symmetry

MDPI

Article

Neural Networks for Radar Waveform Recognition

Ming Zhang, Ming Diao, Lipeng Gao * and Lutao Liu

College of Information and Telecommunication, Harbin Engineering University, Harbin 150001, China; zhangming@hrbeu.edu.cn (M.Z.); diaoming@hrbeu.edu.cn (M.D.); liulutao@hrbeu.edu.cn (L.L.)
* Correspondence: gaolipeng2000@163.com or gaolipeng@hrbeu.edu.cn; Tel.: +86-0451-8251-9804

Academic Editor: Angel Garrido
Received: 6 March 2017; Accepted: 12 May 2017; Published: 17 May 2017

Abstract: For passive radar detection system, radar waveform recognition is an important research area. In this paper, we explore an automatic radar waveform recognition system to detect, track and locate the low probability of intercept (LPI) radars. The system can classify (but not identify) 12 kinds of signals, including binary phase shift keying (BPSK) (barker codes modulated), linear frequency modulation (LFM), Costas codes, Frank code, P1-P4 codesand T1-T4 codeswith a low signal-to-noise ratio (SNR). It is one of the most extensive classification systems in the open articles. A hybrid classifier is proposed, which includes two relatively independent subsidiary networks, convolutional neural network (CNN) and Elman neural network (ENN). We determine the parameters of the architecture to make networks more effectively. Specifically, we focus on how the networks are designed, what the best set of features for classification is and what the best classified strategy is. Especially, we propose several key features for the classifier based on Choi–Williams time-frequency distribution (CWD). Finally, the recognition system is simulated by experimental data. The experiments show the overall successful recognition ratio of 94.5% at an SNR of −2 dB.

Keywords: radar countermeasure; waveform recognition; T-F distribution; convolutional neural network

1. Introduction

Modern radars usually have low instantaneous power, called low probability of intercept (LPI) radars, which are used in electronic warfare (EW). For a radar electronic intelligence (ELINT) system (anti-radar system), analyzing and classifying the waveforms of LPI radars is one of the most effective methods to detect, track and locate the LPI radars [1,2]. Therefore, the second order statistics and power spectral density are utilized in the waveforms' recognition earlier to classify phase shift keying (PSK), frequency shift keying (FSK) and amplitude shift keying (ASK) [3]. Dudczyk presents the parameters (such as pulse repetition interval (PRI), pulse width (PW), etc.) to identify different radar signals [4–7]. Nandi introduces the decision theoretic approach to classify different types of modulated signals [8]. Additionally, the ratio of successful recognition (RSR) is over 94% at a signal-to-noise ratio (SNR) $\geq$15 dB. The artificial neural network is also utilized in the recognition system. The multi-layer perceptron (MLP) recognizer reaches more than 99% recognized performance at SNR $\geq$0 dB [9]. Atomic decomposition (AD) is also addressed in the detection and classification of complex radar signals. Additionally, the receiver realizes the interception of four signals (including linear frequency modulation (LFM), PSK, FSK and continuous wave (CW)) [10]. Time-frequency techniques can increase signal processing gain for the low power signals [11]. In [12], López analyzes the differences among LFM, PSK and FSK based on the short-time Fourier transform (STFT). Additionally, the RSR $\geq$90% at SNR $\geq$0 dB. Lundén [13] introduces a wide classification system to classify the intercepted pulse compression waveforms. The system achieves overall RSR $\geq$98% at SNR $\geq$6 dB. Ming improves the system of Lundén and shows the results in [14]. The sparse classification (SC) based on random projections is proposed in [15]. The approach improves efficiency, noise robustness and information completeness. LFM, FSK and PSK are recognized with RSR ≥90% at SNR ≥0 dB.

We investigate the convolutional neural network (CNN) for radar waveform recognition. CNN has been proposed in image recognition fields [16,17]. Recently, it has been applied for speech recognition [18–21], computer vision [22,23] and handwritten recognition [24–26], etc. Abdel-Hamid introduces the approaches to reduce the further error rate by utilizing CNNs in [27]. Experimental results show that CNNs reduce the error rate by 6%–10% compared with deep neural networks (DNNs) on the speech recognition tasks. In [26], a hybrid model of two superior classifiers CNN and support vector machine (SVM) is discussed. The RSR of the model achieves more than 99.81%, in which SVM performs as a classifier and CNN works as a feature extractor.

In this paper, we explore a wide radar waveform recognition system to classify, but not identify In this paper, the meaning of "classify" is that we distinguish the different types of waveforms. Additionally, "identify" is distinguishing the different individuals of the same type. Twelve types of waveforms (LFM, BPSK, Costas codes, polyphase codes and polytime codes) by using CNN and Elman neural network (ENN) are discussed. We propose time-frequency and statistical characteristic approaches to process detected signals, which transmit in the highly noisy environment. The detected signals are processed into time-frequency images with the Choi–Williams distribution (CWD). CWD has few cross terms for signals, which is a member of the Cohen classes [28]. Time-frequency images show the three main pieces of information of signals: time location, frequency location and amplitude. In the images, time and frequency information is more robust than amplitude. To make the images more suitable for the classifier, a thresholding method is investigated. The method handles the time-frequency images as binary images. After that, binary images are addressed by noise-removing approaches. The final images are used for classification and feature extraction. However, polyphase codes (including Frank code and P1-P4 codes) and LFM are similar to each other. It is difficult to classify them through shapes individually. Therefore, we extract some effective features for further classification of them. Features extraction is from binary images through digital image processing (such as skeleton extraction, Zernike moments [29], principal component analysis (PCA), etc.). The set of features is the input of ENN. Additionally, the output of ENN is the classification result. The entire structure of the classifier consists of two networks, CNN and ENN. CNN is the primary cell of the classifier, and ENN is auxiliary. Binary images are resized for CNN to separate polytime codes (include T1–T4) from the other eight kinds of waveforms. Additionally, we extract features for ENN, which can indicate the eight remaining codes obviously. Only if "others" are selected by the CNN, ENN starts to work (see Figure 2). In the experiments, the recognition system has overall RSR ≥94% at SNR ≥−2 dB.

In this paper, the major contributions can be summarized as follows: (1) build the framework of signals processing; additionally, establish the label data for testing the system; (2) the proposed recognition system can classify as many as 12 kinds of waveforms, which are described in the context; previous articles can seldom reach such a wide range of classification of radar signals; especially, four kinds of polytime codes are classified together for the first time in the published literature; (3) almost all interested parameters and all features will be estimated by received data without a priori knowledge; (4) propose a hybrid classifier that has two different networks (CNN and ENN).

The paper is organized as follows. The structure of the recognition system is exhibited in Section 2. Section 3 proposes the signal model and preprocessing. Section 4 explores the feature extraction, including signal features and image features. Additionally, it lists all features that we need. After that, Section 5 searches the structure of the classifier and describes it in detail. Section 6 shows the experiments. Section 7 draws the conclusions.

2. System View

The entire classification system mainly consists of three components: preprocessing, feature estimation and recognition; see Figure 1. It is an automatic process from the preprocessing part to the recognition part. In the preprocessing part, the received data are transformed into time-frequency images by utilizing CWD transformation. Then, the time-frequency images are transformed into the binary image through image binarization, image opening operation and noise-removing algorithms. In the feature extraction part, we extract effective features to train and test the classifier. Different kinds of waveforms have different shapes in the images. After image processing, the differences of shapes are more significant. CNN has a powerful ability of classification, which distinguishes polytime codes from others. To classify these similar waveforms (such as polyphase codes), we extract features from detected signals and binary images. In the recognition part, all of the waveforms are classified via the proposed classifier based on the extracted features.

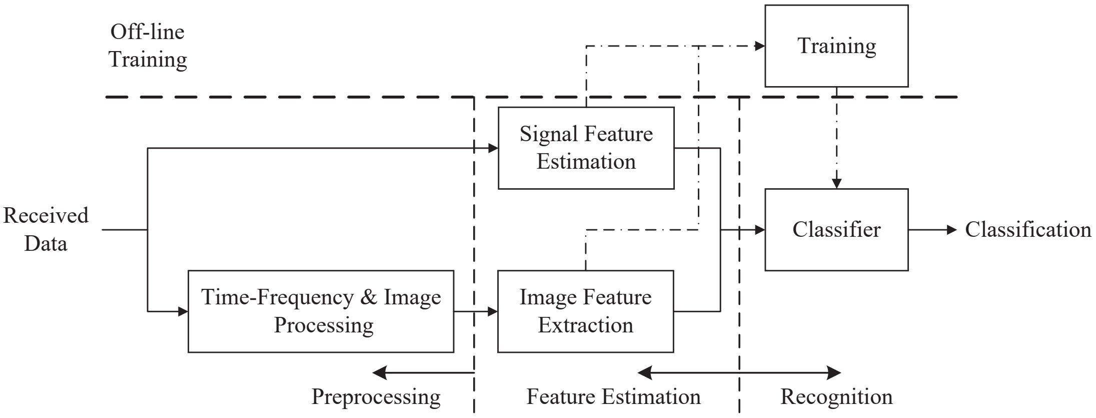

Figure 1. The figure shows the systematic components. Received data is processed in the preprocessing part and feature estimation part to extract features. And the data is classified in the classifier part.

