

Retraction

Retracted: Image Recognition and Simulation Based on Distributed Artificial Intelligence

Complexity

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

Copyright © 2024 Complexity. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

In addition, our investigation has also shown that one or more of the following human-subject reporting requirements has not been met in this article: ethical approval by an Institutional Review Board (IRB) committee or equivalent, patient/participant consent to participate, and/or agreement to publish patient/participant details (where relevant).

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] T. Fan, "Image Recognition and Simulation Based on Distributed Artificial Intelligence," *Complexity*, vol. 2021, Article ID 5575883, 11 pages, 2021.

Research Article

Image Recognition and Simulation Based on Distributed Artificial Intelligence

Tao Fan 

Department of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Tao Fan; 02200001@sues.edu.cn

Received 1 March 2021; Revised 1 April 2021; Accepted 6 April 2021; Published 15 April 2021

Academic Editor: Zhihan Lv

Copyright © 2021 Tao Fan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies the traditional target classification and recognition algorithm based on Histogram of Oriented Gradients (HOG) feature extraction and Support Vector Machine (SVM) classification and applies this algorithm to distributed artificial intelligence image recognition. Due to the huge number of images, the general detection speed cannot meet the requirements. We have improved the HOG feature extraction algorithm. Using principal component analysis (PCA) to perform dimensionality reduction operations on HOG features and doing distributed artificial intelligence image recognition experiments, the results show that the image detection efficiency is slightly improved, and the detection speed is also improved. This article analyzes the reason for these changes because PCA mainly uses the useful feature information in HOG features. The parallelization processing of HOG features on graphics processing unit (GPU) is studied. GPU is used for high parallel and high-density calculations, and the calculation of HOG features is very complicated. Using GPU for parallelization of HOG features can make the calculation speed of HOG features improved. We use image experiments for the parallelized HOG feature algorithm. Experimental simulations show that the speed of distributed artificial intelligence image recognition is greatly improved. By analyzing the existing digital image recognition methods, an improved BP neural network algorithm is proposed. Under the premise of ensuring accuracy, the recognition speed of digital images is accelerated, the time required for recognition is reduced, real-time performance is guaranteed, and the effectiveness of the algorithm is verified.

1. Introduction

With the rapid development of communication technology and computer technology, emerging services such as cloud computing, the Internet of Things, and social networks have promoted the growth of data types and scales in human society at an unprecedented rate [1]. With the explosive growth of data scale and the highly complex data model, the world has therefore entered the era of networked big data. Faced with the ever-increasing data scale, the problem of increasing data scale and limited computing power faced by a single computing unit is gradually exposed. In order to deal with large-scale and complex problems, distributed computing methods came into being. The core idea is to split a problem that requires high computational cost and a lot of time to solve into many simple subproblems and then assign the subproblems to multiple calculations [2]. The unit

processes and finally combines these calculation results to get the final result. As a new distributed computing technology, multiagent system has developed rapidly since its appearance. It has become a way of thinking and tools for complex system analysis and simulation and has gradually penetrated into medical services, smart cities, power systems, national defense construction, etc.

A multiagent system is usually composed of multiple single agents integrating perception, communication, computing, storage, and other capabilities [3]. Each agent detects surrounding environment information through its own sensors or receives information through a communication module, using integration and its own processing. After processing the information, the device interacts with neighbor agents to gradually achieve the goal. Therefore, multiagent systems are usually able to deal with the problems of mutual cooperation in complex environments,

especially for complex problems with spatial distribution characteristics that other methods cannot match [4]. An image is a collection of graphics and images and is an information carrier that contains rich content that humans touch. When human beings recognize an image, they generally recognize it after judging the features it possesses. For a computer, its image recognition process is similar to that of humans, and it also needs to find, process, and extract features for judgment and recognition. Image recognition technology is a research direction of artificial intelligence, and image features are the prerequisite for the use of image recognition technology [5]. After years of research, humans have made more in-depth explorations in the image recognition technology industry and have harvested a lot of results with practical significance, and the application of image recognition technology has become more and more extensive [6].

This article describes HOG features and SVM classification and describes the use of HOG features and SVM classification to locate and type recognition analysis for distributed artificial intelligence image recognition. It is proposed to use PCA to reduce the dimensionality of HOG features and then classify. Specifically, the technical contributions of this article can be summarized as follows.

First, this article proposes to use GPU to accelerate HOG in parallel, introduces the principle architecture and working mechanism of GPU, then conducts research on the parallelization algorithm of HOG feature extraction, and conducts experimental analysis on distributed artificial intelligence image recognition.

Second, we compare the pros and cons of GPU-based HOG feature extraction on the detection effects of distributed artificial intelligence image recognition and original algorithms. On the basis of obtaining the binary image, first we use the projection method to segment, extract each digital image, and then normalize the segmented digital image as the input of the image recognition neural network.

Third, this article proposes an improved BP neural network algorithm for recognizing digital images. The PCA algorithm reduces the dimension of the weight matrix between the input layer and the hidden layer and minimizes the number of neurons in the hidden layer, which is verified by experiments. It can be seen from the experimental results that this algorithm achieves the effect of improving the training speed under the premise of ensuring the accuracy of the digital image.

2. Related Work

Due to the outstanding performance of multiagent distributed optimization in machine learning, signal processing, statistical learning, wireless sensor networks, and other fields, a large number of scientific research scholars have participated in the research of distributed optimization theory and proposed a series of classic distributed optimization algorithm [7, 8]. The current distributed optimization methods mainly include primitive domain method, augmented Lagrangian method, and Newton method. As a classic primitive domain method, gradient method includes

distributed gradient descent algorithm and distributed dual average algorithm. In order to deal with various situations, scholars have improved the algorithm on the original basis [9]. The main advantage of the above gradient-based algorithm is that the algorithm is intuitive and the calculation method is simple [10]. However, the use of attenuation steps to ensure accurate convergence leads to the problem of slow convergence. In order to overcome this drawback, related scholars have accelerated the convergence speed by using a fixed step size, but the accuracy of the algorithm is low [11]. Therefore, the researchers proposed a distributed Nesterov gradient algorithm, which allows the use of a constant step size and one more consistent update at each iteration to achieve faster speed and ensure accurate convergence. Recently, a distributed optimization algorithm based on gradient tracking technology was proposed to ensure the convergence of smooth convex functions and proved the linear convergence when the objective function is strongly convex [12]. The remaining related work includes the expansion of directed networks and the expansion of random or time-varying networks.

