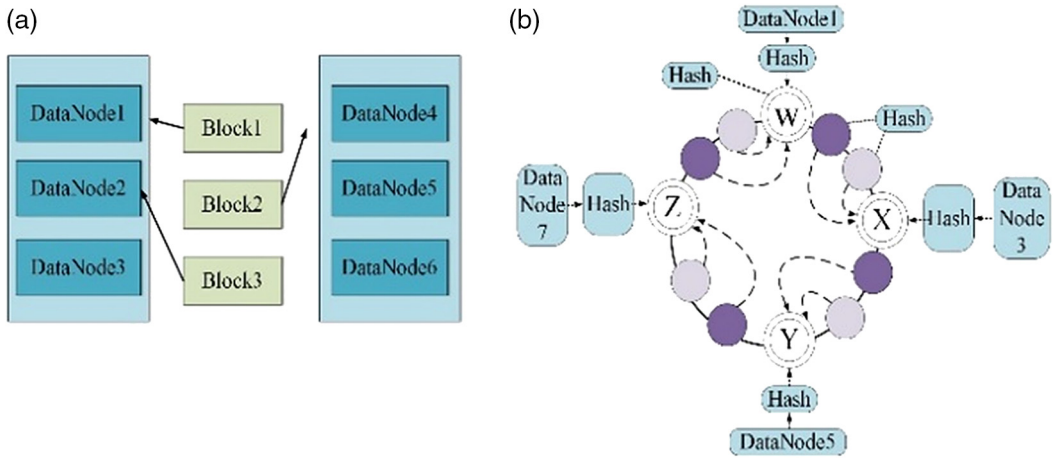

Fig. 2. Data distributed storage and optimization strategy.

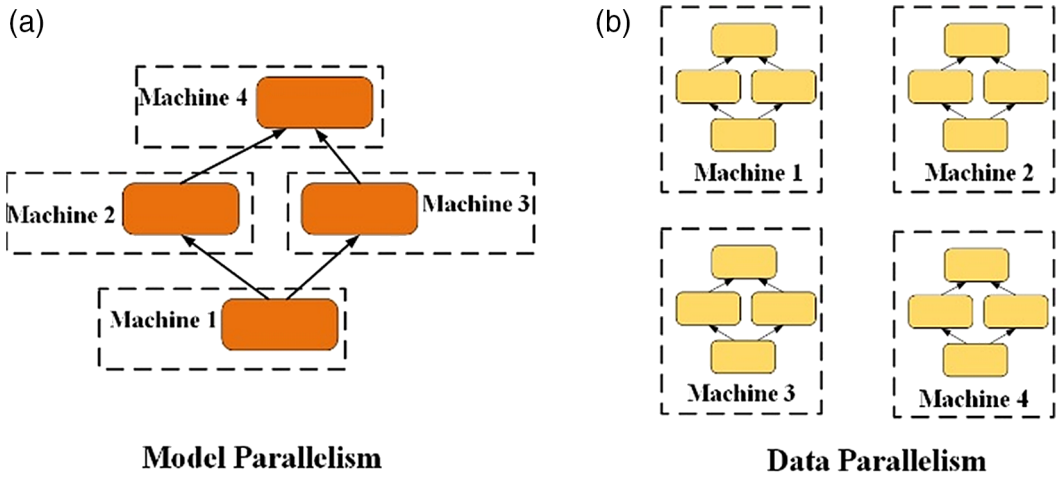

Fig. 3. Two types of distributed training modes.

module Y, and module Z, which correspond to DataNode1, DataNode3, DataNode5, and DataNode7, respectively. The loop of the hash algorithm is completed by calculating the hash value. Through the hash loop algorithm, the data distribution and storage strategy can be optimized. The specific optimization steps are as follows: First, the correlation of the data access information and the number of redundant copies need to be set. The calculation formula of the correlation of the data access information is

$$\theta = \frac{\sum_{i=1}^{n} x_i * a}{n^2 + 1}, \tag{1}$$

where θ represents the correlation between data access information and a represents the number of redundant copies set, we set $a = 3$.

