Ocean Engineering 305 (2024) 117969

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

False alarm suppressing for passive underwater acoustic target detecting with computer visual techniques

Hao Yin [a,b], Chao Li [a,b,*], Haibin Wang [a,b], Fan Yin [a,b], Fan Yang [c]

[a] State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
[b] University of Chinese Academy of Sciences, Beijing 101408, China
[c] Laboratory ImViA, Universite Bourgogne Franche-Comte, Dijon 21078, France

ARTICLE INFO

Keywords:
Underwater signal detection
Computer vision
K-means
Deep learning
Bearing Time Records

ABSTRACT

Given the complexity of marine environments, the detection of underwater acoustic targets frequently relies on manual visual interpretation of the imagery displayed on monitoring screens, which limits the application of related technologies on unmanned platforms. To replace human visual observation, understanding, and reasoning within underwater unmanned equipment, this study explores an intelligent detection method for Bearing Time Records based on computer vision techniques, drawing inspiration from human visual perception in detecting targets on BTR images. Firstly, an unsupervised learning method is employed to extract the bases representing various signal patterns from BTR images, with clear physical meanings. Subsequently, we utilize both algorithm-driven and data-driven methods for automated base classification, and both methods achieve a remarkable 100% accuracy rate in automatic base classification. In comparison to human visual detection, this automated approach exhibits a false alarm rate of less than 3%.

1. Introduction

Underwater acoustic information sensing techniques are challenged by the complex marine environment. Compared to computer algorithms, biosensing systems perceive the world more comprehensively, simply and perfectly, so classical passive underwater acoustic detecting modalities are usually based on the manual decisions of humans. Meanwhile, Autonomous Underwater Vehicles (AUVs) can capture environment information and perform action decisions autonomously, being increasingly widely used to replace humans to complete dangerous missions in extreme ocean environments with low temperatures and high pressures, sparking disruptive and evolutionary progresses in the fields of ocean engineering and military (Department of the navy, 2021a,b). The sophisticated platforms of this type cannot offer enough power, hardware, communication and space resources for manual human interventions any more, forcing us to have to make changes to improve the intelligence of underwater acoustic sensing systems.

Bearing Time Records (BTRs) visualize hydrophone array information by mapping them as a time-space image. Well-trained and experienced operators are used to detecting passive targets by observing it for its advantages of high sensitivity and reliability. Once a trajectory is found in BTR, we can considered that a target locates at the corresponding azimuth and time. Yet, automating this process is far from easy, especially when both the high sensitivity and low false alarm rate are desired. The most rudimentary means to realize it is to detect the energy peaks of the beamforming over targets azimuths. Energy peaks are a type of very important visual pattern on BTRs (Chenhui, 2003; Zheng et al., 2005). It is easy to understand that no matter how low a peak is, it represents a potential target on the screen. The limits of detecting capability can be therefore considered to correspond to the data precision used or the sensitivity of hydrophones. However, underwater environments are often filled with a variety of complex noises, and the beamforming method used may possess side or gating lobe, resulting in false alarms. Even if the false alarm rate is very low, providing AUVs with the wrong information for decision may raise destructive risks, an artificial intelligence system that can replace operators to make reliable posterior decisions is therefore strongly desired.

Up to our knowledges, many efforts have been made to facilitate BTR target detecting tasks. At present, the solution strategy of facilitate BTR target detecting tasks is mainly divided into two categories: target enhancement and decision analysis. The former enhance the BTR trajectories via certain noise filter or normalizer, such as the two-pass mean, the split three-pass mean, the split average exclude average, and the order truncate average normalizers (Struzinski and Lowe, 1984). Shapiro and Green (2000) normalize the spectrogram background of narrowband passive acoustic detection system via the

* Corresponding author at: State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China.
E-mail addresses: yinhao@mail.ioa.ac.cn (H. Yin), chao.li@mail.ioa.ac.cn (C. Li).

https://doi.org/10.1016/j.oceaneng.2024.117969
Received 28 January 2024; Received in revised form 17 April 2024; Accepted 20 April 2024
Available online 1 May 2024

split-window three-pass mean noise spectral estimator and improve it by approximating its bias for the colored-Gaussian-noise case. Carbone and Kay (2012) propose a 3D minimum variance spectral estimating normalizer, and report that it is from 3 dB to over 5 dB more effective than the two-pass split-window normalizer in initializing active sonar tracks depending on the circumstances. Lei et al. (2016) generate the BTR display by using the L1 norm regularization and the total variation regularization to enhance the point-based feature and the region-based feature respectively. Yang (2018b,a) improves the beamforming resolution and mitigates the sidelobes by deconvolving the beam power instead of the complex beam output. Yin et al. (2023) denoising the BTRs by combining the SEED and the BM3D algorithms. Researches demonstrate that these methods can enhance the target trajectories effectively. However, from the algorithmic mechanism, it leads to the following two issues:

(a) Part of the normalizing algorithms are essentially low-pass filter that necessarily loses high-frequency information. The weak energy peaks representing distant or low-noise targets will most likely be ignored.
(b) There is also part of the enhancing algorithms highlight the trajectories via wavelet transforms or feature extractions. Although they can suppress noise while retaining a certain amount of high frequency details, the side and gating lobes are also enhanced so that new false alarms may be caused.

The second category decision analysis method perform detecting and tracking decisions via some tracking before detect method. For example, Saucan et al. (2014) propose a particle filtering algorithm for tracking the direction of arrival (DOA) of multiple echoes impinging on a sonar array. The impulsive nature of the backscattered signals is taken into account and a robust multivariate Laplace distribution is developed. Xin et al. (2015) propose a method of track before detect algorithm based on Hidden Markov Model for BTR weak trajectory detecting and tracking. It is claimed that a gain of 3 dB is obtained (Xin and Luo, 2016). Fan et al. guide the locating information processing of BTRs depending on target motion analyzing results (Yin et al., 2019, 2022). The main issue of this type of approach is to necessitate auxiliary points, which brings us back to the dilemma of energy peak extracting.

Practical applications have shown that bio-vision can better handle the above problems. Human brains can easily complete the missing trajectory information in BTRs or ignore complex interference information based on experience and knowledge. Replicating this decision process in machines in a bionic way is a potential solution to realize high reliable autonomous passive underwater detecting. The main challenge is how to describe the visual patterns of true and false target trajectories programmatically. This paper solve it by using an unsupervised learning method. A priori information is analyzed automatically by using unsupervised learning to establish statistical features of targets in real time, and then they are used to guide to select out the false alarms. An important innovation of this method is that the false-alarm suppressing problem is handled as a binary classification problem after detecting, all the issues in conventional approaches therefore could be circumvented. The sea-trail data evaluation result demonstrates that this is an effective method to improve the reliability of the automatic BTR target detecting techniques.

The remainder of this paper is organized as follows: Section 2 introduces Mechanism of Related Works; Section 3 presents the proposed false alarm suppressing method in detail; Section 4 analyzes the evaluation experiment results; finally, Section 5 gives the final conclusion of this work.

2. Mechanism of related works

Generally speaking, the post-processing of BTRs based on biosensor systems could divided into three steps: color perception, element perceptual grouping and decision.

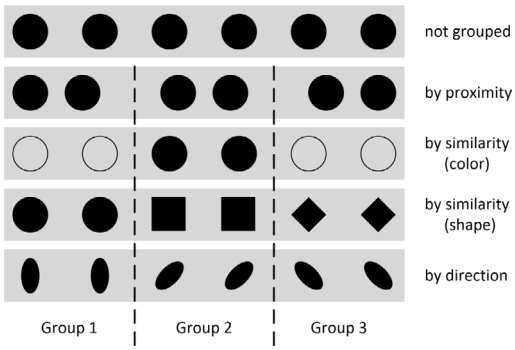

Fig. 1. Element perceptual grouping.

2.1. BTR color perception

There are three types of biosensors in the human visual system than can perceive light weights of different wavelengths, namely trichromacy. The main role in visual BTR observation is played by the cones on the retina, which are categorized into three types depending on the sensed wavelength range: S (short) with maximum sensitivity at around 430 nm, M (medium) at around 560 nm, and L (long) at around 610 nm. The spectral response q_i of the ith sensor can be modeled by integration over a certain range of wavelengths (Milan et al., 2015)

$$q_i = \int_{\lambda_1}^{\lambda_2} I(\lambda) R_i(\lambda) S(\lambda)\, d\lambda, \tag{1}$$

where $R_i(\lambda)$ is the spectral sensitivity of the sensor, $I(\lambda)$ is the spectral density of the illumination, and $S(\lambda)$ describe how the surface patch reflects each wavelength of the illuminating light. The electromagnetic wavelength section visible to humans is from approximately 380 nm to 740 nm, we therefore can visualize the underwater acoustic information by representing the beamforming power via a color space such as RGB (Red, Green and Blue) and HSV (Hue, Saturation and Value).

2.2. Element perceptual grouping

Perceptual grouping is a computer vision principal to aggregate elements provided by low-level operations, which are small blobs to bigger chunks having some meaning (Palmer, 1999). Observing BTRs in small-scale neighborhoods yield basic image elements for high-level understanding. The human ability to group items according to various properties is illustrated in Fig. 1. The roots of this theory are in Gestalt psychology first proposed by Wertheimer in 1912 (King and Wertheimer, 2005). Correct element grouping would constitute new functional units with properties not derivable by summation of its parts, namely patterns take precedence over elements and have properties that are not inherent in the elements themselves.

Underwater acoustic targets appear as typical stripe features on the BTR, which is the functional unit constituted by similar successive striped elements with directions varying continually. Perceived element properties help us to connect them together so yield new properties in a larger scale range such as parallelism, symmetry and continuity as illustrated in Fig. 2.

The functional units and their properties provide us necessary basis for making decisions. Short stripes probably are caused by random underwater noises so could be considered as false alarms. Two parallel long stripes are most likely primary and side lobes, so the weaker one could be ignored. A weak trajectory can be tracked robustly even if it is faintly visible, because the brains would automatically generate and fill in the missing parts.

Building upon color perception and element perceptual grouping, the brain makes judgments to determine whether what is observed is

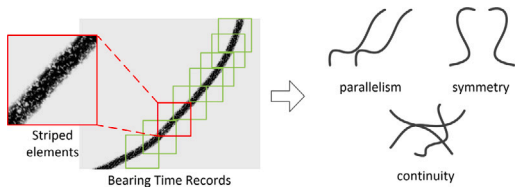

Fig. 2. BTR trajectory element grouping.

a genuine target or a false alarm. These decisions are based on past experiences, the current environmental context, and the brain's holistic understanding of the image.

