

## Research Article

# Water Quality Classification for Inland Lakes and Ponds with Few Color Image Samples Based on Triple-GAN and CSNN

Zan Yang <sup>1</sup>, Wei Nai <sup>2</sup>, Dan Li <sup>1</sup> and Yidan Xing <sup>2</sup>

<sup>1</sup>Faculty of Science, Tongji Zhejiang College, Jiaxing 314051, China

<sup>2</sup>Department of Electronic and Information Engineering, Tongji Zhejiang College, Jiaxing 314051, China

Correspondence should be addressed to Wei Nai; alexni@tongji.edu.cn

Received 16 February 2022; Revised 20 April 2022; Accepted 29 April 2022; Published 13 June 2022

Academic Editor: Jude Hemanth

Copyright © 2022 Zan Yang et al. 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.

Water color is an important representation reflecting the characteristics of its quality in inland lakes or ponds; however, sufficient water color image samples are often difficult to obtain due to the limitation of fishery production. For few color image samples, the existing data enhancement methods based on the depth generation model have the problems of low quality of generated data, difficulty of network training, and so on; moreover, for image classification, traditional methods based on convolutional neural network (CNN) cannot effectively extract the potential manifold structure features in the image and the full connection layer in CNN cannot simulate biological neurons well, resulting in high time cost and low efficiency. In this paper, a water quality classification method has been proposed to solve the above problems, the improved semisupervised triple-generation adversarial network (triple-GAN) algorithm is used to enhance the few water color image samples, and the feature data can then be extracted from enhanced data by manifold learning method t-distributed stochastic neighborhood embedding (t-SNE). Moreover, convolutional spiking neural network (CSNN), in which spiking neural network (SNN) has replaced the original full connection layer of CNN, is used for final water quality classification. The main contribution of this paper is to build a new algorithm framework, introduce triple-GAN and CSNN into the field of classification of few water color image samples for the first time, and make an exploration of integrating artificial intelligence (AI) and water quality analysis problems. By comparing with traditional methods, the proposed method is proved to have the advantages of less time-consuming, low operation cost, and high classification accuracy.

## 1. Introduction

**1.1. Background.** The output of aquatic products in inland lakes and ponds is an important part of fishery production in human society. Common sense tells people that due to the characteristics of geographical property or characteristics of the target breeding fish stocks, the water in different lakes and ponds often has different biological, chemical, and physical characteristics [1]. These characteristics can reflect the current ecological state and the related behavior of fish stocks and can provide an important reference for grasping the state of fishery ecosystem and its sustainable development. It is of no doubt that the water resource quality of inland lakes and ponds has a significant impact on the survival of fish and other biological populations, including

plankton, microorganisms, and zooplankton [2]. In order to meet the requirements of fishery production, it is of great significance to evaluate the water quality of lakes and ponds.

Water color is an important representation reflecting the characteristics of water quality; it can be used as the basis for water quality evaluation [3, 4]. In actual fishery production, water quality is mostly judged based on the observation from naked eyes and experiences; thus, judgment deviation caused by subjectivity is prone to happen and the comparability and interpretability of observation results are poor [5]. Digital image processing technique based on computer vision, combined with the use of artificial intelligence (AI) algorithms, provides a new idea for water quality evaluation. At the same time, it can also make the rapid discrimination of water quality possible. Up to now, a large number of

scholars have studied the water quality evaluation methods for different waters. Also, there have been many reports on employing different types of AI algorithms to solve the problem of water quality evaluation which will be discussed in Section 2 of this paper.

However, when evaluating water quality, especially the water quality of inland lakes and ponds based on AI algorithms under the background of fishery production, there are often some problems in its actual operation, which have not been properly solved yet. First of all, it always takes time to collect samples of different water colors in the same lake or pond area, which makes the color samples provided for water quality analysis often insufficient; moreover, in the process of finding the effective characteristics of water quality, due to the constraints of the objective function in many AI algorithms, the efficiency of employing the algorithms themselves to find the optimal solution based on gradient descent (GD) related methods is not high and even falls into the local optimal solutions.

## 1.2. Objective of the Paper

**1.2.1. Research Motivation.** For the classification of few water color image samples mentioned above, deep generation models (such as generative adversarial network (GAN) or other automatic encoders) are usually used to realize data enhancement. However, traditional deep generation models can only generate fake sample data randomly by noise and the randomness is somehow too strong to approach the real data with high quality. Even if some models introduce semisupervision mechanisms, the structures of generator and discriminator are still solidified, and there is lack of mechanism innovation, the network training is unstable and prone to the problems of gradient disappearance and gradient explosion. There are two reasons for the instability of network training, gradient disappearance, and gradient explosion: (1) the traditional activation function cannot meet the excellent properties of nonlinearity, unsaturated, and differentiability almost everywhere at the same time. After the activation function is nested layer by layer in the deep network, its disadvantages will be amplified accumulatively, which is not conducive to the subsequent parameter optimization. (2) The traditional optimizer of deep generation model generally uses the line search method in convex optimization theory to optimize parameters, such as GD or stochastic GD (SGD). Such methods are often limited to local gradient information and easy to fall into the trap of local extreme value. The closer to the optimal value point, the more serious sawtooth oscillation appears, which is easy to cause the low quality of image generated.

Moreover, water quality images often directly use the convolution layer of convolutional neural network (CNN) for feature extraction, but the potential manifold geometry in the picture (such as algebraic structure, probability structure, and topology) can hardly be extracted effectively. Even if the manifold learning method is used to reduce the dimension of water color image, there is often a lack of parameter optimization method which is coupled with the

optical information of image and depends on the principle of physical optics. In addition, the full connection layer in traditional CNN has many drawbacks: (1) the neuron model is too simple to simulate the changes of membrane potential and spiking releasing process of biological neurons; (2) the output value of neuron is only an approximate representation of neural information, and its inherent limitation is that the time information of a single spiking is not used; (3) a large number of label datasets are usually required to drive the fitting of the network, and thus, the energy consumption is relatively high and efficiency is relatively low; (4) the expansibility of the structure is not strong, especially not conducive to dealing with spatiotemporal events.

**1.2.2. Main Contribution of This Paper.** In this paper, a water quality classification method for inland lakes and ponds just under the condition of few color image samples based on triple-GAN and convolution spiking neural network (CSNN) has been proposed. For this paper, the main contributions are as follows:

- (1) In order to achieve better training effect under the condition of few samples, triple-GAN is used to enhance the collected original data, which abandons the zero-sum game process between generator and discriminator, but adopts the idea that the discriminator adds a supervised learning process based on few samples of real data under a certain category in order to improve the authenticity efficiency of the generated data under a certain category. Also, Xwish activation function is used to replace all of the activation functions in triple-GAN algorithm so as to avoid the defects of gradient disappearance and gradient explosion mentioned before; at the same time, fixed memory step fractional order gradient descent method (FMSGDM) and fixed memory step fractional order gradient ascent method (FMSGAM) based on Caputo derivative are used to replace the traditional GD in solving the loss function so as to avoid falling into the trap of local optimization. Based on the above method, the probability distribution contained in few water color samples can be properly learnt and new high-quality samples under the condition of this probability distribution can be properly generated.
- (2) Manifold learning nonlinear t-distributed stochastic neighbor embedding (t-SNE) is used to reduce the dimension of the original and generated data; on the one hand, the noise in the color samples can be reduced; on the other hand, the advantage is that t-SNE can be utilized to ensure the probability isomorphism of the data manifold structure before and after dimensionality reduction to extract the data features that maintain the manifold structure. In order to match the optical information of water color samples, a new optical optimization algorithm, Gaussian random parameter-based combination optics inspired optimization (GCOIO), for parameter optimization has also been proposed.

- (3) Convolutional spiking neural network (CSNN), in which spiking neural network (SNN) has replaced the original full connection layer of CNN, is used for final water quality classification, so as to better simulate the membrane potential change and spiking releasing process of biological neurons.

**1.3. Paper Organization.** The remainder of this paper is organized as follows: a comprehensive literature review on water quality evaluation related research will be done in Section 2: methodologies, hardware and system development, and the specific research on water quality analysis based on image analysis and artificial intelligence algorithm technology. Then, in Section 3, triple-GAN, which has been introduced in to deal with few samples, will be introduced in detail, including the replacement of its activation function and the optimization of its loss function. In Section 4, the optimal solution of t-SNE and its objective function will be discussed in order to enhance effective features and realize data dimensionality reduction. In Section 5, the detailed process of using CSNN to process the enhanced and dimensionality-reduced data and the final realization of water quality classified evaluation will be discussed in detail. In order to verify the effectiveness and superiority of the proposed method, in Section 6, based on the water sample data of a tilapia aquaculture pond given in literature [5], the proposed method and its key improvements compared with the original triple-GAN and CNN are numerically analyzed in detail. Finally, the conclusion will be drawn in Section 7.

## 2. Literature Review

No matter what kind of water quality is analyzed, the main task is to classify and judge the water quality according to different analysis objectives after processing the collected data or images. As discussed in Section 1, a large number of scholars have carried out relevant research on water quality analysis at present. These studies can be roughly divided into two categories: methodology and system design. In this section, these two categories of research will be reviewed firstly.

**2.1. Methodologies on Water Quality Analysis.** As the water environment contains a large number of chemicals, the traditional water quality analysis methods are mostly based on the chemical perspective and pay attention to the chemical composition of the water. Wang et al. have analyzed the seasonal variation of chemical composition in Taihu Lake, China, based on optically active components in spectral analysis [3]; Abdullah et al. and Ma et al. have analyzed the geochemical characteristics of river water and ground water quality in basin area [6, 7]; Niroumand-Jadidi et al. have tried to acquire the spectra-derived features of water based on multispectral instruments (MSIs) and ocean and land color instruments (OLCIs) [2]; Nayeem et al. have tried to realize rapid determination of the chemical composition in water based on spectral analysis [8].

During the past twenty years or so, scholars have also tried to employ mathematical methodologies into water quality analysis related area. Smith et al. have tried to identify the types of suspended solids in water based on artificial neural network (ANN) in an early stage [9], and Nor et al. have employed multilayer ANN to classify the pollution amount in water [10]; Koponen et al. have employed expert system in handling water quality parameter statistical data [11]; Karami et al. and Shi et al. have both tried to realize water area classification by employing hierarchical cluster analysis (HCA) through environmental variables [12, 13], while Sun et al. have tried to use support vector regression to realize similar object [14]; Zhao et al. and Zhang et al. have both tried to use principal component analysis (PCA) method to find the sources of pollutant in coastal areas and urban rivers [15, 16]. During the past few years, more complex mathematical methodologies, such as unsupervised learning [17] and hybrid algorithm [1, 18, 19], have also been applied in water quality analysis and scholars like Liu et al. have even proposed their own deep learning (DL) based network to predict water quality [20].

Also, there are a bunch of scholars who have tried to go a step further to put methodologies into system design so as to contribute to water quality analysis. Cruz et al. have proposed a process control system for surface water parameter monitoring based on the geology method [21]. McEliece et al. have proposed a model-based decision support system for drinking water evaluation [22].

**2.2. Water Quality Analysis Based on Image Processing and Artificial Intelligence Algorithms.** By considering the actual water quality data collection condition of fishery, most of them are water color analysis after water sample collection. The acquisition of water color image is limited by conditions and often can only reflect the characteristics of water quality locally. Therefore, many scholars have carried out research on image structure extraction and information extraction with incomplete information. Chen and his team members have tried to put forward a variety of image information completion, image information reconstruction, and image information restoration methods, so as to make full use of local information and image semantics and achieve a high success rate of recognition and restoration [23–25], while Xia et al. have proposed a method to track the key target information in the image and achieved a high accuracy [26].