The hybrid classifier consists of two networks, *network*1 and *network*2 ; see Figure 2. The entire classifier can classify 12 different kinds of radar waveforms, which has been mentioned in the writing. *Network*1 is the main network composed of CNN. Its input is a binary image after preprocessing. Additionally, the outputs are five different kinds of classification results. They are four kinds of polytime codes (T1-T4) and others (do not belong to the polytime class). *Network*2 is ENN, which is an auxiliary network. *Network*2 assists the main network (*network*1) in classifying the eight remaining waveforms that do not belong to polytime codes. When the waveform is considered as “others” by *network*1, *network*2 will begin to classify the waveform into one of the eight kinds of waveforms. The proposed structure of the classifier can improve the classified power.

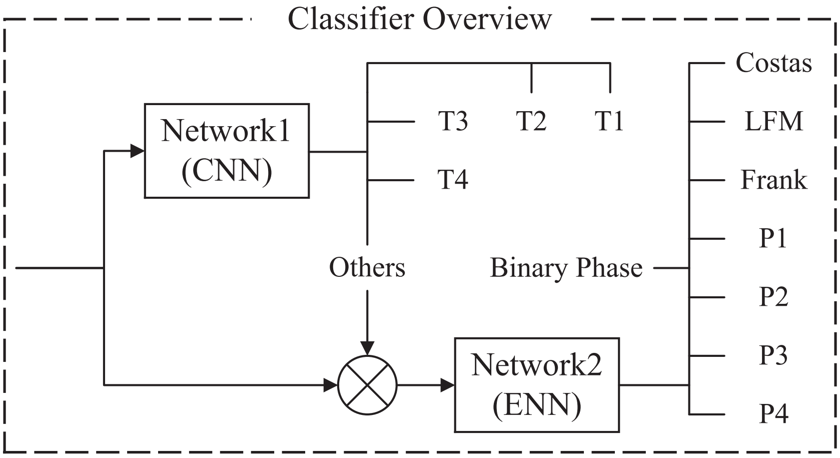

Figure 2. This figure shows the details of the classifier. *Network*1 is the main network composed of CNN and *Network*2 is ENN. *Network*2 assists the main network (*network*1) to complete the classification.

3. Preprocessing

In this section, detected signals are processed into binary images with the Choi–Williams time-frequency distribution.

3.1. Signal Model

We assume the signal is contaminated by additive white Gaussian noise (AWGN). Additionally, the amplitude is constant for time. In summary, the signal model is formulated as follows

$$y(nT) = s(nT) + m(nT) = Ae^{j\phi(nT)} + m(nT) \tag{1}$$

where $s(nT)$ is the n-th sample of complex signals. Additionally, $m(nT)$ is the n-th sample of complex white Gaussian noise (WGN). The variance of WGN equals σ_ε^2. A is the amplitude. However, for the sake of simplicity, we suppose $A = 1$. ϕ is the instantaneous phase of complex signals. To process detected signals from real to complex, Hilbert transform is applied [30].

3.2. Choi–Williams Distribution

The Choi–Williams distribution is a kind of time-frequency distribution, which expresses the details of detected signals. It can reduce the cross terms from the signals obviously.

$$\begin{aligned} C(t,\omega) = &\iiint_\infty e^{j2\pi\xi(s-t)} f(\xi,\tau) \\ &\cdot x(s+\tau/2)x^*(s-\tau/2)e^{-j\omega\tau} d\xi ds d\tau \end{aligned} \tag{2}$$

where ω and t are the axes of frequency and time, respectively. $f(\xi,\tau)$ is a two-dimensional low-pass filter to balance cross terms and resolution. The kernel function is formulated as follows:

$$f(\xi,\tau) = \exp\left[\frac{(\pi\xi\tau)^2}{2\sigma}\right]. \tag{3}$$

σ is the controllable factor. The cross terms will be more obvious with the increase of σ. In this paper, $\sigma = 1$ is applied. In Figure 3, 12 kinds of signals are transformed into time-frequency images through CWD transformation. The work in [31] proposes a new fast calculation of CWD based on standard Fourier transformation (FFT). We could recommend that the number of the sampling is the power of two, such as 256, 512, etc. In this paper, 1024 sampling points are investigated. However, the length of signals is N < 1024 for most of the time. Therefore, zero padding is utilized in the process.

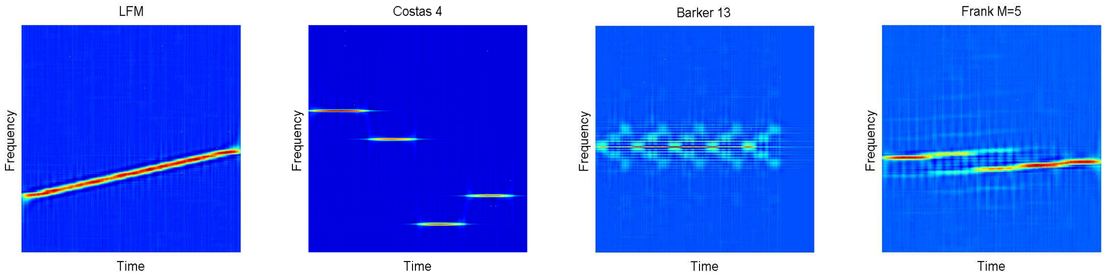

Figure 3. *Cont.*

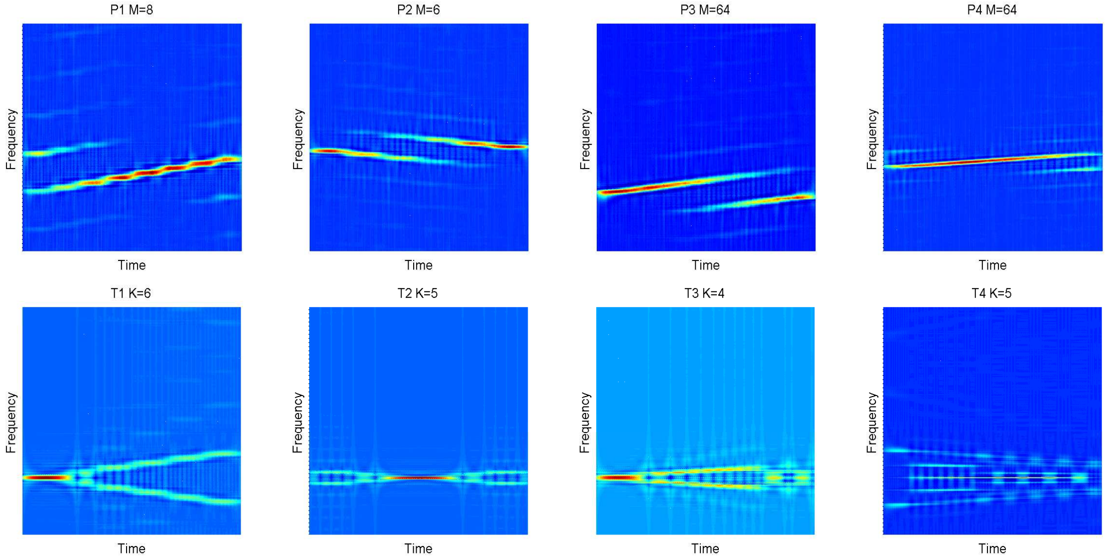

Figure 3. In this figure, different waveform classes are shown, including Linear Frequency Modulation (LFM), Binray Phase Shift Keying (BPSK), Frank, Costas codes, P1-P4 codes and T1-T4 codes. There are significant differences among the Choi-Williams Time-Frequency Distribution (CWD) images.