Second, we calculate the hash values in the cloud cluster one by one and configure them as the interval of the hash cycle. The calculation formula of the node hash value is

$$\delta = \int_{t=1}^{m} \sqrt[2]{d} \otimes b, \tag{2}$$

where δ represents the hash value of the node, d represents the node, m is the number of all nodes, and b represents the calculation parameter of the hash value of the node.

Then, according to the relevance of the data access information, the corresponding data hash value can be calculated, the calculation formula is

$$\eta = \frac{\int_{i=1}^{n} \sqrt[2]{x} \otimes \theta}{n^2}, \tag{3}$$

where λ represents the hash value of data access.

Finally, the storage location of the data can be configured based on the hash value of the node and data access, and the configuration result can be obtained from the following:

$$g(x) = \prod_{i=1,t=1} \delta/\lambda * c, \tag{4}$$

where c is the configuration parameter.

The performance of Re-ID tasks increases linearly with the magnitude of training data. However, when training large-scale pedestrian data and large neural networks, the situation of out of memory is unavoidable. Moreover, for the large-scale datasets such as video-based pedestrian datasets, training the network can even take several days. Experimenters need to constantly try, adjust the model, and change the results. This training speed is unacceptable. Therefore, we need to use distributed training to improve the speed of training.

B. DISTRIBUTED RE-ID MODEL

The performance of Re-ID tasks increases linearly with the magnitude of training data. However, when training large-scale pedestrian data and large neural networks, the situation of out of memory is unavoidable. Moreover, for the large-scale datasets such as video-based pedestrian datasets, training the network can even take several days.

Experimenters need to constantly try, adjust the model, and change the results. This training speed is unacceptable. Therefore, we need to use distributed training to improve the speed of training. We first recognize two distributed training methods, model parallelism and data parallelism, as shown in Fig. 3.

In model parallelism, different machines in a distributed system are responsible for computing different parts of a single network. In data parallelism, different machines have complete copies of the model, each machine is just a different part of the data, and the results of each machine are combined in some way.

Due to the strong nonlinearity of neural networks, the dependence between parameters is much more serious than that of linear models. It cannot be divided simply, and it is impossible to use a technique like linear models to achieve efficient model parallelism through a global intermediate variable. But things always have two sides, and the hierarchical neural network also brings certain convenience to the model parallelism. The network level of Re-ID is

Algorithm 1. Model parallel algorithm.

Input: the entire Re-ID network, pedestrian data set.

Output: parameters after training $\{a'\}$.

1: Divide the network into four layers L according the function.
2: *for* the each layer L *do*
3: Assign it to the nodes in the computer cluster, the parameters are updated to $\{a\}$.
4: *for* the updated parameters a *do*
5: Transmit it to the next node
6: *end for*
7: *end for*

deep. A natural and easy-to-implement model parallel method is that the entire neural network is horizontally divided into K parts, and each working node undertakes one or more layers of computing tasks. The specific implementation algorithm is shown in Alg. 1. We divide the network into layers according to functions. The result of the division is shown in the figure: it is divided into a random sampling layer of the video sequence, a ResNet50 backbone network with nonlocal attention modules, and feature pooling layer. Besides, we divide the ResNet50 backbone network module with the local attention module again, considering the number of nodes of each layer, and balance the computation of each working node as much as possible.

1) VIDEO-BASED RE-ID NETWORK FRAMEWORK. Given an image sequence of any pedestrian, our goal is to use CNN to extract the features of the image, and then perform video-based Re-ID in the feature embedding space. The key to learning the representative features of a sequence is to integrate video features into the features themselves. Therefore, this paper introduces nonlocal attention layers into CNN to explore the spatio-temporal dependence of video sequences. The distributed network framework is shown in Fig. 4.

