



---

*Research article*

## Frequency hopping signal detection based on optimized generalized S transform and ResNet

Chun Li, Ying Chen\* and Zhijin Zhao

School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

\* **Correspondence:** Email: lc1025@hdu.edu.cn.

**Abstract:** The performance of traditional frequency hopping signal detection methods based on time frequency analysis is limited by the tradeoff of time-frequency resolution and spectrum leakage. Machine learning-based frequency hopping signal detection techniques have a high level of complexity. Therefore, this paper proposes a residual network and the optimized generalized S transform to detect frequency hopping signals. First, based on the time-frequency aggregation measure, the generalized S transform parameters  $\lambda$  and  $p$  are optimized using a multi-population genetic algorithm. Second, the optimized generalized S transform is used to determine a signal's time-frequency spectrum, which is then normalized to make this robust to noise power uncertainty. Finally, a residual network structure is designed which receives the time-frequency spectrum. To detect frequency hopping signals, the network automatically learns the time-frequency properties of signals and noise. Simulated findings indicate that the multi-population genetic algorithm not only increases optimization efficiency when compared to a regular genetic algorithm, but also has faster convergence and more stable optimization results. Compared with a hybrid convolutional network/recurrent neural network algorithm, the proposed technique is better at detection and has less computational and storage complexity.

**Keywords:** frequency hopping signal detection; generalized S transform; genetic algorithm; convolutional neural network

---

### 1. Introduction

Frequency hopping (FH) communication technology has the capacity to prevent both interference and interception by randomly and rapidly changing its carrier frequency in a wide frequency band. In

recent years, research on FH signal detection has become a hot topic in communication reconnaissance. Many methods for FH signal detection have been proposed at home and abroad.

Traditional FH signal detection methods include autocorrelation detection [1], signal decomposition [2], channelized receiver detection [3], power spectrum detection, time-frequency analysis, and other methods. The methods proposed in [1–3] all require known parameters such as signal jump speed, which is limited in practical applications. The work in [4,5] uses the difference between the power spectrum changes of fixed frequency and frequency hopping signals to achieve detection, which does not require prior knowledge. However, detailed features of the time-frequency spectrum of the frequency-hopping signal for the above algorithms have not been fully utilized. The time-frequency features of FH signals can be readily obtained via time frequency analysis, which can improve the detection performance. Currently, commonly used time frequency analysis techniques comprise Wigner-Ville distribution (WVD), Wavelet transform (WT), Hilbert-Huang transform (HHT), and short-time Fourier transform (STFT). In [6,7], the time-frequency spectrum of the signal wavelet transform was used, but the selection of the wavelet basis is difficult and greatly affects the detection performance. In [8], the time-frequency diagram of the Hilbert-Huang transform was used to overcome the problem of wavelet base selection, although the computational complexity of the algorithm is very high. In [9–11], the time-frequency spectrum of the short-time Fourier transform was used. However, spectrum leakage and time-frequency resolution are trade-offs that may be present in the aforementioned methods. In [12], the time-frequency resolution of the FH signal was balanced using WVD. This improved the time-frequency local aggregation of the FH signal but introduced a cross interference term, resulting in complex calculations and poor performance.

The time-frequency resolution and time-frequency focus of the S transform (ST) have better advantages than other transforms. This is generally used to process non-stationary signals and has been widely used [13]. However, the fixed Gaussian time window function reduces the flexibility of the S-transform in signal analysis. The generalized S transform (GST) introduces the parameters  $\lambda$  and  $p$  to jointly control the window function. This has better applicability and time-frequency resolution. In [14], the optimal combination weighting method was used to optimize five indicators to obtain the optimal values of  $\lambda$  and  $p$ , but the complexity is high. In [15], the parameters  $\lambda$  and  $p$  were selected according to actual signal characteristics, but this method is subjective and limited. In [16], a standard genetic algorithm (SGA) was used to obtain the optimal values of  $\lambda$  and  $p$ . However, the optimization results were unstable. Taking the time-frequency aggregation as an objective function, a multi-population genetic algorithm (MPGA) is used to obtain the optimal  $\lambda$  and  $p$  values in order to alleviate this problem, and then the GST spectrum is used to detect FH signals.

The creation of detection statistics required by conventional FH signal detection techniques has a significant negative influence on detection performance. Methods for FH signal detection have been developed in [17–19]. The directed gradient histogram properties of the FH signal are shown in the time-frequency diagram; the work in [17,18] used the AdaBoost algorithm and a Support Vector Machine (SVM), respectively, for detection. In [19], a jump in the frequency domain and the continuation of the FH signal in the residence time are used to identify the FH signal, which improved the detection performance. However, the techniques still suffer from spectrum leakage and poor time-frequency resolution. The authors of [20] used the K-means clustering algorithm to correct the time-frequency diagram, and used a convolutional neural network (CNN) to automatically learn time-frequency characteristics. This enhances the time-frequency resolution but has a high complexity. A hybrid convolutional network/recurrent neural network (HCRNN) scheme was proposed in [21].

Initially, by employing numerous parallel short-time Fourier transformations with varied window widths, the signal's time-frequency maps are produced. Then, a number of CNNs are used to extract the attributes and frequency diagrams. Finally, the RNN combines these time-frequency features and finds the FH signal. The HCRNN approach has an improved detection performance and resolves the issues of spectrum leakage and low time-frequency resolution compared to the conventional energy detection methods, the cyclostationary method and the spectrogram method. However, the approach has a very high complexity. To simplify this and further improve the detection performance, in this paper, the time-frequency diagram of the FH signal's characteristics is extracted using the residual network (ResNet) structure which is obtained by the optimized generalized S transform to detect the FH signal.

In summary, the main contributions of this paper include the following aspects:

- A frequency hopping signal detection algorithm is proposed based on the optimized generalized S-transform in order to address the problem that the performance of the traditional S-transform is limited by the fixed Gaussian time window function. Using the time-frequency clustering measure as the criterion, MPGA is used to optimize the parameters  $\lambda$  and  $p$  of the GST.
- A frequency hopping signal detection scheme is proposed based on deep learning which utilizes the optimized GST to obtain the time spectrum of the frequency hopping signal and normalizes the power of the time spectrum to improve robustness to noise power uncertainty. A CNN is designed that has the time-frequency spectrum as input, and this network is utilized to automatically learn the time-frequency characteristics of signals and Gaussian noise to achieve frequency hopping signal detection.
- ResNet is adopted to address the vanishing gradient and exploding gradient problems of CNN models that rely on depth to improve performance. Also, the storage and time complexity of the proposed ResNet network are analyzed and compared with the comparison algorithm to demonstrate that the proposed scheme has better detection performance and lower complexity.
- The frequency hopping signal detection algorithm based on ResNet is evaluated and compared with the HCRNN scheme. First, the detection performance under different time-frequency analysis methods is compared. Then, the ROC AUC measure is used to evaluate the performance of the network and obtain the results of the proposed algorithm and HCRNN under different signal-to-noise ratios and ROC AUC. Simulation results show that the proposed detection method has better performance.

## 2. Generalized S transform and parameter optimization

### 2.1. Generalized S transform

A linear time-frequency analysis technique called the S transform combines the continuous wavelet transform with the short-time Fourier transform. For a given observation signal  $x(t)$ , the generalized S transform is defined as follows [13]:

$$GST(f, t) = \int_{-\infty}^{+\infty} x(\tau) w(\tau - t) e^{-j2\pi f\tau} d\tau \quad (1)$$

where  $\tau$  represents the time-shifting factor,  $j$  is an imaginary unit,  $f$  is the frequency, and  $w(t)$  is the window function, defined as:

$$w(t) = \frac{1}{\sigma(f)\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2(f)}} \quad (2)$$

Here,  $\sigma(f)$  is a scaling factor function,  $\sigma(f) = \frac{1}{\lambda|f|^p}$ , and  $\lambda > 0$ ,  $p > 0$  are the parameters to be determined. When  $\lambda = 1$  and  $p = 1$ , the scaling factor function is  $\sigma(f) = \frac{1}{|f|}$  and the generalized S transform degenerates into the S transform.