Usually, water color images, as the analysis object, can be acquired via many different ways, and till now, various image processing and AI algorithms have been used to realize water quality analysis. The acquisition of satellite remote sensing images has been widely used in water quality monitoring. Based on satellite image data, Ribeiro et al., Dona et al., Nazeer et al., and Batur et al. have employed ANN, expert experience, periodical regression model, and PCA-based response surface regression to realize the estimation and prediction of specific chemical components or organic matter content in water [27–30]. Apart from these reports, some scholars have used neural network (NN) related basic algorithms or hybrid algorithm to solve the

problems of water quality analysis based on images. Isikdogan et al. have used fully CNN to analyze and predict the water availability of surface water mapping image data [31]. Yuan et al. have used a hybrid algorithm of long short-term memory (LSTM) network and neural network (LSTM-NN) to analyze pollutants in water with fish activities [32], and Pu et al. have employed CNN with hierarchical structure to try to establish the relationship between remote sensing images of inland lakes and on-site water quality level [33].

### 3. Data Enhancement Based on Triple-GAN

**3.1. Triple-GAN.** As described in Section 1, by considering that there are always very few samples of water color images which can be applied to fishery, it is necessary to enhance the original data. Conventional data enhancement methods include partial differential equation method based on variational method, GAN algorithm, and variational self-encoder algorithm. The main framework algorithm used in this paper is triple-GAN.

GAN is a kind of unsupervised deep learning algorithm. Its main idea is based on the zero-sum game between generator and discriminator, and its theoretical foundation is Nash equilibrium [34]. The objective function of GAN can be described as

$$\min_G \max_D \{E_{p(x)}[\log D(x)] + E_{p_z(z)}[\log(1 - D(G(z)))]\}. \quad (1)$$

The logic process is that the random noise  $z$  generates the fake sample data  $G(z)$  and then gives it to the discriminator  $D$  to judge the authenticity. Therefore, in order to achieve Nash equilibrium, on the one hand, it is required to minimize  $1 - D(G(z))$ , that is, the closer the false sample data are to the real data, the better; on the other hand, the larger the result of discriminator judging the real sample data, the better. Finally, the objective function is given in the form of cross entropy. However, because of the unstable training of GAN, it is easy to lead to gradient disappearance and gradient explosion, which restrict its generation effect.

On the basis of GAN algorithm, some scholars have proposed conditional GAN (CGAN) [35], category-aware GAN (CatGAN) [36], and triple-GAN [37]. Among them, triple-GAN is a semisupervised deep learning algorithm, which abandons the zero-sum game process between generator and discriminator, but adopts the Nash equilibrium of classifier, generator, and discriminator under category conditions; its objective function is as follows:

$$\min_{C,G} \max_D \{E_{p(x,y)}[\log D(x,y)] + \alpha E_{p_c(x,y)}[\log(1 - D(x,y))] + (1 - \alpha) E_{p_g(x,y)}[\log(1 - D(G(y,z), y))]\}. \quad (2)$$

Even if it can achieve good training results, the problem of gradient disappearance and gradient explosion in traditional GAN still exists. The problem lies in the activation function of neurons in GAN. In order to overcome this

problem, a new depth network activation function, Xwish, is used to replace the original activation function in triple-GAN.

**3.2. Xwish Activation Function.** In the framework of the neurons described in ANN, the target of activation function is to put the weighted sum into a nonlinear function for nonlinearity. To be specific, the weighted sum can be regarded as a linear hyperplane which divides spatial data into two categories, but it is a just linear classifier which does not have universal significance, as most problems in practice are nonlinear. Therefore, activation function is used to make the linear classifier to be nonlinear.

Classical activation functions all have their disadvantages, such as saturation effect on both sides of sigmoid function, a dead zone on the left side of ReLU, and non-differentiable at 0 point of PReLU, ELU, and LReLU. Thus, the network structure in this paper adopts the Xwish activation function introduced in literature [38]. It satisfies the properties of nonlinearity, unsaturation, and differentiability everywhere. The expression of Xwish function can be described as

$$\rho(x) = \frac{x \cdot [\arctan(\beta x) + 0.5\pi]}{\pi}. \quad (3)$$

The expression of its first derivative is

$$\rho'(x) = \frac{1}{\pi} [\arctan(\beta x) + 0.5\pi] + x \cdot \frac{\beta x}{1 + (\beta x)^2}, \quad (4)$$

where  $\beta$  is a hyperparameter, which is unified in this paper as  $\beta = 0.1$ .

**3.3. FMSGDM and FMSGAM.** In the loss function optimization part of triple-GAN, many studies including [37] still use the GD method or stochastic GD (SGD) method; however, both GD and SGD methods are easy to fall into the trap of local optimum, so the FMSGDM introduced in literature [39] has been chosen to replace the original loss function in triple-GAN in this paper.

The advantage of fractional gradient over classical gradient is that it is more sensitive to gradient information and is not easy to fall into the trap of local optimum. Fractional gradient is defined by fractional derivative, which can be divided into three types: Grünwald–Letnikov, Riemann–Liouville, and Caputo. Grünwald–Letnikov derivative is defined by the limit, so it is rarely used in engineering operation. Riemann–Liouville derivative takes the form of integration before derivation, while Caputo derivative takes the form of derivation before integration. As fractional derivative does not satisfy the exchange law of integration and derivation, Riemann–Liouville and Caputo are two derivatives with different structures. Caputo is more suitable for engineering operation, so the FMSGDM discussed in this part is defined by Caputo derivative, and the specific iterative format of the FMSGDM can be described as

$$x_{k+1} = x_k - \mu_{x_{k-K}}^C D_{x_{k-K}}^\alpha f(x) = x_k - \sum_{i=1}^{+\infty} \binom{\alpha-1}{i-1} \frac{f^{(i)}(x)}{\Gamma(i+1-\alpha)} (x_k - x_{k-K})^{i-\alpha}, \quad (5)$$

where  $K \in \mathbb{Z}^+$ ,  $\mu$  is the step size,  $\alpha$  is the order, and  $\Gamma(\cdot)$  is the Gamma function.

The flow of FMSGDM is as follows:

- (1)  $x_1, x_2, \dots, x_{N-1}$  are inputs and  $x_N$  is the output.
- (2)  $\alpha, \mu, N$ , and  $K$  are hyperparameters, which are given manually.
- (3) Perform the following iterative cycle from  $k=0$  to  $k=N-1$ :

$$\frac{C}{\theta_{k-K}^c} D_{\theta_{k-K}^c}^\alpha \left[ \frac{1}{m_d} \left( \sum_{(x_d, y_d)} \log D(x_d, y_d) \right) + \frac{\alpha}{m_c} \sum_{(x_c, y_c)} \log(1 - D(x_c, y_c)) + \frac{1-\alpha}{m_g} \sum_{(x_g, y_g)} \log(1 - D(x_g, y_g)) \right]. \quad (8)$$

- (2) Solve the gradient of classifier parameter  $\theta^c$ :

$$\frac{C}{\theta_{k-K}^c} D_{\theta_{k-K}^c}^\alpha \left[ \frac{\alpha}{m_c} \sum_{(x_c, y_c)} \log p_c \left( \frac{y_c}{x_c} \right) \log(1 - D(x_c, y_c)) \right]. \quad (9)$$

- (3) Solve the gradient of class-conditional generator parameter  $\theta^g$ :

$$\frac{C}{\theta_{k-K}^g} D_{\theta_{k-K}^g}^\alpha \left[ \frac{1-\alpha}{m_g} \sum_{(x_g, y_g)} \log(1 - D(x_g, y_g)) \right]. \quad (10)$$

Then, the value of gradients can be put into FMSGDM to let the problem be solved.

The pseudocode of the training process of triple-GAN with Xwish activation function, FMSGDM, and FMSGAM is as follows:

**FOR** number of training iterations, **DO**

- (1) Sample a batch of pairs  $(x_g, y_g) \sim p_g(x, y)$  of size  $m_g$ , a batch of pairs  $(x_c, y_c) \sim p_c(x, y)$  of size  $m_c$ , and a batch of labelled data  $(x_d, y_d) \sim p(x, y)$  of size  $m_d$ .
- (2) Update discriminator  $D$  (with network activation function Xwish) by FMSGAM, and the fixed memory step fractional order gradient of  $D$  is calculated by  $\frac{C}{\theta_{k-K}^d} D_{\theta_{k-K}^d}^\alpha [1/m_d (\sum_{(x_d, y_d)} \log D(x_d, y_d)) + \alpha/m_c \sum_{(x_c, y_c)} \log(1 - D(x_c, y_c)) + 1 - \alpha/m_g \sum_{(x_g, y_g)} \log(1 - D(x_g, y_g))]$ .
- (3) Update classifier  $C$  (with network activation function Xwish) by FMSGDM, and the fixed memory step fractional order gradient of  $C$  is calculated by  $\frac{C}{\theta_{k-K}^c} D_{\theta_{k-K}^c}^\alpha [\alpha/m_c \sum_{(x_c, y_c)} \log p_c(y_c/x_c) \log(1 - D(x_c, y_c))]$ .

$$\zeta = \sum_{i=1}^{+\infty} \binom{\alpha-1}{i-1} \frac{f^{(i)}(x)}{\Gamma(i+1-\alpha)} (x_k - x_{k-K})^{i-\alpha}, \quad (6)$$

$$x_{k+1} = x_k - \mu \zeta, \quad (7)$$

until the termination conditions are met. When the minus sign in equation (7) has been changed to the plus sign, the FMSGAM can be acquired.

Furthermore, it only needs to solve all types of gradients of the loss function in triple-GAN:

- (1) Solve the gradient of discriminator parameter  $\theta^d$ :

- (4) Update class-conditional generator  $G$  (with network activation function Xwish) by FMSGDM, and the fixed memory step fractional order gradient of  $G$  is calculated by  $\frac{C}{\theta_{k-K}^g} D_{\theta_{k-K}^g}^\alpha [1 - \alpha/m_g \sum_{(x_g, y_g)} \log(1 - D(x_g, y_g))]$ .

**END FOR**

## 4. Data Dimensionality Reduction Based on t-SNE

As the water color image data are usually a bit too uniform, in order to facilitate the effective feature extraction in the subsequent classification step employing deep networks, dimensionality reduction of the data should be done so as to make the features in the water color image to be more sparse and obvious. In this paper, t-SNE described in literature [40] is used to reduce the dimensionality of data.

**4.1. t-SNE.** t-SNE is based on stochastic neighbor embedding (SNE), while SNE is an unsupervised nonlinear dimensionality reduction method in the field of machine learning [41]. The basic idea of SNE is to let the nearest points expressed by probability in the high-dimensional space to be projected into low-dimensional space. Assuming that there are two points  $\phi_i$  and  $\phi_j$  in the high-dimensional space and the conditional probability that  $\phi_j$  is close to  $\phi_i$  is  $p_{j|i}$ , the spatial Euclidean distance can be transformed into a probability value with the help of Gaussian distribution, namely,

$$p_{j|i} = \frac{\exp(-\|\phi_i - \phi_j\|/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\phi_i - \phi_k\|/2\sigma_i^2)}, \quad (11)$$

where  $\sigma_i^2$  is the variance of the normal distribution centered at  $\phi_i$ , after  $\phi_i$  and  $\phi_j$  are projected, the corresponding points in the low-dimensional space are  $\Phi_i$  and  $\Phi_j$ , and the probability value formula is

$$q_{j|i} = \frac{\exp(-\|\Phi_i - \Phi_j\|)}{\sum_{k \neq i} \exp(-\|\Phi_i - \Phi_k\|)}. \quad (12)$$

According to the probability value, the discrete probability distributions corresponding to high-dimensional space and low-dimensional space are  $P_i$  and  $Q_i$ . In order to ensure the effectiveness of the algorithm, the optimal parameters in equations (11) and (12) must be found, so that the two distributions are close enough, that is, some measure should be taken to investigate the similarity of the two probability distributions. In machine learning, Kullback-Leibler (KL) divergence is generally used to measure the difference between two probability distributions, and its formula is

$$\text{KL}(p\|q) = \sum_x p(x) \ln \frac{p(x)}{q(x)}. \quad (13)$$

It is hoped that the smaller the KL divergence is, the better. Therefore, the algorithm is finally transformed into the problem of finding the minimum value of the following objective function:

$$L(\Phi_i) = \sum_{i=1}^n \text{KL}(P_i\|Q_i) = \sum_{i=1}^n \sum_{j=1}^n p_{j|i} \ln \frac{p_{j|i}}{q_{j|i}}, \quad (14)$$

where  $n$  is the number of samples. The above optimization problem can be solved by GD or SGD.