2.3. Functional unit constitutions of BTR targets

The keys of the proposed false alarm suppressing method is to build proper functional units with the properties that can distinguish true and false alarms. BTR target patterns possess obvious stripe properties, which are described simply by the width w, the length l and the direction β in a double-dimension space. In the case of this paper, the spatial coordinates of the image correspond to the azimuths and time, respectively, so we can easily build up their relationships with the underlying physical variables. w is essentially the primary-lobe width of direction arrival estimations, l is the length of observation time, and β is the azimuth variation direction of targets.

According to the classical beamforming theory, trajectories in BTR image can be modeled mathematically. Let us take a uniform linear array (ULA) as an example, its beamforming signal $s(t)$ is

$$\begin{aligned} s(t) &= \sum_{i=1}^{N} s_i(t) \\ &= \sum_{i=1}^{N} A\cos[2\pi f t + (i-1)\phi] \\ &= A\cos[2\pi f t + \frac{N-1}{2}\phi]\frac{\sin\frac{N}{2}\phi}{\sin\frac{1}{2}\phi}, \end{aligned} \tag{2}$$

with

$$\phi = 2\pi\frac{d}{\lambda}\sin\theta, \tag{3}$$

where $s_i(t)$ is the signal received by the ith array unit, A is the signal amplitude, f is the signal frequency, t is the time, N is the array unit number, d is the array unit interval, λ is the signal wavelength and θ is the independent variable of azimuth.

The directivity function $D(\theta)$ of Conventional Beamforming (CBF) is therefore given by

$$D_{CBF}(\theta) = A\left|\frac{\sin(N\pi\frac{d}{\lambda}\sin\theta)}{N\sin(\pi\frac{d}{\lambda}\sin\theta)}\right| \tag{4}$$

(4) describes the directivity of ULAs oriented to $\theta_0 = 0°$. If other look-directions are desired, a delay of $\tau_i(\theta_0) = 2\pi(i1)\frac{d}{\lambda}\sin(\theta_0)$ will occur for the ith array unit, and the directivity function of CBF becomes:

$$D_{CBF}(\theta) = A_0\left|\frac{\sin[N\pi\frac{d}{\lambda}(\sin\theta-\sin\theta_0)]}{N\sin[\pi\frac{d}{\lambda}(\sin\theta-\sin\theta_0)]}\right| \tag{5}$$

The directivity function $D(\theta)$ of the MVDR (Minimum Variance Distortionless Response) beamformer algorithm depends on the covariance matrix of the received signals. The weight vector $\mathbf{W}_{MVDR}(\theta_0)$ and the array manifold vector $\mathbf{a}(\theta)$ for ULAs are defined as follows:

$$\mathbf{w}_{MVDR}(\theta_0) = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta_0)}{\mathbf{a}^H(\theta_0)\mathbf{R}^{-1}\mathbf{a}(\theta_0)} \tag{6}$$

$$\mathbf{a}(\theta) = [1, e^{j2\pi\frac{d}{\lambda}\sin\theta}, e^{j2\pi 2\frac{d}{\lambda}\sin\theta}, \ldots, e^{j2\pi(N-1)\frac{d}{\lambda}\sin\theta}]^T \tag{7}$$

where R is the covariance matrix of the received signals, and $\mathbf{a}(\theta)$ is the array manifold vector for the desired signal direction θ. The directivity function $D_{MVDR}(\theta)$ of MVDR beamformer algorithm is therefore given as:

$$D_{MVDR}(\theta) = \left|\mathbf{w}_{MVDR}^H\mathbf{a}(\theta)\right| \tag{8}$$

$$\mathbf{a}(\theta,\tau) = [1, e^{j2\pi\frac{d}{\lambda}(\sin\theta-\sin\theta_0+\tau d)}, \ldots, e^{j2\pi(N-1)\frac{d}{\lambda}(\sin\theta-\sin\theta_0+\tau d)}]^T \tag{9}$$

The definition of directivity functions demonstrate that its square is the power of beamforming at t, which is BTR's pixel value, so we can model the functional units of ULAs for single-target scenarios as:

$$\begin{aligned} F_{CBF}(\vartheta,\tau) &= |D_{CBF}(\vartheta,\tau)|^2 \\ &= \left|A_0\frac{\sin[N\pi\frac{d}{\lambda}(\sin(\theta-\vartheta)-\sin\theta_0(t-\tau))]}{N\sin[\pi\frac{d}{\lambda}(\sin(\theta-\vartheta)-\sin\theta_0(t-\tau))]}\right|^2. \end{aligned} \tag{10}$$

$$\begin{aligned} F_{MVDR}(\vartheta,\tau) &= |D_{MVDR}(\vartheta,\tau)|^2 \\ &= \left|\left(\frac{\mathbf{R}^{-1}\mathbf{a}(\theta_0(t-\tau))}{\mathbf{a}^H(\theta_0(t-\tau))\mathbf{R}^{-1}\mathbf{a}(\theta_0(t-\tau))}\right)^H \right. \\ &\quad \left. \times\left[1, \quad \ldots, \quad e^{j2\pi(N-1)\frac{d}{\lambda}\sin(\theta-\vartheta)}\right]^T\right|^2 \end{aligned} \tag{11}$$

(10) and (11) considers the functional unit as a region of interest (ROI) of BTRs at $\langle\theta,t\rangle$.

Similarly, the functional units of other array shapes that has regular directivity functions, such as circular and arc arrays, can be obtained. According to the above beamforming equations, a connection is established between the BTR image and the beamforming, allowing us to relate the underlying physical variables to the BTR image.

3. Method

The overall framework of the proposed method is shown in Fig. 3. We conduct peak extraction on the original BTR image. Subsequently, elements are constructed, followed by the construction of bases. Finally, we applied automated base template selection, resulting in target detection outcomes after false alarm suppressing. In this paper, an "element" refers to a collection of pixels observed in the actual BTR image, while a "base" serves as an abstraction of these elements, aiming to represent their fundamental characteristics. Specifically, within the context of the beamforming algorithms mentioned in (10) and (11), a "base" can be regarded as a "functional unit".

3.1. Base extraction using unsupervised learning

3.1.1. Open peak extraction

Color resolution plays a crucial role in image processing and computer vision. However, human color perception has inherent limitations. Historically, our conventional approach to peak extraction often involved the use of fixed thresholds, which lacks adaptability. To overcome the constraints imposed by human color resolution, it is imperative to use open peak extraction method that fully utilize the power of computers in order to surpass the limitations of human color resolution.

The subject of this paper is to suppress the false alarms of BTRs with high sensitivity. Our idea is to realize it via distinguishing after detecting. Briefly, energy peaks are first extracted with loose constraints:

$$p(\theta_i) = \begin{cases} true, & B(\theta_{i-1}) < B(\theta_i) < B(\theta_{i+1}), \\ false, & \text{otherwise}, \end{cases} \tag{12}$$

where $p(\theta_i)$ is the logical detecting result at the observation direction θ_i, whose value is $true$ if an energy peak exists, otherwise $false$. $B(\theta_i)$ is the beamforming power at θ_i. Next, false alarms are selected out via some strategy which mimics human vision systems.

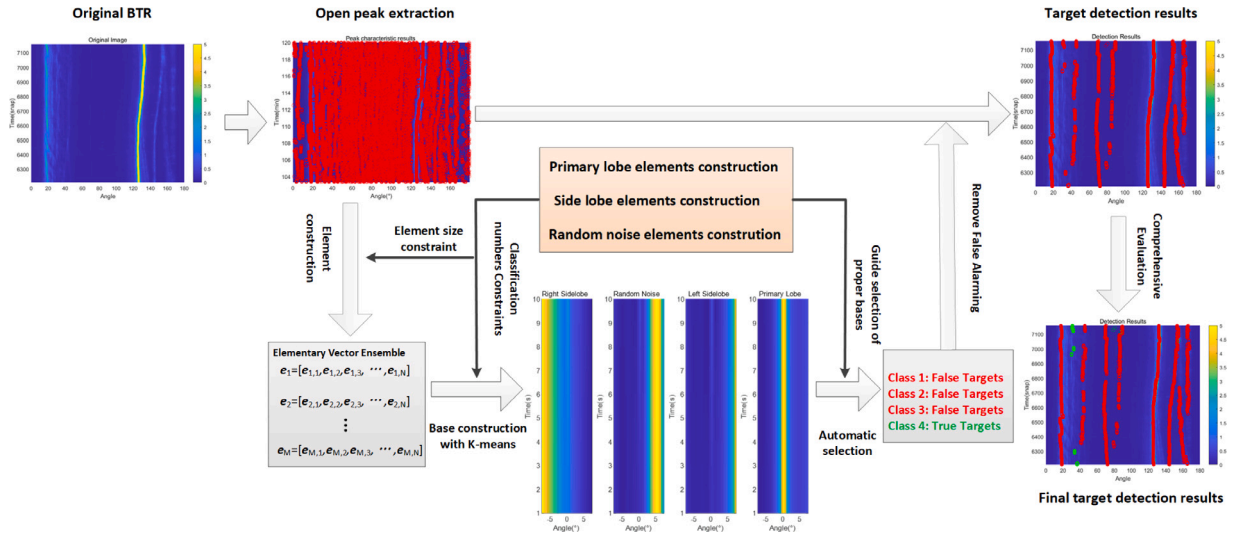

Fig. 3. Technical method.

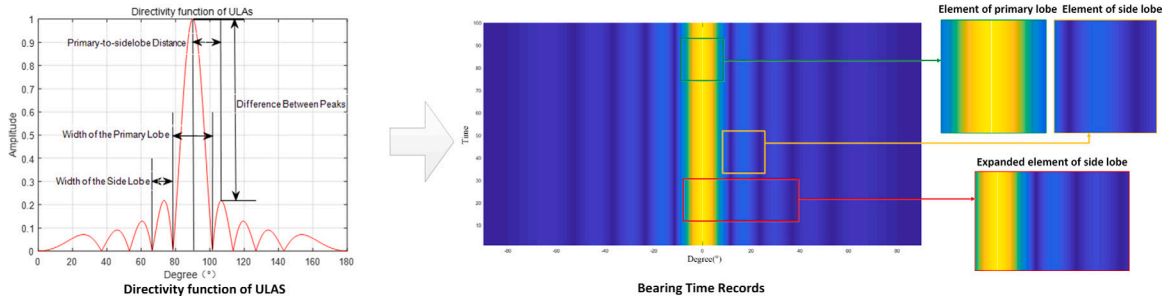

Fig. 4. Element size and beamforming algorithms.