Researchers in the field of distributed optimization of existing multiagent systems have proposed a series of distributed optimization algorithms that adapt to different complex situations for different network communication mechanisms, constraint problems, and update mechanisms to solve the global optimal solution of the target problem [13]. However, the research work on the improvement strategy of the distributed optimization algorithm of the multiagent system is in its infancy, and the relevant theories are not complete. Therefore, how to further improve the convergence speed and stability of the algorithm without reducing the performance of the original algorithm and without affecting the scope of application of the original algorithm has become a very practical research direction. In the centralized convex optimization method, a method that is widely used to improve the convergence rate of a gradient optimization algorithm is called the momentum method. Its essence is a first-order optimization method that adds momentum to the classical gradient descent method. Different from the centralized algorithm, in the distributed optimization environment, not only the optimization goal but also the consistency of the local solutions must be achieved, which poses new challenges to the improvement of the original algorithm [14]. Related scholars combined the heavy-ball momentum method in centralized optimization with the distributed gradient tracing algorithm (Am algorithm) using row random and column random double matrices and proposed the ABm algorithm, which successfully integrated the centralized first-order optimization method used in distributed algorithms [15]. Through theoretical analysis and simulation experiments, relevant scholars have proved that the algorithm after adding the momentum term can converge to the global optimal solution at a linear speed faster than the original algorithm [16]. Based on the Nesterov method, the researchers proposed two distributed optimization algorithms under a fixed undirected network [17]. The simulation results proved the proposed distributed Nesterov gradient algorithm (D-NG

algorithm) and the uniform iterative distributed Nesterov gradient algorithm (D-NC algorithm). Compared with the distributed dual average algorithm and distributed sub-gradient algorithm, it has advantages in convergence speed and can converge to the global optimal solution faster. Aiming at the optimization problem in the directed network environment, related scholars combined the Nesterov method and the gradient tracing algorithm to propose a new distributed optimization algorithm [18]. This algorithm not only improves the convergence speed of the algorithm by introducing a momentum term, but also allows each agent to independently design step length. When the objective function is strongly convex and smooth, as long as the step size is selected from a given interval, the algorithm will converge to the global optimal solution at a linear speed.

Usually the mathematical models established for highly maneuvering targets have strong nonlinear and non-Gaussian characteristics. In recent years, many scholars have devoted themselves to studying the filtering problem of this type of system and have obtained some achievements [19]. The particle filter algorithm has become a hot topic of state estimation under non-Gaussian nonlinear conditions. The particle filter algorithm is based on Monte Carlo sampling of state variables and calculates the important weights of each particle to infer the state estimation at the next moment. The algorithm is not affected by nonlinear and non-Gaussian conditions and has a better effect on target tracking with strong mobility. Particle filtering obtains a huge swarm of particles through sampling technology, and iterative recursive operations are performed on each particle in the tracking process, which leads to a large increase in the calculation of the algorithm, and some particles also have particle degradation problems in the process. At present, on the basis of the particle filter algorithm, combined with the advantages of interactive multimodels to describe the motion characteristics of maneuvering targets, the researchers have proposed a particle filter algorithm based on interactive multimodels [20]. In the framework of interactive multimodels, multiple models are used to establish a mathematical model of highly maneuvering target motion, each of which describes all possible states of target motion. On this basis, particle filtering is used to estimate the target state for each model, and finally the weighted sum of the state estimation values of all models is used as the filtering output. This technology uses particle filtering for multiple models, which doubles the complexity of the calculation. In practical applications, these data need to be stored, which increases the burden on the system and is difficult to implement in real time in the actual system [21–24].

3. Image Target Detection and Recognition Based on HOG and SVM

3.1. Target Detection Based on HOG Features and SVM Classification. Histogram of Oriented Gradients (HOG) feature is a feature descriptor used for object detection in computer vision and image processing. It composes features by calculating and counting the histogram of the gradient direction of the local area of the image. HOG is an operator

that describes the shape and edge characteristics of objects. The sudden changes in the color, gray, or texture of the image will cause the discontinuity of the local features and cause the edge characteristics to change significantly. You firstly divide the image into small connected areas, call them cell units, then collect the gradient or edge direction histogram of each pixel in the cell unit, and finally combine these histograms to form a feature descriptor.

We calculate the gradient (including size and direction) of each pixel in the image to be processed. First, we need to calculate the gradient of the abscissa and ordinate directions and then obtain the gradient direction value of each pixel. The gradient formula of pixels (x, y) in the image is

$$\begin{aligned} G_x(x, y) &= F(x, y) - F(x - 2, y), \\ G_y(x, y) &= F(x, y + 2) - F(x, y). \end{aligned} \quad (1)$$

The calculation formulas for the gradient magnitude and gradient direction at pixel (x, y) are

$$\begin{aligned} G(x, y) &= [G_x(x, y)^2 + G_y(x, y)^2]^{(1/2)}, \\ \theta(x, y) &= \arctan \left[\frac{G_y(x, y)}{G_x(x, y)} \right]. \end{aligned} \quad (2)$$

We divide all gradient directions into 9 bins; that is, there are 9-dimensional feature vectors as the horizontal axis of the histogram, and the accumulated gradient value corresponding to the angle range is the vertical axis, and then we divide the image into several cell units and set the size of each cell unit. For 10×10 pixels, 2×2 cell units are made into a block, each area includes 36-dimensional feature vectors, and an image with a size of 120×160 pixels has 48 areas, that is, 1728-dimensional feature vectors.

Support Vector Machine (SVM) is a pattern recognition method that is very suitable for solving nonlinear small sample, high-dimensional space recognition problems. It is developed on the basis of statistical learning theory. SVM is a learning method that minimizes structural risks.