As mentioned before, t-SNE improves SNE algorithm. First,  $p_{j|i}$  and  $q_{j|i}$  are not necessarily equal; thus, symmetry cannot be ensured. Therefore, equations (11) and (12) can be changed as the form defined by joint probability as

$$p_{ij} = \frac{\exp(-\|\phi_i - \phi_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\phi_i - \phi_k\|^2 / 2\sigma_i^2)}, \quad (15)$$

$$q_{ij} = \frac{\exp(-\|\Phi_i - \Phi_j\|)}{\sum_{k \neq i} \exp(-\|\Phi_i - \Phi_k\|)}. \quad (16)$$

However, equation (15) is prone to the problem of abnormal values, so equation (16) can be further transformed into

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}. \quad (17)$$

Also, the modified SNE is called symmetrical SNE. It should be pointed out that the definition based on Gaussian distribution will lead to congestion, which means that the points after dimensionality reduction will gather together and will be difficult to distinguish. By considering that t-distribution has a long tail effect, which can help the points after dimensionality reduction to be more sparse with more

obvious characteristics, it is necessary to change the definition of probability value in low-dimensional space to

$$q_{ij} = \frac{(1 + \|\Phi_i - \Phi_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\Phi_i - \Phi_k\|^2)^{-1}}. \quad (18)$$

Till now, the original SNE has been transformed as t-SNE. t-SNE itself still has defects, as its essence is to solve the optimal parameter problem of minimizing KL divergence, and KL divergence is not a real distance which does not satisfy the symmetry of the distance axiom. In order to increase the scalability of the algorithm, t-SNE is further improved in this paper. KL divergence in t-SNE is replaced by JS divergence as

$$\text{JS}(P\|Q) = \frac{1}{2} \text{KL}\left(P\left\|\frac{P+Q}{2}\right.\right) + \frac{1}{2} \text{KL}\left(Q\left\|\frac{P+Q}{2}\right.\right). \quad (19)$$

From equation (19), we can get the optimized objective function as

$$\begin{aligned} \tilde{L}(\Phi_i) &= \sum_i \text{JS}(P_i\|Q_i) \\ &= \frac{1}{2} \sum_i \left[ \text{KL}\left(P_i\left\|\frac{P_i+Q_i}{2}\right.\right) + \text{KL}\left(Q_i\left\|\frac{P_i+Q_i}{2}\right.\right) \right]. \end{aligned} \quad (20)$$

**4.2. GCOIO.** Usually, GD or SGD can be used to solve the objective function, but GD and SGD are easy to fall into the trap of local optimum. Therefore, in this paper, Gaussian random parameter-based combination optics inspired optimization (GCOIO) has been used to replace GD-related optimization method in t-SNE.

Optics inspired optimization (OIO) is a heuristic optimization algorithm [42]; its basic idea is to regard the optimization function as a mirror, the convex part of the function as a convex mirror, and the concave part of the function as a concave mirror. Also, each initial solution is regarded as an initial optical source point. In this way, after the light of each optical source point is reflected by the function mirror, the upright or inverted image that has been scaled down can be obtained by the uneven nature of the reflecting surface, and these series of image points can be used as the initial optical source points for the next step of optimization. As OIO algorithm is also prone to fall into local optimum, and the algorithm is easy to get too premature. Rotation OIO (ROIO) and combination OIO (COIO), which is the combination of OIO and ROIO, have also been proposed so as to improve the operation performance.

No matter which type the OIO-related algorithms mentioned above are, uniform distribution is used to construct the position information of random optical source points; however, uniform distribution has weak randomness and degree of freedom. In this paper, the statistical simulation method Box-Muller transform is used to transform

the uniform distribution into Gaussian distribution, which can help to increase the randomness of the algorithm, and constructs GCOIO.

**4.2.1. Basic OIO Model.** Within the definition domain of the optimization function, the initial optical source point  $O_j^t = [o_{j1}^t, o_{j2}^t, \dots, o_{jn}^t]$  and another variable, vertex coordinates  $F_{i_k}^t$ , are randomly generated. When  $i_k \neq j$ , there is  $F_{i_k}^t = O_j^t$ .

There are 2 processing methods at this time: (1) if  $\tilde{f}(O_j^t) < \tilde{f}(F_{i_k}^t)$  (where  $\tilde{f}$  is the focal length), the corresponding function segment is regarded as a convex function model; (2) if  $\tilde{f}(O_j^t) > \tilde{f}(F_{i_k}^t)$ , then the corresponding function segment is regarded as a concave function model.

Suppose that the mirror function of the  $j$ th optical source point is a concave function and the function position  $s_{j,i_k}^t$  of the optical source point is a random number in  $U[\tilde{f}(O_j^t), \tilde{f}(O_j^t) + d_\infty]$  (where  $d_\infty$  is infinite). In this paper, Box-Muller transform in statistical simulation is used:

$$\begin{cases} X = \cos(2\pi Z_1) \sqrt{-2 \ln Z_2} \\ Y = \sin(2\pi Z_1) \sqrt{-2 \ln Z_2} \end{cases}, Z_1, Z_2 \sim \text{Uniform}[0, 1], \quad (21)$$

where  $Z_1$  and  $Z_2$  obey  $[0, 1]$  uniform distribution.

When transforming uniform distribution  $U[\tilde{f}(O_j^t), \tilde{f}(O_j^t) + d_\infty]$  to the Gaussian distribution  $G[\tilde{f}(O_j^t), \tilde{f}(O_j^t) + d_\infty]$ , during the initial operation, there is

$$d_\infty = \left| \max_{j=1, \dots, n} \{\tilde{f}(O_j^t)\} \right|, \quad (22)$$

where  $n$  is the number of optical source points. According to the imaging method discussed in reference [42], the object distance and the radius of the surface, namely,  $p_{j,i_k}^t$  and  $r_{j,i_k}^t$ , can be obtained, and the center of the surface  $m_{j,i_k}^t$  is also a random number in  $G[\tilde{f}(O_j^t), \tilde{f}(O_j^t) + d_\infty]$ .

Assuming that the function in front of the optical source points is a convex function, the function position  $s_{j,i_k}^t$  of the  $j$ -th optical source point is a random number in  $G[\tilde{f}(O_j^t), \tilde{f}(O_j^t) + d_\infty]$ , so it is easy to obtain the object distance, and the radius of the surface and the center of the surface are also random numbers of  $G[\tilde{f}(O_j^t), \tilde{f}(O_j^t) + d_\infty]$ .

**4.2.2. Iteration of New Optical Source Points.** As discussed above, Gaussian random parameter based OIO has been proposed in this paper to increase the randomness in optimizing the objective function of t-SNE.

In the way of optical source point iteration in Gaussian random parameter based OIO (GOIO), the image length of the  $j$ -th optical source point is  $HI_{j,i_k}^t$ , which is the search step. The position of the  $j$ -th light source point in the  $t$ -th iteration is

$$I_{j,i_k}^t = F_{i_k}^t + HI_{j,i_k}^t \frac{(O_j^t - F_{i_k}^t)}{\|O_j^t - F_{i_k}^t\|} = F_{i_k}^t - \frac{r_{i_k}^t}{2p_{j,i_k}^t - r_{i_k}^t} (O_j^t - F_{i_k}^t), \quad (23)$$

where  $I_{j,i_k}^t$  is the new search solution.

In the way of optical source point iteration in Gaussian random parameter based ROIO (GROIO), the search direction is determined by the first law of light reflection and the objective function is optimized by rotating the position of the image around the main axis. The position of the image of the  $j$ -th light source point in the  $t$ -th iteration is

$$I_{j,i_k}^t = F_{i_k}^t - HO_{j,i_k}^t \frac{q_{j,i_k}^t}{p_{j,i_k}^t} \frac{F_{i_k}^t - \text{Refl}_{O_j^t}(F_{i_k}^t)}{\|F_{i_k}^t - \text{Refl}_{O_j^t}(F_{i_k}^t)\|}, \quad (24)$$

where  $HO_{j,i_k}^t$  is the object length,  $q_{j,i_k}^t$  is the image distance, and  $\text{Refl}_{O_j^t}(F_{i_k}^t)$  is the reflection image point of  $F_{i_k}^t$ .

Based on the discussion above, GCOIO can be obtained by combining the solution update method of GOIO with the solution update method of GROIO. The position of the  $j$ -th light source point in the  $t$ -th iteration is

$$I_{j,i_k}^t = F_{i_k}^t - HO_{j,i_k}^t \frac{q_{j,i_k}^t}{p_{j,i_k}^t} \left( \lambda_j^t \frac{(O_j^t - F_{i_k}^t)}{\|O_j^t - F_{i_k}^t\|} + (1 - \lambda_j^t) \frac{F_{i_k}^t - \text{Refl}_{O_j^t}(F_{i_k}^t)}{\|F_{i_k}^t - \text{Refl}_{O_j^t}(F_{i_k}^t)\|} \right), \quad (25)$$

where  $\lambda_j^t$  is a Gaussian random number on the interval  $[0, 1]$ . In this paper, GCOIO is used to solve the optimization part of t-SNE.

## 5. Water Quality Classification Based on CSNN

The data after dimensionality reduction are then given to deep network for classification. The deep network referred to in this paper is based on CNN framework. Compared with biological NN, traditional NN has inherent limitations in information processing: on the one hand, the neuron model is too simple; on the other hand, the neural spiking is not connected with time [43]. Different from the classical CNN, Xwish activation function is also used between the convolution layer and the pooling layer, and the traditional network is replaced by the spiking neural network in the full connection layer. Thus, in this paper, the above network form is called CSNN.

**5.1. SNN.** A large number of neuroscience experiments show that the visual and auditory nervous systems of many organisms use the time of action potentials (i.e., neural spikings) issued by neurons to encode information [44, 45], and the third-generation ANN model-spiking neural network (SNN) has come into being due to the requirements of actual situations. The main idea of SNN is to use time coding for information transmission and processing and directly use the neural spiking releasing time of neurons as the input and output of the network model, so as to realize the efficient processing of information, which is more pressing into the actual biological neural system than the traditional neural network model [46, 47].

The full connection layer of CSNN in this paper adopts the multilayer SNN structure based on spatiotemporal interest points (STIPs) to replace the original second-generation neural network. Of course, the learning process of this method is still the process of supervised learning algorithm.

**5.1.1. Learning Rules of STIP.** The framework of STIP learning rules is as follows: discrete spiking sequences are transformed into continuous functions through the representation of spiking sequence inner product and can be interpreted as specific neurophysiological signals. The purpose of mapping the spiking sequence set to the reproducing kernel Hilbert space corresponding to the kernel function is to unify the representation of the spiking sequence and the formal definition of the similarity measure of the spiking sequence [48, 49]. By defining the error function of spiking sequence varying with time and the relationship between neuron input spiking sequence and output spiking sequence, the learning of complex spatio-temporal model of spiking sequence is constructed and the inner product form of synaptic weight learning rule is given.