When performing open peak extraction, it is necessary to set a crucial parameter to define the width of the main lobe in beamforming algorithms, which is referred to as the "minimum peak separation", which dictates the minimum horizontal distance between peaks in each row of the image data. In essence, it ensures that each detected peak is separated by at least the predetermined "minimum peak separation" number of pixels from its nearest neighbor. From the perspective of beamforming, this can be regarded as the width of the main lobe, indicating the minimum resolution that the primary energy concentration region should maintain.

For the CBF method, W_{main} is defined as twice the position of the first zero-crossing point of the directivity function $D(\theta)$:

$$W_{main} = 2\arcsin\left(\frac{\lambda}{Nd}\right) \tag{13}$$

((13)) describes the primary lobe width W_{main}.

For the MVDR beamforming algorithm, which relies on the signal-to-noise ratio (SNR), a fuzzy recognition approach can be employed to define fuzzy sets for the primary lobe width W_{main}.

3.1.2. Element construction

After extracting peaks using open peak extraction methods, each of these peaks represents valuable physical information. The challenge lies in how to effectively construct visual units from these peaks. Visual units serve as essential tools for establishing associations between peaks within images and their underlying physical meaning, which are called as "elements". To do this, it is essential to select the length of the element to minimize the interference of random noise. Random noise typically appears as isolated, discontinuous points of low intensity rather than continuous segments of a signal. Therefore, it is common to select a length based on empirical values to avoid mistaking isolated, discontinuous noise points for actual targets.

As shown in Fig. 4, the target trajectory representation exhibits continuous portions with high intensity, while the sidelobe representation is typically characterized by lower but continuous intensity portions on either side of the primary lobe. The difference in signal intensity between the target trajectory element and the sidelobe element is reflected in the peak differences. To minimize sidelobe interference, the selection of the element width should be equal to twice the primary-to-sidelobe distance.

For the CBF method, to calculate the primary-to-sidelobe distance, we can find the position of the maximum value of the first sidelobe, it is necessary to differentiate $D(\theta)$ with respect to θ, neglecting the absolute value and the constant factor A.

$$\frac{d}{d\theta}\left(\frac{\sin(N\pi d/\lambda\sin(\theta))}{N\sin(\pi d/\lambda\sin(\theta))}\right) = \frac{\pi d\left(N\cos\left(\frac{\pi N d\sin(\theta)}{\lambda}\right) - \frac{\sin\left(\frac{\pi N d\sin(\theta)}{\lambda}\right)}{\tan\left(\frac{\pi d\sin(\theta)}{\lambda}\right)}\right)\cos(\theta)}{N\lambda\sin\left(\frac{\pi d\sin(\theta)}{\lambda}\right)} \tag{14}$$

(14) describes the derivative equation of the directivity function for the CBF algorithm.

The position of the first sidelobe's maximum value is the first positive root of this equation, which needs specific parameters and numerical methods to solve. Let us denote the position of the maximum value of the first sidelobe obtained through numerical methods as D_{side}, which is called as the primary-to-sidelobe distance.

The MVDR beamforming algorithm relies on the SNR, a fuzzy recognition approach can be employed to define fuzzy sets for D_{side}. Through

the application of fuzzy logic rules and defuzzification methods, the optimal element width can be determined. This method guarantees that the selected element width accurately adapts to the characteristics of both the primary lobe and sidelobe.

The width of element $\Delta\theta$ is then defined as:

$$\Delta\theta = 2D_{side}$$

3.1.3. Base construction

In typical cases, the primary lobe of targets is usually described in BTR images as a series of continuous and relatively distinct patterns. Conversely, sidelobes tend to emerge on both sides of the primary lobe, presenting as continuous yet slightly lower-intensity features. On the other hand, noise may shown as dispersed or faint, discontinuous patterns. To accurately extract and classify these different types of signal patterns in BTR images, we plan to employ a clustering approach, which falls under the domain of unsupervised learning and involves grouping data points into similar clusters to identify various signal patterns. K-means is a widely employed and robust unsupervised learning method. It commonly utilizes the Euclidean distance as a kernel function to measure the dissimilarity among samples. In the context of the current situation, we intend to use the Pearson correlation coefficient to measure the similarity among various patterns within the primary lobe, sidelobes, and random noise in BTR images, which better captures the morphological similarities between different patterns, rather than solely focusing on their absolute positional information (Zhao et al., 2011).

To begin with, we employ primitive dimensions to extract fundamental elements from the identified peaks in the BTR images. This process leads to the formation of a primitive set, represented as $E = e_1, e_2, \ldots, e_m$.

(1) Initialization

We randomly select n elements from the element set $E = e_1, e_2, \ldots, e_m$ as the initial cluster centers, where n represents the predefined number of clusters.

$$C^{(0)} = \{c_1^{(0)}, c_2^{(0)}, \ldots, c_n^{(0)}\} \tag{15}$$

where $c_n^{(0)}$ is the cluster center.

(2) Element assignment

For each element e_i in the element set, calculate its Pearson correlation coefficient $r(e_i, c_j^{(t)})$ with each cluster center $c_j^{(t)}$. Assign element e_i to the cluster center with the highest correlation coefficient.

$$r(e_i, c_j^{(t)}) = \frac{\sum_{k=1}^{N}(e_i^{(k)} - \bar{e}_i)(c_j^{(t,k)} - \bar{c}_j^{(t,k)})}{\sqrt{\sum_{k=1}^{N}(e_i^{(k)} - \bar{e}_i)^2 \sum_{k=1}^{N}(c_j^{(t,k)} - \bar{c}_j^{(t)})^2}} \tag{16}$$

(16) describes the computation of the Pearson correlation coefficient. Where $e_i^{(k)}$ represents the k-th feature value of element sample e_i, $c_j^{(t,k)}$ is the k-th feature value of cluster center $c_j^{(t)}$, $\bar{e}_i$ is the mean of element sample e_i, and $\bar{c}_j^{(t)}$ is the mean of cluster center $c_j^{(t)}$. This correlation coefficient measures the similarity between element samples and cluster centers, enabling element samples to be assigned to the cluster center with the highest correlation coefficient.

$$y_i^{(t)} = \arg\max_j r(e_i, c_j^{(t)}) \tag{17}$$

where $y_i^{(t)}$ is sample label.

(3) Update cluster centers

For each cluster j, its new center is computed as the average of all data points assigned to it.

$$c_j^{(t+1)} = \frac{1}{|S_j|} \sum_{i:y_i^{(t)}=j} d_i \tag{18}$$

where S_j represents the set of element samples assigned to cluster j, and $|S_j|$ is the number of elements assigned to cluster j.

(4) Convergence check

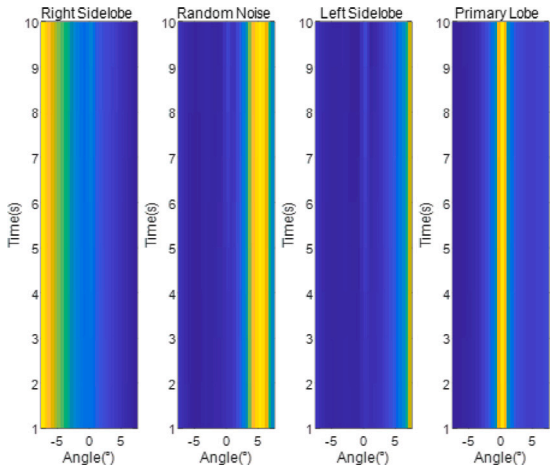

Fig. 5. Four base templates.

Algorithm convergence is evaluated by examining changes in cluster centers across successive iterations.

$$\forall j : \|c_j^{(t+1)} - c_j^{(t)}\| < \varepsilon \tag{19}$$

(19) describes the convergence criteria of this algorithm, where $c_j^{(t)}$ and $c_j^{(t+1)}$ denote the pattern centers after the tth and (t+1)-th iterations, respectively, ε represents a local minimum value. If the new cluster centers are the same as or change very little compared to the previous ones, the algorithm converges; otherwise, return to step 2.

(5) Output results

After completing the clustering process, we acquire cluster labels $y_i^{(t)}$ for each individual sample element, as well as the ultimate cluster centroids $C(t)$. These cluster centroids can be employed to characterize the attributes of the cluster.

In theory, cluster centers should include four distinct classes: the true target (the primary lobe), the left side lobe, the right side lobe, and random noise. Due to the diverse characteristics of random noise, it is also possible to consider further dividing it into more classes, with the simplest situation involving four classes. As shown in Fig. 5, each of these four base classes reflects different types of target clustering characteristics, namely the right side lobe, random noise, the left side lobe, and the primary lobe. Each base shows distinct visual features that correspond to their underlying beamforming theory. Table 1 provides a detailed explanation of the visual characteristics of each base with their beamforming theory.

3.2. Base classification

Upon the completion of visual unit construction, each type of visual unit corresponds to a distinct real-world physical concept. This raises a question: How to automatically select the specific visual units required for a given task? To achieve the goal of automated base selection, two methods are proposed in this paper: one driven by algorithms, and the other driven by data, employing neural networks to learn base features. After getting the four bases, the next step is to select the base which best matches the primary lobe of the target. This automated selection not only improve efficiency but also guarantees that the selected visual units are contextually relevant to the physical aspects of the task. As a result, it facilitates a more precise interpretation and analysis of the information contained within the image. One essential assumption that needs to be emphasized is that among the four classes of base templates, only one class of base template represents the true target, while the other three types of base templates represent false alarms.

Table 1
Visual descriptions of four base templates.

	Visual characteristics	Beamforming mechanism
Right Sidelobe	Prominent bright yellow area on the left edge of images. Darker blue tones along the central line and right edge.	Bright yellow line represents the primary lobe of the beam, with sidelobes to the right.
Random Noise	Predominantly blue color in images.	Lack of directional signal enhancement.
Left Sidelobe	Prominent bright yellow area on the right edge of images. Darker blue tones along the central line and right edge.	Bright yellow line represents the primary lobe of the beam, with sidelobes to the right.
Primary Lobe	Bright yellow central line with significantly reduced energy on both sides.	Bright yellow line near the central axis signifies the primary lobe of the beam.

3.2.1. Algorithm-driven base automatic classification

Usually, the base of the primary lobe of the target shows higher intensity in the central region and lower intensity on both sides, a base automatic selection method is therefore proposed. It relies on the measurement of similarity between the centerline and the sides. More precisely, this similarity is calculated by analyzing the central line and the average values on both sides of the base, thereby automatically selecting the most representative base.