Substitute the window function to get the GST of  $x(t)$  as follows [13]:

$$GST(f, t) = \int_{-\infty}^{+\infty} x(\tau) \frac{\lambda|f|^p}{\sqrt{2\pi}} e^{-\frac{(\tau-t)^2 \lambda^2 |f|^{2p}}{2}} e^{-j2\pi f \tau} d\tau \quad (3)$$

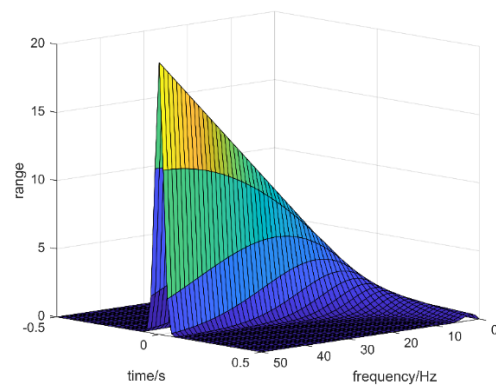
The GST is completely reversible, and its inverse transformation is shown in Eq (4):

$$x(t) = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} GST(\tau, f) d\tau \right] e^{j2\pi f t} df \quad (4)$$

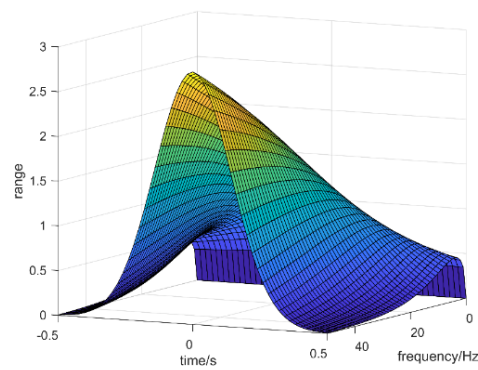
The parameters  $\lambda$  and  $p$  have a large impact on the performance of the time-frequency analysis. The time-frequency three-dimensional diagrams of the window function with different values of  $\lambda$  and  $p$  can be seen in Figure 1.

From Figure 1 it can be seen that when  $\lambda = 1$ ,  $p = 1$ , the Gaussian window  $w(t)$  widens with increasing frequency. When  $\lambda = 1$ ,  $p < 1$ , the influence of the frequency on the window scale is weakened, that is, the higher the frequency, the wider the window width. When  $\lambda < 1$ ,  $p = 1$ , the widening rate of the Gaussian window with the increase of frequency is much slower than for  $\lambda = 1$ ,  $p = 1$ . From this it can be observed that the Gaussian window of the generalized S transform can not only adjust the time window width according to the frequency, but also alter the time window's width's rate of change, and the determination of parameters  $\lambda$  and  $p$  is very important for the effectiveness of signal analysis. Thus, MPGA is used to optimize these parameters. The specific steps are as follows: Take the logarithm of the time-frequency aggregation degree proposed in this paper as the optimization function; use the multi-population genetic algorithm to evolve and search using this function; and finally obtain the optimal value, the value at the highest frequency aggregation degree, and output the optimal parameters  $\lambda$  and  $p$  required in this paper through decoding.

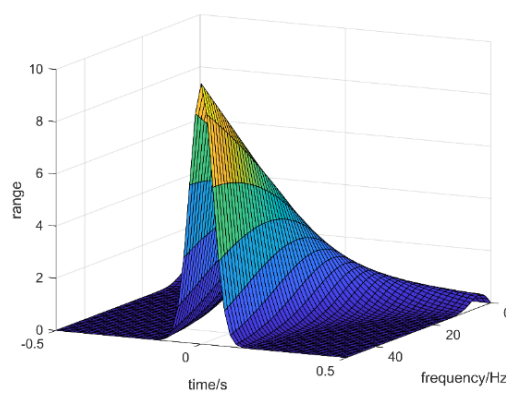




(a)  $\lambda = 1, p = 0.5$



(b)  $\lambda = 1, p = 0.5$



(c)  $\lambda = 0.5, p = 1$

**Figure 1.** Three-dimensional time-frequency diagram of the window function with different parameters.

## 2.2. Determination of parameters $\lambda$ and $p$

### 2.2.1. Evaluation function

Time-frequency aggregation is employed to measure the performance of the time-frequency analysis and this is used as the optimization index in this paper. Let the time-frequency spectrum of observation signal  $x(t)$  be  $P_x(t, f) = |GST_x(t, f)|^2$ , the time-frequency aggregation [15] is defined as:

$$y(x) = \left( \sum_{n=1}^N \sum_{k=1}^K |P_x(n, k)|^{\frac{1}{r}} \right)^r \quad (5)$$

where  $P_x(n, k)$  is the discrete form of the time-frequency spectrum  $P_x(t, f)$ , with frequency dimension  $k = 1, 2, \dots, K$ , and time dimension  $n = 1, 2, \dots, N$ ;  $r$  is a constant. When  $r > 1$ , the smaller the  $y$  value, the better the time-frequency aggregation; when  $0 < r \leq 1$ ,  $y$  is greatly affected by concentrated components in the time-frequency distribution, and it is insensitive to components with poor aggregation.

### 2.2.2. Multi-population genetic algorithm

A genetic algorithm has a good global search ability which can quickly search for all solutions in the solution space without falling into the trap of local optima. By utilizing its inherent parallelism, it can facilitate distributed computing and accelerate the solution speed. However, the local search ability of genetic algorithms is poor, resulting in a more time-consuming and inefficient search in the later stages of evolution. In practical applications, genetic algorithms are prone to premature convergence problems. The selection method used to preserve the best individuals while maintaining population diversity has always been a difficult issue.

The SGA was inspired by the biological world's natural selection and evolution mechanisms. It is a closely related, random search technique for the adaptive global optimization of probabilities. The parameters are encoded as chromosomes. The genetic operations used include selection, crossover and mutation. After several evolutionary iterations, chromosomes that meet the optimization goal are finally obtained. To address the premature convergence problem of SGA, when the aggregation of the time-frequency matrix is obtained in this paper, MPGA is used as shown in Figure 2 to optimize the parameters of GST. MPGA introduces SGA to multiple populations for simultaneous optimization search, and every population is connected via a migration operator to actualize the coevolution of many populations. Various populations are offered various command parameters for accomplishing distinct searches. The best individual in each evolutionary generation of each population is saved in the essence population. The optimal solution is the comprehensive result of the coevolution of multiple populations, which is used to decide algorithm convergence. The parameter optimization algorithm is designed as follows.

Parameters  $\lambda$  and  $p$  are used as chromosomes of the population and are binary encoded.

The chromosomes of the population are randomly initialized and divided into  $M$  independent subpopulations, each subpopulation has  $G$  individuals.

#### 1) Fitness function

The fitness function is utilized in order to assess the group members' level of competence and the logarithm of the time-frequency aggregation is taken, as shown in Eq (6); scaling down can enhance

the algorithm's performance. The larger the value of the fitness function, the better the individual is.

$$R(x) = \ln[y(x)] \quad (6)$$

## 2) Evolution operations

The definition of the selection operation is the probability-based selection of exceptional individuals from the old population to create a new population. An individual's likelihood of selection is influenced by their fitness value.