**5.1.2. Spiking Frequency Coding Method.** In the spiking frequency coding method, the amount of information transmitted by a specific spiking mode response does not have to be proportional to the number of spiking in the coding time window. The spiking activity response below the reference frequency can also transmit a large amount of information. The mutual information transmitted by a specific symbol depends on the degree of distinction between the symbol and all other possible symbols. Common spiking coding methods include frequency coding based on spiking counting, frequency coding based on spiking density, frequency coding based on group activity, precise spiking timing coding, first spiking trigger time coding, delay phase coding, and spiking sequence coding. Since the first spiking trigger time coding method has been widely used in the data coding of SNN [50–52], it is used as the spiking coding method here in CSNN in this section.

**5.1.3. Transform of Spiking Sequence.** The input and output of spiking neurons are expressed as spiking sequence coding information. For a given spiking sequence  $s(t)$ , a specific smooth function can be selected, and the spiking sequence can be uniquely transformed into a continuous function by convolution operation:

$$f_s(t) = s(t) * h(t) = \sum_{\tau=1}^M h(t - t^\tau), \quad (26)$$

where  $t^\tau$  indicates the releasing time of the  $\tau$ -th spiking in the neuron and  $M$  is the total number of spiking. For the spiking sequence released in time interval  $\Gamma = [0, T]$ , it is assumed that the spiking sequence input by presynaptic neurons is  $s_i(t) \in S(\Gamma)$ , where  $i = 1, 2, \dots, \bar{N}$ , and  $\bar{N}$  represents the number of presynaptic input neurons. The spiking sequence output by postsynaptic neurons is  $s_a(t) \in S(\Gamma)$ . By employing equation (26), the relationship expression between multiple input spiking sequences and output spiking sequences in time  $t$  after conversion can be expressed as

$$f_{s_a}(t) = \sum_{i=1}^{\bar{N}} w_i f_{s_i}(t), \quad (27)$$

where weight  $w_i$  represents the link strength between presynaptic neuron  $i$  and postsynaptic neuron, namely, the weight of synaptic.

**5.1.4. Learning Rules of Synaptic Weights.** The key of constructing spiking neuron supervised learning algorithm is to define the error function of spiking sequence and the learning rules of synaptic weights. The error of spiking neuron at time  $t$  can be defined as the square of the difference between the output spiking sequence  $s_a$  and corresponding function  $f_s(t)$  of the target spiking sequence  $s_d \in S(\Gamma)$ :

$$E(t) = \frac{1}{2} [f_{s_a}(t) - f_{s_d}(t)]^2. \quad (28)$$

Therefore, the total error of spiking neuron at time  $\Gamma$  is  $\hat{E} = \int_{\Gamma} E(t) dt$ . The gradient of synaptic weight can be calculated by employing spiking sequence error function. The change of synaptic weight  $\Delta w_i$  from presynaptic neuron  $i$  to postsynaptic neuron can be calculated as

$$\Delta w_i = -\eta \nabla \hat{E}_i, \quad (29)$$

where  $\eta$  represents the learning rate and  $\nabla \hat{E}_i$  represents the gradient calculation value of spiking sequence error function  $\hat{E}$  for synaptic weight  $w_i$ , which can be expressed as the integral of the derivative of error function  $E(t)$  to weight  $w_i$  in time interval:

$$\nabla \hat{E}_i = \int_{\Gamma} \frac{\partial E(t)}{\partial w_i} dt. \quad (30)$$

Using the error function and chain rule, the derivative of the error function  $E(t)$  to the synaptic weight  $w_i$  in time  $t$  can be obtained according to (27) and (28):

$$\frac{\partial E(t)}{\partial w_i} = \frac{\partial E(t)}{\partial f_{s_a}(t)} \frac{\partial f_{s_a}(t)}{\partial w_i} = [f_{s_a}(t) - f_{s_d}(t)] f_{s_i}(t), \quad (31)$$

where  $s_i \in S(\Gamma)$  represents the spiking sequence released by neuron  $i$ . According to equation (30), the gradient of error function  $\hat{E}$  for synaptic weight  $w_i$  can be calculated as

$$\nabla \hat{E}_i = \int_{\Gamma} [f_{s_a}(t) - f_{s_d}(t)] f_{s_i}(t) dt = F(s_a, s_i) - F(s_d, s_i), \quad (32)$$

where  $F(s_i, s_j)$  represents the inner product of 2 spiking sequences  $s_i$  and  $s_j$ ; its definition is

$$F(s_i, s_j) = \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \int_{\Gamma} h(t - t_i^m) h(t - t_j^n) dt = \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \kappa(t_i^m, t_j^n), \quad (33)$$

where  $t_i^m$  and  $t_j^n$  represent the releasing time corresponding to the 2 spiking sequences  $s_i$  and  $s_j$ , respectively, and  $N_i$  and  $N_j$  represent the number of spiking corresponding to the 2 spiking sequences  $s_i$  and  $s_j$ , respectively.  $\kappa(\cdot, \cdot)$  is the kernel function, which is generally required to have the characteristics of symmetry, translation invariance, and positive definite, and Gaussian kernel is taken in this paper.



According to the above derivation process, the following synaptic learning rules can be obtained:

$$\Delta w_i = -\eta \left[ F(s_a, s_i) - F(s_d, s_i) \right] = \eta \left[ \sum_{g=1}^{F_d} \sum_{f=1}^{F_i} \kappa(t_d^g, t_i^f) - \sum_{h=1}^{F_a} \sum_{f=1}^{F_i} \kappa(t_a^h, t_i^f) \right], \quad (34)$$

where  $t_i^f$ ,  $t_a^h$ , and  $t_d^g$  represent the releasing time of the corresponding spiking in the input spiking sequence  $s_i$ , the actual output spiking sequence  $s_a$ , and the target spiking sequence  $s_d$ , respectively;  $F_i$ ,  $F_a$ , and  $F_d$  represent the number of spiking contained in the spiking sequences  $s_i$ ,  $s_a$ , and  $s_d$ , respectively.

**5.1.5. Self-Adaptive Learning Rate.** The value of learning rate has a direct impact on network training time and training accuracy. A too small learning rate will lead to too slow convergence speed of the algorithm and play a negative role in the effective updating of weights; a too high learning rate will easily lead to too much vibration in the learning process and affect the efficiency of the algorithm. Therefore, it is necessary to define the self-adaptive learning rate to improve the adaptability of the learning algorithm to the training process.

Here, the definition of adaptive learning rate in the CSNN is as follows:

$$\eta = \begin{cases} (1+c)\eta^*, & v < v_{\min}, \\ \eta^*, & v_{\min} \leq v \leq v_{\max}, \\ \frac{\eta^*}{(1+c)}, & v > v_{\max}, \end{cases} \quad (35)$$

where  $c$  is the scaling factor,  $\eta^*$  is the benchmark learning rate, namely, the value of learning rate in benchmark frequency range, and  $v$  is the releasing frequency of neuron spiking sequence; the self-adaptive learning rate  $\eta$  can be adjusted according to different spiking releasing frequencies, and its benchmark frequency interval is  $[v_{\min}, v_{\max}]$ . When  $v \in [v_{\min}, v_{\max}]$ , scaling factor  $c = 1$ ; otherwise, the expression of  $c$  is

$$c = \begin{cases} \frac{v_{\min} - v}{v_{\max} - v_{\min}}, & v < v_{\min}, \\ \frac{v - v_{\max}}{v_{\max} - v_{\min}}, & v > v_{\max}. \end{cases} \quad (36)$$

**5.1.6. Similarity Measurement of Spiking Sequence.** In order to evaluate the final learning performance, the approximation between the actual spiking sequence and the target output spiking sequence after learning needs to be measured; it can be understood as measuring the distance between the

two groups of spiking sequences. By Cauchy-Schwarz inequality

$$F^2(s_i, s_j) = F(s_i, s_i)F(s_j, s_j), \forall s_i, s_j \in S(\Gamma), \quad (37)$$

the similarity measurement  $S$  of the two spiking sequences can be obtained as follows:

$$S = \frac{\langle f_{s_i}(t), f_{s_j}(t) \rangle}{\|f_{s_i}(t)\| \|f_{s_j}(t)\|} = \frac{F(s_i, s_j)}{\sqrt{F(s_i, s_i)} \sqrt{F(s_j, s_j)}}, \quad (38)$$

where  $\|f_s(t)\|$  is the Euclidean norm of the spiking sequence; the more similar the two spiking sequences are, the closer  $S$  is to 1.

**5.2. Multilayer SNN Based on STIP.** In the above discussion, the basic theory of single neuron has been prepared. For the needs of water color image classification, the multilayer feed forward SNN based on STIP rules needs to be introduced. Combining the contents described in Sections 5.1.4 and 5.1.5, its learning rules can be described as follows:

- (1) The learning rule of synaptic weight between neurons in the last hidden layer and output layer is

$$\begin{aligned} \Delta w_{ho} &= -\eta [F(s_o^a, s_h) - F(s_o^d, s_h)] \\ &= \eta \left[ \sum_{m=1}^{F_o^d} \sum_{n=1}^{F_h} \kappa(t_o^m, t_h^n) - \sum_{m=1}^{F_o^a} \sum_{n=1}^{F_h} \kappa(\tilde{t}_o^m, t_h^n) \right]. \end{aligned} \quad (39)$$

- (2) The learning rules of synaptic weights between input layer or hidden layer and hidden layer neurons are

$$\begin{aligned} \Delta w_{ih} &= -\eta \sum_{o=1}^{N_o} [F(s_o^a, s_i) - F(s_o^d, s_i)] w_{ho} \\ &= \eta \sum_{o=1}^{N_o} \left[ \sum_{m=1}^{F_o^d} \sum_{n=1}^{F_i} \kappa(t_o^m, t_i^n) - \sum_{m=1}^{F_o^a} \sum_{n=1}^{F_i} \kappa(\tilde{t}_o^m, t_i^n) \right] w_{ho}, \end{aligned} \quad (40)$$

where  $t_i^n$  and  $\tilde{t}_i^n$  represent the releasing time of the corresponding spiking in the input spiking sequence  $s_i$  and the spiking sequence  $s_h$  issued by the hidden layer neuron, respectively;  $t_o^m$  and  $\tilde{t}_o^m$  represent the releasing time of the corresponding spiking in the actual output spiking sequence  $s_o^a$  and the target spiking sequence  $s_o^d$  of the  $o$ -th neuron in the output layer, respectively;  $F_i$ ,  $F_h$ ,  $F_o^a$ , and  $F_o^d$  represent the number of spiking contained in the spiking sequences  $s_i$ ,  $s_h$ ,  $s_o^a$ , and  $s_o^d$ , respectively;  $N_o$  is the number of neurons in the output layer.

The pseudocode of the algorithm training process is as follows:

Note: for precise in expression, the spiking sequence encoded by the characteristic data of each sample in the training set is recorded as  $s_i$  and the spiking sequence encoded by the label data of each sample in the training set is recorded as  $s_o^d$ . In order to distinguish  $s_i$  and  $s_o^d$  of each

training sample, mark  $(s_i, s_o^d)_\xi$  has been introduced to represent the training data pair composed of the characteristic spiking sequence of the  $\xi$ -th training sample and the tag (target) spiking sequence (Algorithm 1).

## 6. Numerical Analysis

There are only 5000 water color picture data of a tilapia aquaculture pond for the study in this paper [5]. In order to meet the requirements of training in the subsequent CNN, the data need to be enhanced and increased to 20000 by employing triple-GAN. The activation function of triple-GAN adopts Xwish, and the optimization part is FMSGDM. t-SNE is used to reduce the dimension of 20000 data and to strengthen the features; its optimization part adopts GCOIO. After dimensionality reduction, the data are subject to feature extraction and further dimensionality reduction compression through several convolution layers and pooling layers. Xwish is still selected as the nonlinear activation function between convolution layer and pooling layer. Furthermore, the data are spiking encoded and sent to SNN, and finally the classified category is output through spiking decoding. The flowchart of this method is shown in Figure 1.

First, the original water color image data are processed. Each image in the small sample is outputted as red, green, and blue (RGB) color channels. Figure 2 shows the original sample water color image, and Figure 3 shows its corresponding RGB three-color channel output result.