(1) Define the centerline and the two sides of the base template:

$$\begin{aligned} \bar{l} &= \frac{1}{m}\sum_{i=1}^{m} l(i) \\ \bar{L} &= \frac{1}{m(\frac{n}{2}-1)}\sum_{i=1}^{m}\sum_{j=1}^{\frac{n}{2}-1} L(i,j) \\ \bar{R} &= \frac{1}{m(\frac{n}{2})}\sum_{i=1}^{m}\sum_{j=\frac{n}{2}+1}^{\mathrm{n}} R(i,j) \end{aligned} \tag{20}$$

where $\bar{l}$, $\bar{L}$, and $\bar{R}$ denote the average values for the template's centerline, left region, and right region.

(2) Measure the similarity between the centerline and its two sides:

$$s(i) = \mid \bar{l} - \frac{\bar{L}+\bar{R}}{2} \mid \tag{21}$$

(3) Identify the highest similarity measure and select the base which corresponds to it:

$$S_{max} = \max_i s(i) \tag{22}$$

3.2.2. Data-driven base automatic classification

This paper implements three distinct neural network models for base classification: Artificial Neural Network (ANN), Deep Neural Network (DNN), and Convolutional Neural Network (CNN). Following this, we will perform a comparative evaluation of their respective performance.

(a) ANN
The Artificial Neural Network is a computational framework that emulates the functions of biological neural networks, primarily comprising an input layer, a hidden layer, and an output layer. Each of these layers consists of interconnected neurons, or processing units, that convey and process information through weighted connections. The input layer receives external data, the hidden layer is responsible for data processing, and the output layer provides the predictive results of the model. The training process of an ANN involves optimization algorithms (such as gradient descent) and backpropagation mechanisms, aimed at adjusting the network's weight parameters to minimize discrepancies in output.

(b) DNN
The Deep Neural Network, evolving from the architectural foundation of ANN, incorporate multiple hidden layers, thereby establishing a more profound network structure. This enhanced depth enables DNN to recognize and learn more complex data features. Specifically, DNN increase their learning and understanding capabilities of inherent data patterns and connections by adding more layers and neurons. This deep learning model excels in processing high-dimensional data and performing complex tasks, though they need greater computational resources and extensive training data.

(c) CNN
The Convolutional Neural Network is a deep learning model specifically designed for processing data with a grid structure, like images. The core characteristic of CNN is the use of convolutional layers, which apply a set of filters to the input data to effectively capture local features. The architecture of CNN typically includes multiple convolutional layers, activation functions, pooling layers (for reducing feature dimensions), and fully connected layers. This design allows CNN to efficiently extract complex spatial features from input data, while improving computational efficiency through weight sharing and feature downsampling.

The Cross Entropy Loss and Rectified Linear Unit (ReLU) are used as the loss function and activation function of the networks in this work, respectively.

(a) Weighted cross entropy loss
Cross Entropy Loss is a common cost function for binary classification problems. It measures the performance of a classification model whose output is a probability value p between 0 and 1. For a binary classification (where $y \in \{0, 1\}$),

$$L(y,p) = -y\log(p) - (1-y)\log(1-p) \tag{23}$$

(23) demonstrates that the lower the loss, the better the model's prediction accuracy.

(b) Rectified Linear Unit (ReLU)
The ReLU function is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero, which introduces non-linearity into the output of a neuron.

$$f(x) = \max(0, x) \tag{24}$$

In the architectures of ANN, DNN, and CNN, the ReLU is employed as the activation function in hidden layers to infuse essential non-linearity. In the output layer, activation functions are typically not used because they output logits for the cross-entropy loss function, which applies the softmax function for binary classification tasks.

4. Simulations

In this section, we experimentally validated the technique in SNR situations in the following simulations, respectively. The frequency of signal is 500 Hz, and the sampling frequency is 2000 Hz. The receiving array is a 128-units horizontal linear sonar array, with a spacing of 1.5 m between the array unit. The measuring signal angle ranges from 0 to 180 degrees, with a precision of 0.5 degrees.

We evaluate the performance of the proposed method within different SNR environments. The noise used in this experiment is Gaussian white noise. The target moves at a constant speed from 80 degrees to 80 degrees, and MVDR is used for beamforming. First, a set of experiments

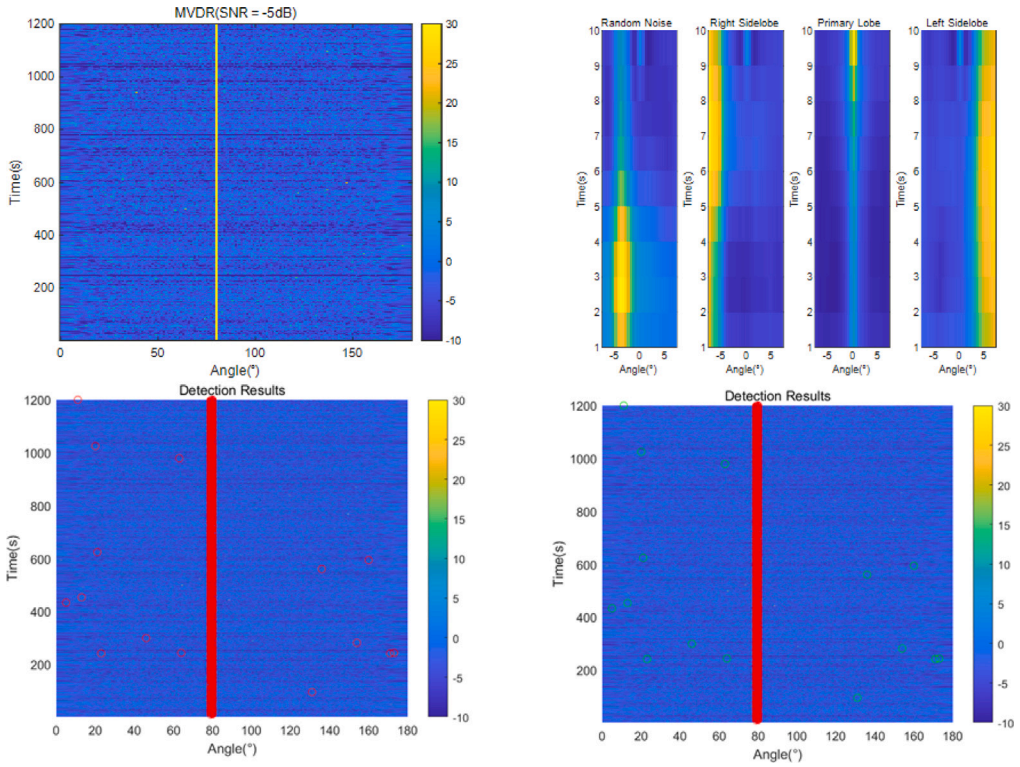

Fig. 6. SNR = −5 dB.

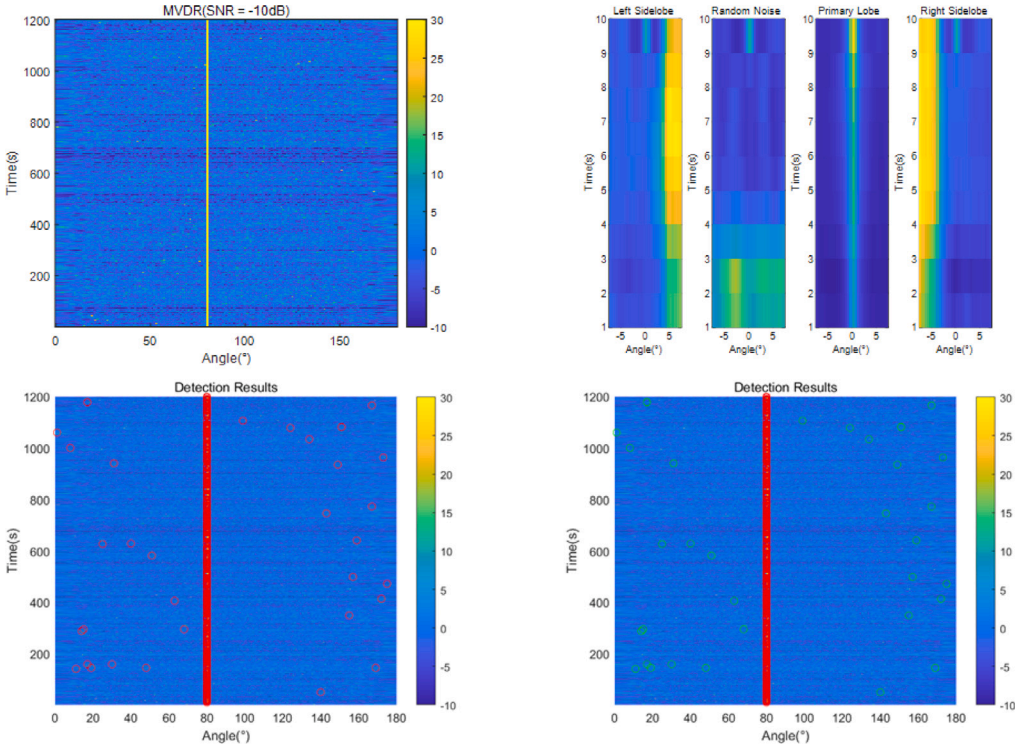

Fig. 7. SNR = −10 dB.

is required to SNR = −5 dB. According to fuzzy recognition approach, we set the primary lobe width W_{main} to 7 units, equivalent to 3.5°; and the element width $\Delta\theta$ to 31 units, equivalent to 15.5°. The four bases obtained are the Random Noise, Right Sidelobe, Primary Lobe, and Left Sidelobe, respectively. The final target detection results are shown in Fig. 6. By manually annotating false alarms through visual observation, we found 15 green circles out of a total of 1191 circles, resulting in a false alarm rate of 1.2438%, while the accuracy of target detection is 98.7562%.