SVM itself is a linear classifier; it also solves linearly separable problems at first and then expands to nonlinear separable problems. The main method to solve the problem of nonlinear inseparability is to use nonlinear mapping to map inseparable samples from low-dimensional space to linearly separable samples. Because SVM has great advantages in solving small sample, nonlinear, and high-dimensional recognition problems, and its mathematical form is simple, and there are few artificial parameters, SVM has been applied to image processing, data mining, and other aspects. The basic model of SVM is a linear classifier, and this classifier is characterized by maximizing the interval in the feature space to find the optimal separation hyperplane.

3.2. Parallel Design of HOG Feature Extraction Algorithm. The GPU is characterized by high-speed parallel computing. In parallel computing, we often hand over the denser data calculations in the program to the GPU and encapsulate the execution process into a function. This function is the kernel

function. GPU uses more two threads to run the kernel function in parallel, and after each operation is completed, the running result is copied to the system memory, and finally the CPU does the finishing work.

The existing HOG feature extraction algorithm takes more time. The main reason is that the HOG feature extraction algorithm is a computationally intensive algorithm, and it is called a lot of times in the detection system. An image with a size of 190*290 has HOG. The feature needs to be calculated about 3000 times, which takes about 286 ms, and the number of images we need to detect is very large, and the total calculation time becomes longer. In this section, the HOG feature extraction algorithm is parallelized in order to calculate all the features in the image at one time, thereby improving operating efficiency. The HOG feature extraction process is shown in Figure 1.

The process of parallelized improvement of HOG feature extraction algorithm is as follows: first we use the CPU to input the image and perform preprocessing, then send the collected data to the device, and then use the GPU for gradient calculation, construction of directional gradient histogram and histogram normalization, etc. We then return the extracted feature value to the host side and finally use the CPU to output the feature.

We transfer the image from the host memory to the GPU register to reduce the input and output communication between the GPU and the CPU. Compared with the classic HOG gradient calculation, this module has parallelized improvements to grayscale and color images. In the CUDA platform, we use one thread block to calculate the gradient value of a pixel block, and each thread block has 512 threads, and each thread is used to complete the gradient calculation of a pixel. Therefore, after executing a thread block, the gradient calculation of a maximum of 512 pixels can be completed. In addition, in order to obtain a faster memory access speed, we store the calculation process of the pixel block in the corresponding thread block shared memory.

Using GPU-based principle operations and CUDA kernel functions, the statistics of the gradient histogram is divided into two stages: in the first stage, we calculate the gradient histogram of the data corresponding to each parallel thread block, because the execution of each thread block is mutually exclusive, so we can calculate the gradient histogram in shared memory. The second stage is to add each element on the temporary gradient histogram of the thread block to the final gradient histogram. The specific method we implemented to allocate parallel resources is as follows: a thread block in CUDA corresponds to a cell in the HOG feature extraction algorithm. We can execute gradient histograms of several pixels in parallel in CUDA.

According to a thread corresponding to a normalized pixel block, the contrast is normalized for each block. If each block contains m histograms, and each histogram contains n bins, then $m*n$ threads are required for normalization. We use the parallel accumulation algorithm to calculate the sum of squares of all bin values in the block. We can perform reduction calculations on all threads in parallel, repeat

execution to merge two values in one thread into one value, reduce the values in all threads to one value, and write it into a block. The schematic diagram of the interactive mode of distributed artificial intelligence image recognition is shown in Figure 2.

4. Distributed Image Intelligent Recognition Algorithm Based on Neural Network

4.1. Research on Image Segmentation Methods. In this paper, preprocessing operations such as fuzzy image removal, image denoising, and image enhancement are performed on the acquired original image to achieve the effect of removing noise, enhancing the region of interest, and improving image quality. For the digital display images acquired by the robot at night, the preprocessing operation achieves the effect of clear and easy-to-read images.

The next step is to segment the image and extract every digital image in the image. Firstly, the image is binarized, and the connected domain labeling is performed on the binarized image, and the interference part caused by factors such as reflection in the image is removed to obtain a binary image. The image result is shown in Figure 3.

After the preprocessing is over, the digital image can be extracted. Horizontal and vertical projection methods are commonly used methods for segmenting digital images. The basic idea of the algorithm is to first perform inverse color processing on the binary image, make the background white and the black digital part as the target area, then project the inverted image in the horizontal and vertical directions, and finally analyze the two directions. The change of the projection value determines the specific position of the number. We traverse the binary image and perform the reverse color operation on each pixel to get the preprocessed image I before segmenting the image. We set the number of rows of the image as H and the number of columns as W . In the image I , the pixel value of the i -th row and j -th column is represented as $s(i, j)$. According to the definition, the horizontal projection $f(i)$ and the vertical projection $g(j)$ are, respectively,

$$\begin{aligned} f(i) &= \prod_{j=1}^W s(i, j), \\ g(j) &= \prod_{i=1}^H s(i, j). \end{aligned} \quad (3)$$

According to the calculated vertical projection, from left to right, the projection value of each coordinate in the image is detected. When there is a sudden change from 0 to the first nonzero, it is the left boundary of the first digital image. We continue to detect to the right, and the first sudden change with a value of 0 appears, which is the right boundary of the first digital image. By analogy, the coordinates and width of each image can be detected. Similarly, the length of the image area can be detected according to the obtained horizontal projection. We obtain the horizontal and vertical coordinates and width of each digital image and then cut out the complete digital image according to the result.