It can be seen from Figure 3 that the image edge data are invalid, and as the color data are relatively uniform, the central pixel image data have been intercepted via Python program. Figure 4 shows the effect after the original screenshot, and Figure 5 is the RGB three-color channel output result of the screenshot.

Then, the processed data are inputted into triple-GAN for data enhancement. In the process of data enhancement network training, triple-GAN is set to use 3 different activation functions including Xwish, Sigmoid, and ReLU, respectively, for comparison, as shown in Figures 6 and 7.

It can be clearly seen from Figures 6 and 7 that after 10000 iterations of training, the accuracy of Sigmoid activation function is the lowest, which is 83.76%, and the network is difficult to converge. The network using ReLU function as the activation function has higher accuracy and faster convergence speed, and the accuracy is 94.46%. By introducing Xwish activation function, it improves the drawbacks of traditional activation functions, such as easy to saturate on both sides and cause gradient explosion and gradient disappearance (when Sigmoid is used) and easy to fall into dead zone, resulting in neurons unable to learn effective features (when Sigmoid is used). By combining the advantages of two kinds of common activation functions, Xwish activation function can enhance the training efficiency and accuracy of the algorithm, and the training accuracy of the network is the highest, reaching 98.12%, 14.36% higher than Sigmoid and 3.66% higher than ReLU. The network has the fastest convergence speed and the highest accuracy, which reflect the characteristics of high efficiency and accuracy of Xwish function.

In order to focus on the optimization performance of FMSGDM, its time efficiency and iteration speed have also been investigated in this section. In the performance analysis shown later in this section, a hyperparameter  $\tau$  and a performance evaluation function  $\rho_s(\tau)$  which can be recognized as the standard indicator of the possibility of solution defined in literature [53] have been chosen to show the performance results. For convenience of expression, the number of iterations is recorded as NIT. It can be seen from Figure 8 that FMSGDM is superior to GD and SGD in CPU time performance, and it can also be seen from Figure 9 that the iteration time performance of FMSGDM is also superior to GD and SGD.

In order to evaluate the image enhancement quality of triple-GAN proposed in this paper, two classical data enhancement indexes, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), are introduced. PSNR is the most common and widely used image objective evaluation index. However, it is based on the error between corresponding pixels, that is, the image quality evaluation based on error sensitivity. Although it is not completely consistent with the visual quality seen by human eyes, it is still used as the baseline for other indicators. The unit of PSNR is dB. The larger the value of PSNR is, the smaller the distortion is in the image.

As the visual characteristics of human eyes are not considered (human eyes are more sensitive to contrast differences with low spatial frequency and are more sensitive to brightness contrast differences than chromaticity, the perception result of human eyes to an area will be affected by its surrounding adjacent areas, etc.), the evaluation results are often inconsistent with people's subjective feelings. Therefore, SSIM is also introduced to evaluate the enhanced data objectively and comprehensively together with PSNR. SSIM evaluates the similarity of two images from brightness, contrast, and structure and evaluates the distortion by sensing the structure information, which is closer to the human eye. The value range of SSIM is [0,1]. The larger the SSIM value is, the smaller the distortion is in the image.

GAN, CGAN, CatGAN, and triple-GAN have been used to increase 5000 groups of original water color image samples to 20000 groups, and the average PSNR and SSIM of each method have been calculated. The results are shown in Table 1. As PSNR higher than 40 dB and SSIM value closer to 1 also indicate excellent image quality, it can be seen from Table 1 that the enhancement quality of triple-GAN image is very high.

After data enhancement, the enhanced data are then sent to t-SNE for dimension reduction and feature enhancement. The original 2-dimensional image matrix will be reduced to 1-dimensional vector. The optimization performance of GCOIO has been focused. It can be seen from Figure 10 that the CPU time performance of GCOIO is better than that of GD and SGD, and it can also be seen from Figure 11 that the iteration time performance of GCOIO is also superior to GD and SGD.

Finally, the one-dimensional data are input into the one-dimensional CSNN architecture. There are in total 8 layers of convolution and pooling, in which the activation

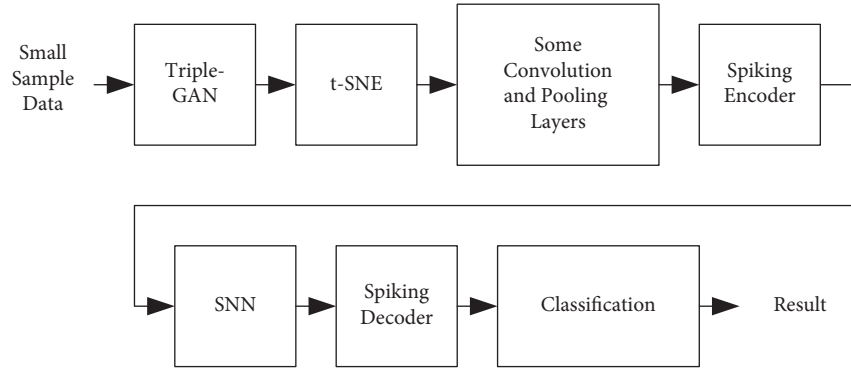


FIGURE 1: Flowchart of the method proposed in this paper.

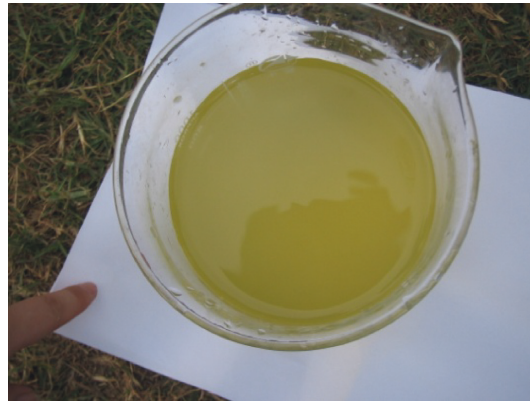


FIGURE 2: Original sample water color image.

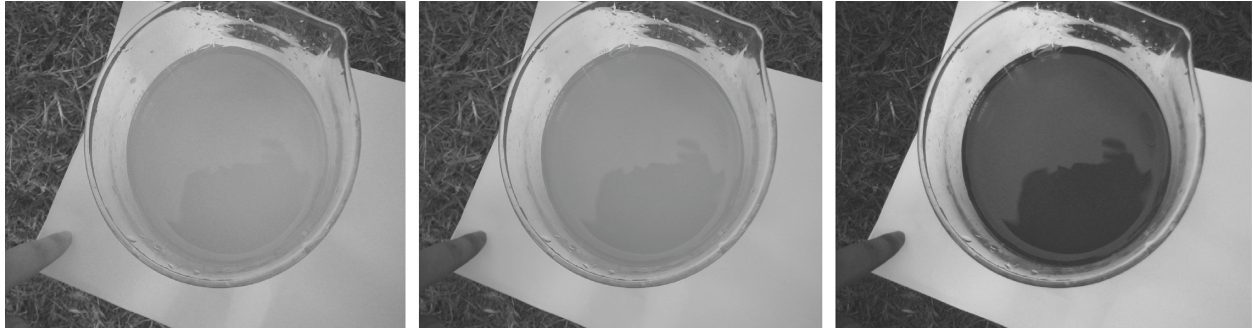


FIGURE 3: RGB output of the original sample water color image.

function is Xwish. The data are spiking encoded after passing through the convolution pooling layer and input to the 5-layer SNN based on STIP constructed. The output spiking of the network layer is decoded to obtain the category label.

Figure 12 shows the spiking sequence learning process of spiking neurons. In Figure 12, “ $\nabla$ ” represents the target output spiking sequence, “ $\Delta$ ” represents the output spiking sequence before spiking neuron learning, and the black dot represents the actual output spiking sequence of some learning cycles in the learning process. It can be seen from the learning process that the neuron learns the desired

output spiking sequence from the initial output spiking sequence after about 46 cycles. Figure 13 shows the change of learning accuracy curve in the learning process. It can be observed that, after 46 cycles, the actual output is the same as the target spiking sequence, that is, the similarity measure  $S=1$ . Figures 14 and 15 show the changes of 500 synaptic weights before and after learning, respectively. In the process of learning, the variation range of neuron synaptic weight is  $[0, 0.2]$ . Through the analysis of spiking sequence learning process, it can be seen that SNN can learn the spatiotemporal patterns of complex spiking sequences and has good learning ability.



FIGURE 4: Original sample water color image after screenshot.



FIGURE 5: RGB output of the original sample water color image after screenshot.

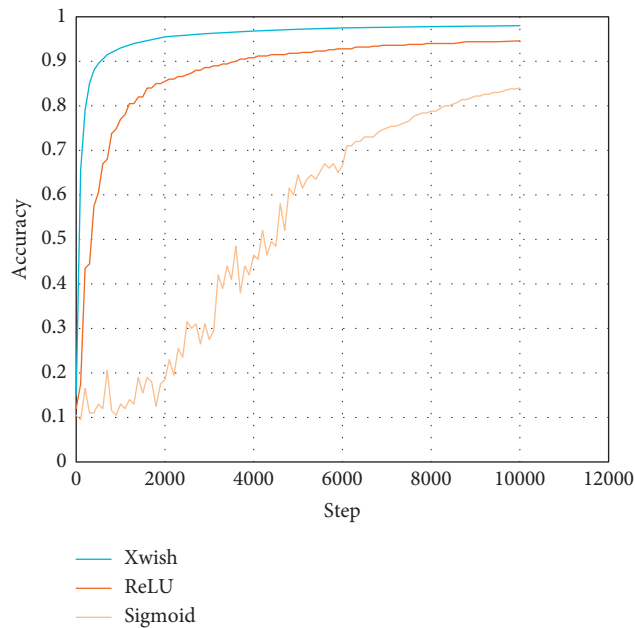


FIGURE 6: Accuracy curve for each activation function in CSNN.

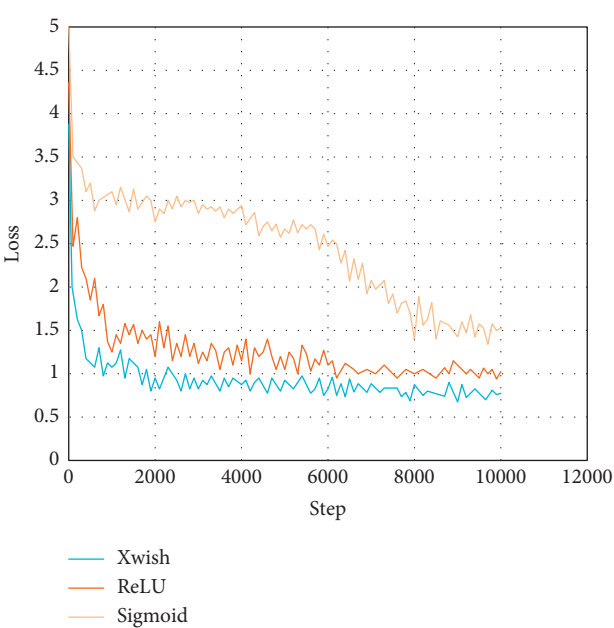


FIGURE 7: Loss curve for each activation function in CSNN.