3.3. Binary Image

In this part, the detected signals are processed into binary images with the global thresholding algorithm [32]. Before it is done, we need to resize the time-frequency images to reduce the computational load, in which the resized size is $N \times N$. The algorithm is organized as follows.

a. Normalize the resized images $G(x,y) \in [0,1]$, i.e.,

$$G(x,y) = \frac{CWD_{N\times N}(x,y) - \min CWD_{N\times N}(x,y)}{\max(CWD_{N\times N}(x,y) - \min CWD_{N\times N}(x,y))};$$

b. Estimate the threshold T of $G(x,y)$, i.e.,

$$T = \frac{\max G(x,y) + \min G(x,y)}{2};$$

c. Separate the image into two pixel groups G_1 and G_2; G_1 includes all pixels that values $> T$, and G_2 includes others;
d. Calculate the average value μ_1 and μ_2 of two pixel groups G_1 and G_2, respectively;
e. Update the threshold, i.e.,

$$T = \frac{\mu_1 + \mu_2}{2};$$

f. Repeat (b)–(e), until the δT is smaller than 0.001, i.e.,

$$\delta T = T_{now} - T_{before};$$

g. Compute $B(x,y)$ as follows:

$$B(x,y) = \begin{cases} 1 & G(x,y) > T \\ 0 & others \end{cases};$$

h. Output $B(x,y)$.

3.4. Noise Removed

After the operation of binarization, meanwhile, there are some isolated noises and processed noises in $B(x,y)$. Isolated noises come from the noisy environment. In the binary image, isolated noises are groups of pixels that have no fixed shape. Processed noises are generated from the CWD kernel especially. In the binary image, they are straight lines, long but thin. The length is more than 50 pixels, but the width of lines is less than three pixels. Image opening operation (erosion followed by dilation) algorithms are proposed to remove the processed noises. Additionally, the size of the operational kernel is 3 × 3. It is effective to remove the shape whose width is less than three pixels. For isolated noises, we count the number of pixels of signals or noises. In this paper, the groups are removed, in which sizes are smaller than 10% of the largest one. The content of removing noise is introduced in Figure 4. The finished binary images are used in CNN and feature extraction.

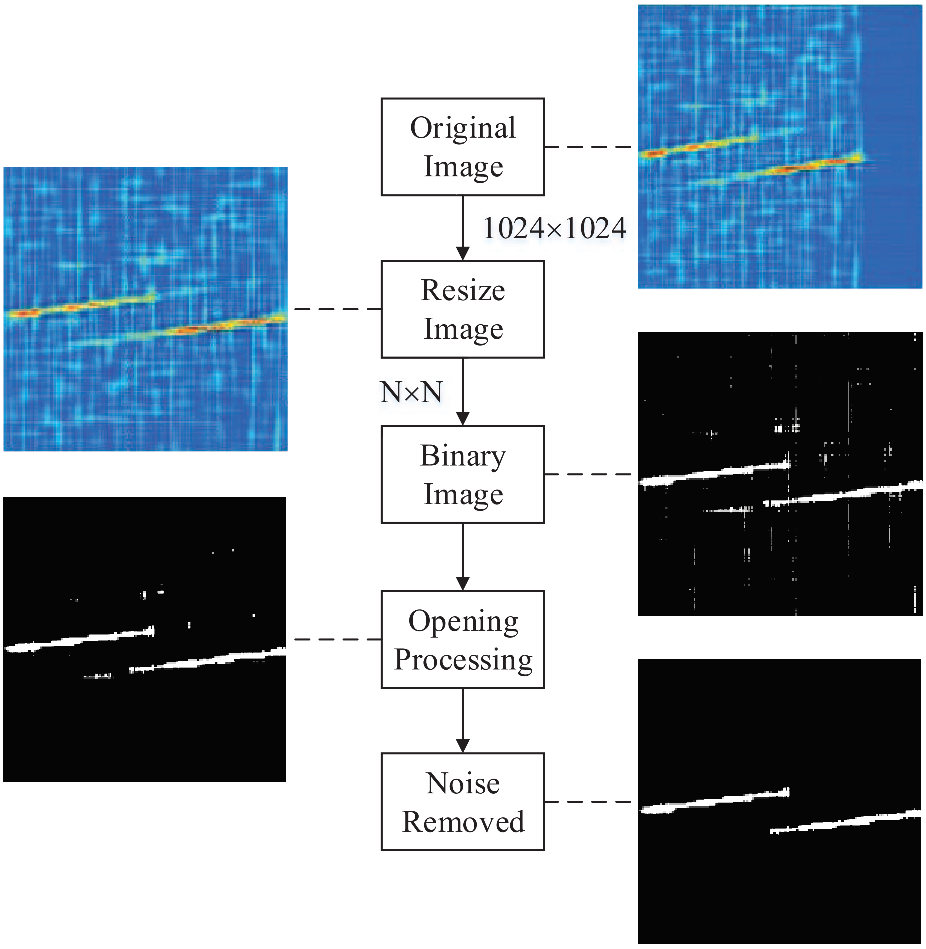

Figure 4. In this figure, we exhibit the processing with P3 code at an signal-to-noise ratio (SNR) of −4 dB.

4. Feature Extraction

In this section, we extract some useful features and build a feature vector for ENN in order to assist the CNN to complete recognition. The section consists of two parts, including signal features and image features. The features, which we can estimate or calculate from detected signals directly, belong to signal features. Similarly, image features include the features that are extract from binary images. Table 1 lists the signal features and image features that are used in *network*2.

Table 1. List of the features for *network*2.

Index	Description	Symbol
1	Moment (1-order)	$\hat{M}_{10}$
2	Moment (2-order)	$\hat{M}_{20}$
3	Cumulant (2-order)	$\hat{C}_{20}$
4	PSD maximum (1-order)	γ_1
5	PSD maximum (2-order)	γ_2
6	Std. of phase	$\hat{\sigma}_\phi$
7	Std. of frequency	$\hat{\sigma}_f$
8	No. of objects (20%)	N_{obj1}
9	No. of objects (50%)	N_{obj2}
10	CWD time peak location	t_{max}
11	Std. of object width	$\hat{\sigma}_{obj}$
12	Maximum of PCA degree	$\hat{\theta}_{max}$
13	Std. of f_n	$\hat{\sigma}_{Wf}$
14	Autocorrelationof f_n	r
15	FFT of correlationf_n	$\hat{a}_{max}$
16	Pseudo-Zernike moment (2-order)	$\hat{Z}_{20}$
17	Pseudo-Zernike moment (3-order)	$\hat{Z}_{30}$
18	Pseudo-Zernike moment (4-order)	$\hat{Z}_{22}$
19	Pseudo-Zernike moment (4-order)	$\hat{Z}_{31}$
20	Pseudo-Zernike moment (5-order)	$\hat{Z}_{32}$
21	Pseudo-Zernike moment (6-order)	$\hat{Z}_{33}$
22	Pseudo-Zernike moment (7-order)	$\hat{Z}_{43}$

PSD: Power Spectral Density. PCA: Principal Component Analysis. FFT: Fast Fourier Transformation. Std.: Standard Deviation.

4.1. Signal Features

In this part, the features are extracted from signals based on signal processing approaches.

4.1.1. Based on the Statistics

We estimate the n-order moment of complex signals as follows:

$$\hat{M}_{nm} = \left| \frac{1}{N} \sum_{k=0}^{N-1} y^{n-m}(k) (y^*(k))^m \right| \tag{4}$$

where $(*)$ is the conjugated symbol and N is the sample number. We utilize absolute values to ensure that the estimated values are invariant constants when the signal phase rotates. $\hat{M}_{10}$ and $\hat{M}_{20}$ are calculated by Equation (4).

The n-order cumulant is given by [33,34]:

$$\hat{C}_{nm} = \left| \frac{1}{N} \sum_{k=0}^{N-1} (y(k) - \hat{M}_{10})^{n-m} (y^*(k) - \hat{M}_{10})^m \right| \tag{5}$$

where, the same as context, $\hat{M}_{10}$ is from Equation (4).

4.1.2. Based on the Power Spectral Density

Before estimation of Power Spectral Density (PSD), the detected signals should be normalized as follows:

$$\tilde{y}(k) = \frac{y(k)}{\sqrt{\hat{M}_{21} - \sigma_\varepsilon^2}} \tag{6}$$

where $\hat{M}_{21}$ is obtained from Equation (4) and $y(k)$ is the k-th sample. The variance of additive noise σ_ε^2 can be obtained in [35].

The PSD are calculated as follows:

$$\gamma_m = \frac{1}{N} \max_n \left\{ \frac{1}{N} \left| \sum_{k=0}^{N-1} \tilde{y}^m(k) e^{-j2\pi nk/N} \right|^2 \right\} \tag{7}$$

where $\tilde{y}(k)$ is from Equation (6).

4.1.3. Based on the Instantaneous Properties

Instantaneous properties are the essential characteristics of detected signals. They can distinguish frequency modulated signals from phase modulated signals effectively. In this paper, we estimate the instantaneous frequency and instantaneous phase from samples. The standard deviation of instantaneous phase is addressed in [9]. For brevity, $\phi(k) = \tan^{-1}[\mathrm{Im}(y(k))/\mathrm{Re}(y(k))]$ is applied; where, Re and Im are the real and imaginary parts of complex signals, respectively. The standard deviation of instantaneous phase is given by:

$$\hat{\sigma}_\phi = \sqrt{\frac{1}{N}\left(\sum_{k=0}^{N-1} \phi^2(k)\right) - \left(\frac{1}{N}\sum_{k=0}^{N-1} |\phi(k)|\right)^2} \tag{8}$$

where N is the sample number. ϕ is the instantaneous phase with the range of $[-\pi, \pi]$.