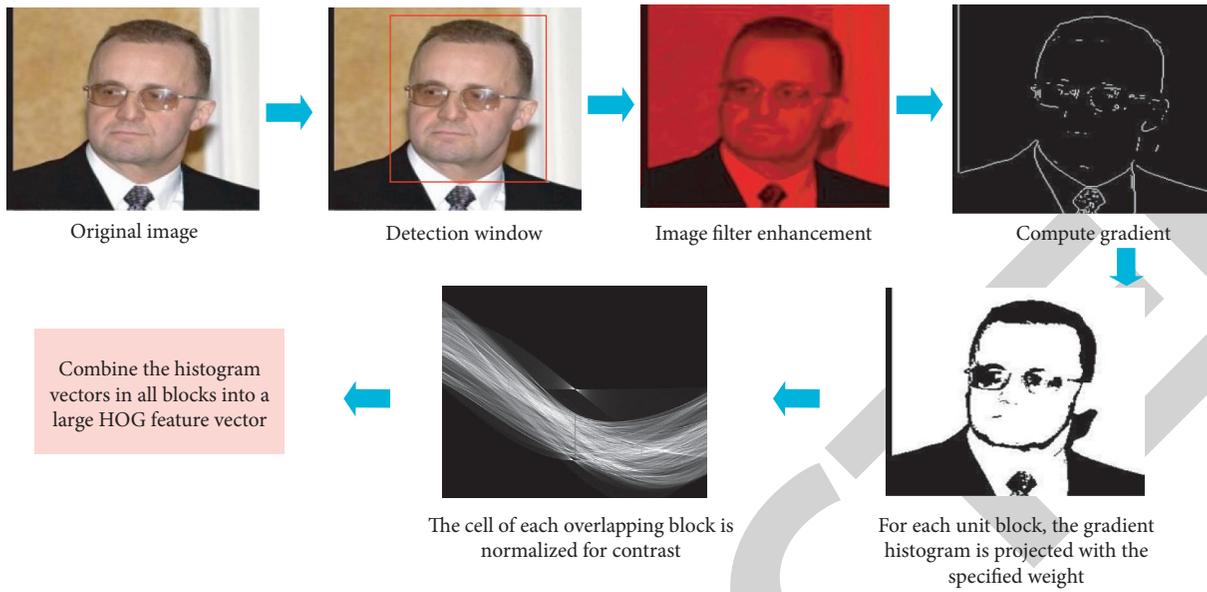


FIGURE 1: HOG feature extraction process.

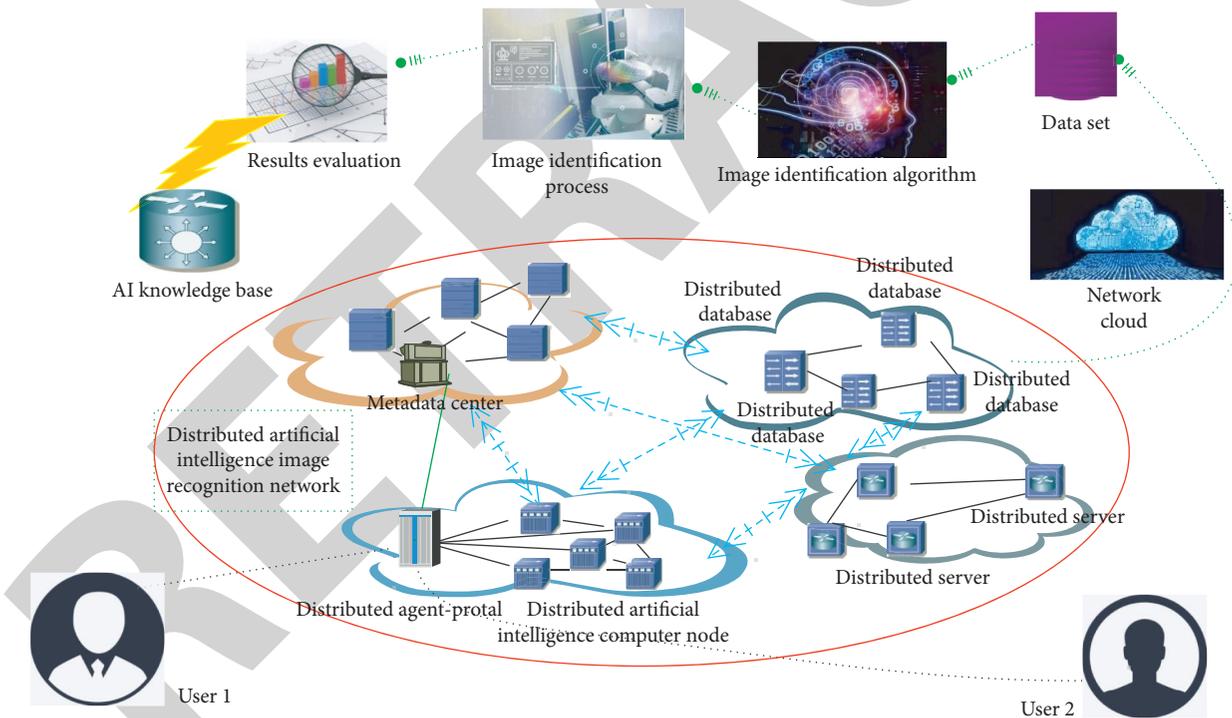


FIGURE 2: Schematic diagram of the interactive mode of distributed artificial intelligence image recognition.



FIGURE 3: Binary image after filtering connected components.

4.2. Scaling Normalization. Before recognizing a digital image, it is necessary to normalize a single image after segmentation. If this operation is not performed, the input signals of all samples may be positive when training the neural network, resulting in the simultaneous change of the weight matrix connecting the input layer and the hidden layer, and the training speed will be very slow. In order to avoid this situation, the input image is scaled and normalized, so that the mean value of the input signal of all samples is close to 0, and the mean square error is small, thus speeding up the training network.

Based on empirical data, scaling a single image after segmentation to a size of 30*10 is beneficial to image processing and recognition. According to this principle, the scale to be scaled is

$$s = \min\left(\frac{30}{H}, \frac{10}{W}\right). \quad (4)$$

H is the number of rows of the image and W is the number of columns of the image. According to the above formula, the size of the zoomed image is

$$\begin{cases} H = H \cdot s, \\ W = W \cdot s. \end{cases} \quad (5)$$

In the process of applying BP neural network for recognition, inputting the normalized digital image can accelerate the convergence speed when training the network.