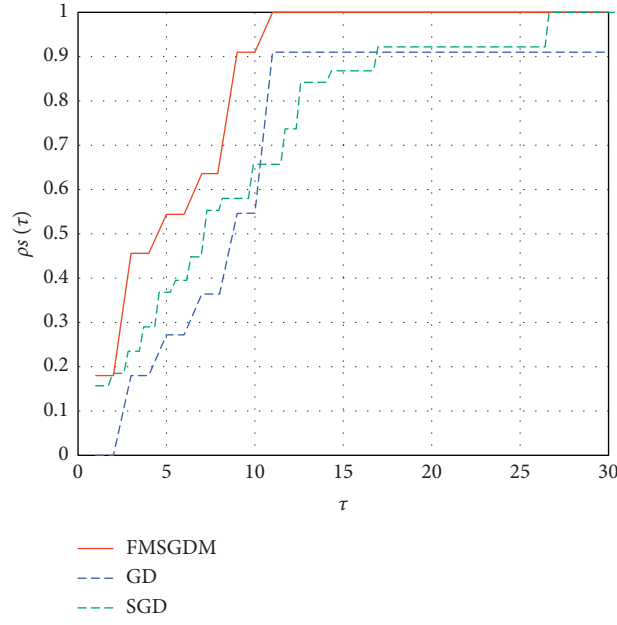


FIGURE 8: Performance of CPU time for FMSGDM, GD, and SGD.

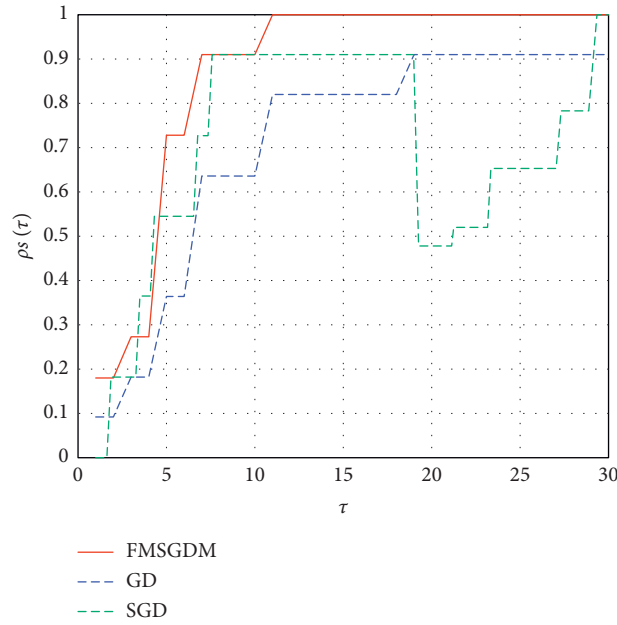


FIGURE 9: Performance of NIT time for FMSGDM, GD, and SGD.

**INPUT:** input the training set data  $(s_i, s_o^d)_i, i = 1, \dots, \tilde{\xi}$  and the reference learning rate  $\eta^*$  into SNN, where  $\tilde{\xi}$  is the number of training samples;  
**REPEAT**  
**FOR** Each training set data  $(s_i, s_o^d)_i, i = 1, \dots, \tilde{\xi}$   
 Calculate the actual output pulse sequence  $s_o^a$  of SNN from the input spiking sequence  $s_i$ ;  
 Calculate the error between the actual output spiking sequence  $s_o^a$  and the target spiking sequence  $s_o^d$  of CSNN:  
 $E(t) = 1/2 \sum_{o=1}^{N_o} [f_{s_o^a}(t) - f_{s_o^d}(t)]^2$ ;

ALGORITHM 1: Continued.

Calculate the learning rate  $\eta$  according to the benchmark learning rate  $\eta^*$ ;  
 Update the parameters using GD method according to the error;  
 Synaptic weights between neurons in the last hidden layer and output layer:  
 $\Delta w_{ho} = -\eta [F(s_o^a, s_h) - F(s_o^d, s_h)] = \eta [\sum_{m=1}^{F_o^d} \sum_{n=1}^{F_h} \kappa(t_o^m, t_h^n) - \sum_{m=1}^{F_o^a} \sum_{n=1}^{F_h} \kappa(\hat{t}_o^m, t_h^n)];$   
 Synaptic weight between neurons in the last hidden layer and output layer:  
 $\Delta w_{ih} = -\eta \sum_{o=1}^{N_o} [F(s_o^a, s_i) - F(s_o^d, s_i)] w_{ho} = \eta \sum_{o=1}^{N_o} [\sum_{m=1}^{F_o^d} \sum_{n=1}^{F_h} \kappa(t_o^m, t_i^n) - m = 1 \sum_{m=1}^{F_o^a} \sum_{n=1}^{F_h} \kappa(\hat{t}_o^m, t_i^n)] w_{ho};$   
 END  
**UNTIL** The number of iterations has been reached;  
**OUTPUT** Trained SNN.

ALGORITHM 1: The pseudocode of the algorithm training process.

TABLE 1: The average PSNR and SSIM value of different data enhancement methods.

Method	PSNR	SSIM
GAN	27.83	0.42
CGAN	35.96	0.73
CatGAN	39.12	0.79
Triple-GAN	42.13	0.86

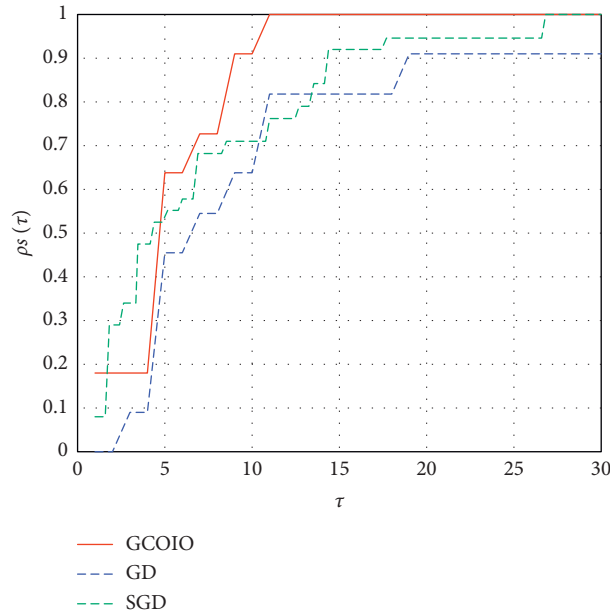


FIGURE 10: Performance of CPU time for GCOIO, GD, and SGD.

Then, the performance of the method proposed in this paper is compared with the existing popular water quality image classification methods. The comparison algorithms include logistic regression (LR) combined with multi-classification strategy OvO, random forest (RF), naive Bayes (NB), and support vector machine (SVM) combined with multiclassification strategy OvO; deep learning methods, including multilayer perception (MLP), DenseNet homologous variant models, DenseNet-121, DenseNet-169, and DenseNet-201, and ResNet homologous variant models, ResNet-34, ResNet-50, and ResNet-101.

Different from the evaluation indexes of the two classification problems, the commonly used evaluation indexes

of the two classification problems' algorithm include accuracy rate, precision rate, recall rate, and F1 score. The water color image classification problem in this paper is a five-class classification problem. Multiclassification problem standard evaluation indexes Macro-F1 and Micro-F1, which can comprehensively reflect the accuracy rate and recall rate of each specific classification, have been used in this paper. In Table 2, the evaluation index parameters have been shown, in which the higher the values of Macro-F1 and Micro-F1, the better the classification quality. It can be seen that the values of Macro-F1 and Micro-F1 of the method proposed in this paper are higher than those of other methods, which shows its high accuracy and stability.

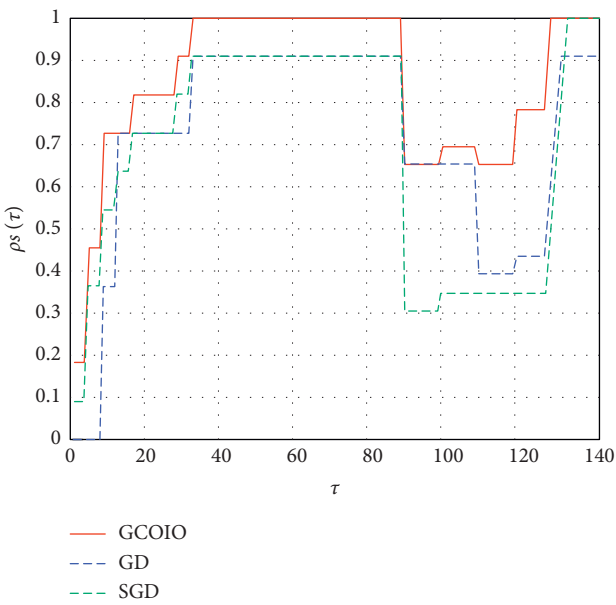


FIGURE 11: Performance of NIT time for GCOIO, GD, and SGD.

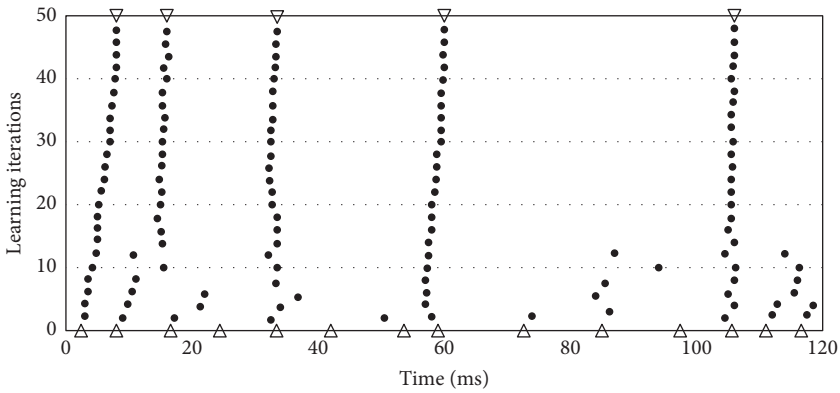


FIGURE 12: Learning process of spiking sequence.

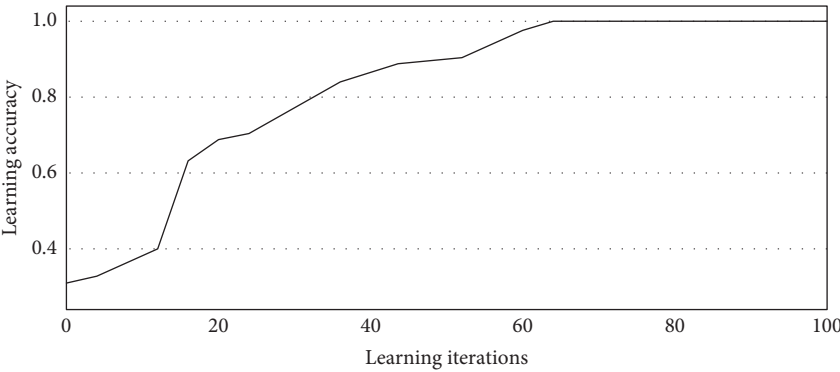


FIGURE 13: Changing of learning accuracy curve during the learning process.

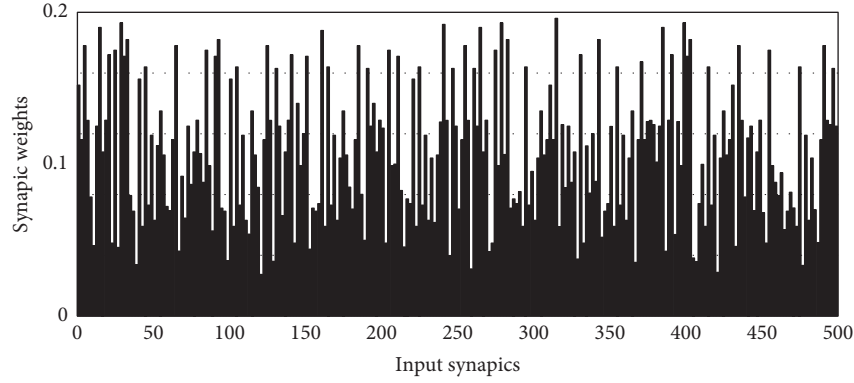


FIGURE 14: Synaptic weights before learning.