In SGA, the genetic algorithm's capacity for global search is defined by crossover, which is the key operator for creating new individuals. The ability of the genetic algorithm's local search to find new individuals is determined by the mutation operator, which is an auxiliary operator. If the crossover probability  $P_c$  and the mutation probability  $P_m$  are different, the optimization results will be different. Because the genetic process of each population in the MPGA algorithm is parallel and independent, different  $P_c$  and  $P_m$  are selected to ensure that the evolution of each population is different. Therefore,  $P_c$  and  $P_m$  can be calculated by Eq (7):

$$\begin{cases} P_c = P_{c0} + d_c \times r_{rand}(M, 1) \\ P_m = P_{m0} + d_m \times r_{rand}(M, 1) \end{cases} \quad (7)$$

where  $P_{c0}$  and  $P_{m0}$  are the initial crossover probability and mutation probability, and the random number generator  $r_{rand}$  produces numbers with a uniform distribution between  $[0, 1]$ .

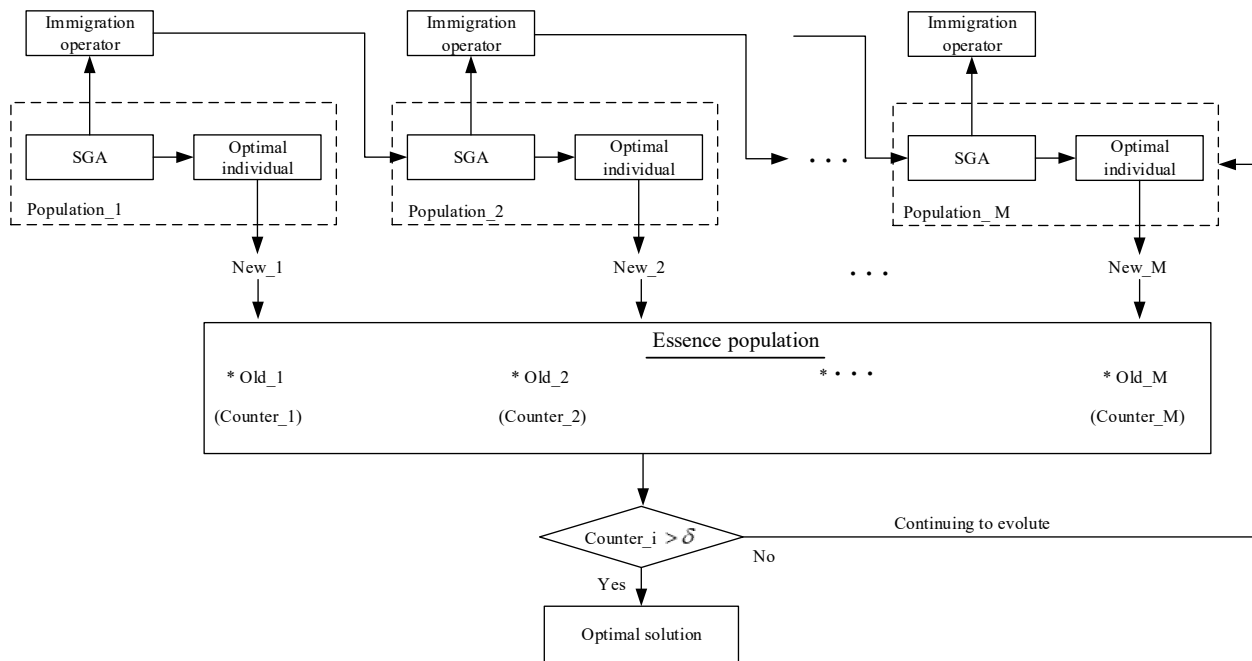
## 3) Migration operation

Different from SGA, various groups of MPGA are connected by the migration operator. Following certain evolutions, the best member of the current subpopulation replaces the worst member of the target subpopulation. For example, in population 1, the best member replaces the worst member in population 2, and in population N, the best member replaces the worst member in population 1.

## 4) Essence population and optimal solution

Unlike other populations, the best individuals of each subpopulation in each evolutionary generation make up the essence population, and it has  $M$  individuals. The essence population no longer carries out genetic operations such as selection, crossover, and mutation. It is only responsible for recording and saving the optimal individuals of each generation, so that the algorithm's evolution phase can fully retain the best possible result.

The minimum holding generation of the best individual in the essence population is denoted as  $\delta$ , and is used to decide the termination of evolution. The  $M$  counters are used to record the generation of individuals in the essence population. After each evolution of subpopulations, it is judged whether the new optimal value from each subpopulation is the same as the old optimal value recorded in the essence population. If it is different, the old optimal value will be updated. If it is the same, the new optimal value is abandoned and the counter of the old optimal value will be increased by one. When one of the counters is greater than  $\delta$ , the iteration is stopped and the optimal parameters  $\lambda$  and  $p$  are obtained. Otherwise, evolution will continue until the iteration termination condition is met. The genetic algorithm's knowledge gain during the evolution phase is fully utilized by this termination condition.



**Figure 2.** Schematic diagram of the MPGA algorithm structure.

### 3. Detection of signals based on GST and deep learning

The detection of FH signals can be viewed as a binary classification issue. In order to detect signals, deep learning is employed to extract the time-frequency spectrum properties of the generalized S transform of the FH signal.

#### 3.1. FH signal and generalized S transform spectrum

The FH signal  $s_1(t)$  is defined as

$$s_1(t) = \vartheta(t) \times A_a \times \sum_{i=1}^I \text{rect}_{T_h}(t - iT_h) e^{j(2\pi f_i t + \varphi_i)} \quad (8)$$

where  $I$  represents all hops throughout the observation time, the FH cycle is  $T_h$ , the convoluted baseband envelope is  $\vartheta(t)$ , the FH signal's amplitude is  $A_a$ , the carrier frequency and phase are  $f_i$  and  $\varphi_i$  of hop  $i$ , respectively.  $\text{rect}_{T_h}(t) = \begin{cases} 1, & t \in [0, T_h] \\ 0, & \text{others} \end{cases}$ .

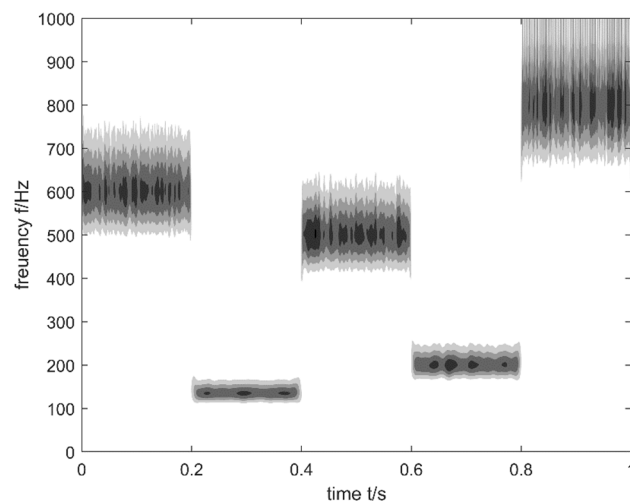
Let  $H_0$  represent the hypothesis that there is simply noise, assumption of FH signal in noise  $H_1$ . The observed signal intercepted by the receiver can be expressed as:

$$x(t) = \begin{cases} n(t) & H_0 \\ s_1(t) + n(t) & H_1 \end{cases} \quad (9)$$

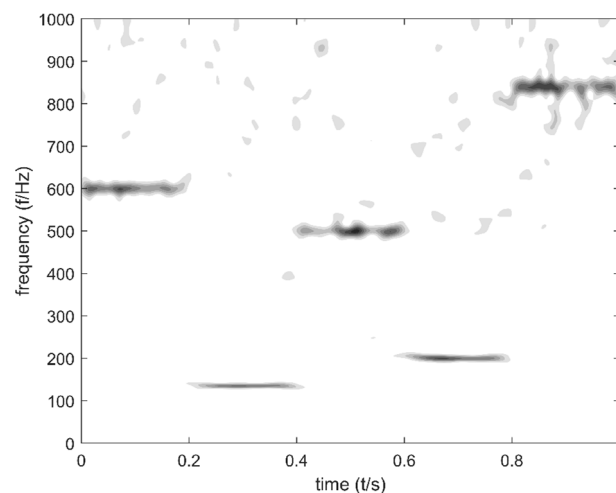
where  $n(t)$  is white Gaussian noise. The generalized S transform of the observation signal represents the binary hypothesis test model as follows:

$$GST_x(f, t) = \begin{cases} GST_v(f, t) & H_0 \\ GST_s(f, t) + GST_v(f, t) & H_1 \end{cases} \quad (10)$$

Figure 3(a),(b) show the time-frequency spectra of the standard S transform and the generalized S transform of an FH signal, respectively. Figure 3 demonstrates that the signal and noise display different time-frequency distribution characteristics. It is focused on the high-frequency region of the noise, and the FH information is scattered over a wide frequency and temporal range. The standard S transform is sensitive to noise, and the time-frequency aggregation of GST is better than the standard S transform.