4.3. Derivation of Improved BP Algorithm Based on PCA. For image recognition, the BP neural network is used for recognition accuracy. The more the number of hidden layer neurons, the more accurate the training result, but the longer the time consumed; on the contrary, the smaller the number of neurons, the shorter the training time; therefore, it is necessary to reduce the number of hidden layer neurons as much as possible on the premise of ensuring the training accuracy and reduce the training time. Trial and error method to determine the number of hidden layers is currently a common method. This method is highly accurate, but the workload is too large and multiple trials are time-consuming and labor-intensive. Aiming at the problem of image recognition, this paper proposes an improved BP neural network algorithm to process images; that is, principal component analysis (PCA) is used to optimize the number of neurons in the middle layer. The derivation process of the improved BP neural network algorithm based on PCA is shown below.

We define the network input layer node as x_i and the hidden layer node as y_j . The hidden layer node output is

$$y_j = f\left(\sum_i x_i \cdot w_{ji}\right) - \theta_j. \quad (6)$$

In the formula, w_{ji} is the network weight between the input layer node and the hidden layer node, θ_j represents the activation function, and the sigmoid function is generally used. All w_{ji} constitute the weight matrix W . The matrix W is

expanded into a vector W through the determined complete orthogonal normalized vector system u_j :

$$W = \sum_{j=1}^{+\infty} u_j \times (m_j + 1), \quad (7)$$

where

$$m_j = u_j \times u_j^T \times m_j = W \times u_j^T. \quad (8)$$

After orthogonal vector basis decomposition, d finite terms are used to estimate the vector and Z is used to represent the estimator of W . The calculation formula is as follows:

$$Z = \prod_{j=1}^d u_j \times m_j. \quad (9)$$

The mean square error of the calculation formula is as follows:

$$\Delta E = \sum_d^{\infty} (u_j \times u_j^T \times W \times W^T). \quad (10)$$

Using Lagrangian multiplier method can minimize the mean square error; the solution formula is as follows:

$$g(u_j) = \sum_d^{\infty} (u_j \times u_j^T \times R) - \sum_d^{\infty} u_j \times \lambda_j \times u_j^T. \quad (11)$$

After obtaining the derivative of u_j ,

$$u_j \times R = \lambda_j \times u_j. \quad (12)$$

When the vector estimation formula satisfies the above formula, the mean square error obtained is the smallest, namely,

$$\Delta E = \sum_d^{\infty} u_j \times u_j^T \times R = \sum_d^{\infty} \lambda_j. \quad (13)$$

According to the above derivation, W with the smallest mean square error is

$$W = \prod_{j=0}^{d-1} u_j \times m_{j-1}. \quad (14)$$

4.4. Design of Image Recognition Algorithm. We use the improved BP neural network algorithm derived in the previous section to train and recognize the segmented digital image: first we determine the number of input and output data. The data information of the image normalized to 32*14 is used as input, and the number of units in the input layer is 448. The output layer is 0-9, that is, 10 numbers, so the output layer is 10 units. The node value of the hidden layer is first initialized by the empirical formula, and the final value is determined by the PCA method so that the BP neural network achieves the characteristics of both accuracy and speed during training. The algorithm flowchart is shown in Figure 4.

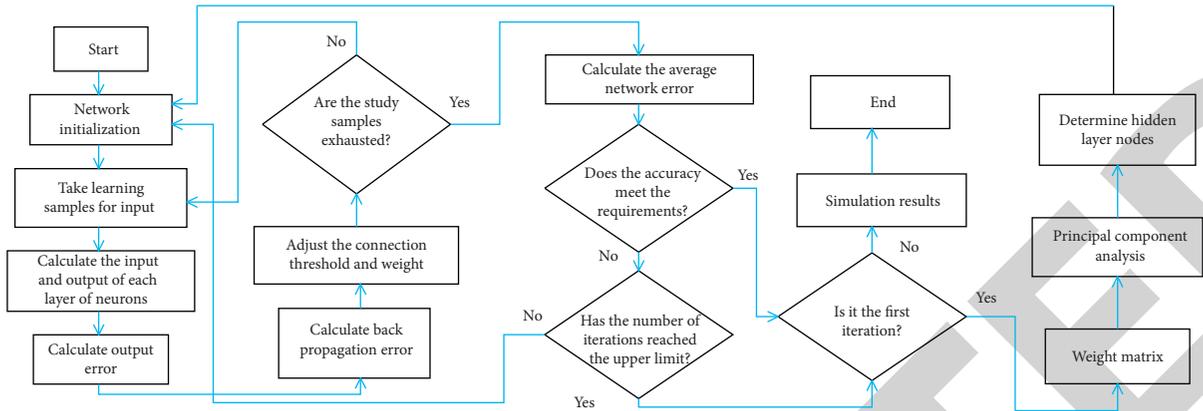


FIGURE 4: Improved BP algorithm flowchart.

The improved BP algorithm based on PCA mainly reduces the dimension of the weight matrix between the input layer and the hidden layer. The larger the number of hidden layer neurons, the higher the training accuracy, but after the number of hidden layer neurons reaches a certain value, as the number of neurons increases, the increase in training accuracy becomes smaller and smaller. It takes longer and longer. Therefore, using the PCA algorithm to reduce the dimension of the weight matrix can reduce the number of neurons in the hidden layer, shorten the training time, and speed up the training while ensuring the accuracy of training.

5. Simulation Experiment and Result Analysis

5.1. Experimental Simulation of GPU Working Mechanism. Graphics processing unit (GPU) is a powerful computing device. It is a processor dedicated to processing graphics and images. The GPU architecture is different from the central processing unit (CPU). The CPU is a multicore structure, it has hundreds of thousands of many-core structures, and it supports parallel processing of tens of thousands of threads. GPU has been used for high parallelization of floating-point operations from the beginning, with the goal of having higher performance and speed than CPU in terms of parallelization, matrix, and vector operations.

Unlike the CPU, the GPU was originally designed for high parallel computing and high intensive computing, so most of the transistors in the GPU are used for data processing, especially suitable for processing single instruction multiple data streams (SIMD), while the CPU is mainly used for non-data caching or flow control. Each computing core of the GPU performs the same program, so the GPU does not need complicated processes to control, and the program runs in parallel on the computing core, and the memory access delay is scheduled or hidden by the thread block, which causes the GPU to not need a large amount of data cache. The complicated process control and large data cache are necessary for the CPU. The disadvantage of GPU is that GPU-accelerated computing is at the cost of offloading overhead, and compared with CPU, due to the large number of computing cores, the operating frequency of GPU is

relatively low. We need to weigh these issues in order to benefit from GPU programming.