Instantaneous frequency estimation is more complex than instantaneous phase. We describe the method in several steps to make it clear.

a. Calculate $\phi(k)$;

b. Calculate $\phi_u(k)^\bigstar$ from $\phi(k)$;

c. Calculate $f(k)^{\bigstar\bigstar}$, i.e.,

$$f(k) = \phi_u(k) - \phi_u(k-1);$$

d. Calculate μ_f, i.e.,

$$\mu_f := \frac{1}{N} \sum\nolimits_{k=0}^{N-1} f(k);$$

e. Normalize the instantaneous frequency $\tilde{f}(k)$,

$$\tilde{f}(k) = (f(k) - \mu_f)/(\max|f(k) - \mu_f|)$$

f. Output the standard deviation of instantaneous frequency $\hat{\sigma}_f$,

$$\hat{\sigma}_f = \sqrt{\frac{1}{N}\left(\sum_{k=0}^{N-1} \tilde{f}^2(k)\right) - \left(\frac{1}{N}\sum_{k=0}^{N-1} |\tilde{f}(k)|\right)^2}. \tag{9}$$

★ $\phi_u(k)$ is the unwrapped phase of $\phi(k)$. When the absolute jumps from $\phi(k)$, we can add ±2π to recover the consecutive phase.

★★ In the sequence of $f(k)$, some spikes are created by processing. We use the median filter algorithm with window size of five to smooth the spikes.

4.2. Image Features

In this part, we extract the features based on binary images. The number of objects (N_{obj}) is a key feature. For instance, Costas codes have more than three objects, but Frank code and P2 have two. Additionally, P1, P4 and LFM only have one. We estimate two features N_{obj1} and N_{obj2}. N_{obj1} is the number of objects, the sizes of the pixels of which are more than 20% of the size of the largest object. Likewise, $N_{obj2} \geq 50\%$.

The maximum energy location in time domain is also a feature, i.e.,

$$t_{max} = \frac{1}{N-1} \arg\max_{t} \{C_{N\times N}(t,\omega)\} \tag{10}$$

where $C_{N\times N}(t,\omega)$ is the resized time-frequency image and N is the sample number.

The standard deviation of the width of signal objects ($\hat{\sigma}_{obj}$) can describe the concentration of signal energy. The feature is estimated as follows.

- Repeat for every object, do $k = 1, 2, ..., N_{obj1}$;

  1. Retain the k-th object and remove others, called $B_k(x,y)$;
  2. Estimate the principal components of $B_k(x,y)$;
  3. Rotate★ the $B_k(x,y)$ until the principal components are vertical; record as $B_{k'}(x,y)$;
  4. Sum the vertical axis, i.e.,
  $$v(x) = \sum_{y=0}^{N-1} B_{k'}(x,y),$$
  $x = 0, 1, 2, ..., N-1$;
  5. Normalize $v(x)$ as follows
  $$\hat{v}(x) = \frac{v(x)}{\max\{v(x)\}};$$
  6. Estimate the standard deviation of $\hat{v}(x)$, i.e.,
  $$\hat{\sigma}_{k,obj1} = \sqrt{1/N \sum_x \hat{v}^2(x) - \left(1/N \sum_x \hat{v}(x)\right)^2},$$
  where N is the sample number;

- Output the rotation degree $\hat{\theta}_{max}$, which performs Step (c) at the maximum object.
- Output the average of the $\hat{\sigma}_{k,obj1}$, i.e.,

$$\hat{\sigma}_{obj} = (1/N_{obj1}) \sum_{k=1}^{N_{obj1}} \hat{\sigma}_{k,obj1}. \tag{11}$$

★ Nearest neighbor interpolation is applied in rotation processing.

P2 has a negative slope in five types of polyphase codes. Therefore, the feature $\hat{\theta}_{max}$ can classify P2 from others easily. The feature shows the angle between the maximum object and the vertical direction. It can be obtained from the calculation of $\hat{\sigma}_{obj}$ easily.

Next, we retain the maximum object in the binary image, but others are removed. The skeleton of the object is extracted by utilizing the image morphology method. Additionally, the linear trend of the object is also estimated based on minimizing the square errors method at the same time. Subtract the linear trend from the skeleton to achieve the difference vector f_n. The standard deviation of f_n is estimated as:

$$\hat{\sigma}_{Wf} = \sqrt{\frac{1}{M-1}\sum_{k=1}^{M} f_n^2(k) - \left(\frac{1}{M-1}\sum_{k=1}^{M} f_n(k)\right)^2} \tag{12}$$

where M is the sample number of f_n.

Some features are extracted by using autocorrelation of f_n, i.e., $c(m) = \sum_k f_n(k) f_n(k-m)$, $m = 0, 1, ..., N-1$. The autocorrelation method makes differences more significant among stepped waveforms (P1, Frank code) and linear waveforms (P3, P4, LFM). See Figure 3 for more details.

The ratio of the maximum value and sidelobe maximum value of $c(m)$ is formulated as:

$$r = \frac{N\max_{m \in [m_0, N-1]} c(m)}{(N - m_1) \max c(m)} \tag{13}$$

where m_0 is the value corresponding to the minimum of $c(m)$ and m_1 is the value corresponding to the maximum of $c(m)$ in the location of $[m_0, N-1]$.

We estimate the maximum of the absolute of FFT operation $\hat{a}_{max}$ as follows:

$$\hat{a}_{max} = \max\{\text{abs}\left[\text{FFT}(c_{normal}(m))\right]\} \tag{14}$$

where $c_{normal}(m)$ is normalized from $c(m)$ and $c_{normal}(m) \in [-1, 1]$.

Pseudo-Zernike moments are invariant for topological transformation [36], such as rotation, translation, mirroring and scaling. They are widely applied in pattern recognition [37–39]. The n-order image geometric moments are calculated as:

$$m_{pq} = \sum_x \sum_y B(x, y) x^p y^q \tag{15}$$

where $B(x, y)$ is from Section 3.3. The central geometric moments for scale and translation invariant are given by:

$$G_{pq} = \frac{1}{m_{00}^{(p+q+2)/2}} \sum_x \sum_y B(x, y)(x - \bar{x})^p (y - \bar{y})^q \tag{16}$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$.

The scale and translation invariant radial geometric moments are shown as:

$$R_{pq} = \frac{1}{m_{00}^{(p+q+3)/2}} \sum_x \sum_y B(x, y) \left(\tilde{x}^2 + \tilde{y}^2\right)^{1/2} \tilde{x}^p \tilde{y}^q \tag{17}$$

where $\tilde{x} = x - \bar{x}$ and $\tilde{y} = y - \bar{y}$.

Then, the pseudo-Zernike moments can be estimated as follows:

$$\begin{aligned} Z_{nm} = & \frac{n+1}{\pi} \sum_{\substack{s=0 \\ \overline{n-s-m=even}}}^{n-|m|} \sum_{a=0}^{k} \sum_{b=0}^{m} (-j)^b \\ & \begin{pmatrix} k \\ a \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} D_{nms} G_{2k-2a+m-b, 2a+b} \\ & + \frac{n+1}{\pi} \sum_{\substack{s=0 \\ \overline{n-s-m=odd}}}^{n-|m|} \sum_{a=0}^{d} \sum_{b=0}^{m} (-j)^b \\ & \begin{pmatrix} d \\ a \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} D_{nms} R_{2d-2a+m-b, 2a+b} \end{aligned} \tag{18}$$

where $k = (n - s - m)/2$, $d = (n - s - m - 1)/2$ and:

$$D_{nms} = (-1)^s \frac{(2n + 1 - s)!}{s!(n - |m| - s)!(n + |m| + 1 - s)!}. \tag{19}$$

At last, $\hat{Z}_{nm}$ is estimated, i.e., $\hat{Z}_{nm} = \ln|Z_{nm}|$. The members of pseudo-Zernike moments include $\hat{Z}_{20}$, $\hat{Z}_{22}$, $\hat{Z}_{30}$, $\hat{Z}_{31}$, $\hat{Z}_{32}$, $\hat{Z}_{33}$ and $\hat{Z}_{43}$.

5. Classifier

In Section 4, we complete the resized binary image labels for CNN and the feature vector extraction with 22 elements for ENN. In this section, we describe the structure of two networks in detail.