Second, a set of experiments is required to SNR = −10 dB. According to fuzzy recognition approach, we set the primary lobe width W_{main} to 5 units, equivalent to 2.5°; and the element width $\Delta\theta$ to 31 units,

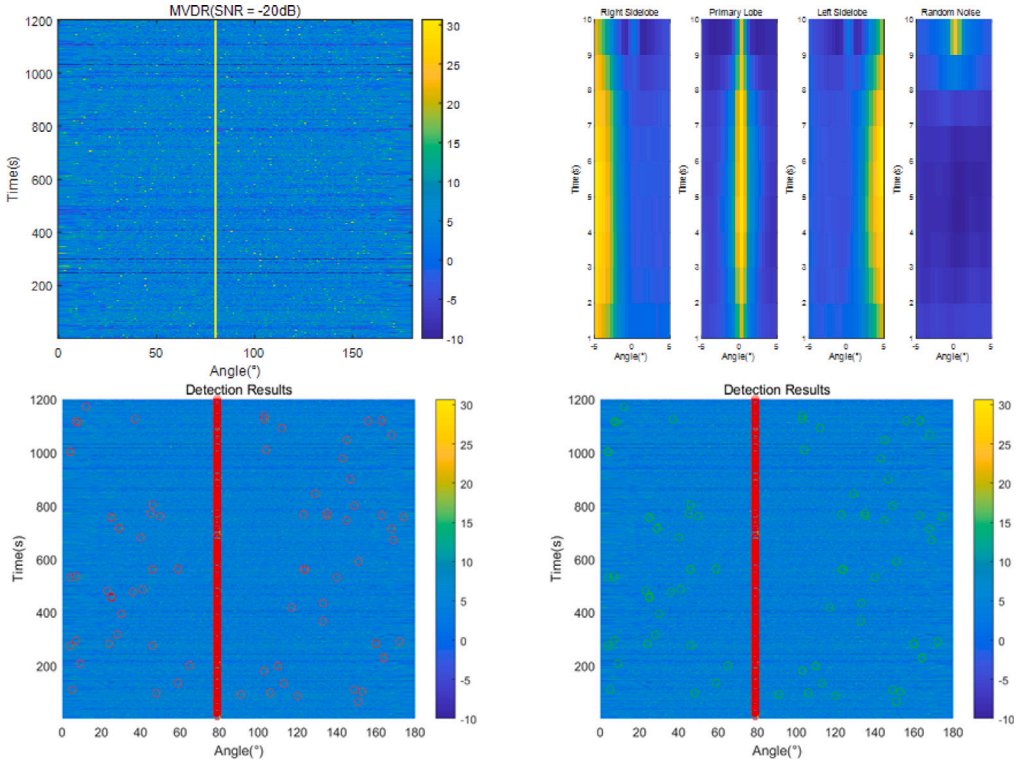

Fig. 8. SNR = −20 dB.

equivalent to 15.5°. The four bases obtained are the Right Sidelobe, Primary Lobe, Left Sidelobe and Random Noise, respectively. The final target detection results are shown in Fig. 7. By manually annotating false alarms through visual observation, we found 34 green circles out of a total of 1144 circles, resulting in a false alarm rate of 2.97%, while the accuracy of target detection is 97.03%.

Finally, a set of experiments is required to SNR = −20 dB. According to fuzzy recognition approach, we set the primary lobe width W_{main} to 3 units, equivalent to 1.5°; and the element width $\Delta\theta$ to 21 units, equivalent to 10.5°. The four bases obtained are the Left Sidelobe, Random Noise, Primary Lobe, and Right Sidelobe, respectively. The final target detection results are shown in Fig. 8. By manually annotating false alarms through visual observation, we found 81 green circles out of a total of 1184 circles, resulting in a false alarm rate of 6.841%, while the accuracy of target detection is 93.159%.

The experimental results indicate that even under extremely low SNR conditions, the proposed technique achieves high accuracy in target detection by employing a fuzzy recognition approach and appropriately setting the primary lobe width W_{main} and the element width $\Delta\theta$. Particularly noteworthy is the achievement of target detection accuracies of 98.7562%, 97.03%, and 93.159% at SNRs of −5 dB, −10 dB, and −20 dB respectively, when compared to human visual capabilities. Upon careful analysis, we believe that the false alarm rate primarily originates from the unsupervised learning process during the base construction process. These results demonstrate the robustness and effectiveness of the technique in complex noisy environments. Especially under extreme conditions, such as an SNR of −20 dB, relatively high accuracy is maintained, suggesting the potential applicability of this technique in real-world complex scenarios, offering a reliable solution for automated target detection tasks.

5. Sea-trial experiments

5.1. Dataset description

The experiments of this section evaluate the method by using sea-trial data, and the data are collected in the South China Sea, in the summer of 2021. The sonar array utilized in the experiment is a towed horizontal line array with 256 elements, with a unit distance of 1.5 m, and the detection signal's frequency band is below 1000 Hz, all data were processed using the MVDR method for array signal processing. To evaluate the false alarm rate of this method in the same and different sea areas, the original samples were divided into two parts. The first dataset, referred to as "Dataset A", originated from Sea Area A, containing 27,394 snapshots; the second dataset, "Dataset B", came from Sea Area B, with 12,543 snapshots. A detailed description of the original sample datasets is presented in Table 2 and the sound speed profiles for areas A and B are shown in Fig. 9.

Table 2
Parameters of Dataset A and B.

	Dataset A	Dataset B
Duration	27 394	12 543
Snapshot Interval	1s	1s
Snapshot Period	0s	0s
Number of Array Elements	256	256
Element Spacing	1.5 m	1.5 m
Frequency	Below 1000 Hz	Below 1000 Hz
Beamforming Algorithm	MVDR	MVDR
Water Depth	4360.2 m	2139.7 m

For this study, we selected snapshots 6200 to 7200 from Dataset A as the evaluation samples. Moreover, to accurately assess the performance of the neural network method in automatically classifying elements, it was necessary to construct and manually label each set of elements within the dataset. To fulfill this requirement, we chose Dataset A and created 1824 groups of labeled sample sets, each element of size is $[1, 310]$. The specific construction method will be discussed in detail in the section on element classification. To provide a more comprehensive evaluation of the neural network method, it was essential not to limit the test to the labeled Dataset A only. Therefore, we conducted an additional experimental assessment in different sea areas, but at the same frequency. We selected snapshots 6600 to 8600 from Dataset B as the evaluation samples. The evaluation samples from

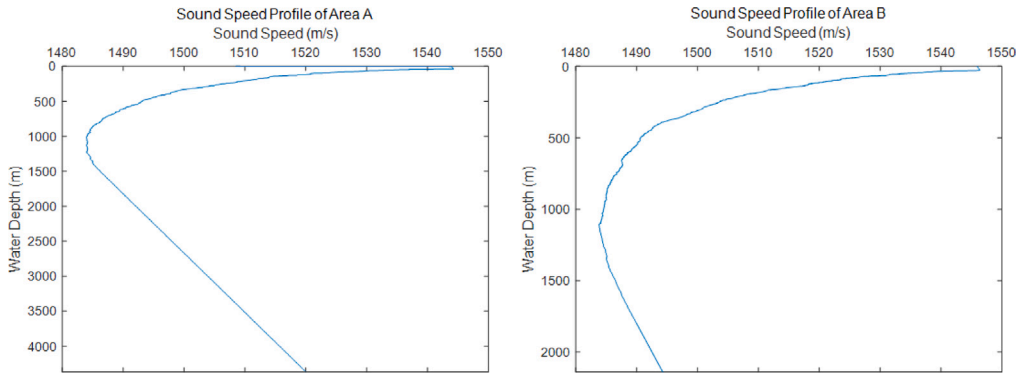

Fig. 9. Sound speed profile of area A and B.

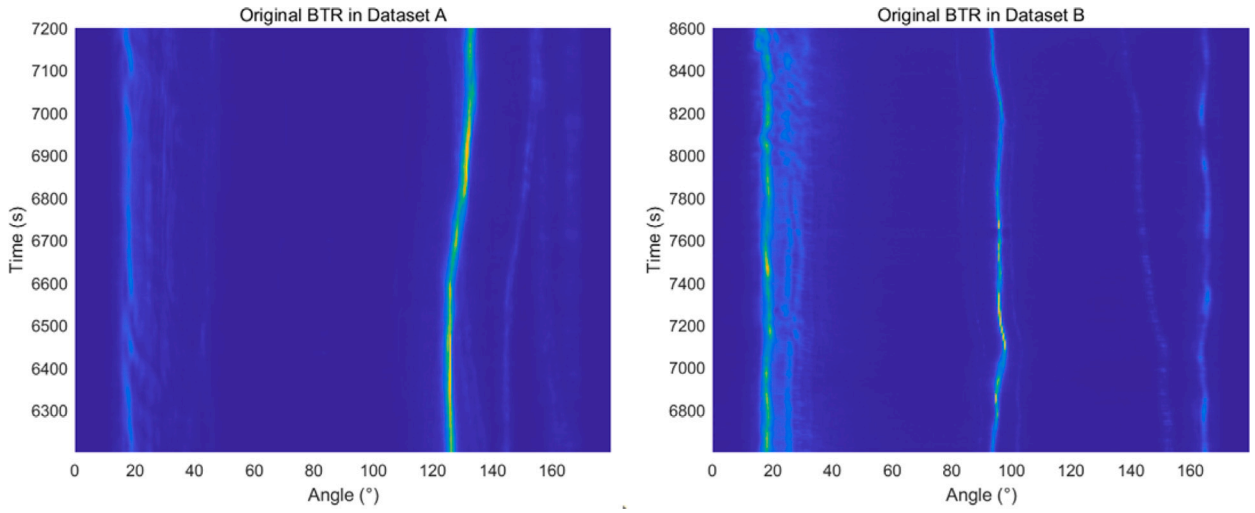

Fig. 10. Original BTR in Dataset A and B.

Datasets A and B are visualized in Fig. 10, the horizontal axis represents the angle, covering a total of 180 angles which correspond to 360 pixel values and the vertical axis indicates time, measured in snapshots or seconds.

5.2. Baseline for comparison: fixed threshold peak extraction

In the traditional field of image processing, the task of target detection typically involves two key steps: the application of filtering algorithms for image preprocessing, and the execution of fixed-threshold peak detection to identify targets within the image. The filtering algorithms used during the preprocessing phase, including but not limited to Gaussian filtering, median filtering, and mean filtering, aim to remove noise components from the image while retaining as much structural information as possible crucial for subsequent target detection steps. As a foundational technique for target identification, fixed-threshold peak detection serves as a straightforward and effective baseline method. This method involves setting a predetermined threshold and then evaluating the intensity values of each pixel in the image. If a pixel's value exceeds this threshold, it is considered part of a target. Mathematically, for each pixel $I(x, y)$ in the image I, the detection result $D(x, y)$ can be represented as:

$$D(x,y) = \begin{cases} 1 & \text{if } I(x,y) > T \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

Where T represents the preset threshold, although this strategy is conceptually straightforward, its significant limitation primarily manifests in the selection of a fixed threshold, which often relies on subjective experience rather than on rigorous mathematical derivation, thereby lacking adaptability.