The GPU is characterized by multicore high-speed parallel computing. Most of the GPU's internals are floating-point arithmetic units. On the other hand, its internal controllers and caches are relatively small, which leads to the lack of logic judgment functions in the GPU itself. Processing is slower, so we use GPU to perform repetitive parallel calculations rather than as a logic controller in supercomputing. The computer delegates other decision-making functions to the CPU. Figure 5 shows the performance comparison chart of GPU and CPU.

5.2. Determination of the Number of Neurons in the Hidden Layer and Comparative Experiments. According to the designed improved BP algorithm, the neural network is trained, and the optimal number of hidden layer neurons is 20. We use MATLAB's Neural Network Training (nntool) toolbox to train BP neural networks with 10, 20, and 30 hidden layer neurons. During the training process, the relationship between the Mean Square Error (MSE) reached by the verification set and the number of training steps is shown in Figure 6.

Figure 6 shows the training results when the number of hidden layer neurons obtained by the improved BP algorithm is 10, 20, and 30 hours, respectively. It can be seen from the experimental results that when the number of neurons in the hidden layer is 20, the smaller mean square error of the validation set is obtained when the number of training steps is 940. When the number of neurons in the hidden layer is set to 10 and 30, respectively, if the mean square error of the corresponding verification set is to reach the same value as the mean square error of the verification set when the number of hidden layer neurons is 20, the number of steps has increased. As shown in Figure 6, under the condition that the mean square error of the verification set is certain, when the number of hidden layer neurons is 20, the number of steps required is the least. Running the same number of steps, the hidden layer of 20 achieves the highest accuracy. Therefore, when the number of hidden layers is 20, the digital image can be recognized quickly and accurately.

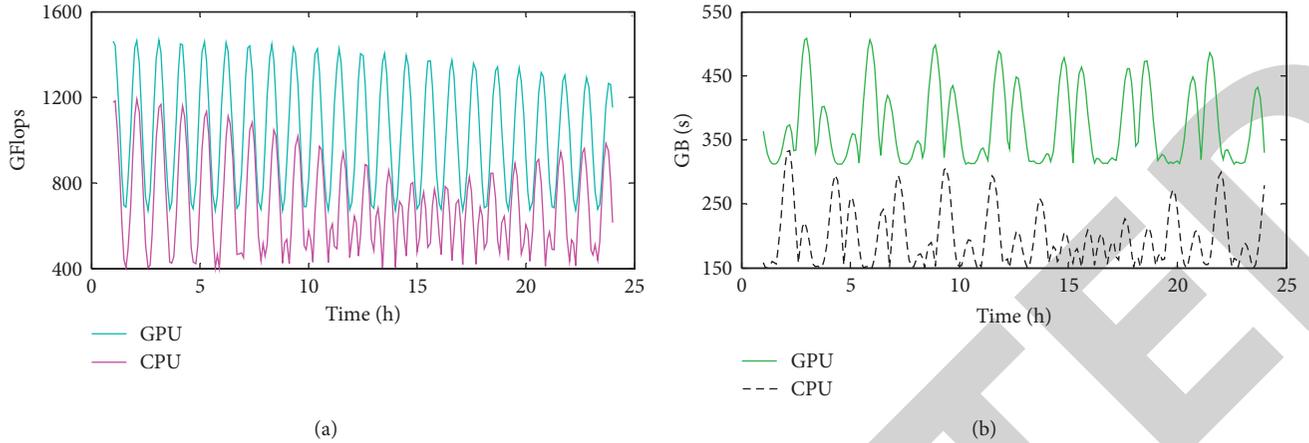


FIGURE 5: GPU and CPU performance comparison chart. (a) Comparison of double-precision floating-point number calculation performance. (b) Memory bandwidth comparison.

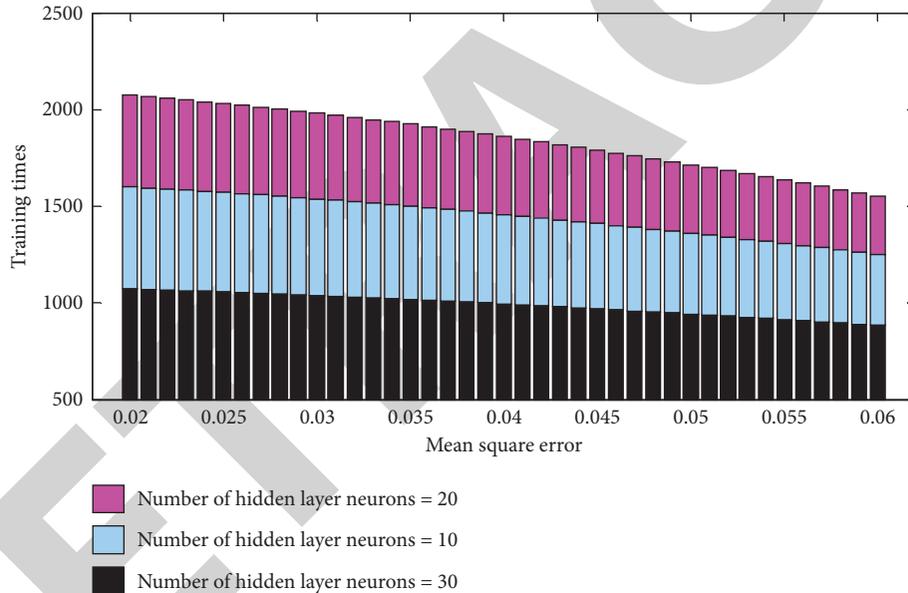


FIGURE 6: Comparison of the relationship between the number of errors of different hidden layers and the number of training steps.

5.3. Distributed Artificial Intelligence Image Recognition Experiment Based on HOG Feature Extraction of PCA Dimensionality Reduction. We use PCA to perform dimensionality reduction experiments on HOG feature dimensions, and the results obtained with the same image size are shown in Figure 7. It can be seen from Figure 7 that when the HOG takes the same feature dimension, the image recognition rate after dimensionality reduction is improved by using PCA to reduce the dimensionality compared with the unreduced detection result.