5.1. CNN

CNN is a new neural network, which has a special structure for image feature extraction. Different from traditional network, the input of CNN is a two-dimensional feature (image). The convolution layers can extract information, and pooling layers reduce computer load effectively. CNN is not a full connected network, which is similar to the cerebral cortex. The architecture of the CNN model is shown in Figure 5. CNN has the hierarchical architecture [40]. Hence, we describe the neural architecture as follows.

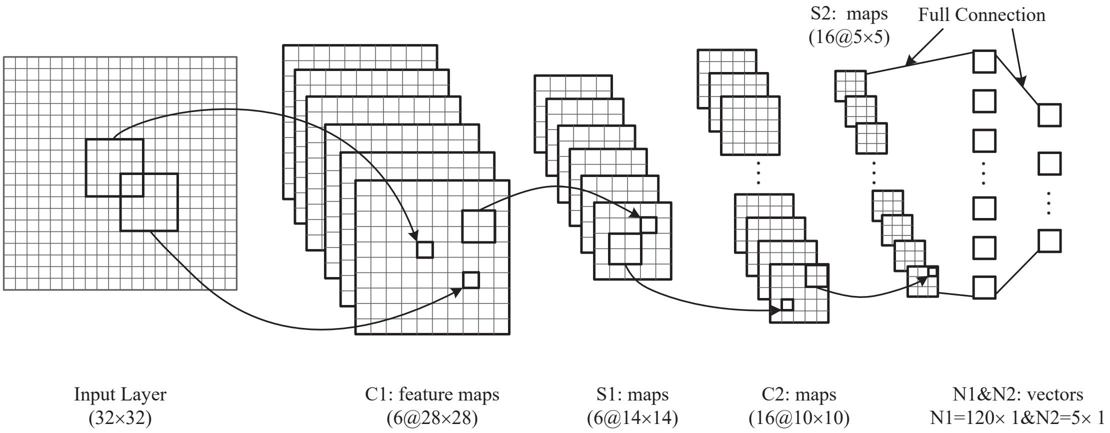

Figure 5. The figure shows the structure of Convolutional Neural Network (CNN). The input image are processed in the hidden layers and classified in the out layer.

a. The input layer is a binary image, which is from Section 3.4. To reduce the computer load, we resize the image to 32 × 32 with the nearest neighbor interpolation algorithm.

b. The first hidden layer C_1 is a convolutional layer, which has six feature maps. Different feature maps require a different convolutional kernel. C_1 has six convolutional kernels with a size of 5 × 5. We utilize $C_1(m,n,k)$ to represent the value of the k-th feature map at position (m,n) in the C_1 layer.

c. The second hidden layer S_1 is a down-sampling layer with six feature maps. In S_1, every feature value is the average of four adjacent elements in C_1. We denote $S_1(m,n,k)$ as the context. Further, we have:

$$\begin{aligned} S_1(m,n,k) = \ & \mathrm{mean}(C_1(2m-1,2n-1,k), C_1(2m-1,2n, \\ & k), C_1(2m,2n-1,k), C_1(2m,2n,k)). \end{aligned} \tag{20}$$

The size of feature maps in S_1 reduces to $1/4$, compared with feature maps of C_1.

d. C_2 is a convolutional layer with 16 different kernels. It is not fully connected with the S_1 layer [41]. The connection details are described in Table 2. $C_2(m,n,k)$ is also utilized to describe the neurons in this layer. For the α-th column in Table 2, we mark row indices by $\beta_{\alpha,0}, \beta_{\alpha,1}, \cdot, \beta_{\alpha,p-1}$. For instance, if $\alpha = 7$, then we will get parameters as follows: $p = 4$, $\beta_{7,0} = 1$, $\beta_{7,1} = 2$,

$\beta_{7,2} = 3$, $\beta_{7,3} = 4$. Further, the size of the convolutional kernel is $p \times 5 \times 5$. K_α is the α-th kernel. Additionally, we have:

$$C_2(m,n,\alpha) = \sum_{r=0}^{p-1} \sum_{m_0=0}^{4} \sum_{n_0=0}^{4} [\, S_1(m+m_0, n+n_0, \beta_{\alpha,r}) \\ \times K_\alpha(5-m_0, 5-n_0, p-1-r)\,]. \tag{21}$$

For example, for the zeroth column, $p = 3$, $\alpha_{00} = 0$, $\alpha_{01} = 1$, $\alpha_{02} = 2$, and we also have:

$$C_2(m,n,0) = \sum_{r=0}^{p-1} \sum_{m_0=0}^{4} \sum_{n_0=0}^{4} [\, S_1(m+m_0, n+n_0, \beta_{0,r}) \\ \times K_0(5-m_0, 5-n_0, p-1-r)\,]. \tag{22}$$

e. Similar to S_1, this layer is a down-sampling layer, called S_2. S_2 has 16 feature maps. To follow the context in Equation (20), we donate:

$$S_2(m,n,k) = \operatorname{mean}(C_2(2m-1, 2n-1, k), C_2(2m-1, 2n, \\ k), C_2(2m, 2n-1, k), C_2(2m, 2n, k)). \tag{23}$$

f. The connection between S_2 and N_1 is a full connection. Each kernel in N_1 will be connected with all of the feature maps in S_2. There are 120 kernels in this layer. Additionally, the size of the kernel is 5×5, which means the output is a column vector with the size of 120×1. We describe $N_1(\lambda)$ as the λ-th feature map of N_1 and K_λ as the λ-th kernel. Then, we have:

$$N_1(\lambda) = \sum_{r=0}^{15} \sum_{m_0=0}^{4} \sum_{n_0=0}^{4} [\, S_2(m_0, n_0, r) \\ \times K_\lambda(5-m_0, 5-n_0, 15-r)\,]. \tag{24}$$

g. Finally, the connected style between N_1 and output layer is fully connected. There are five neurons (defined by the classes we want to classify) in the output layer with the *sigmoid* function.

Table 2. Connection detail about S_1 and C_2.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	✓				✓	✓	✓			✓	✓	✓	✓		✓	✓
1	✓	✓				✓	✓	✓			✓	✓	✓	✓		✓
2	✓	✓	✓				✓	✓	✓			✓		✓	✓	✓
3		✓	✓	✓			✓	✓	✓	✓			✓		✓	✓
4			✓	✓	✓			✓	✓	✓	✓		✓	✓		✓
5				✓	✓	✓			✓	✓	✓	✓		✓	✓	✓

5.2. ENN

The three-layer ENN is utilized in the paper for signal classification. The connections, which connect different hidden layers or output layers, have different weights [42]. At every time step, the input is propagated in a feed-forward fashion and the feedback of the output. Additionally, the error back propagation (BP) learning algorithm is also utilized [43]. The connection results in that the context units always maintain a copy of the previous values of hidden units. Thus, the network can keep the past state, which is useful for applications such as sequence prediction [44–46]. In Figure 6, there are 46 neurons in the hidden layer. For the input and output layer, the number of neurons is determined by the dimension of input and output vectors. Sigmoid function $f(x) = 1/(1+e^{-x})$ is proposed in every layer. In [47], Sheela discusses the different methods to fix the neurons number H_{num} of hidden layers. In this paper, a simple formula is given by:

$$H_{num} = \frac{C \times X + 0.5 \times C \times (X^2 + X) - 1}{C + X} \tag{25}$$

where X is the dimension of the feature vector. C is the number of categories. The proposed formula cannot determine the optimal number of hidden layer completely. We may fine-tune the number in some situations. Forty-six neurons of hidden layers are applied in this paper.

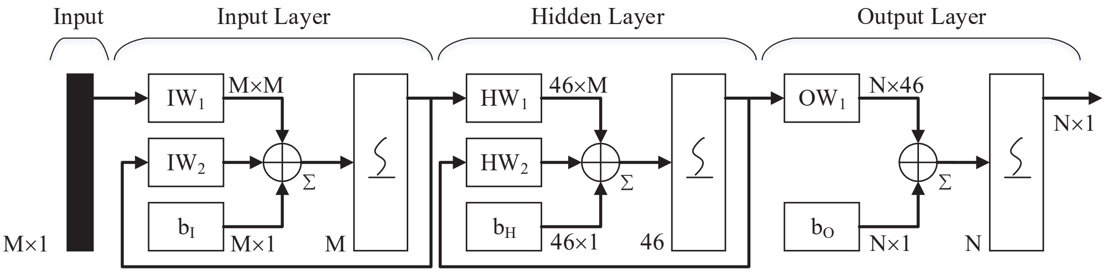

Figure 6. This figure shows the structure of Elman neural network. It is a 3 layers network that has feedback loops. So, the network can keep the past state, which is useful for waveform classification.

6. Simulation Results and Discussion

In this section, the performance of the proposed recognition system is analyzed by utilizing simulated data. The section consists of three parts, including creating the simulated data, discussing the relationship between SNR and RSR, depicting the accurate rate of robustness and summarizing the experiments.