(a) Standard S transform

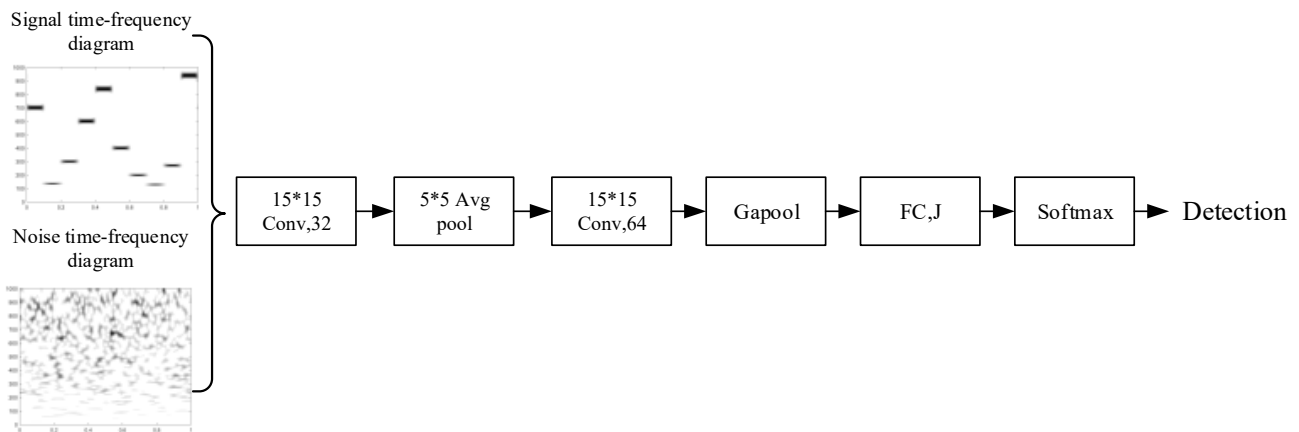


(b) Generalized S transform

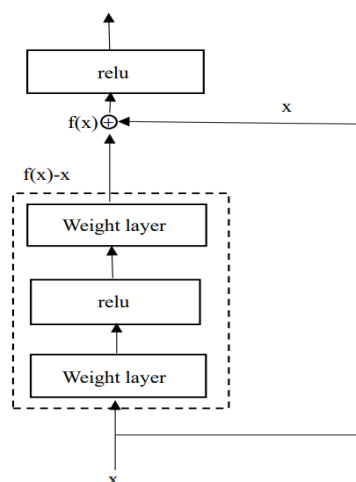
**Figure 3.** Time frequency diagram of the standard S transform and generalized S transform.

### 3.2. Network model

A CNN is a neural network specifically designed to handle data with a grid-like structure. Convolution networks are those neural networks that use convolution operations to replace general matrix multiplication operations in at least one layer of the network. The CNN network model and parameters designed in this article are shown in Figure 4, which mainly includes the convolutional layer, pooling layer, fully connected layer, etc. Here,  $\{15 * 15 \text{ Conv}, 32\}$  is used, which means that the convolutional kernel size is  $15 \times 15$ , the convolutional layer has 32 convolutional kernels. Using  $\{5 * 5 \text{ Avgpool}\}$ , the default step size is 1, and the pooling size is  $5 \times 5$  average pooling layers. “Gapool” is global average pooling that is introduced to avoid overfitting. FC represents the fully connected layer, and parameter J represents the number of output neurons. In this article, this is set to two. The designed model adopts two convolutional layers, passing through a fully connected layer with two neurons, and finally obtaining the detection result through the Softmax function.



**Figure 4.** CNN network structure.



**Figure 5.** Basic residual block structure.

Usually, increasing the depth of the network model can improve the performance of CNN, but

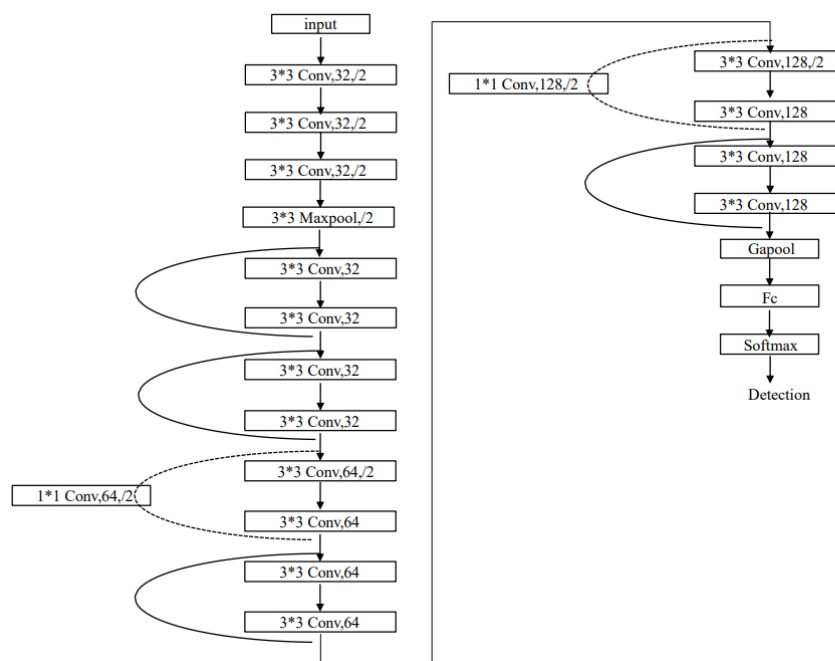
this may lead to the vanishing gradient or exploding gradient problems. This article uses ResNet to address this. ResNet is stacked by residual basic unit blocks; the basic unit block structure is shown in Figure 5. Assume that the input is  $x$  and the ideal mapping to learn is  $f(x)$  which serves as the activation function's input. The portion of the dotted circle within the residual block structure must match the identity mapping's residual mapping,  $f(x) - x$ .

Figure 6 shows the ResNet model and parameter details for this paper. The network consists of three convolution layers, a maximum pooling layer, a global average pooling layer, six residual blocks, and a fully connected layer. The parameters of the convolution layer are the size, type, number, and step of the convolution kernels; for example,  $\{3 * 3 \text{ Conv}, 32, /2\}$  denotes that the convolution kernel size is  $3 \times 3$ , the convolution kernel count is 32, and the step size is 2. The default value of the step size is 1. The entire convolution layer's operational process is:

$$X^{(L)} = W^{(L)} \otimes X^{(L-1)} + b^{(L)} \quad (11)$$

where the layer serial number is  $L$ ,  $X^{(L)}$  is the feature map that this layer outputs,  $X^{(L-1)}$  is the preceding layer's feature map output,  $W^{(L)}$  is this layer's convolution kernel weight, and  $b^{(L)}$  is the offset value of this layer. The activation function can increase the nonlinearity of the network. The neural network can indefinitely approximate any nonlinear function since the input layer introduces nonlinear variables to neurons. Each convolution layer has a ReLU layer coupled to it as the activation function:

$$\text{relu}(X) = \max(0, X) \quad (12)$$



**Figure 6.** The ResNet network structure.

The parameters of the pooling layer include the pool size, pool type, and step size, respectively. For example,  $\{3 * 3, \text{Maxpool}, /2\}$  indicates that the maximum pooling layer with a step size of 2 is

adopted. As before, “Gapool” refers to the avoidance of overfitting by pooling global averages. The fully connected layer is denoted by “Fc”. An additional  $1 \times 1$  convolution layer (the dotted line) is introduced to increase the input dimension, so as to address the problem that the front and back dimensions of these residual blocks are inconsistent.