Restricted random sampling. To balance speed and accuracy, we use restrictive random sampling [39] [40] to deal with long-distance time structures. Given an input video, we divide it into T parts $\{P_t\}_{t=[1,T]}$ by equal duration. During training, we randomly sample an

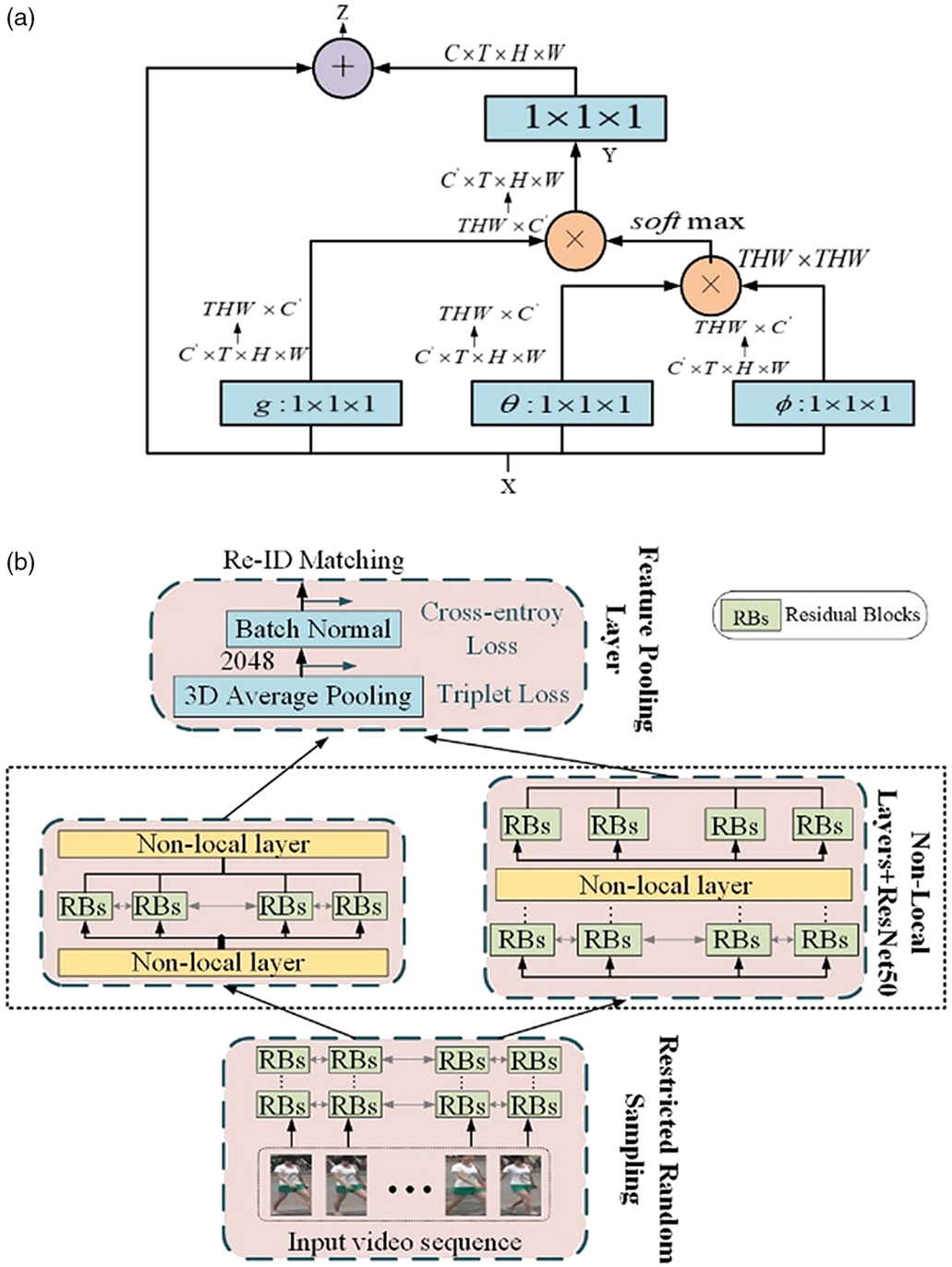

Fig. 4. Re-ID network structure.

image in each part. In the test, we used the first picture of each part. The video is then represented by an ordered set of sample frames.