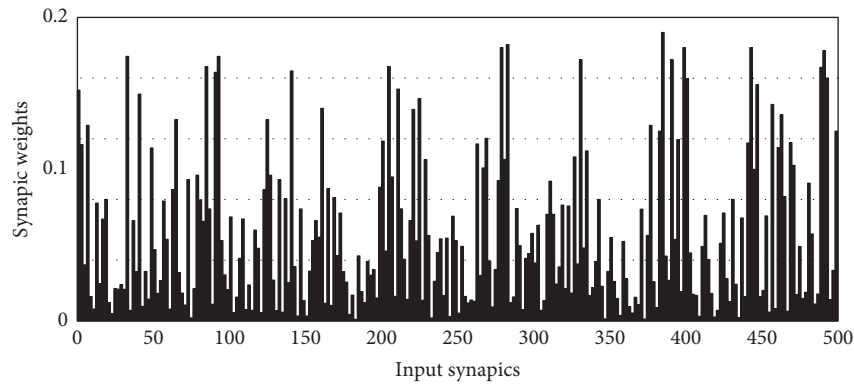


FIGURE 15: Synaptic weights after learning.

TABLE 2: Performance comparison of various methods in water quality classification.

Method	Macro-F1 (%)	Micro-F1 (%)
LR + OvO	72.8	73.1
RF	74.2	73.8
SVM + OvO	75.1	74.7
NB	73.3	73.6
MLP	78.7	79.0
DenseNet-121	92.3	93.2
DenseNet-169	92.7	92.4
DenseNet-201	92.6	93.1
ResNet-34	88.3	91.1
ResNet-50	90.6	90.3
ResNet-101	91.7	92.0
Method proposed in this paper	93.4	94.5

## 7. Conclusions

In this paper, a new algorithm framework for few color samples' quality classification for inland lakes or ponds has been proposed, in which triple-GAN and SNN have been introduced into the field of water quality problems for the first time. The proposed method has made an active theory exploration of integrating AI and water quality analysis problems. In order to effectively evaluate the classification efficiency of the method proposed in this paper, multi-classification problem standard evaluation indexes Macro-

F1 and Micro-F1 have been introduced to evaluate its performance, and its advantages can be easily seen via the final results. It can be concluded that the better performance acquired by employing the method proposed in this paper benefits from two aspects: on the one hand, from the spiking sequence learning process of spiking neurons, the change of learning accuracy curve in the learning process, and the change of synaptic weight before and after learning, it can be seen that the SNN in the full connection layer of the proposed method has the advantages of high neuron biological simulation, low energy consumption, and high efficiency; on the other hand, the number of layers of CSNN proposed in this paper is lower than that of the deep learning algorithm involved in the comparison, but the experimental effect is better than the latter. It can be concluded that the t-SNE algorithm based on GCOIO proposed in this paper has significant feature extraction ability.

## Nomenclature

$z$ :	The latent variable (Gaussian random noise)
$p(x, y)$ :	True data joint distribution
$p(x)$ :	True marginal distribution of random variable $x$
$p(y)$ :	True marginal distribution of random variable $y$
$p(y x)$ :	The probability of occurrence of $y$ on the premise of $x$ about true data $(x, y)$



$p(x y)$ :	The probability of occurrence of $x$ on the premise of $y$ about true data $(x, y)$	${}^C_{x_{k-K}}D^\alpha_x$ :	The $\alpha$ -th order Caputo derivative by taking $x_{k-K}$ as the initial iteration point
$p_z(z)$ :	The prior distribution over the latent variables	$N$ :	Subscript of final output iteration point in fixed memory step fractional order descent method
$p_g(x)$ :	The marginal distribution of $x$ defined by the generator	$K$ :	Subscript of existing iteration points in the fixed memory step fractional order gradient descent method
$p_g(x y)$ :	Assume that $x$ is transformed by the latent style variable $z$ given the label $y$ , namely, $x = G(y, z)$ , $z \sim p_z(z)$	$\zeta$ :	Fixed memory step size fractional order
$p_g(x, y)$ :	The joint distribution formed by $p(y)$ $p_g(x, y)$ , where a fake input-label pair can be sampled from $G$ by first drawing $y \sim p(y)$ and then drawing $x y \sim p_g(x y)$	$f(x)$ :	Network loss function
$p_c(y x)$ :	The conditional distribution defined by the classifier	$\theta^d$ :	Parameter of discriminator in triple generative adversarial network (triple-GAN)
$p_c(x, y)$ :	The joint distribution formed by $p(x)p_c(y x)$ , where $x \sim p(x)$ , a fake label $y$ is produced by $C$ according to the conditional distribution $p_c(y x)$	$\theta^c$ :	Parameter of classifier in triple-GAN
$C$ :	Classifier that approximately characterizes the conditional distribution $p_c(y x) \approx p(y x)$	$\theta^g$ :	Parameter of generator in triple-GAN
$G$ :	Class-conditional generator that (approximately) characterizes the conditional distribution in the other direction $p_g(x y) \approx p(x y)$	$(x_d, y_d)$ :	Data feature and label involved in discriminator
$D$ :	Discriminator that distinguishes whether a pair of data $(x, y)$ comes from the true distribution $p(x, y)$	$(x_c, y_c)$ :	Data feature and label involved in classifier
$E_{p(x)}$ :	The expectation under the condition of true data probability density $p(x)$	$(x_g, y_g)$ :	Data feature and label involved in generator
$E_{p_z(z)}$ :	The expectation under the condition of fake sample data probability density $p_z(x)$	$m_d$ :	The number of data involved in discriminator
$E_{p(x,y)}$ :	The expectation under the joint probability density $p(x, y)$ of true data $x$ and label data $y$	$m_c$ :	The number of data involved in classifier
$E_{p_z(z)p(y)}$ :	The expectation under the condition that the probability density of fake sample data and the probability density label data are independent	$m_g$ :	The number of data involved in generator
$E_{p_c(y x)}$ :	The expectation under the condition that the conditional probability density $p_c(y x)$ with true data $x$ as the premise and clustering classification label $y$ as the result	$\phi_i$ :	Sample points in high-dimensional space
$E_{p_c(y G(z))}$ :	The expectation under the condition that conditional probability density $p_c(y G(z))$ with generated fake sample data $G(z)$ as the premise and clustering classification label $y$ as the result	$\Phi_i$ :	Sample points in low-dimensional space corresponding to high-dimensional space after dimensionality reduction
$E_{p_c(x,y)}$ :	The expectation under the joint probability density $p_c(x, y)$ of true data $x$ and its label data $y$ in the classifier module	$p_{j i}$ :	Conditional probability of $\phi_j$ near $\phi_i$ in high-dimensional space
$E_{p_g(x,y)}$ :	The expectation under the joint probability density $p_g(x, y)$ of true data $x$ and its label data $y$ in the generator module	$q_{j i}$ :	Conditional probability of $\Phi_j$ near $\Phi_i$ in low-dimensional space
$\rho(x)$ :	Xwish activation function	$P_i$ :	Probability distribution of points in high-dimensional space
$\beta$ :	Hyperparameter in Xwish activation function	$Q_i$ :	Probability distribution of points in low-dimensional space
$\mu$ :	Step size in the fractional order descent method with fixed memory step size	$KL(\cdot  \cdot)$ :	KL divergence
$\alpha$ :	Order in the fractional order descent method with fixed memory step size	$JS(\cdot  \cdot)$ :	JS divergence
$\Gamma(\cdot)$ :	Gamma function	$O_j^t$ :	The position of the $j$ -th optical source point in $n$ -dimensional space at the $t$ -th iteration
$x_{k-K}$ :	Varying initial instant	$F_{i_k}^t$ :	Vertex coordinates at the $t$ -th iteration
		$\tilde{f}$ :	Focal length
		$s_{j,i_k}^t$ :	The function position of the $j$ -th optical source point when its mirror function is a concave function at the $t$ -th iteration
		$Z_1, Z_2$ :	Random numbers obeying uniform distribution in $[0, 1]$
		$n$ :	Number of optical source points
		$p_{j,i_k}^t$ :	The object distance of the $j$ -th optical source point at the $t$ -th iteration
		$r_{j,i_k}^t$ :	The surface radius of the $j$ -th optical source point at the $t$ -th iteration
		$m_{j,i_k}^t$ :	The surface center of the $j$ -th optical source point at the $t$ -th iteration
		$HI_{j,i_k}^t$ :	The image length of the $j$ -th optical source point at the $t$ -th iteration
		$I_{j,i_k}^t$ :	The image position of the $j$ -th optical source point at the $t$ -th iteration
		$r_{i_k}^t$ :	The curvature radius, whose center is on the principal axis passing through $F_{i_k}^t$

$HO_{j,i_k}^t$ :	The object length of the $j$ -th optical source point at the $t$ -th iteration
$q_{j,i_k}^t$ :	The image distance of the $j$ -th optical source point at the $t$ -th iteration
$RefI_{O_j}(F_{i_k}^t)$ :	The reflection image point of $F_{i_k}^t$ at the $t$ -th iteration
$\lambda_j^t$ :	Gaussian random number in $[0, 1]$
$s(t)$ :	A given spiking sequence
$H$ :	Specific smoothing function
$t^r$ :	Releasing time of the $t$ -th spiking of neuron
$S(\Gamma)$ :	All spiking sequences released within the time interval $\Gamma = [0, T]$
$M$ :	Total number of spiking
$\bar{N}$ :	Number of presynaptic input neurons
$s_i(t)$ :	Spiking sequence released by input neuron $i$
$s_a(t)$ :	Spiking sequence output by postsynaptic neurons
$f_{s_i}(t)$ :	Input spiking sequence transformed by convolution operation
$f_{s_a}(t)$ :	Output spiking sequence transformed by convolution operation
$w_i$ :	The link strength between presynaptic neurons and postsynaptic neurons, that is, synaptic weight
$s_d$ :	Target spiking sequence
$E(t)$ :	Error of spiking neuron at time $t$
$\hat{E}$ :	Total error of spiking neuron at time $\Gamma$
$\Delta w_i$ :	Changes in synaptic weights from presynaptic neurons $i$ to postsynaptic neurons
$\eta$ :	Learning rate
$\eta^*$ :	Benchmark learning rate
$\nabla \hat{E}_i$ :	Gradient value of spiking sequence error function $\hat{E}$ for synaptic weight $w_i$
$F(\cdot, \cdot)$ :	Inner product of two spiking sequences
$t_i^m, t_j^n$ :	Releasing times corresponding to spiking sequence $s_i$ and $s_j$
$N_b, N_j$ :	The numbers of spiking corresponding to spiking sequences $s_i$ and $s_j$
$\kappa(\cdot, \cdot)$ :	Kernel function
$t_i^f, t_a^h, t_d^g$ :	Releasing times corresponding to the input spiking sequence $s_i$ , the actual output spiking sequence $s_a$ , and the target spiking sequence $s_d$
$F_b, F_a, F_b$ :	The numbers of spiking corresponding to input spiking sequence $s_i$ , the actual output spiking sequence $s_a$ , and the target spiking sequence $s_d$
$c$ :	Scaling factor
$v$ :	The frequency of releasing neuron spiking sequence
$S$ :	Similarity measurement parameter
$\Delta w_{ho}$ :	Synaptic weights between neurons in the last hidden layer and output layer
$\Delta w_{ih}$ :	Synaptic weights between neurons in the input layer and hidden layer
$t_i^n, t_h^n$ :	Releasing times corresponding to the input spiking sequence $s_i$ and the spiking sequence $s_h$ released by hidden layer neurons

$s_o^a, s_o^d$ :	Actual output spiking sequence and the target spiking sequence $s_o^a$ and the of the $o$ -th neuron in the output layer
$t_o^m, \hat{t}_o^m$ :	Releasing times corresponding to the actual output spiking sequence and the target spiking sequence $s_o^d$ of the $o$ -th neuron in the output layer
$F_h, F_o^a, F_o^d$ :	The numbers of spiking corresponding to spiking sequence $s_h$ released by hidden layer neurons, the actual output spiking sequence $s_o^a$ , and the target spiking sequence $s_o^d$
$N_o$ :	Number of neurons in the output layer.