In the process of applying filtering algorithms for image preprocessing, selecting the appropriate filter size becomes a critical factor affecting the image quality. This study analyzed BTR image data from snapshots number 6200 to 7200 in dataset A, with the original image shown in Fig. 10. As shown in Fig. 11, Gaussian filtering, median filtering, and mean filtering with sizes of 5 × 5 and 10 × 10 were applied to the original image. The results shows that larger filter size, covering a wider neighborhood, thus achieving more smoothing effects. However, this may also lead to blurring of the edges and fine details in the image. Gaussian filtering, with its weights decreasing from the center to the periphery according to a Gaussian distribution, effectively reduces image noise while relatively preserving the edges and structural information. Median filtering maintains clear edge details in the image, while mean filtering, although performing well in reducing random noise, may blur sharp edges and fine details when larger filter sizes are chosen.

In the field of target detection, the selection of filter size and peak detection parameters is critically important for the performance of the detection algorithm. Through the visualization results shown in Figs. 12 and 13, this study analyzed the effects of target detection on BTR images after applying Gaussian, median, and mean filters of size 10 × 10. In Fig. 12, with the minimum peak height (minHeight) set to 0, the detection algorithm was highly sensitive, resulting in a large number of false alarms across all minimum peak distance (minDistance) parameter settings. Conversely, Fig. 13 demonstrates that increasing

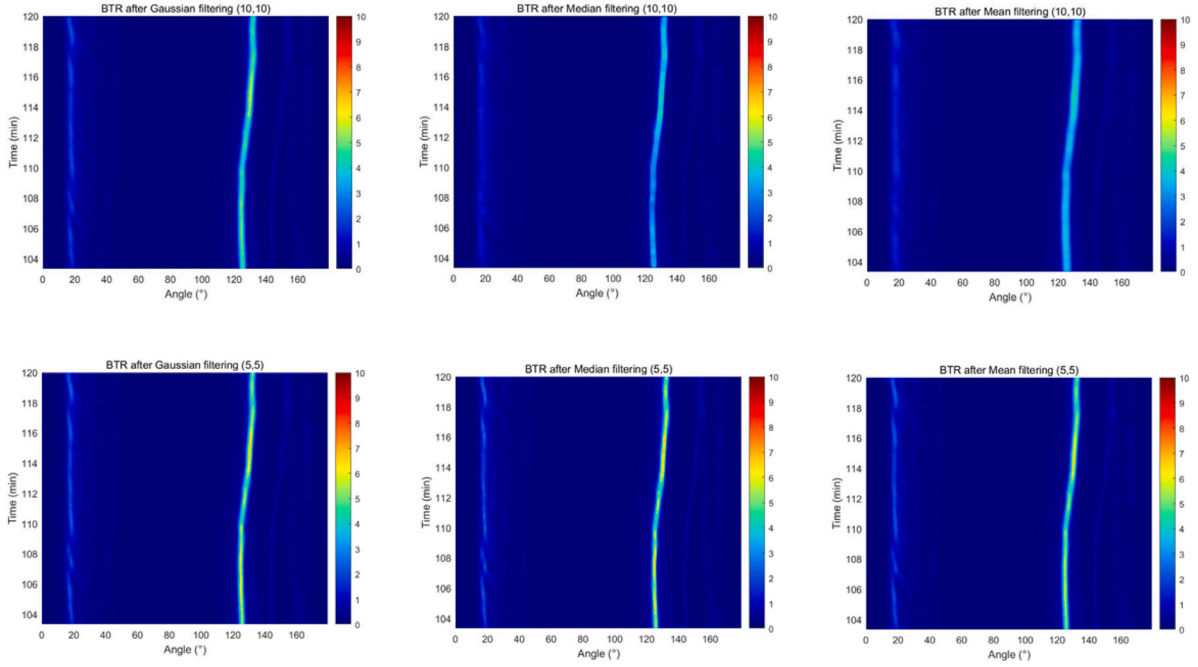

Fig. 11. Original BTR in A area after Gaussian filtering, Median filtering, and Mean filtering (with filter sizes of 5 × 5 and 10 × 10).

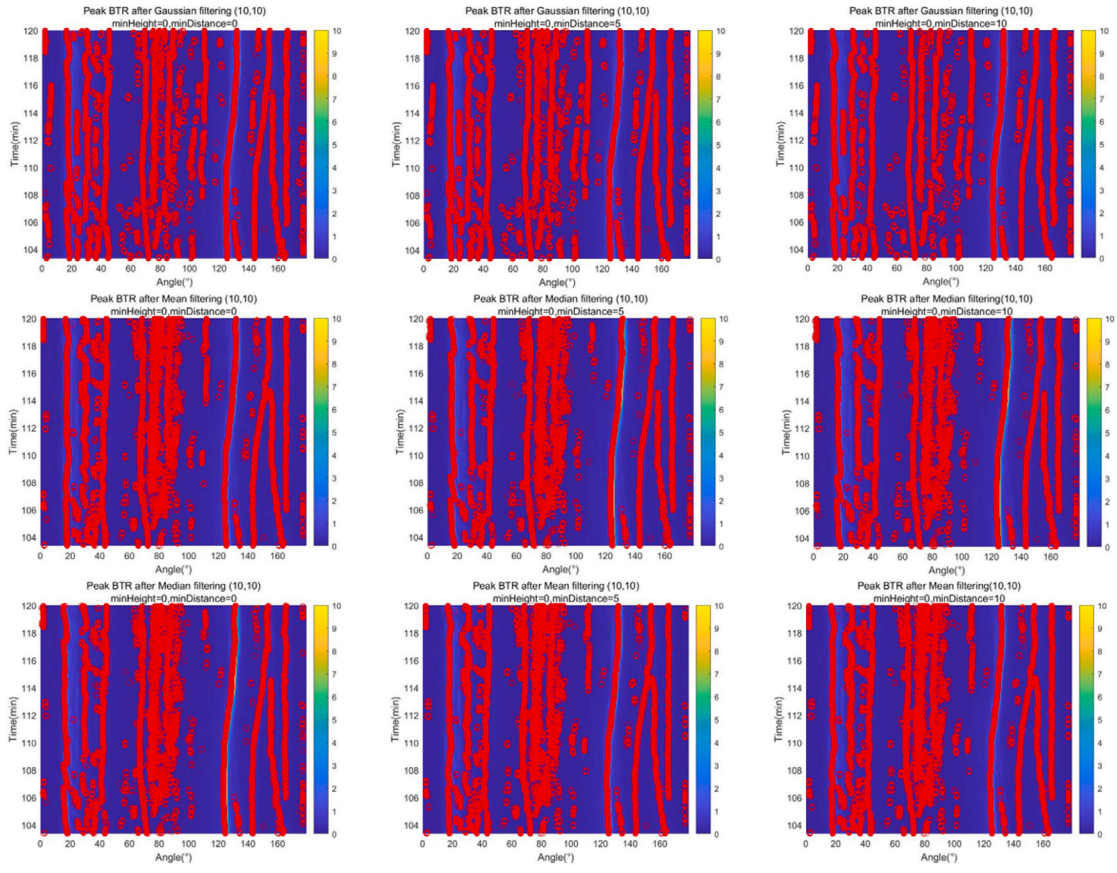

Fig. 12. Detection results in A area (with filter size = 10 × 10, minHeight = 0, minDistance = 0, 5, 10).

the minimum peak height to 0.1 improved the algorithm's discrimination ability, especially when the minimum peak distance was set to 10, effectively reducing the number of false alarms. Although Gaussian filtering partially suppressed false alarms, it fails to effectively reduce the sidelobe region on the left side of the target trajectories, and two potential targets were missed in the main lobe region in the center of the images. The analysis of Fig. 12 and Fig. 13 highlights that the outcome of target detection is significantly limited by the parameters

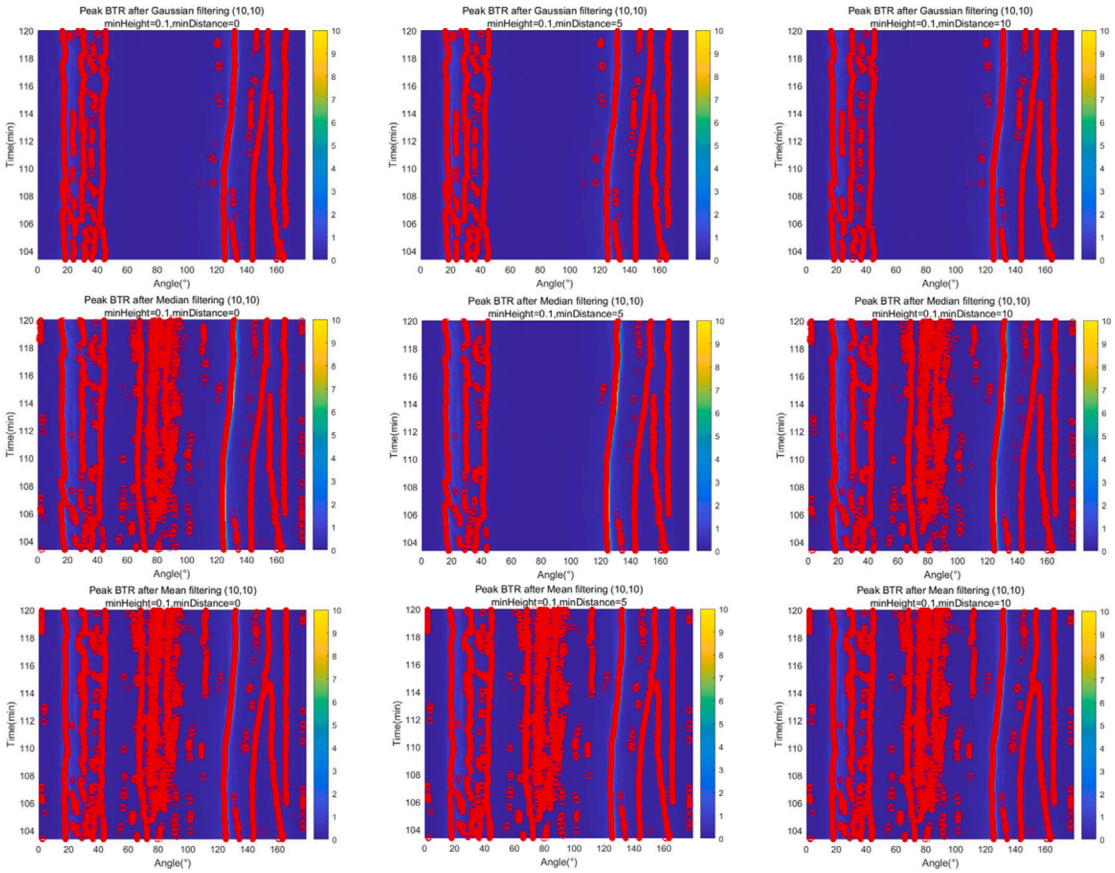

Fig. 13. Detection results in A area (with filter size = 10 × 10, minHeight = 0.1, minDistance = 0, 5, 10).

set for peak detection, describing the inherent challenge of using fixed threshold detection methods: the difficulty of achieving both high sensitivity and low false alarm rates simultaneously.