Nonlocal attention network. To embed video characteristics into features, we introduced the nonlocal layer proposed by Wang *et al*. [41] into the backbone CNN. The structure of the nonlocal layer is shown in Fig. 4(a). The feature tensor $X \in \mathrm{R}^{C\times T\times H\times W}$ is obtained from a series of T feature graphs of size $C\times T\times H\times W$. We hope to exchange information between all spatial positions and the features of the frame. Let $x_i \in \mathrm{R}^C$ be sampled from X, then the corresponding output $y_i \in \mathrm{R}^C$ for nonlocal operation can be defined as follows:

$$y_i = \frac{1}{\sum_{\forall j} e^{\theta(x_i)^T \phi(x_j)}} \sum_{\forall j} e^{\theta(x_i)^T \phi(x_j)} g(x_j), \tag{5}$$

where $i, j=[1,THW]$ indexes the feature map and all frames. We first use a linear transformation function θ,ϕ,g into a lower dimensional embedding space $\mathrm{R}^{C'}$. Then, we calculate the corresponding of each x_i by weighting all x_j using embedding Gaussian instantiation. Equation (5) in the nonlocal layer is a self-attention mechanism. By mapping Y to the transformation W_Z (1×1×1 convolution) of the original feature space R^C, the entire nonlocal layer is finally defined as $Z = W_Z Y + X$, where the output of the nonlocal operation is added to the original feature tensor X. The reason for this nonlocal operation is that when extracting features at a specific location at a specific time, the network should consider the temporal and spatial dependencies within the sequence by focusing on the nonlocal environment. In our Re-ID network, we use five nonlocal layers embedded in our backbone CNN network (ResNet-50) to understand the semantic relations presented in the sequence.

Feature pooling layer. After the image sequence passes through the backbone CNN and the nonlocal attention layer, we use the feature pool layer to obtain the final features of Re-ID. We apply 3D Average Pooling (3DAP) along the space and time dimensions, aggregate the output features of each image into a representative vector, and then go through a batch normalization (BN) layer. We train the network by jointly optimizing cross-entropy loss and soft-margin batch-hard triplet loss. Optimizing the cross-entropy loss of the final feature before BN while optimizing the triplet loss of the feature will produce the best Re-ID performance.

The algorithm of the Re-ID network training process and testing process are shown below:

Algorithm 2. Model parallel algorithm.

Input: a sequence of images V
divide V in equal T chunks
2: Select training set and test set from T chunks through restricted random sampling
for epoch←1, predetermined epoch *do*
4: Feature set←five layers ResNet50 (training set)
New feature set←five layers nonlocal & ResNet50(Feature set)
6: Feature vector←flatten (feature map)
Normalized feature vector←batch normalization (feature vector)
8: Triplet loss←triplet loss (feature vector, label)
Cross entropy loss←cross entropy loss (normalized feature vector, label)
10: Total loss←triplet loss+ cross entropy loss
Backward (total loss)
Update parameters (optimizer)
end for

The above is the division of the entire network, and the entire operation process is serial. The latter nodes need to wait for the previous nodes to perform operations before they can proceed. However, the latter nodes do not prevent the previous nodes from performing the next batch of training when they are performing operations. Therefore, we can regard model parallelism as the work of a pipeline, with each node performing its own duties.

IV. EXPERIMENT

We evaluate our approach on the current largest video-based person Re-ID dataset MARS in terms of accuracy and training time. We deployed the network according to the proposed distributed algorithm and compared the training time and number of iterations with the centralized network. To prove that the distributed algorithm we proposed can set a larger batch size and consume less time during training.

A. EXPERIMENTAL SETUP

1) DATASET. With a size of 6.3 G, MARS is one of the largest video-based pedestrian data sets. The dataset has two folders called bbox train and bbox test. In the bbox-train folder, there are 625 pedestrian IDs, a total of 8298 tracks, and a total of 509,914 pictures. In the bbox-test folder, there are 636 pedestrian IDs, a total of 12,180 tracks, and a total of 681,089 pictures. MARS is a huge dataset.