## Data Availability

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

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the Zhejiang Soft Science Research Project under Grant no. 2019C35092, the Science and Technology Commission of Shanghai Foundation under Grant no. 19DZ1205804, the Jiaxing Public Welfare Research Project under Grant nos. 2020AY10033, 2020AY30025, and 2021AY10079, and the General Scientific Research Fund of Zhejiang Provincial Education Department under Grant no. Y202147878.

## References

- [1] C. Qi, S. Huang, and X. Wang, "Monitoring water quality parameters of Taihu Lake based on remote sensing images and LSTM-RNN," *IEEE Access*, vol. 8, Article ID 188068, 2020.
- [2] M. Niroumand-Jadidi, F. Bovolo, and L. Bruzzone, "Novel spectra-derived features for empirical retrieval of water quality parameters: demonstrations for OLI, MSI, and OLCI sensors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10285–10300, 2019.
- [3] S. Wang, J. Li, Q. Shen, B. Zhang, F. Zhang, and Z. Lu, "MODIS-based radiometric color extraction and classification of inland water with the fore-ule scale: a case study of lake Taihu," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 2, pp. 907–918, 2015.
- [4] T. J. Malthus, R. Ohmsen, and H. J. v. d. Woerd, "An evaluation of citizen science smartphone Apps for inland water quality assessment," *Remote Sensing*, vol. 12, no. 10, p. 1578, 2020.
- [5] L. Zhang, L. Wang, L. Tan, and J. Su, *Python Practice on Data Analysis and Mining*, pp. 195–203, China Machine Press, Beijing, China, 2015.
- [6] T. O. Abdullah, S. S. Ali, N. A. Al-Ansari, and S. Knutsson, "Hydrogeochemical evaluation of groundwater and its

- suitability for domestic uses in Halabja Saidadiq Basin, Iraq,” *Water*, vol. 11, no. 4, p. 690, 2019.
- [7] L. Ma, J. Abuduwaili, Y. Li, S. Abdyzhaparuulu, and S. Mu, “Hydrochemical characteristics and water quality assessment for the upper reaches of syr darya river in aral sea basin,” *Water*, Central Asia, vol. 11, no. 9, 2019.
  - [8] H. Nayeem, A. Syed, and M. Z. A. Khan, “Towards development of a simple technique based on wavelength specific absorption for quality measurement of flowing water,” *IEEE Sensors Journal*, vol. 20, no. 24, Article ID 14780, 2020.
  - [9] P. R. Smith, D. A. Green, and R. Naimimohasses, “Some investigations on neural processing of scattered light in water quality assessment,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 261–266, 1994.
  - [10] A. S. M. Nor, M. Faramarzi, M. A. M. Yunus, and S. Ibrahim, “Nitrate and sulfate estimations in water sources using a planar electromagnetic sensor array and artificial neural network method,” *IEEE Sensors Journal*, vol. 15, no. 1, pp. 497–504, 2015.
  - [11] S. Koponen, K. Kallio, J. Pulliainen, J. Vepsäläinen, T. Pyhälähti, and M. Hallikainen, “Water quality classification of lakes using 250-m MODIS data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 4, pp. 287–291, 2004.
  - [12] J. Karami, A. Alimohammadi, and S. Modabberi, “Analysis of the spatio-temporal patterns of water pollution and source contribution using the MODIS sensor products and multivariate statistical techniques,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 4, pp. 1243–1255, 2012.
  - [13] K. Shi, Y. Li, Y. Zhang, L. Li, H. Lv, and K. Song, “Classification of inland waters based on Bio-Optical properties,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 2, pp. 543–561, 2014.
  - [14] D. Sun, Y. Li, Q. Wang et al., “Hyperspectral remote sensing of the pigment C-Phycocyanin in turbid inland waters, based on optical classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 3871–3884, 2013.
  - [15] S. Zhao, J. Chen, Q. Jin et al., “Improved principal component-fuzzy comprehensive assessment coupling model for urban river water quality: a case study in Chongqing, China,” *Water*, vol. 12, no. 5, p. 1375, 2020.
  - [16] X. Zhang, Y. Zhang, Q. Zhang et al., “Evaluation and analysis of water quality of marine aquaculture area,” *International Journal of Environmental Research and Public Health*, vol. 17, no. 4, p. 1446, 2020.
  - [17] Q. Guan, L. Feng, and X. Kuang, “Optical classifications of Poyang Lake water and long-term dynamics based on MERIS observations,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 12, pp. 4591–4603, 2018.
  - [18] Y. Liu, Y. Liang, K. Ouyang, S. Liu, D. Rosenblum, and Y. Zheng, “Predicting Urban Water Quality with Ubiquitous Data - a Data-Driven Approach,” *IEEE Transactions on Big Data*, vol. 8, 2020.
  - [19] L. Sheng, J. Zhou, X. Li, Y. Pan, and L. Liu, “Water quality prediction method based on preferred classification,” *IET Cyber-Physical Systems: Theory & Applications*, vol. 5, no. 2, pp. 176–180, 2020.
  - [20] J. Liu, C. Yu, Z. Hu et al., “Accurate prediction scheme of water quality in smart mariculture with deep Bi-S-SRU learning network,” *IEEE Access*, vol. 8, Article ID 24784, 2020.
  - [21] D. Venancio Cruz, M. Rivelino Gomes de Oliveira, M. Cunha Filho, and D. Venancio da Cruz, “Monitoring pH with quality control based on Geostatistics methodology,” *IEEE Latin America Transactions*, vol. 14, no. 12, pp. 4787–4791, 2016.
  - [22] R. McEliece, S. Hinz, J. M. Guarini, and J. Coston-Guarini, “Evaluation of nearshore and offshore water quality assessment using UAV multispectral imagery,” *Remote Sensing*, vol. 12, no. 14, 2020.
  - [23] Y. Chen, H. Zhang, L. Liu et al., “Research on image inpainting algorithm of improved total variation minimization method,” *Journal of Ambient Intelligence and Humanized Computing*, 2021.
  - [24] Y. Chen, L. Liu, V. Phonevilay et al., “Image super-resolution reconstruction based on feature map attention mechanism,” *Applied Intelligence*, vol. 51, no. 7, pp. 4367–4380, 2021.
  - [25] Y. Chen, L. Liu, J. Tao et al., “The improved image inpainting algorithm via encoder and similarity constraint,” *The Visual Computer*, vol. 37, no. 7, pp. 1691–1705, 2021.
  - [26] R. Xia, Y. Chen, and B. Ren, “Improved anti-occlusion object tracking algorithm using Unscented Rauch-Tung-Striebel smoother and kernel correlation Filter,” *Journal of King Saud University - Computer and Information Sciences*, 2022.
  - [27] H. M. Campos Ribeiro, A. d. C. Almeida, B. R. P. D. Rocha, and A. V. Krusche, “Water quality monitoring in large reservoirs using remote sensing and neural networks,” *IEEE Latin America Transactions*, vol. 6, no. 5, pp. 419–423, 2008.
  - [28] C. Dona, J. M. Sanchez, V. Caselles, J. A. Dominguez, and A. Camacho, “Empirical relationships for monitoring water quality of lakes and reservoirs through multispectral images,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 5, pp. 1632–1641, 2014.
  - [29] M. Nazeer and J. E. Nichol, “Combining landsat TM/ETM+ and HJ-1 A/B CCD sensors for monitoring coastal water quality in Hong Kong,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1898–1902, 2015.
  - [30] E. Batur and D. Maktav, “Assessment of surface water quality by using satellite images fusion based on PCA method in the Lake Gala, Turkey,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 5, pp. 2983–2989, 2019.
  - [31] F. Isikdogan, A. C. Bovik, and P. Passalacqua, “Surface water mapping by deep learning,” *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 11, pp. 4909–4918, 2017.
  - [32] F. Yuan, Y. Huang, X. Chen, and E. Cheng, “A biological sensor system using computer vision for water quality monitoring,” *IEEE Access*, vol. 6, Article ID 61535, 2018.
  - [33] F. Pu, C. Ding, Z. Chao, Y. Yu, and X. Xu, “Water-quality classification of inland lakes using Landsat8 images by Convolutional Neural Networks,” *Remote Sensing*, vol. 11, no. 14, p. 1674, 2019.
  - [34] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, pp. 2672–2680, Montreal, Canada, October, 2014.
  - [35] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” 2014, <https://arxiv.org/abs/1411.1784>.
  - [36] J. T. Springenberg, “Unsupervised and semi-supervised learning with categorical generative adversarial networks,” 2015, <https://arxiv.org/abs/1511.06390>.
  - [37] C. Li, K. Xu, J. Liu, J. Zhu, and B. Zhang, “Triple Generative Adversarial Networks,” 2019, <https://arxiv.org/abs/1703.02291>.
  - [38] Y. Liu, T. Wang, and X. Xu, “New adaptive activation function for deep learning neural networks,” *Journal of Jilin University (Earth Science Edition)*, vol. 57, no. 4, pp. 857–859, 2019.

- [39] Y. Wei, Y. Kang, W. Yin, and Y. Wang, "Design of Generalized Fractional Order Gradient Descent Method," 2018, <https://arxiv.org/abs/1901.05294>.
- [40] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning*, vol. 9, pp. 2579–2625, 2008.
- [41] G. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Proceedings of the 15th International Conference on Neural Information Processing System (NIPS)*, pp. 857–864, Vancouver, Canada, June, 2002.
- [42] A. Hussein-zadeh Kashan, "A new metaheuristic for optimization: optics inspired optimization (OIO)," *Computers & Operations Research*, vol. 55, pp. 99–125, 2015.
- [43] X. Lin and X. Wang, *Principle and Application of Spiking Neural Network*, Science Press, Beijing, China, 2018.
- [44] K. I. Mcanally and J. F. Stein, "Auditory temporal coding in dyslexia," *Proceedings of the Royal Society of London - Series B: Biological Sciences*, vol. 263, no. 1373, pp. 961–965, 1996.
- [45] F. d. r. Theunissen and J. P. Miller, "Temporal encoding in nervous systems: a rigorous definition," *Journal of Computational Neuroscience*, vol. 2, no. 2, pp. 149–162, 1995.
- [46] S. Ghosh-dastidar and H. Adeli, "Spiking neural networks," *International Journal of Neural Systems*, vol. 19, no. 04, pp. 295–308, 2009.
- [47] L. F. Abbott, B. DePasquale, and R.-M. Memmesheimer, "Building functional networks of spiking model neurons," *Nature Neuroscience*, vol. 19, no. 3, pp. 350–355, 2016.
- [48] X. Lin, X. Wang, and X. Dang, "A new supervised learning algorithm," *Acta Electronica Sinica*, vol. 44, no. 12, pp. 2877–2886, 2016.
- [49] X. Wang, "Supervised Learning in Spiking Neural Networks with Inner Product of Spike Trains," M. S. thesis, College of Computer Science & Engineering, Northwest Normal University, Lanzhou, China, 2015.
- [50] H. C. Tuckwell and F. Y. M. Wan, "Time to first spike in stochastic Hodgkin-Huxley systems," *Physica A: Statistical Mechanics and its Applications*, vol. 351, no. 2-4, pp. 427–438, 2005.
- [51] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Brain-like system for audiovisual person authentication based on time-to-first spike coding," in *Proceedings of the Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*, pp. 384–412, IGI Global Press, Hershey, PA, USA, January, 2011.
- [52] X. Lin, N. Zhang, W. Cui, and L. Feng, "Image segmentation method of spiking neural network based on TTFS coding," *Computer Engineering*, vol. 41, no. 8, pp. 227–232, 2015.
- [53] D. Li and D. Zhu, "An affine scaling interior trust-region method combining with nonmonotone line search filter technique for linear inequality constrained minimization," *International Journal of Computer Mathematics*, vol. 95, no. 8, pp. 1494–1526, 2018.