5.3. Target detection with computer vision techniques

Traditional peak detection methods inherently face limitations in their physical mechanisms, which fails to resolve the inherent contradiction between high sensitivity and low false alarm rates. Additionally, the approach of using a fixed threshold is flawed, as determining an appropriate threshold is challenging and prone to variability with environmental changes. Even when an ostensibly suitable threshold is established, it might not effectively suppress false alarms originating from sidelobes, potentially leading to missed detection of real targets.

5.3.1. Base construction

To comprehensively evaluate the impact of different parameter settings on experimental results, we set numerous parameters for the width of main lobe W_{main} in open peak extraction and the width of element $\Delta\theta$ in element construction. Specifically, we set W_{main} to 0, 5, and 10 units, equivalent to 0°, 2.5°, and 5° in terms of angles; simultaneously, we evaluated two parameter settings for the width of element $\Delta\theta$, 21 and 31 units, corresponding to angles of 10.5° and 15.5°, respectively. As shown in Fig. 14, it is evident that the best target detection performance is achieved when the width of main lobe W_{main} is set to 2.5° and the width of element $\Delta\theta$ is set to 15.5°. Based on the comparison of experimental results under the various parameter settings, we have set W_{main} to 5 units, equivalent to 2.5°, and selected $\Delta\theta$ is 31 units, corresponding to 15.5°. The original BTR image and the open peak detection result image are shown in Fig. 15. Following this evaluation, to prevent the misidentification of random noise as true targets, an empirical length of 10 units is commonly selected for the element, representing 10 s. Furthermore, an element width, $\Delta\theta$, of 31 units is determined, representing an azimuthal field of view of 15.5 degrees.

Concerning the number of cluster centers, theoretically, there should be four classes, the primary lobe of the target, left sidelobe, right sidelobe, and random noise. Due to the diverse characteristics of random noise, the number of cluster centers can also be further subdivided into more categories. In this experiment, we consider the number of cluster centers into 4, 5, and 6 classes, providing specific visual representations of each base element in the BTR images.

When the number of cluster centers is set to 4, we can construct four bases, with the physical meaning from left to right being random noise, right sidelobe, left sidelobe, and primary lobe. The representation of these bases on the BTR image is shown in Fig. 16.

When the number of cluster centers is set to 5, we can construct five bases, with the physical meaning from left to right being left sidelobe, primary lobe, right sidelobe, class 1 random noise and class 2 random noise. The representation of these bases on the BTR image is shown in Fig. 17.

When the number of cluster centers is set to be 6, we can construct 6 bases, with the physical meaning from left to right being right sidelobe, left sidelobe, class 1 random noise, class 1 primary lobe, class 2 random noise, and class 2 primary lobe. The representation of these bases on the BTR image is shown in Fig. 18. However, it is worth noting that when there are 6 cluster centers, the primary lobe is divided into two classes. As a result, the typical choice for the number of cluster centers is usually 4 or 5, depending on the specific situation.

5.3.2. Base automatic classification

We conduct the experiments using two types of methods. Firstly, based on the algorithm-driven method, we calculate the similarities between the centerline of the base and the average values of the two

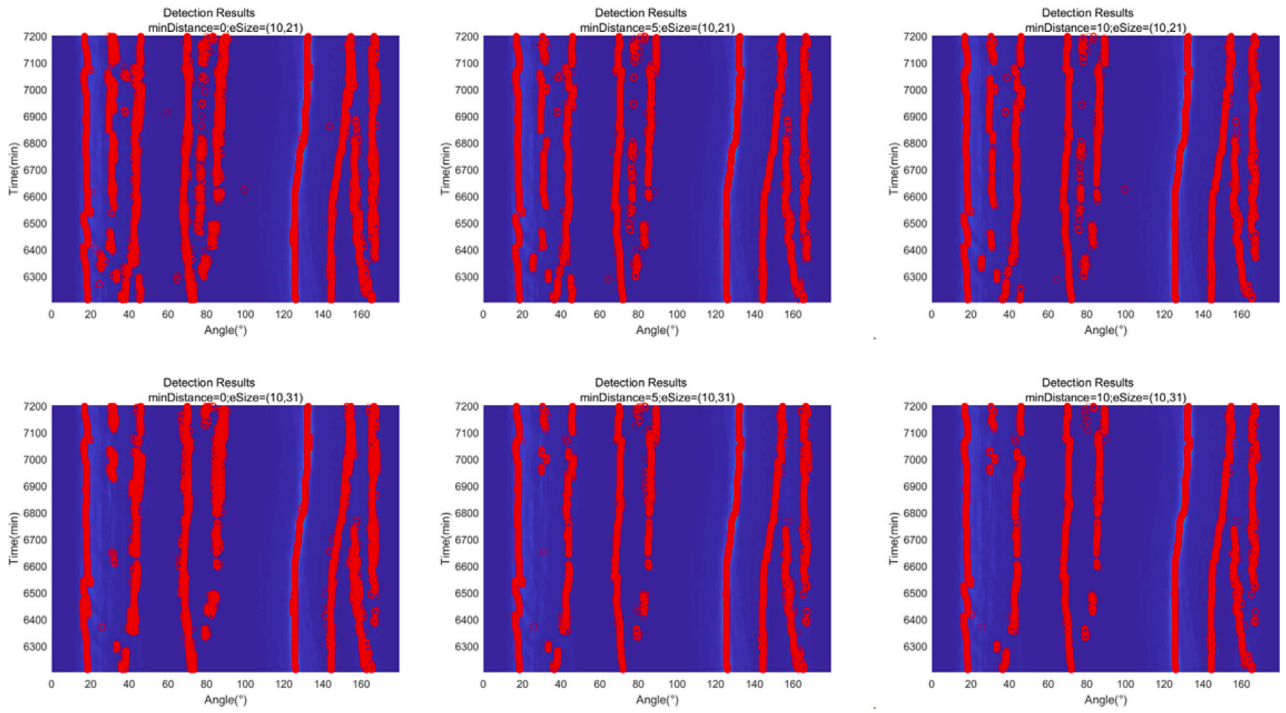

Fig. 14. Target Detection Results with Various Parameter Settings (W_{main} at 0°, 2.5°, and 5°; $\Delta\theta$ at 10.5° and 15.5°).

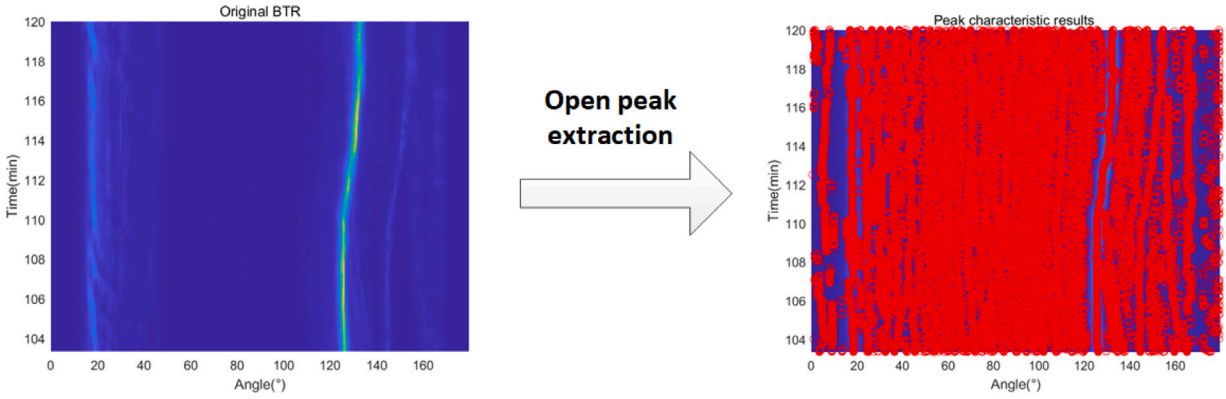

Fig. 15. Original BTR and peak BTR in A area.

side regions. Secondly, with the data-driven approach, we employed ANN, DNN, and CNN to automatically select bases for the clustering centers of the elements. Subsequently, we compared the convergence rates of these three types of neural networks. Additionally, we applied a labeled dataset to BTR images from both the same and different sea areas, all with the same frequency, to observe accuracy. It is worth noting that in our experiments, only four bases were generated each time, with one representing a real target, while the other three represented false alarms. To automatically classify bases using neural network methods, it is necessary to construct datasets in advance and manually label each set of bases. In the process of constructing manually labeled experiments, we selected Dataset A, which comprises a set of 27,394 snapshots of BTR images. To organize the dataset, it was divided into subsets, each containing 600 consecutive snapshots with a step size of 60 snapshots. Within each data subset, four base templates were generated. Specifically, we manually labeled the base template corresponding to the primary lobe as "1", while marking the other three classes as "0". Simultaneously, the generated sample set needed to match the size of subsequent experimental tasks. According to the principles mentioned in the blur estimation, the primary lobe width was set to 5 units, and the element size when constructing the elements was set to $[10, 31]$, we will explain in more detail about how parameters are chosen in next section. This process resulted in the creation of a total of 1,824 sets of labeled sample sets with a size of $[1, 310]$.