6.1. Production of Simulated Signals

In this part, simulated signals are created. In addition, the SNR is proposed as $\text{SNR} = 10\log_{10}(\sigma_s^2)/(\sigma_\varepsilon^2)$; where σ_s^2 and σ_ε^2 are the variances of the signal and noise, respectively. Every signal has different parameters that need to be set. We denote a uniform variable $\text{U}(\cdot)$ based on the sample rate. For example, we assume that the original frequency (f_0) is 1000 Hz and the sample rate (f_s) is 8000 Hz. Then, the uniform result is $f_0 = \text{U}(f_0/f_s) = \text{U}(1/8)$. Meanwhile, $\text{U}(1/8, 1/4)$ expresses the random variable that belongs to $[\text{U}(1/8), \text{U}(1/4)]$. And in this paragraph, $[1, 3]$ also represents a set that includes $\{1, 2, 3\}$. Table 3 lists the parameters of the waveforms. For LFM, the sample points change from 500–1000 randomly. Additionally, the range of bandwidth (Δf) is $\text{U}(1/16, 1/8)$, so is the initial frequency (f_0). For BPSK, the cycle number per phase code (cpp) and the code periods' number (N_p) are $[1, 5]$and $[100, 300]$, respectively. The length of the Barker codes is selected from $\{7, 11, 13\}$ randomly. The carrier frequency is $\text{U}(1/8, 1/4)$. For the Costas codes, the fundamental frequency (f_{min}) is $\text{U}(1/24)$. Additionally, the frequency changed number is $[3, 6]$. For the Frank code, frequency steps (M) are in the range of $[4, 8]$. Polyphase codes have the same types of parameters as the Frank code. For polytime codes, the range of segments number (k) and overall code duration (T) are $[4, 6]$ and $[0.07, 0.1]$, respectively.

Table 3. List of the parameters of simulated signals.

Signal Waveforms	Parameters	Uniform Ranges
-	Sampling rate (f_s)	$\mathrm{U}(1)$
LFM	Samples number (N)	$[500, 1000]$
	Bandwidth (Δf)	$\mathrm{U}(1/16, 1/8)$
	Initial frequency (f_0)	$\mathrm{U}(1/16, 1/8)$
BPSK	Cycles per phase code (cpp)	$[1, 5]$
	Number of code periods (N_p)	$[100, 300]$
	Barker codes	$\{7, 11, 13\}$
	Carrier frequency (f_c)	$\mathrm{U}(1/8, 1/4)$
Costas codes	N	$[512, 1024]$
	Fundamental frequency (f_{min})	$\mathrm{U}(1/24)$
	Number changed	$[3, 6]$
Frank and P1 codes	f_c	$\mathrm{U}(1/8, 1/4)$
	cpp	$[1, 5]$
	Frequency steps (M)	$[4, 8]$
P2 code	f_c	$\mathrm{U}(1/8, 1/4)$
	cpp	$[1, 5]$
	M	$2 \times [2, 4]$
P3 and P4 codes	f_c	$\mathrm{U}(1/8, 1/4)$
	cpp	$[1, 5]$
	M	$2 \times [16, 35]$
T1-T4 codes	Number of segments (k)	$[4, 6]$
	Overall code duration (T)	$[0.07, 0.1]$

6.2. Experiment with SNR

In this part, we depict the relation between SNR and RSR in Figure 7. There are 1000 labels in each waveform class. Twenty percent of the labels are utilized for testing and 80% for training. The result is compared with Lundén's system [13] and our previous work [14], both of which are wide systems in waveform classification.

Figure 7 plots the experimental results of RSR with different SNR. Twelve kinds of waveforms and the "overall" are provided. The solid line shows the proposed system, and the dotted lines represent others. For LFM and P4, the proposed approach provides better performance than Lundén's, especially at low SNR, but poorer than the previous work, although the difference is not too much. For BPSK and Costas codes, the three RSRs almost have similar results, and all of them are at a high level. For Frank and P2, the results of the proposed method and previous work are alike and higher than Lundén's. In the simulation of P1, the proposed method is the best when the SNR is more than -2 dB. The results of P3 are similar to P1; proposed method performs well at high SNR. For polytime codes, the proposed approach also has excellent RSRs. It benefits from the outstanding design of pre-processing and the high RSR of the classifier. Finally, the overall RSR has been raised by 20% in the proposed approaches, compared to Lundén's and previous work. At SNR of -2 dB, the overall probabilities are still more than 90%. Table 4 exhibits the confusion table of 12 kinds of waveforms at the SNR of -2 dB. As Table 4 shows, the waveforms of P3 and P4 are not "always" classified correctly. For P3, most of the errors are classified into the Frank code. Meanwhile, most of the errors of P4 are classified into LFM. However, the two pairs are very similar; see Figure 3.

Table 4. Confusion matrix for the system at an SNR of −2 dB. The overall Ratio of Successful Recognition (RSR) is 94.5%.

	T1	T2	T3	T4	LFM	Costas	BPSK	Frank	P1	P2	P3	P4
T1	99.5	0	0	0	0	0	0	0	0	0	0	0
T2	0	98.5	0	0	0	0	0	0	0	0	0	0
T3	0.5	0	96	0	0	0	0	0	0	0	0	0
T4	0	0	0	98.5	0	1.5	2	0	0	0	0	0
LFM	0	0	0	0	89.5	0	0	0	1	0	2	15.5
Costas	0	0	0	0	0	97.5	0	0	0	1.5	1	0
BPSK	0	0	0.5	0	0	0	98	1	0	0	0	0.5
Frank	0	0	0	0	1	1	0	90	4	7	11	1.5
P1	0	0	0	0	0.5	0	0	0	87.5	0	0	7.5
P2	0	0	0	0	1	0	0	6.5	5.5	90	3	5.5
P3	0	1.5	3.5	1.5	0	0	0	2.5	1	1.5	78.5	0.5
P4	0	0	0	0	8	0	0	0	1	0	4.5	69