2) IMPLEMENTATION DETAIL. We conducted experiments in single-GPU mode and multiGPU parallel mode, respectively. First, put the baseline and nonlocal CNN networks on a single GPU for training, respectively. When training the baseline on a single card, the batch size is set to 2, 3, 4, and 6. The batch size of the nonlocal CNN is set to 2 and 3. Because the nonlocal CNN model is too large, it can only be at most 3. In the subsequent multi-GPU parallel experiment, we took into account the reality and used three GPUs and six GPUs for training. The model is divided into three or six

TABLE 1. Accuracy of ResNet50 in three training modes.

One GPU			Three GPUs			Six GPUs		
Batch size	R1	mAP	Batch size	R1	mAP	Batch size	R1	mAP
2	0.86	0.78	6	0.86	0.77	12	0.87	0.79
3	0.86	0.81	9	0.88	0.82	18	0.88	0.83
4	0.87	0.80	12	0.87	0.80	24	0.88	0.82
6	0.88	0.80	15	0.87	0.80	30	0.89	0.82

TABLE II. Accuracy of nonlocal CNN in three training modes.

One GPU			Three GPUs			Six GPUs		
Batch size	R1	mAP	Batch size	R1	mAP	Batch size	R1	mAP
2	0.86	0.78				12	0.87	0.79
3	0.86	0.81	9	0.88	0.82	18	0.89	0.83

pieces and loaded into the GPU. Before each epoch, model parameters are copied according to the number of GPUs used. When each epoch runs, the batch size input each time is divided into 3 or 6 equally and submitted to each GPU to run. The model on each GPU will pass the learned parameters to the next node after running, until all nodes are trained, and a new model is generated, and the next epoch is ready to start. For each model on the GPU, the batch size input each time is the set batch size/3 or batch size/6.

TABLE III. Time of ResNet50 in three training modes.

One GPU		Three GPUs		Six GPUs	
Batch size	**Time**	**Batch size**	**Time**	**Batch size**	**Time**
2	19,373 (5.4 h)	6	10,237 (2.8 h)	12	7688 (2.1 h)
3	18,740 (5.2 h)	9	8750 (2.4 h)	18	6656 (1.8 h)
4	18,675 (5.18 h	12	8929 (2.48 h)	24	6250 (1.7 h)
6	18,161 (5.0 h)	15	8091 (2.2 h)	30	5813 (1.6 h)

TABLE IV. Time of nonlocal CNN in three training modes.

One GPU		Three GPUs		Six GPUs	
Batch size	**Time**	**Batch size**	**Time**	**Batch size**	**Time**
2	40,768 (11.3 h)			12	18,046 (5.0 h)
3	39,218 (10.9 h)	9	16,310 (4.5 h)	18	10,813 (3.0 h)