The time-frequency matrix generated by the GST transform serves as the network’s input. Taking the noise signal’s power uncertainty into account, power normalization is applied to the time-frequency matrix and the dimension of the matrix is  $101 \times 200$ . The final detection results are obtained through the full connected layer with two neurons and the Softmax function.

### 3.3. Network training and detection rules

Suppose there are  $n$  pairs of training data  $\{(p_{\text{train}}^{(1)}, q_{\text{train}}^{(1)}), (p_{\text{train}}^{(2)}, q_{\text{train}}^{(2)}), \dots, (p_{\text{train}}^{(n)}, q_{\text{train}}^{(n)})\}$ ;  $p_{\text{train}}^{(k)}$  represents the  $k$ -th signal sample’s time-frequency matrix,  $q_{\text{train}}^{(k)}$  is the real label of  $p_{\text{train}}^{(k)}$ .  $p_{\text{train}}^{(k)}$  passes multi-layer pooling and convolution; the relationship of the final mapping in the forward propagation is as follows:

$$\hat{q}_{\text{train}}^{(k)} = g_{W,b}(p_{\text{train}}^{(k)}) \quad (13)$$

where  $b$  and  $W$  are the parameters of the network to be trained, the biases and weights, respectively, and  $\hat{q}_{\text{train}}^{(k)}$  is the output mapping of  $p_{\text{train}}^{(k)}$  through the network.

The cross-entropy loss function is used. In Eq (14), and to reduce the discrepancy between the ultimate anticipated worth and its actual worth, the backpropagation technique is used to modify  $b$  and  $W$  in the network layer by layer:

$$\text{Loss} = -\frac{1}{K} \sum_{k=1}^K \sum_{i=1}^2 q_{\text{train},i}^{(k)} \log(\hat{q}_{\text{train},i}^{(k)}) \quad (14)$$

where  $K$  is the size of the small batch. The Softmax layer is selected as the output layer, and a probability is displayed for the multi-classification result:

$$f(x_i) = \frac{e^{x_i}}{\sum_{i=1}^J e^{x_i}} \quad (15)$$

where  $J$  denotes the number of classes. In this paper,  $J$  is set to two.

Typically, for signal detection problems, the probability of detection  $P_d$  and the probability of false alarm  $P_f$  are employed to assess the effectiveness of signal detection.

$$\begin{aligned} P_d &= P(H_1|H_1) \\ P_f &= P(H_1|H_0) \end{aligned} \quad (16)$$

Considering that  $f(x_i)$  is the outcome of the signal sample  $x_i$ ’s passage through the Softmax layer, the following judgment rule is used:



$$\begin{cases} f(x_i) \geq 1 - \mu & H_1 \\ f(x_i) < 1 - \mu & H_0 \end{cases} \quad (17)$$

where  $\mu$  is the judgment threshold, which can be adjusted depending on the likelihood of a false alarm.

### 3.4. Complexity analysis

#### 3.4.1. Analysis of network calculation complexity

The time complexity of each convolution layer is  $S_{\text{conv}} \sim O(N_T^L \times N_F^L \times N_K^L \times N^L)$ , where  $N_T^L \times N_F^L$ ,  $N^L$ , and  $N_K^L$  represent the size of the input characteristic graph of the layer  $L$  convolution layer, the number of convolution kernels, and the size of the convolution kernel, respectively. The time complexity of the ReLU layer is  $S_{\text{relu}} \sim O(N_T^L \times N_F^L)$ . The complexity of the pooling layer is  $S_{\text{pooling}} \sim O(N_T^L \times N_F^L \times N_P/N_D)$ , where  $N_P$  is the length of the pooled filter, and  $N_D$  is the downsampling multiple. The maximum number of convolution cores in the ResNet network structure used in this paper is limited; the network's overall computational complexity is  $O(K \times N)$  when the input signal's matrix dimension is  $K \times N$ .

#### 3.4.2. Analysis of network storage complexity

Storage complexity is a measure of storage space occupation, consisting of two components: the network's overall settings and each layer's output characteristic graph. The total number of parameters is the sum of the weight parameter of the network layer, that is, the volume of the model. The parameter quantity of the network model is  $S_{\text{Mod}} \sim O(\sum_{L=1}^D N_K^L \times N_K^L \times C_{L-1} \times C_L)$ , where the total number of model layers is  $D$ ;  $C_{L-1}$  and  $C_L$  are the number of input and output channels of the current network layer, respectively. The input and output feature maps need only be temporarily stored during the calculation of the convolution layer, and the input storage units are released after the calculation is completed, so as to recycle in turn. Therefore, the feature map used in the reasoning process has double the amount of storage space as the biggest feature map in the network. The total number of parameters of the designed ResNet network are 711, 488, and the size of the largest feature map is  $\frac{K}{2} \times \frac{N}{2} \times 32$ .

Therefore, the storage complexity of the network is  $711, 488 + K \times \frac{N}{2} \times 32$ .

## 4. Performance analysis

### 4.1. Data generation

The zero IF signal is obtained after the received signal passes through a down-converter,

therefore, the following signal parameters are chosen. The frequency of the FH signal is set to [700, 135, 300, 600, 840, 400, 199, 128, 270, 940] Hz, and the FH period is 0.1 s. The length of signal is 2000 and the sampling frequency is 2000 Hz. The dimension of the time-frequency diagram after the generalized S transform is  $101 \times 200$ . The training set and test set are generated by MATLAB. The training data and test data contain 500 and 200 samples, respectively, under each SNR of -30 dB to 0 dB with a space of 2 dB. Under the same conditions, Gaussian white noise samples are generated with the same dimension and quantity as the signal data set, which have training and test subsets. The training set contains 8000 samples of both noise and signal. The number of noisy samples and signal samples in the training set is 3200.

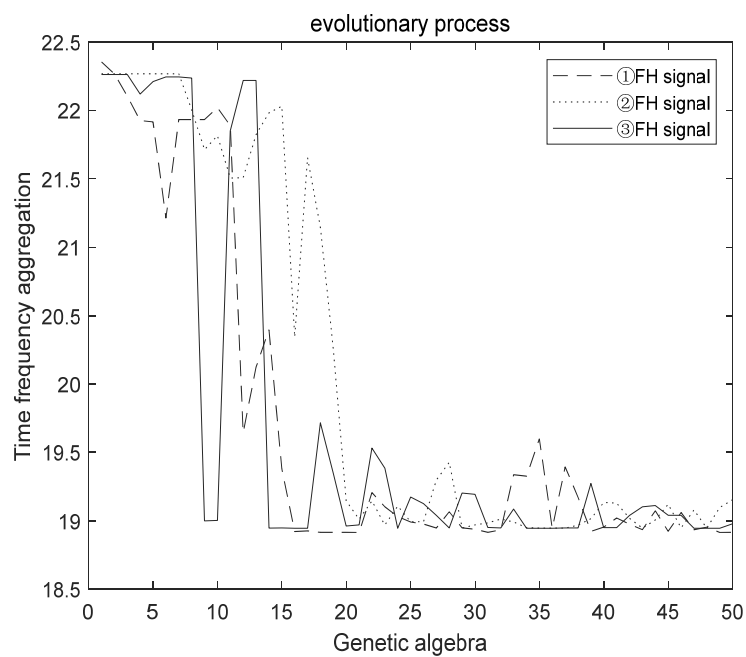
#### 4.2. Simulation results

Each training iteration's mini-batch size is 128 and 20 rounds make up the network's training cycle. The Adam optimizer is used. The learning rate begins at 0.01 while the network is being trained and drops by one-tenth every three cycles. Finally,  $\delta = M$ .