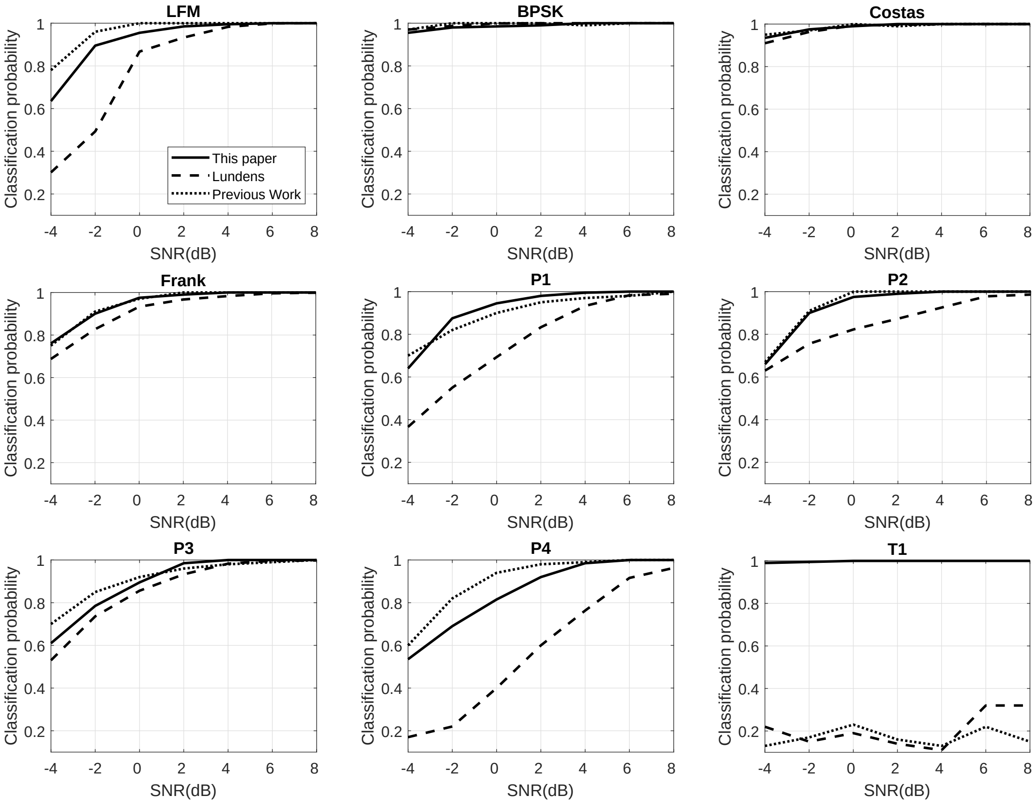

Figure 7. *Cont.*

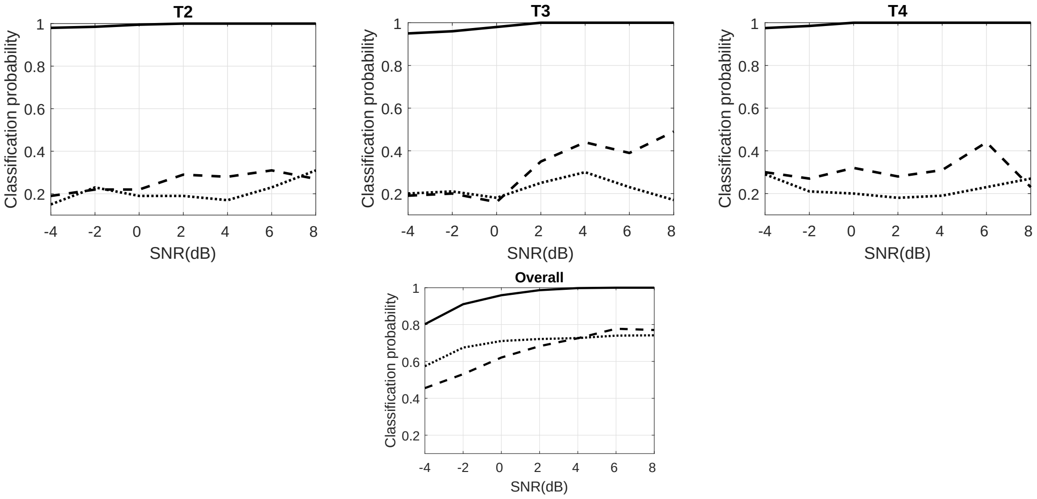

Figure 7. This figure depicts the different probabilities of 12 types of radar waveforms with testing data. SNR: Signal-to-noise ratio.

6.3. Experiment with Robustness

The robustness of proposed approaches is explored in different training samples. There are 900 label samples in each waveform for training and 100 labels for testing. Afterwards, the training samples will be increased from 100–900 with a step of 200. Meanwhile, the experiment will be repeated for three times in the condition of SNR = −4 dB, 0 dB and 6 dB.

Figure 8 plots the impact of training samples on successful recognition with three conditions of SNR. In general, it is positively correlated between training samples and successful recognition. When the samples are less than 500, the successful recognition increases obviously. However, when the samples are more than 500, the successful recognition is substantially retained. It means that the proposed approaches are able to work well in a small number of samples.

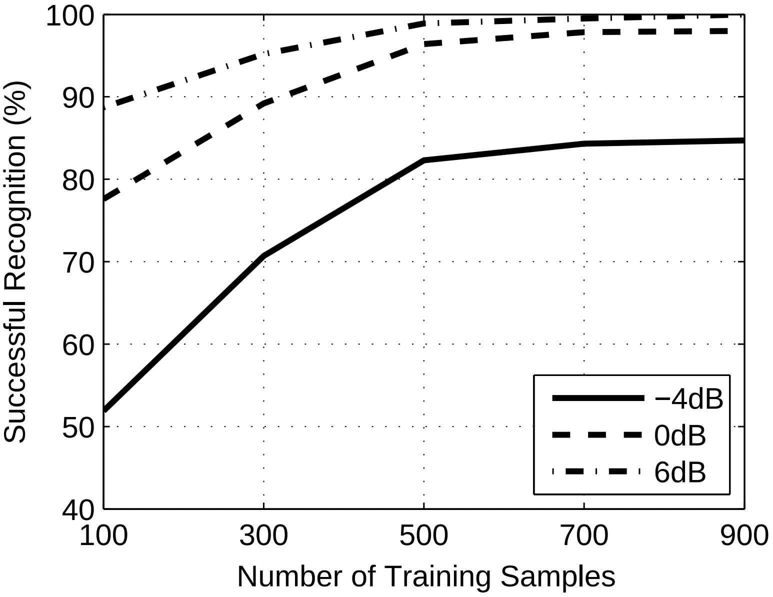

Figure 8. The figure shows the successful recognition ratio of different numbers of samples.

6.4. Experiment with Computational Burden

Computational burden is also an important issue for the classification system. We measure the time of the proposed method and compare it with [13,14] in the same conditions. Three different SNRs, −4 dB, −0 dB and 6 dB are tested, and each test repeats 50 times to calculate the average value. Table 5 shows the testing environment, and Table 6 demonstrates the testing results, respectively.

Table 5. The testing environment.

Item	Model/Version
CPU	E5-1620v2 (Intel)
Memory	16GB (DDR3 @ 1600 MHz)
GPU	NVS315 (Quadro)
MATLAB	R2012a

CPU: Central Processing Unit. GPU: Graphics Processing Unit. MATLAB is a software produced by the MathWorks, Inc. that located in Natick, Massachusetts, United States.

Table 6. Computational burden test (this paper/previous work/Lundén; unit: s).

	LFM	BPSK	Costas	Frank
−4 dB	54.798/55.302/85.331	51.117/51.132/82.553	54.463/54.798/84.735	56.115/56.221/86.132
0 dB	54.336/55.096/85.152	50.860/50.903/82.094	54.108/54.255/84.113	55.704/55.909/85.754
6 dB	53.983/54.887/84.755	50.378/50.875/81.598	53.766/53.842/83.795	55.368/55.806/85.389
	P1	**P2**	**P3**	**P4**
−4 dB	58.887/58.889/88.112	55.559/55.759/86.739	58.386/58.522/87.117	54.105/54.732/85.079
0 dB	58.398/58.519/87.847	55.308/55.431/86.180	58.106/58.310/86.869	53.858/54.338/84.787
6 dB	57.792/58.034/87.106	54.668/55.307/85.848	57.707/57.802/86.478	53.501/54.196/84.503
	T1	**T2**	**T3**	**T4**
−4 dB	53.781/54.086/83.308	52.896/53.117/85.401	55.269/55.887/86.249	56.703/56.861/85.322
0 dB	53.266/53.799/83.011	52.715/52.980/85.166	54.523/55.300/86.093	56.359/56.622/85.054
6 dB	52.823/53.201/82.799	52.107/52.741/84.455	54.396/54.916/85.702	55.993/56.279/84.811

In Table 6, the proposed method and previous work spend less than 60 s, while Lundén's more than 80 s; because Lundén's method has more calculations, we do not need to compute, such as the Wigner–Ville distribution, peak search and data driven, etc. We also improve the effectiveness of the system and reduce the consumption of time compared with previous work. In the same type of waveform, the highest SNR has the least time. In the different types of waveforms, BPSK is easiest to calculate, but P3 code is the opposite. However, overall, the change of cost is not obvious. The proposed method is stable, and different waveforms or SNR also have little effect on the computational burden of the classification system.

7. Conclusions

In this paper, an automatic system to realize the recognition of radar signal waveforms is proposed. We build the processing flow for detected signals by utilizing signal and image processing algorithms. Using these methods, the signal waveforms are fully represented via sets of feature vectors and binary images. The vectors and images are classified into 12 types in the classifier. The simulation results show that the overall RSR is more than 94% at SNR ≥ -2 dB. Additionally, the processes of feature extraction and noise removed make the system robust. When the sample labels are more than 500, the successful recognition is substantially retained. At last, the computational burden is tested. The proposed method is stable in different waveforms or SNR and spends less time than Lundén's method and our previous work.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61201410, 61571149 and the Fundamental Research Funds for the Central Universities HEUCF170802. The authors would like to thank the associate editor and reviewers for useful feedback.

Author Contributions: M.Z. and M.D. conceived of and designed the experiments. M.Z. performed the experiments. M.Z. and LT.L. analyzed the data. LP.G. contributed analysis tools. M.Z. wrote the paper.

Conflicts of Interest: All co-authors have seen and agree with the contents of the manuscript and declare no conflict of interest. We certify that the submission is original work and is not under review at any other publication. The founding sponsors have no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

LPI	Low probability of intercept
EW	Electronic warfare
ELINT	Electronic intelligence
PSK	Phase shift keying
FSK	Frequency shift keying
ASK	Smplitude shift keying
PRI	Pulse repetition interval
PW	Pulse width
RSR	Ratio of successful recognition
SNR	Signal to noise ratio
MLP	Multi-layer perceptron
AD	Atomic decomposition
LFM	Linear frequency modulation
CW	Continuous wave
STFT	Short time Fourier transform
SC	Sparse classification
CNN	Convolutional neural network
DNN	Deep neural network
SVM	Support vector machine
BPSK	Binary phase shift keying
ENN	Elman neural network
CWD	Choi–Williams time-frequency distribution
PCA	Principal component analysis
AWGN	Additive white Gaussian noise
WGN	White gaussian noise
FFT	Fast Fourier transformation
PSD	Power spectral density
CPU	Central Processing Unit
GPU	Graphics Processing Unit

References

1. Adamy, D.L. *EW 102: A Second Course in Electronic Warfare*; Artech House: Norwood, MA, USA, 2004.
2. Adamy, D.L. *EW 101: A First Course in Electronic Warfare*; Artech House: Norwood, MA, USA, 2000.
3. Nandi, A.; Azzouz, E. Automatic analogue modulation recognition. *Signal Process.* **1995**, *46*, 211–222.
4. Dudczyk, J.; Kawalec, A. Specific emitter identification based on graphical representation of the distribution of radar signal parameters. *Bull. Pol. Acad. Sci. Tech. Sci.* **2015**, *63*, 391–396.
5. Dudczyk, J.; Kawalec, A. Fast-decision identification algorithm of emission source pattern in database. *Bull. Pol. Acad. Sci. Tech. Sci.* **2015**, *63*, 385–389.