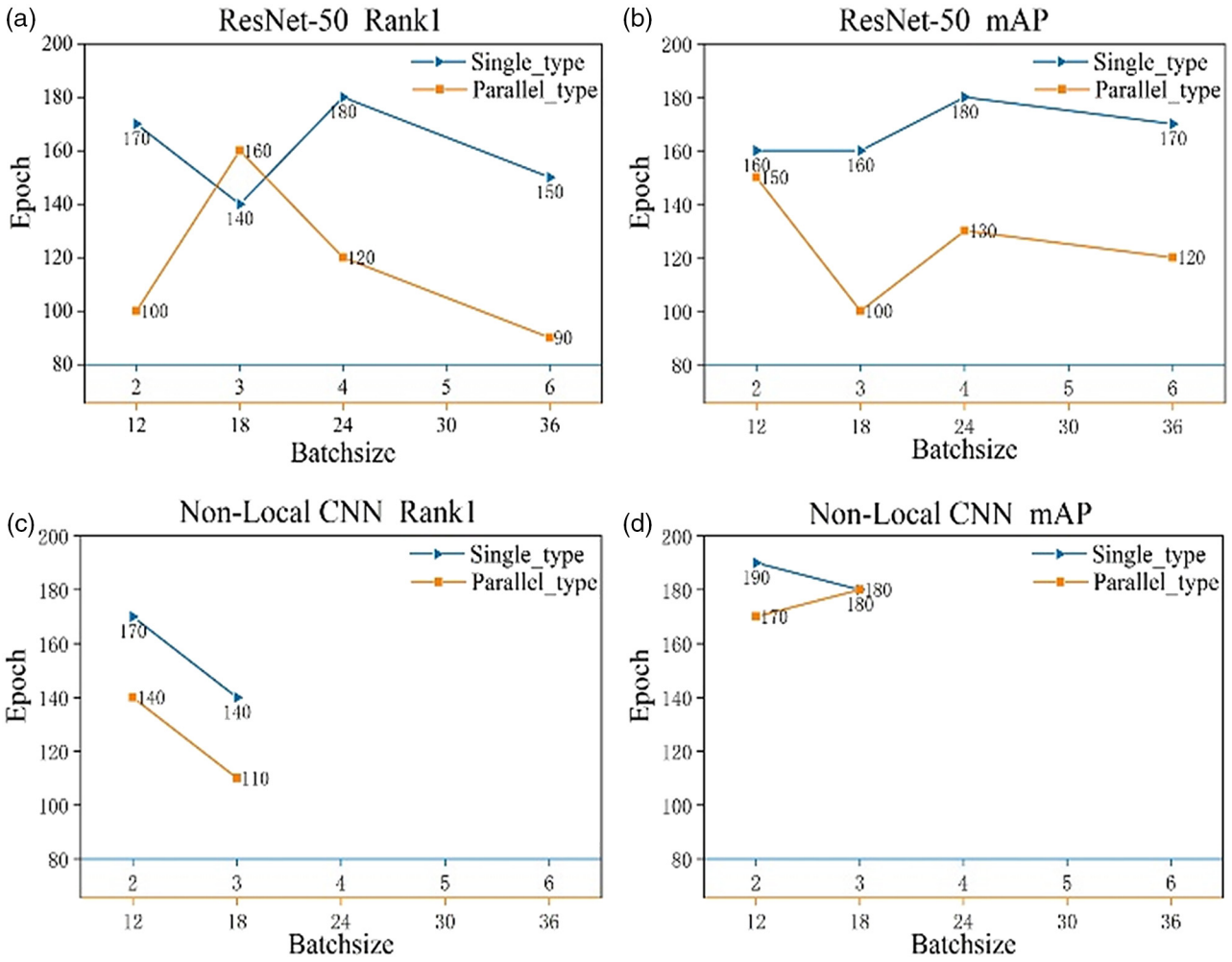

Fig. 5. Number of iterations in two training modes.

B. OVERALL PERFORMANCE AND COMPARISON RESULTS

1) TRAINING TIME ANALYSIS. As shown in Tables I–IV, we show the results of training time and accuracy of ResNet50 and nonlocal CNN on single GPU and multi-GPU. When training the baseline with a single GPU, batch size can only take values of 2, 3, 4, and 6. The batch size of nonlocal CNN can only be 2 and 3. If it is too large, it will overflow. When using parallel-type training, batch size can be larger. In terms of accuracy, the performance achieved by distributed training and centralized training is the same, and distributed training will be even higher. In terms of training time, training on a single GPU takes a lot of time. However, after the distributed deployment, the training time is reduced. It takes even 11 h to train nonlocal CNN on a single GPU, which is unacceptable. After the model is trained in parallel, it can be seen that the training time is reduced by half.

2) ITERATION ANALYSIS. As shown in Fig. 5, for baseline or nonlocal CNN iterative training, the number of iterations required for training on a single GPU is usually higher than that for distributed training. In other words, we train the network until the optimal performance is reached. With distributed training, we only need less epoch numbers.

V. CONCLUSION

Aiming at the problem that the pedestrian data set in the actual environment is huge and difficult to store, and the deep training of the network layer number consumes a lot of time. We have proposed distributed storage and distributed training methods. The data are divided into data blocks and replicated through a redundancy mechanism, and then stored on different nodes. When training the network, the parallel mode of the model is used to greatly shorten the training time while ensuring accuracy.