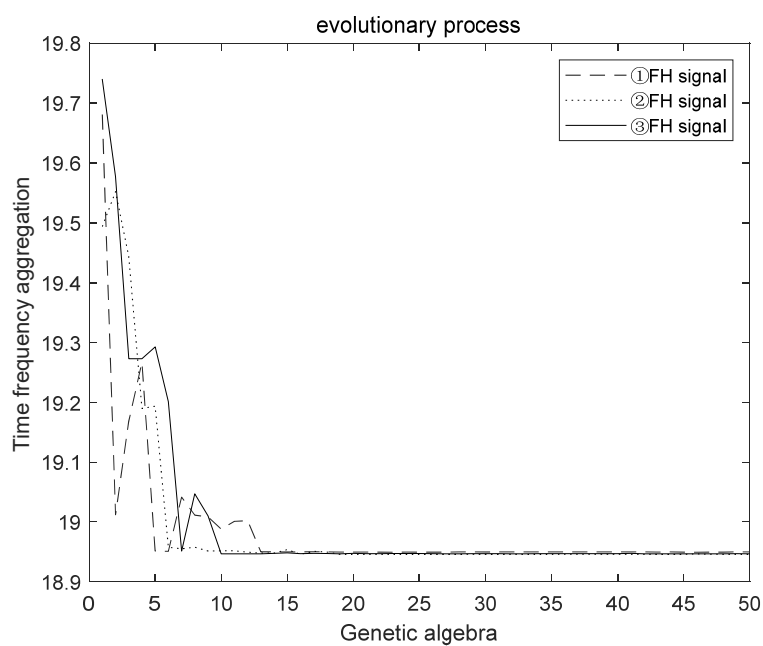
##### 1) Performance comparison of MPGA and SGA optimization

Three FH signals with a frequency range of 100–1000 Hz and cycles of 0.1 s, 0.1 s and 0.05 s are randomly generated, respectively, denoted as FH signals ①, ② and ③, respectively. The parameters of MPGA are set as follows:  $M$  and  $G$  are 10 and 100, respectively; the length of the chromosomes is 20, the fixed crossover probability  $P_c$  is within the interval [0.7, 0.9], the mutation probability  $P_m$  is within the interval [0.001, 0.05], and the parameter  $r$  of the time-frequency aggregation degree is taken as 2. The crossover probability of the SGA algorithm is 0.7 and the mutation probability is 0.05. The other parameters are the same as those of the MPGA algorithm.

The relation curves between the time-frequency aggregation degree and the iteration number obtained using the SGA and MPGA algorithms are shown in Figure 7. The optimal values of the time-frequency aggregation degree obtained by SGA and MPGA and the corresponding parameters  $\lambda$  and  $p$  are shown in Tables 1 and 2, respectively. It can be seen from Figure 7(a) and Table 1 that the optimization results obtained using the SGA algorithm for different frequency hopping signals are different and unstable, and the algorithm may not converge to an optimal value. Figure 7(b) and Table 2 indicate that the optimal solution obtained using MPGA for different frequency hopping signals is essentially the same, and MPGA converges to the optimal value after about 15 iterations. This is due to the MPGA algorithm's employment of many populations to concurrently explore the solution space collaboratively and considers both local and global search. Because of this, the sensitivity of the results to the parameters of the genetic algorithm is greatly reduced. Therefore, the parameters  $\lambda$  and  $p$  are selected as 1.7 and 0.6, respectively.



(a) Evolution of the SGA algorithm



(b) Evolution of the MPGA algorithm

**Figure 7.** Evolution of the different genetic algorithms.

**Table 1.** Optimal value of  $y(x)$  using SGA and corresponding parameters  $\lambda$  and  $p$ .

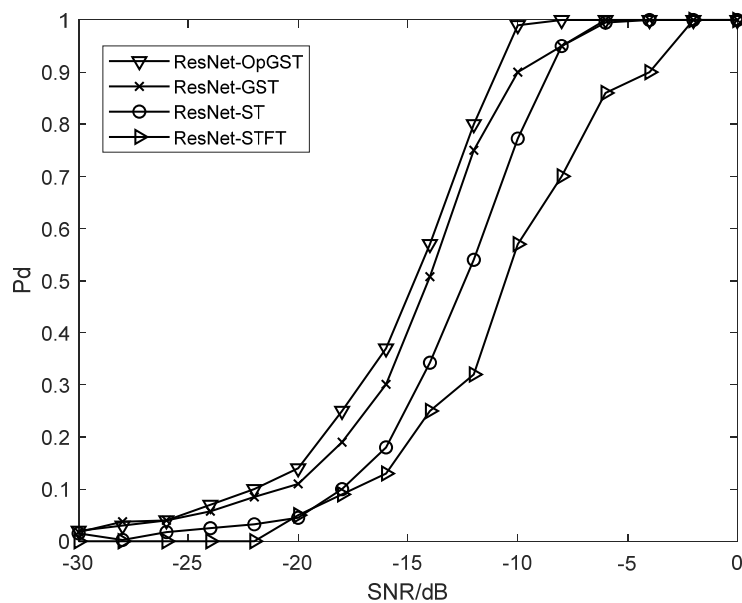
FH signal	$\lambda$	$p$	Optimal value of $y(x)$
① FH signal	1.6299	0.4273	18.9468
② FH signal	1.7692	0.5458	19.1537
③ FH signal	1.5213	0.5128	18.9579

**Table 2.** Optimal value of  $y(x)$  using MPGA and corresponding parameters  $\lambda$  and  $p$ .

FH signal	$\lambda$	$p$	Optimal value of $y(x)$
① FH signal	1.6983	0.6043	18.9497
② FH signal	1.7006	0.6034	18.9498
③ FH signal	1.7011	0.5991	18.9468

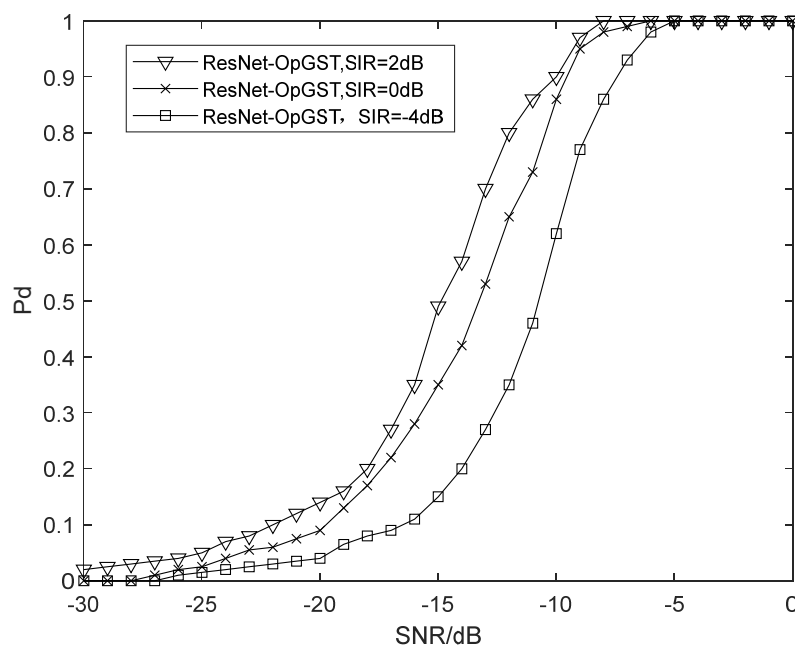
## 2) Influence of GST on detection performance

The time-frequency diagram samples for training and testing are obtained under the STFT with a 128-bit window and a sliding step of 96, the standard S transform, the generalized S transform with parameters  $\lambda$  and  $p$  of 1.3 and 0.1, and the generalized S transform with optimized parameters (OpGST), respectively. The probability of false alarm is 0.01 and the FH signal parameters are the same as those in Section 4.1. The designed ResNet network is combined with STFT, standard ST, GST, and OpGST; these are abbreviated as ResNet-STFT, ResNet-ST, ResNet-GST, and ResNet-OpGST, respectively. The effectiveness of the four algorithms at detecting FH signals is shown in Figure 8. The ResNet-OpGST algorithm's detection performance is the best, followed by ResNet-GST, and ResNet-STFT is the worst. Therefore, the following simulations all adopt the generalized S transform of parameter optimization.