6. Dudczyk, J. A method of feature selection in the aspect of specific identification of radar signals. *Bull. Pol. Acad. Sci. Tech. Sci.* **2017**, *65*, 113–119.
7. Dudczyk, J. Radar Emission Sources Identification Based on Hierarchical Agglomerative Clustering for Large Data Sets. *J. Sens.* **2016**, *2016*, 1879327.
8. Nandi, A.K.; Azzouz, E.E. Algorithms for automatic modulation recognition of communication signals. *IEEE Trans. Commun.* **1998**, *46*, 431–436.
9. Wong, M.D.; Nandi, A.K. Automatic digital modulation recognition using artificial neural network and genetic algorithm. *Signal Process.* **2004**, *84*, 351–365.
10. López-Risueño, G.; Grajal, J.; Sanz-Osorio, Á. Digital channelized receiver based on time-frequency analysis for signal interception. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 879–898.
11. Cohen, L. Time-frequency distributions-a review. *Proc. IEEE* **1989**, *77*, 941–981.
12. Lopez-Risueno, G.; Grajal, J.; Yeste-Ojeda, O. Atomic decomposition-based radar complex signal interception. *IEE Proc. Radar Sonar Navig.* **2003**, *150*, 323–331.
13. Lundén, J.; Koivunen, V. Automatic radar waveform recognition. *IEEE J. Sel. Top. Signal Process.* **2007**, *1*, 124–136.
14. Ming, Z.; Lutao, L.; Ming, D. LPI Radar Waveform Recognition Based on Time-Frequency Distribution. *Sensors* **2016**, *16*, 1682.
15. Ma, J.; Huang, G.; Zuo, W.; Wu, X.; Gao, J. Robust radar waveform recognition algorithm based on random projections and sparse classification. *IET Radar Sonar Navig.* **2013**, *8*, 290–296.
16. Lawrence, S.; Giles, C.L.; Tsoi, A.C.; Back, A.D. Face recognition: A convolutional neural-network approach. *IEEE Trans. Neural Netw.* **1997**, *8*, 98–113.
17. LeCun, Y.; Huang, F.J.; Bottou, L. Learning methods for generic object recognition with invariance to pose and lighting. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004.
18. Sainath, T.N.; Kingsbury, B.; Saon, G.; Soltau, H.; Mohamed, A.R.; Dahl, G.; Ramabhadran, B. Deep convolutional neural networks for large-scale speech tasks. *Neural Netw.* **2015**, *64*, 39–48.
19. Yoshioka, T.; Karita, S.; Nakatani, T. Far-field speech recognition using CNN-DNN-HMM with convolution in time. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia, 19–24 April 2015; pp. 4360–4364.
20. Swietojanski, P.; Arnab, G.; Steve, R. Convolutional neural networks for distant speech recognition. *IEEE Signal Process. Lett.* **2014**, *21*, 1120–1124.
21. Abdel-Hamid, O.; Mohamed, A.R.; Jiang, H.; Penn, G. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In Proceedings of the Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 4277–4280.
22. Lin, T.Y.; RoyChowdhury, A.; Maji, S. Bilinear CNN models for fine-grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 2015; pp. 1449–1457.
23. Vaddi, R.S.; Boggavarapu, L.N.P.; Vankayalapati, H.D.; Anne, K.R. Comparative analysis of contrast enhancement techniques between histogram equalization and CNN. In Proceedings of the 2011 Third International Conference on Advanced Computing, Chennai, India, 14–16 December 2011; pp. 106–110.
24. Zhou, M.K.; Zhang, X.Y.; Yin, F.; Liu, C.L. Discriminative quadratic feature learning for handwritten Chinese character recognition. *Pattern Recognit.* **2016**, *49*, 7–18.
25. Akhand, M.A.H.; Rahman, M.M.; Shill, P.C.; Islam, S.; Rahman, M.H. Bangla handwritten numeral recognition using convolutional neural network. In Proceedings of the 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 21–23 May 2015; pp. 1–5.
26. Niu, X.X.; Suen, C.Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognit.* **2012**, *45*, 1318–1325.
27. Abdel-Hamid, O.; Mohamed, A.R.; Jiang, H.; Deng, L.; Penn, G.; Yu, D. Convolutional neural networks for speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 1533–1545.
28. Feng, Z.; Liang, M.; Chu, F. Recent advances in time–frequency analysis methods for machinery fault diagnosis: A review with application examples. *Mech. Syst. Signal Process.* **2013**, *38*, 165–205.

29. Chen, B.; Shu, H.; Zhang, H.; Chen, G.; Toumoulin, C.; Dillenseger, J.L.; Luo, L.M. Quaternion Zernike moments and their invariants for color image analysis and object recognition. *Signal Process.* **2012**, *92*, 308–318.
30. Xu, B.; Sun, L.; Xu, L.; Xu, G. Improvement of the Hilbert method via ESPRIT for detecting rotor fault in induction motors at low slip. *IEEE Trans. Energy Convers.* **2013**, *28*, 225–233.
31. Pace, P.E. *Detecting and Classifying Low Probability of Intercept Radar*, 2nd ed.; Artech House: Norwood, MA, USA, 2009.
32. Gonzalez, R.C. *Digital Image Processing*; Pearson Education: Delhi, India, 2009.
33. Zhu, Z.; Aslam, M.W.; Nandi, A.K. Genetic algorithm optimized distribution sampling test for M-QAM modulation classification. *Signal Process.* **2014**, *94*, 264–277.
34. Ozen, A.; Ozturk, C. A novel modulation recognition technique based on artificial bee colony algorithm in the presence of multipath fading channels. In Proceedings of the 2013 36th International Conference on Telecommunications and Signal Processing (TSP), Rome, Italy, 2–4 July 2013; pp. 239–243.
35. Stoica, P.; Moses, R.L. *Spectral Analysis of Signals*; Pearson/Prentice Hall Upper: Saddle River, NJ, USA, 2005.
36. Bailey, R.R.; Srinath, M. Orthogonal moment features for use with parametric and non-parametric classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 389–399.
37. Chen, Z.; Sun, S.K. A Zernike moment phase-based descriptor for local image representation and matching. *IEEE Trans. Image Process.* **2010**, *19*, 205–219.
38. Tan, C.W.; Kumar, A. Accurate iris recognition at a distance using stabilized iris encoding and Zernike moments phase features. *IEEE Trans. Image Process.* **2014**, *23*, 3962–3974.
39. Honarvar, B.; Paramesran, R.; Lim, C.L. Image reconstruction from a complete set of geometric and complex moments. *Signal Process.* **2014**, *98*, 224–232.
40. Lauer, F.; Suen, C.Y.; Bloch, G. A trainable feature extractor for handwritten digit recognition. *Pattern Recognit.* **2007**, *40*, 1816–1824.
41. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
42. Johnson, A.E.; Ghassemi, M.M.; Nemati, S.; Niehaus, K.E.; Clifton, D.; Clifford, G.D. Machine learning and decision support in critical care. *Proc. IEEE* **2016**, *104*, 444–466.
43. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828.
44. Lin, W.M.; Hong, C.M.; Chen, C.H. Neural-network-based MPPT control of a stand-alone hybrid power generation system. *IEEE Trans. Power Electron.* **2011**, *26*, 3571–3581.
45. Lin, C.M.; Boldbaatar, E.-A. Autolanding control using recurrent wavelet Elman neural network. *IEEE Trans. Syst. Man Cybern.-Syst.* **2015**, *45*, 1281–1291.
46. Yin, S.; Yang, H.; Gao, H.; Qiu, J.; Kaynak, O. An adaptive NN-based approach for fault-tolerant control of nonlinear time-varying delay systems with unmodeled dynamics. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *PP*, 1–12.
47. Sheela, K.G.; Deepa, S. Review on methods to fix number of hidden neurons in neural networks. *Math. Probl. Eng.* **2013**, *2013*, 425740.