Figure 8 shows the missed detection rate and false detection rate of the image in the case of dimensional changes. It can be seen from the figure that the missed detection rate and false detection rate of the image obtained by using

different dimensional HOG features for dimensionality reduction are different. It can be concluded that the use of PCA to reduce the dimensionality of HOG features can improve the detection rate and detection efficiency of target detection to a certain extent, but the effect is not particularly significant.

5.4. Image Recognition Effect Comparison Test. According to the improved BP algorithm and processing flow, this paper imports 1000 digital images in batches, and the recognition results obtained are shown in Figure 9. The experimental results show that the improved BP neural network algorithm improves the accuracy of image recognition.

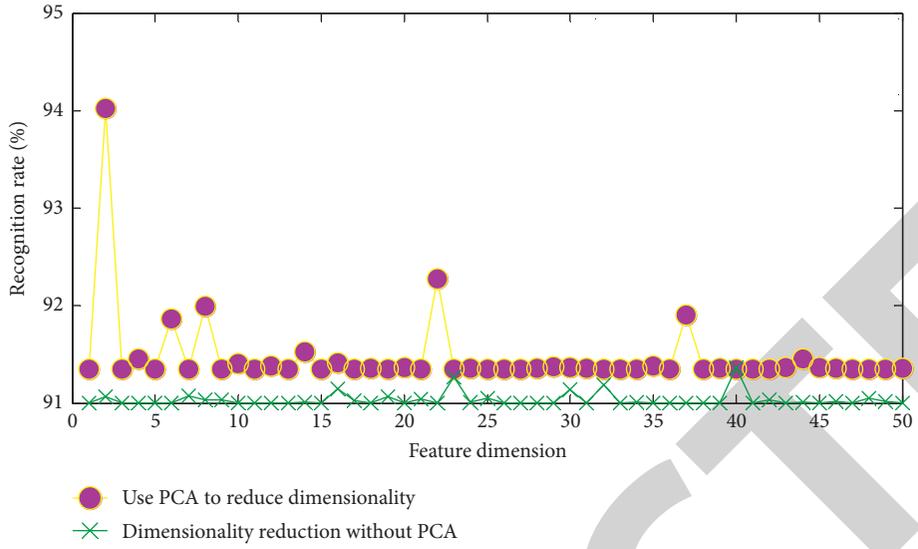


FIGURE 7: Image recognition result after dimensionality reduction based on PCA.

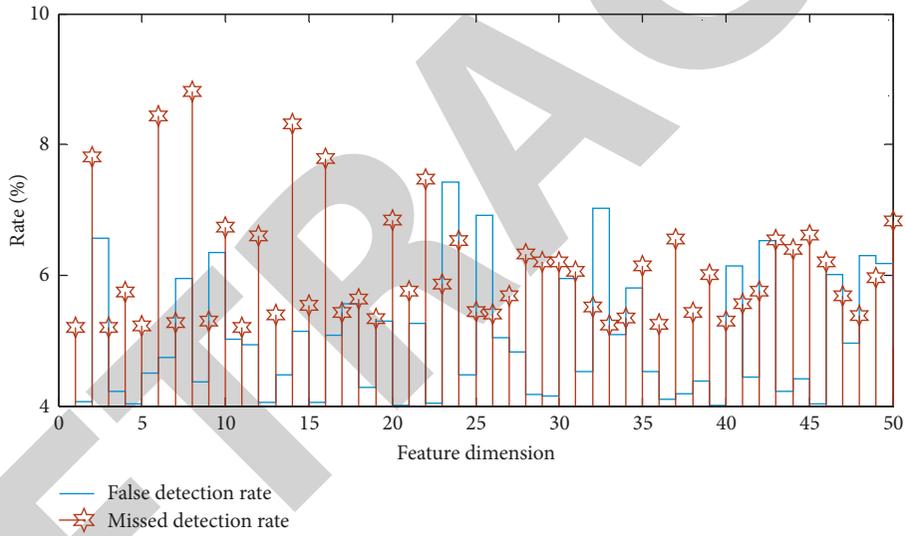


FIGURE 8: The missed detection rate and false detection rate of the image under the change of dimensionality.

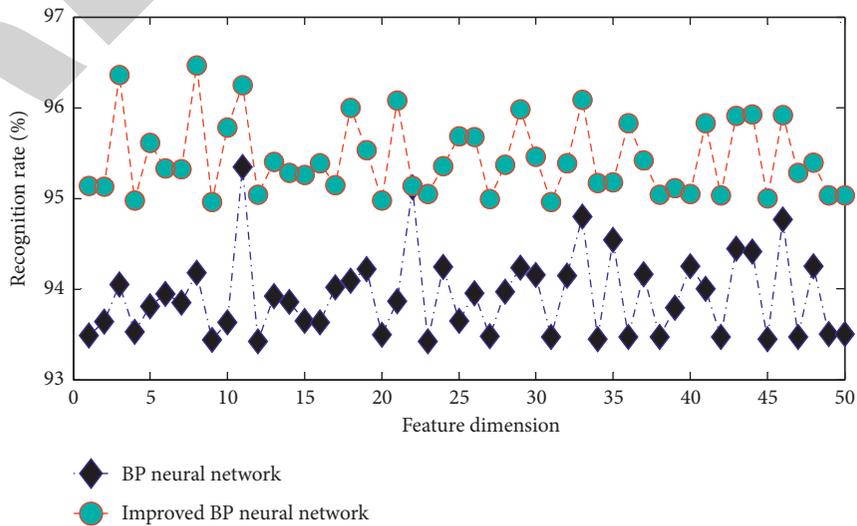


FIGURE 9: Comparison of the recognition effect of arbitrary 1000 test images.