**Figure 8.** FH signal detection performance for different transformations.

### 3) Detection performance under interference

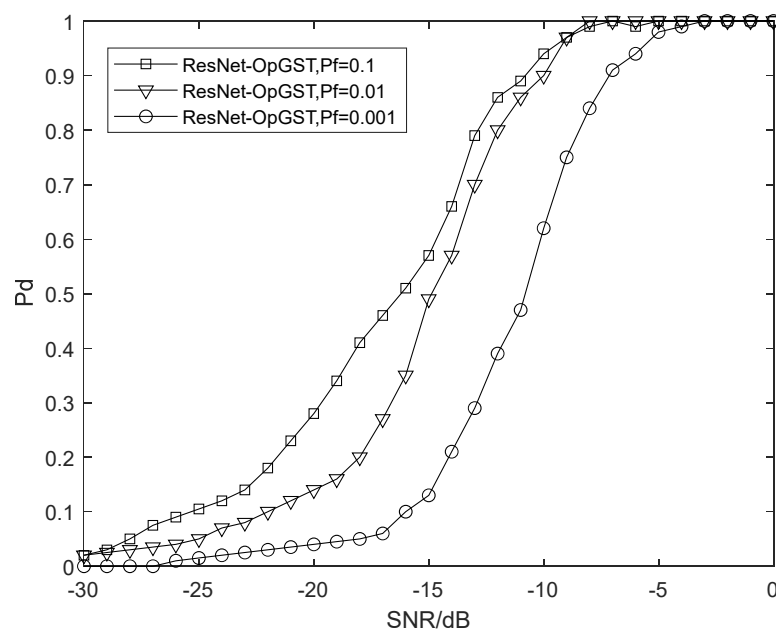
The frequencies of the fixed frequency interference signals are 710 kHz, 300 kHz, and 225 kHz. The SIR is set to -4 dB, 0 dB, and 2 dB, respectively, with the same data sample and frequency hopping signal parameters as (2). The detection performance of the ResNet-OpGST method proposed in this article is shown in Figure 9. As shown in the figure, the higher the SIR, the better the detection performance. When the SNR is -10 dB and the SIR is 2 dB, the detection probability of the ResNet-OpGST algorithm reaches 90%; the detection probability of ResNet-OpGST when the SIR is 0 dB is 86%; the detection probability of ResNet-OpGST with a SIR of -4 dB is 62%.



**Figure 9.** FH signal detection performance for different SIR values.

### 4) Detection performance for different false alarm probabilities

A SIR of 2 dB is selected and the false alarm probabilities are set to 0.001, 0.01, and 0.1; the data samples and frequency hopping signal parameters are the same as (2). The detection performance of ResNet-OpGST is shown in Figure 10. As shown in this figure, the higher the false alarm probability, the higher the detection probability. When the SNR is -10 dB and the false alarm probability is 0.1, the detection probability of ResNet-OpGST reaches 94%; the detection probability of ResNet-OpGST is 90% when the false alarm probability is 0.01; ResNet-OpGST with a false alarm probability of 0.001 has a detection probability of 62%.

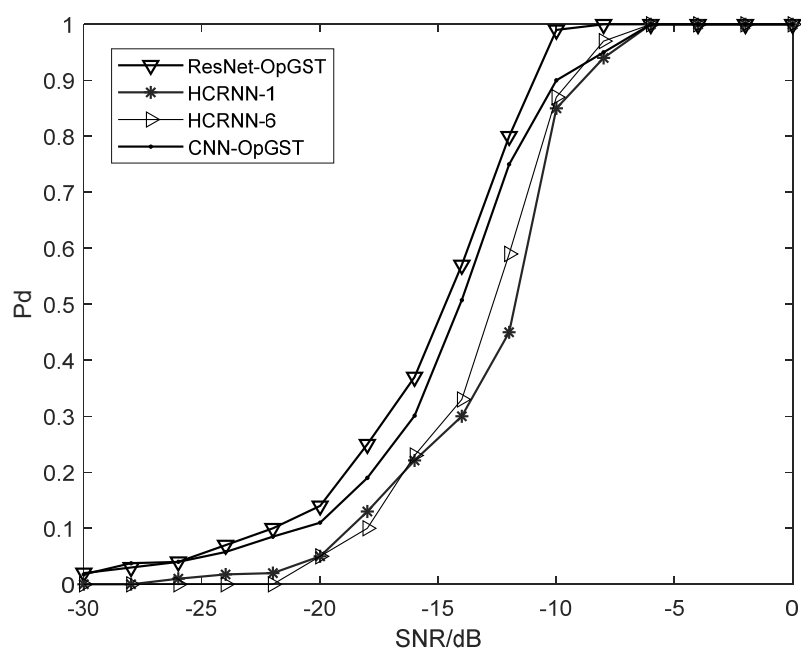


**Figure 10.** FH signal detection performance for different false alarm probabilities.

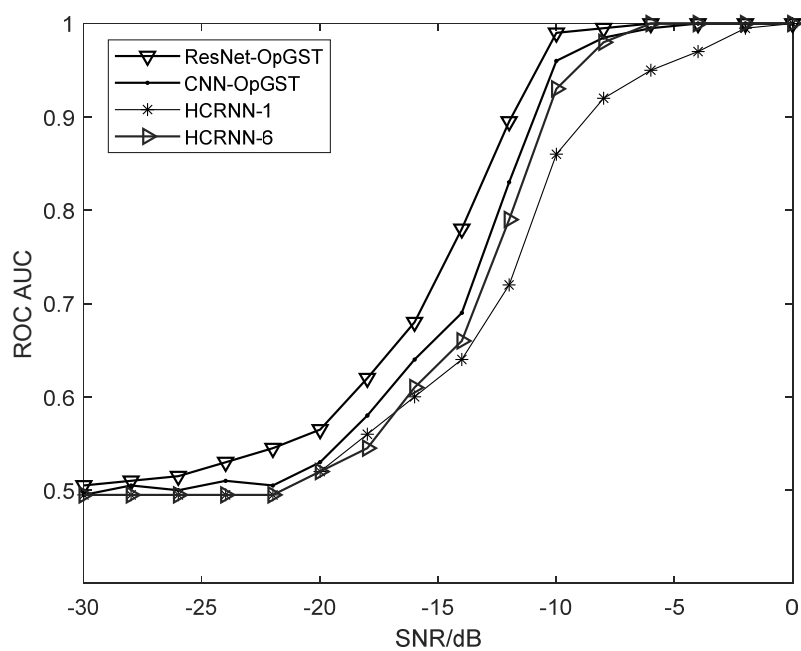
#### 5) Algorithm performance comparison

The detection algorithm used in this article is abbreviated here as CNN-OpGST. This section compares the ResNet-OpGST detection algorithm and CNN-OpGST proposed in this paper with the algorithm in [21]. The technique used six different types of mixing window lengths, including 1024, 512, 256, 128, 64, and 32, and spectral dimensions of  $1024 \times 768$ ,  $512 \times 384$ ,  $256 \times 192$ ,  $128 \times 96$ ,  $64 \times 48$ , and  $32 \times 24$ ; this is shortened to HCRNN-6 in [21]. The method that used a spectral dimension of  $128 \times 96$  and a window length of 128 in [21] is shortened to HCRNN-1.