ACKNOWLEDGMENT

This research was founded by the Common Key Technology Innovation Special of Key Industries of Chongqing Science and Technology Commission under Grant No. cstc2017zdcy-zdyfX0067.

REFERENCES

[1] Y. Chen, *Service-oriented Computing and System Integration: Software, IoT, Big Data, and AI as Services*, 7th ed., Kendall Hunt Publishing, Dubuque, IA, US, 2020.

[2] W. Tsai, G. Qi, and Y. Chen, "A cost-effective intelligent configuration model in cloud computing," in *32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Macau, 2012, pp. 400–408.

[3] W. Tsai and G. Qi, "DICB: Dynamic intelligent customizable benign pricing strategy for cloud computing," in *IEEE Fifth Int. Conf. Cloud Comput.*, Honolulu, HI, USA, 2012, pp. 654–661.

[4] W. Tsai, G. Qi, and Y. Chen, "Choosing cost-effective configuration in cloud storage," in *11th IEEE ISADS*, Mexico City, Mexico, 2013, pp. 1–8.

[5] D. Wang *et al.*, "Cloud computing for high performance image analysis on a national infrastructure," in *13th IEEE/ACM, Cloud, and Grid Computing*, Delft, 2013, pp. 172–173.

[6] G. Qi, Z. Zhu, K. Erqinhu, Y. Chen, Y. Chai, and J. Sun, "Fault-diagnosis for reciprocating compressors using big data and machine learning," *Simul. Modell. Pract. Theory*, vol. 80, pp. 104–127, 2018.

[7] S. Kang, K. Kim, and K. Lee, "Tablet application for satellite image processing on cloud computing platform," in *IEEE IGARSS*, Melbourne, VIC, 2013, pp. 1710–1712.

[8] A. Lage-Freitas, R. P. Ribeiro, N. D. C. Oliveira, and A. C. Frery, "An automatic deployment support for processing remote sensing data in the cloud," in *IEEE IGARSS*, Valencia, 2018, pp. 2054–2057.

[9] Q. Zou, "Research on cloud computing for disaster monitoring using massive remote sensing data," in *2nd IEEE ICCCBDA*, Chengdu, 2017, pp. 29–33.

[10] Y. Zhuang *et al.*, "Efficient and robust large medical image retrieval in mobile cloud computing environment," *Inf. Sci.*, vol. 263, pp. 60–86, 2014.

[11] T. Wen *et al.*, "Multiswarm artificial bee colony algorithm based on spark cloud computing platform for medical image registration," *Comput. Methods Prog. Biomed.*, vol. 192, pp. 105432, 2020.

[12] W. Wang *et al.*, "Towards a pervasive cloud computing based food image recognition," in *IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber Phys. Soc. Comput.*, Beijing, China, 2013, pp. 2243–2244.

[13] G. Qi, W. Tsai, W. Li, Z. Zhu, and Y. Luo, "A cloud-based triage log analysis and recovery framework," *Simul. Modell. Pract. Theory*, vol. 77, pp. 292–316, 2017.

[14] Z. Zhu, G. Qi, M. Zheng, J. Sun, and Y. Chai, "Blockchain based consensus checking in decentralized cloud storage," *Simul. Modell. Pract. Theory*, vol. 102, pp. 101987, 2020.

[15] G. Qi, G. Hu, X. Wang, N. Mazur, Z. Zhu, and M. Haner, "EXAM: A framework of learning extreme and moderate embeddings for person Re-ID," *J. Imaging*, vol. 7, no. 1, p. 6, 2021

[16] Y. Sun *et al.*, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *ECCV,* Munich, Germany, 2018, pp. 480–496.

[17] C. Su *et al.*, "Pose-driven deep convolutional model for person re-identification," in *16th IEEE ICCV,* Venice, 2017, pp. 3960–3969.