6. Conclusion

Because the current HOG feature extraction algorithm consumes a lot of time, and there are a lot of redundant features, the current algorithm has low detection accuracy and detection efficiency for the detection target. This paper proposes to use GPU and CPU parallelism to improve the existing algorithm. This article introduces the principles and implementation steps of algorithms such as HOG feature, SVM, and PCA and conducts experiments on distributed artificial intelligence image recognition based on HOG feature extraction and SVM classification. We analyze the experimental results of the detection algorithm based on HOG feature extraction and SVM classification to find out the shortcomings and defects. It is proposed to use PCA to perform dimensionality reduction operations on HOG features and perform experiments on distributed artificial intelligence image recognition. The analysis of the experimental results shows that using PCA to reduce the dimensionality of HOG features improves the accuracy and detection efficiency of distributed artificial intelligence image recognition, and the improved effect can meet the actual needs of image recognition. We use GPU to accelerate HOG in parallel. After analyzing the results, it is concluded that the detection rate of the HOG feature extraction algorithm accelerated by GPU is greatly increased, which can basically meet the needs of image recognition. An image recognition method for digital display instruments is proposed, which accelerates the recognition speed of digital images and reduces the time required for recognition on the basis of ensuring accuracy. Finally, the effectiveness of the algorithm is verified by experiments.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Key R&D Program of China (2019YFB1802700).

References

- [1] B. Maschler, S. Kamm, N. Jazdi, and M. Weyrich, "Distributed cooperative deep transfer learning for industrial image recognition," *Procedia CIRP*, vol. 93, pp. 437–442, 2020.
- [2] M. Chen, S. Lu, and Q. Liu, "Uniform regularity for a Keller-Segel-Navier-Stokes system," *Applied Mathematics Letters*, vol. 107, Article ID 106476, 2020.
- [3] Z. Zhang, C. Lai, H. Liu, and Y.-F. Li, "Infrared facial expression recognition via Gaussian-based label distribution learning in the dark illumination environment for human emotion detection," *Neurocomputing*, vol. 409, pp. 341–350, 2020.
- [4] Y. Xu, Z. Dai, J. Wang, Y. Li, and H. Wang, "Automatic recognition of palaeobios images under microscope based on machine learning," *IEEE Access*, vol. 8, pp. 172972–172981, 2020.
- [5] R. Mayer and H.-A. Jacobsen, "Scalable deep learning on distributed infrastructures," *ACM Computing Surveys*, vol. 53, no. 1, pp. 1–37, 2020.
- [6] R. Mühlhoff, "Human-aided artificial intelligence: or, how to run large computations in human brains? Toward a media sociology of machine learning," *New Media & Society*, vol. 22, no. 10, pp. 1868–1884, 2020.
- [7] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–33, 2020.
- [8] R. C. Chen, C. Dewi, W. W. Zhang et al., "Integrating gesture control board and image recognition for gesture recognition based on deep learning," *International Journal of Applied Science and Engineering*, vol. 17, no. 3, pp. 237–248, 2020.
- [9] C. Mo and W. Sun, "Point-by-point feature extraction of artificial intelligence images based on the Internet of Things," *Computer Communications*, vol. 159, pp. 1–8, 2020.
- [10] J. Liu, J. Tan, J. Qin et al., "Smoke image recognition method based on the optimization of SVM parameters with improved fruit fly algorithm," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 8, pp. 3534–3549, 2020.
- [11] R. Vinuesa, H. Azizpour, I. Leite et al., "The role of artificial intelligence in achieving the Sustainable Development Goals," *Nature Communications*, vol. 11, no. 1, pp. 1–10, 2020.
- [12] J. Zhang, J. Liu, B. Pan, and Z. Shi, "Domain adaptation based on correlation subspace dynamic distribution alignment for remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 11, pp. 7920–7930, 2020.
- [13] D. S. Shakyia, "Analysis of artificial intelligence based image classification techniques," *Journal of Innovative Image Processing*, vol. 2, no. 1, pp. 44–54, 2020.
- [14] L. Pan, C. Li, Y. Zhou, B. Xiong, and R. Chen, "A combinatorial convolutional neural network of double subnets for food-ingredient recognition," *International Journal of Embedded Systems*, vol. 13, no. 4, pp. 439–448, 2020.
- [15] A. Ikeda, H. Nosato, Y. Kochi et al., "Support system of cystoscopic diagnosis for bladder cancer based on artificial intelligence," *Journal of Endourology*, vol. 34, no. 3, pp. 352–358, 2020.
- [16] S. Kalra, H. R. Tizhoosh, S. Shah et al., "Pan-cancer diagnostic consensus through searching archival histopathology images using artificial intelligence," *NPJ Digital Medicine*, vol. 3, no. 1, pp. 1–15, 2020.
- [17] K. Alexopoulos, N. Nikolakis, and G. Chryssoulouris, "Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 5, pp. 429–439, 2020.
- [18] U. Paschen, C. Pitt, and J. Kietzmann, "Artificial intelligence: building blocks and an innovation typology," *Business Horizons*, vol. 63, no. 2, pp. 147–155, 2020.
- [19] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "DeTrac: transfer learning of class decomposed medical images in convolutional neural networks," *IEEE Access*, vol. 8, pp. 74901–74913, 2020.

- [20] N. Abderrahmane, E. Lemaire, and B. Miramond, "Design space exploration of hardware spiking neurons for embedded artificial intelligence," *Neural Networks*, vol. 121, pp. 366–386, 2020.
- [21] H. Zhu, "Big data and artificial intelligence modeling for drug discovery," *Annual Review of Pharmacology and Toxicology*, vol. 60, no. 1, pp. 573–589, 2020.
- [22] F. Schwendicke, W. Samek, and J. Krois, "Artificial intelligence in dentistry: chances and challenges," *Journal of Dental Research*, vol. 99, no. 7, pp. 769–774, 2020.
- [23] O. Debauche, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, and F. Lebeau, "Edge computing and artificial intelligence semantically driven. Application to a climatic enclosure," *Procedia Computer Science*, vol. 175, pp. 542–547, 2020.
- [24] C. S. Pitt, A. S. Bal, and K. Plangger, "New approaches to psychographic consumer segmentation: exploring fine art collectors using artificial intelligence, automated text analysis and correspondence analysis," *European Journal of Marketing*, vol. 54, no. 2, pp. 305–326, 2020.