The detection performance of FH signals attained by the ResNet-OpGST algorithm, HCRNN-6, and HCRNN-1 are shown in Figure 11. At the same time, to assess the algorithm's performance, the ROC AUC measure is used. The best performance is shown by the ROC AUC being 1, while the lowest performance is indicated by the ROC AUC being 0.5. Figure 11 shows that ResNet-OpGST has the best performance, followed by CNN-OpGST, and HCRNN-1 has the worst performance. According to the complexity analysis in Section 3.4, the dimension size of the input matrix of ResNet-OpGST is  $101 \times 200$  and the dimension size of the HCRNN-1 input matrix is  $128 \times 96$ ; the time complexity of ResNet-OpGST is similar to that of HCRNN-1 and the network storage complexity of ResNet-OpGST is 1,037,888 and that of HCRNN-1 is 652,416. However, the HCRNN-6 algorithm uses a 6-way parallel CNN network to extract spectrum features, so its computational complexity and storage complexity are much higher than the ResNet-OpGST algorithm.



(a) Detection performance curves



(b) FH detection ROC AUC

**Figure 11.** Comparison of algorithm performance.

## 5. Conclusions

Based on the time-frequency aggregation measure, the parameters of the generalized S transform are determined by using a multi-population genetic algorithm. Low time-frequency resolution and spectrum leakage are issues that can be addressed using the generalized S transform. The designed ResNet network can extract the spectrum characteristics of the generalized S transform, and accurately find the FH signal. Simulation results show that stable GST parameters  $\lambda$  and  $p$  can be obtained quickly via the multi-population genetic algorithm, and the proposed ResNet-OpGST algorithm has better detection performance and lower complexity than the compared algorithm (HCRNN) [21].

## Acknowledgments

This research is jointly supported by the National Natural Science Foundation of China (U19B2016).

## Conflict of interest

The authors declare there is no conflict of interest.

## References

1. A. Polydoros, K. Woo, LPI detection of frequency-hopping signals using autocorrelation techniques, *IEEE J. Sel. Areas Commun.*, **3** (1985), 714–726. <https://doi.org/10.1109/JSAC.1985.1146255>
2. Y. Zhou, R. Zhao, Existence detection of differential frequency hopping signal (in Chinese), *Railway Trans.*, **3** (2002), 40–44.
3. R. A. Dillard, G. M. Dillard, Likelihood-ratio detection of frequency-hopped signals, *IEEE Trans. Aerosp. Electron. Syst.*, **32** (1996), 543–553. <https://doi.org/10.1109/7.489499>
4. X. Gao, D. Li, N. Li, C. Chen, Algorithm for frequency-hopping signals detection based on suppressing power spectrum (in Chinese), *J. Jilin Univ.*, **3** (2008), 238–243. <https://doi.org/10.3969/j.issn.1671-5896.2008.03.003>
5. X. Liu, Frequency hopping signal detection based on power spectrum cancellation algorithm (in Chinese), *Instrum. Meas.*, **11** (2017), 69–73. <https://doi.org/10.3969/j.issn.1003-7241.2017.11.017>
6. X. Wu, W. Guo, W. Cai, X. Shao, Z. Pan, A method based on stochastic resonance for the detection of weak analytical signal, *Talanta*, **61** (2003), 863–869. [https://doi.org/10.1016/S0039-9140\(03\)00371-0](https://doi.org/10.1016/S0039-9140(03)00371-0)
7. M. Fargues, H. Overdyk, Wavelet-based detection of frequency hopping signals, in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers*, (1977), 515–519.
8. Y. Zheng, X. Chen, R. Zhu, Frequency hopping signal detection based on wavelet decomposition and Hilbert-Yellow transform (in Chinese), *Mod. Phys. Lett. B*, **31** (2017), 132–135. [https://doi.org/10.13873/j.1000-9787\(2017\)09-0132-04](https://doi.org/10.13873/j.1000-9787(2017)09-0132-04)



9. W. Fan, P. Xu, X. Dai, A hop generation method in frequency hopping signal acquisition system based on time frequency diagram (in Chinese), *J. Appl. Sci.*, **23** (2006), 557–562. <https://doi.org/10.3969/j.issn.0255-8297.2005.06.002>
10. Y. Lv, Y. Yi, Y. Lu, Frequency hopping signal detection technology based on overlapping sliding window time frequency analysis (in Chinese), *Electron. Inf. Countermeas. Technol.*, **35** (2020), 25–29. <https://doi.org/10.3969/j.issn.1674-2230.2020.02.007>
11. J. Liu, Z. Zhao, Y. Cao, X. Ye, L. Wang, Blind detection of multi-frequency hopping signals based on time-frequency analysis (in Chinese), *Signal Process.*, **37** (2021), 763–771. <https://doi.org/10.16798/j.issn.1003-0530.2021.05.009>
12. J. Du, J. Liu, F. Qian, A new method for time-frequency analysis of frequency hopping signals (in Chinese), *J. China Acad. Electron. Sci.*, **4** (2009), 576–579. <https://doi.org/10.3969/j.issn.1673-5692.2009.06.005>
13. R. Lowe, Localization of the complex spectrum: the S transform, *IEEE Trans. Signal Process.*, **44** (1996), 998–1001. <https://doi.org/10.1109/78.492555>
14. G. Lv, H. Liu, X. Ye, H. Yuan, Z. Geng, An improved S transform method for voltage sag detection based on optimal combination weighting (in Chinese), *Elect. Meas. Instrum.*, **57** (2020), 47–52. <https://doi.org/10.19753/j.issn1001-1390.2020.15.008>
15. F. Zhang, X. Chen, X. Luo, J. Zhang, H. Xu, Improved window parameter optimization S transform and its application in river detection (in Chinese), *Oil Geophys. Prospect.*, **56** (2021), 809–814. <https://doi.org/10.13810/j.cnki.issn.1000-7210.2021.04.014>
16. X. Yu, Research on communication signal analysis method based on Generalized S transform, *Harbin Eng. Univ.*, **1** (2018).
17. D. Sun, Y. Wang, W. Wang, D. Wei, Automatic detection model of frequency hopping signal based on hog (in Chinese), *Commun. Technol.*, **51** (2018), 758–762. <https://doi.org/10.3969/j.issn.1002-0802.2018.04.002>
18. M. Zhang, W. Wang, J. Ren, D. Wei, W. Huang, Z. Yang, et al., Detection and recognition algorithm of frequency hopping signal based on HOG-SVM, *J. Inf. Secur.*, **5** (2020), 62–77. <https://doi.org/10.19363/J.cnki.cn10-1380/tn.2020.05.06>
19. J. Hou, Z. Yao, J. Yang, Y. Li, Z. Wang, A fast detection method of frequency hopping signal based on K-means clustering (in Chinese), *Telecommun. Eng.*, **2021** (2021), 1–9.
20. Y. Wang, S. He, C. Wang, Z. Li, J. Li, H. Dai, et al., Detection and parameter estimation of frequency hopping signal based on the deep neural network, *Int. J. Electron.*, **109** (2022), 520–536. <https://doi.org/10.1080/00207217.2021.1914190>
21. K. Lee, S. Oh, Detection of frequency-hopping signals with deep learning, *IEEE Commun. Lett.*, **24** (2020), 1042–1046. <https://doi.org/10.1109/LCOMM.2020.2971216>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)