[18] Z. Zhu *et al.*, "Pose-driven deep convolutional model for person re-identification," in *IEEE/CVF CVPR,* CA, 2019, pp. 2342–2351.

[19] C. Mao *et al.*, "Multi-channel pyramid person matching network for person re-identification," in *32nd AAAI,* LA, 2018, pp. 7243–7250.

[20] R. R. Varior *et al.*, "Gated siamese convolutional neural network architecture for human re-identification," in *14th ECCV,* Amsterdam, The Netherlands, 2016, pp. 791–808.

[21] F. Schroff *et al.*, "Facenet: A unified embedding for face recognition and clustering," in *IEEE CVPR,* Boston, MA, US, 2015, pp. 815–823.

[22] H. Liu *et al.*, "End-to-end comparative attention networks for person re-identification," *IEEE Trans. Image Process.,* vol. 26, pp. 3492–3506, 2017.

[23] D. Cheng *et al.*, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *IEEE CVPR,* Las Vegas, NV, US, 2016, pp. 1335–1344.

[24] W. Chen *et al.*, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *IEEE CVPR,* Honolulu, HI, US, 2017, pp. 1320–1329.

[25] R. Zhao *et al.*, "Unsupervised salience learning for person re-identification," in *26th IEEE CVPR,* Portland, OR, US, 2013, pp. 3586–3593.

[26] C. Liang *et al.*, "A unsupervised person re-identification method using model based representation and ranking," in *23rd ACM,* Brisbane, Australia, 2015, pp. 771–774.

[27] X. Ma *et al.*, "Person re-identification by unsupervised video matching," *Pattern Recognit.,* vol. 65, pp. 197–210, 2016.

[28] H. Wang *et al.*, "Towards unsupervised open-set person re-identification," in *IEEE ICIP,* Phoenix, AZ, US, 2016, pp. 769–773.

[29] H. Li, Y. Chen, D. Tao, Z. Yu, and G. Qi, "Attribute-aligned domain-invariant feature learning for unsupervised domain adaptation person re-identification," *IEEE Trans. Inf. Forensics Secur.,* vol. 16, pp. 1480–1494, 2021.

[30] Y. Lin *et al.*, "A bottom-up clustering approach to unsupervised person re-identification," in *33rd AAAI,* Honolulu, HI, US, 2019, pp. 8738–8745.

[31] P. Peng *et al.*, "Unsupervised cross-dataset transfer learning for person re-identification," in *IEEE CVPR,* Las Vegas, NV, US, 2016, pp. 1306–1315.

[32] Y. Fu *et al.*, "Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification," in *ICCV,* Seoul, Korea, 2019, pp. 6112–6121.

[33] N. Mclaughlin *et al.*, "Recurrent convolutional network for video-based person re-identification," in *IEEE CVPR,* Las Vegas, NV, US, 2016, pp. 1325–1334.

[34] T. Wang *et al.*, "Person re-identification by discriminative selection in video ranking," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 38, pp. 2501–2514, 2016.

[35] J. You *et al.*, "Top-push video-based person re-identification," in *IEEE CVPR,* Las Vegas, NV, US, 2016, pp. 1345–1353.

[36] Y. Liu *et al.*, "Quality aware network for set to set recognition," in *IEEE CVPR,* Honolulu, HI, US, 2017, pp. 4694–4703.

[37] G. Song *et al.*, "Region-based quality estimation network for large-scale person re-identification," in *32nd AAAI,* New Orleans, LA, US, 2018, pp. 7347–7354.

[38] J. Zhu, J. Hu, M. Zhang, Y. Chen, and S. Bi, "A fog computing model for implementing motion guide to visually impaired," *Simul. Modell. Pract. Theory,* vol. 101, pp. 102015, 2020.

[39] S. Li *et al.*, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *IEEE/CVF CVPR,* Salt Lake City, UT, US, 2018, pp. 369–378.

[40] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV,* Amsterdam, The Netherlands, 2016, pp. 20–36.

[41] X. Wang *et al.*, "Non-local neural networks," in *IEEE CVPR,* Salt Lake City, UT, 2018, pp. 7794–7803.