Given the limited sample size of 1824 data sets, we employ a five-fold cross-validation approach in our experiments. In each training round, the dataset is divided into five subsets, with four of them used for training and one reserved for validation. This methodology ensured that the model possessed robust generalization capabilities across distinct data subsets. Throughout the training process, we recorded both training and validation losses to monitor the model's learning progress and guard against overfitting. Additionally, we documented the model's accuracy on the validation set, serving as a crucial performance metric. Acknowledging the data imbalance issue in our dataset, which comprises a total of 1824 data samples, with only 456 samples labeled as true base templates, while the remaining 1368 samples are labeled as false alarm base templates, we select to apply a weighted loss function. This function assigns higher weights to the class with fewer samples, providing a more effective solution to the dataset imbalance. The weights are typically calculated based on the number of samples

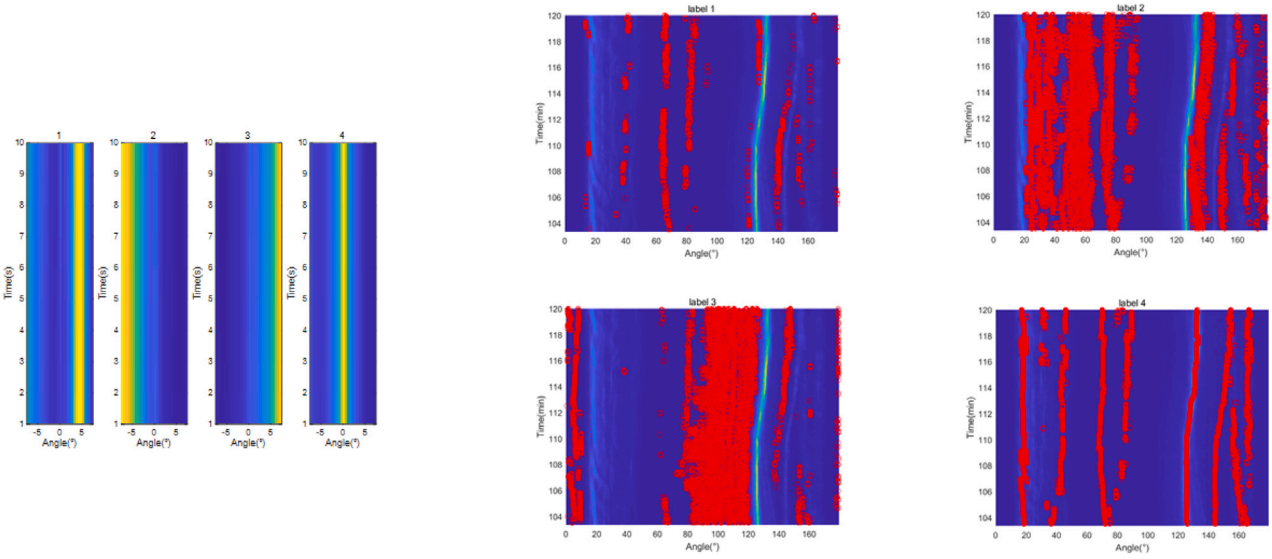

Fig. 16. Four-base representation.

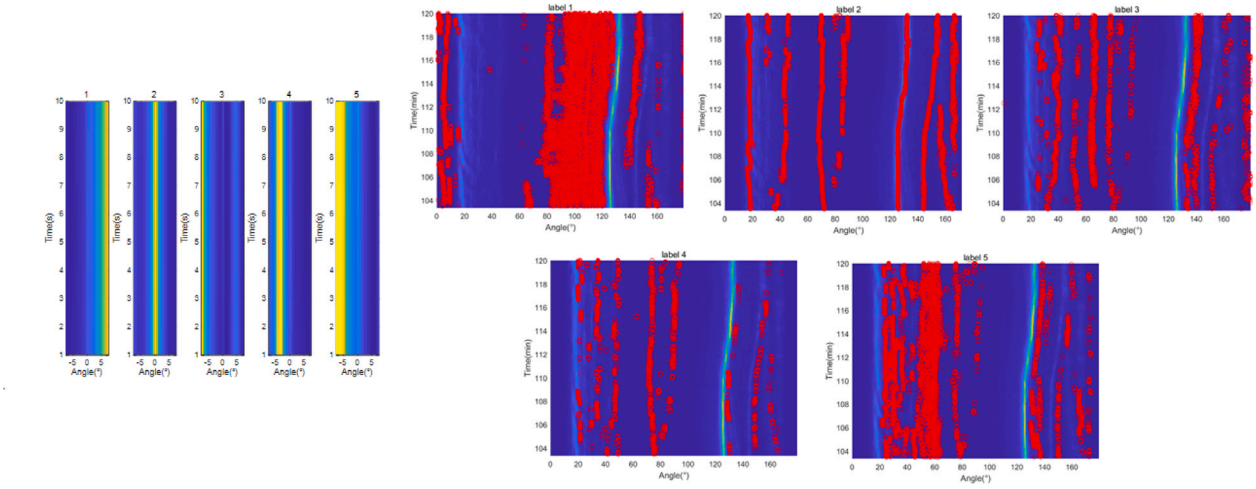

Fig. 17. Five-base representation.

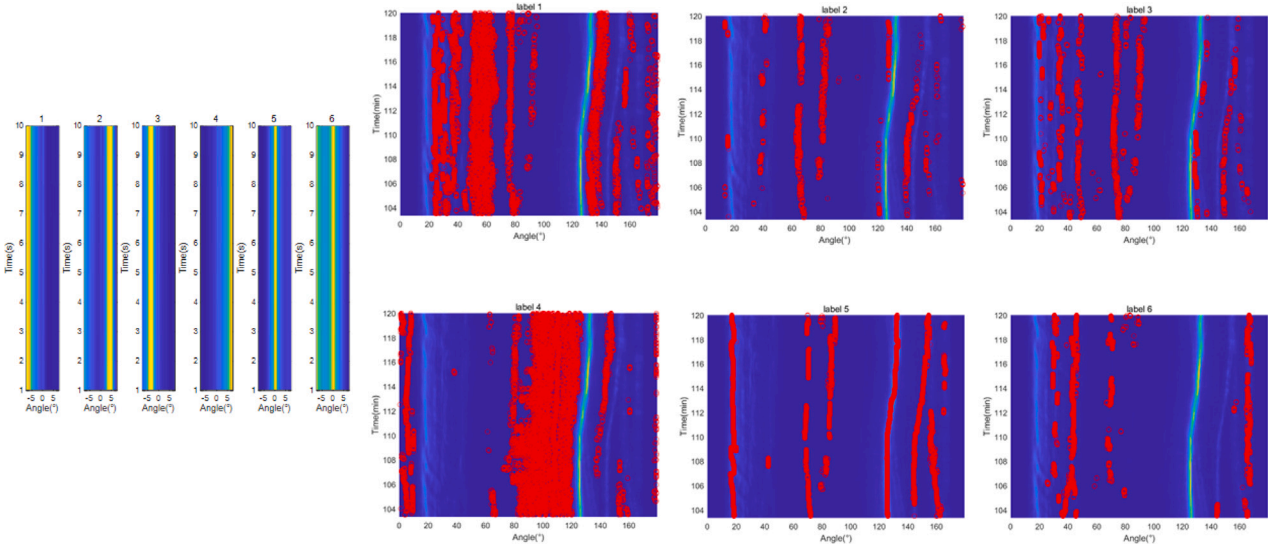

Fig. 18. Six-base representation.

Table 3
Neural network architectures: ANN, DNN, and CNN.

Network	Layer	Layer Type	Parameters	Activation Function	Notes
ANN (Artificial Neural Network)					
ANN	Input	–	1 × 310	–	Input layer
ANN	Hidden Layer 1	Fully Connected	Input: 310, Output: 256	ReLU	–
ANN	Output Layer	Fully Connected	Input: 256, Output: 2	–	No activation (logits output)
DNN (Deep Neural Network)					
DNN	Input	–	1 × 310	–	Input layer
DNN	Hidden Layer 1	Fully Connected	Input: 310, Output: 512	ReLU	–
DNN	Hidden Layer 2	Fully Connected	Input: 512, Output: 256	ReLU	–
DNN	Hidden Layer 3	Fully Connected	Input: 256, Output: 128	ReLU	–
DNN	Dropout	Dropout	Dropout Rate: Variable	–	Applied before output layer
DNN	Output Layer	Fully Connected	Input: 128, Output: 2	–	No activation (logits output)
CNN (Convolutional Neural Network)					
CNN	Input	–	10 × 31	–	Input layer
CNN	Conv Layer 1	Convolution	In Channels: 10, Out Channels: 16, Kernel Size: 5	ReLU	Padding: 2
CNN	Conv Layer 2	Convolution	In Channels: 16, Out Channels: 32, Kernel Size: 5	ReLU	Padding: 2
CNN	Conv Layer 3	Convolution	In Channels: 32, Out Channels: 64, Kernel Size: 5	ReLU	Padding: 2
CNN	Flatten	–	–	–	Flatten for fully connected
CNN	Hidden Layer	Fully Connected	Input: 64 × 31, Output: 128	ReLU	–
CNN	Dropout	Dropout	Dropout Rate: Variable	–	Applied before output layer
CNN	Output Layer	Fully Connected	Input: 128, Output: 2	–	No activation (logits output)

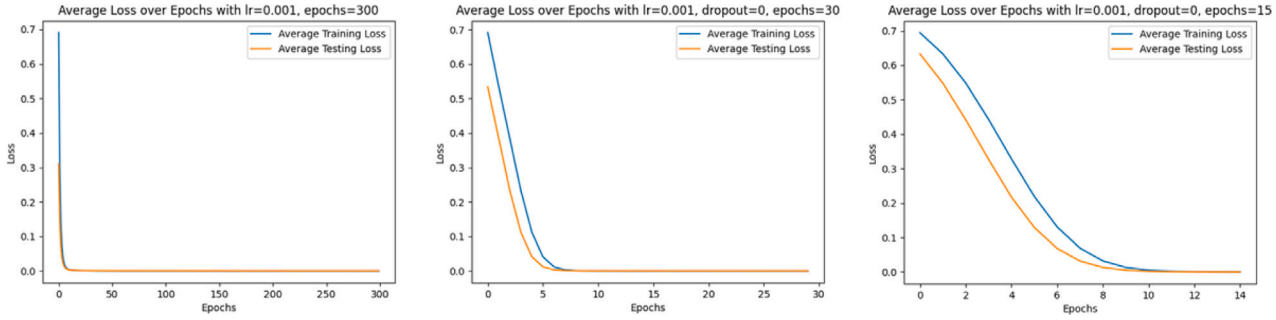

Fig. 19. Average loss of ANN, DNN, CNN.

in each class.

$$w_c = \frac{N}{N_c} \tag{26}$$

where, N is the total number of samples, N_c is the number of samples of class c.

The parameters for each layer of ANN, DNN, and CNN are specified in Tables 3. When the learning rate is set to 0.001, ANN, DNN, and CNN converge after 300, 30, and 15 iterations, respectively. Furthermore, all three networks achieve 100% accuracy under these conditions. When the learning rate is set to the same value, lr=0.001, the loss functions for these three network structures are illustrated in Fig. 19, it can be observed that CNN exhibits the fastest convergence speed and the highest efficiency.

The experiments have shown that using the base automatic classification method (both algorithm-driven method and data-driven method) can achieve a 100% accuracy at the same frequency, both in the same sea area and across different sea area.

5.3.3. Performance evaluations

The experimental data used in this study were collected from two distinct marine areas. These datasets have been accordingly segmented into two scenarios, facilitating the evaluation of the novel method's performance. In the first scenario, the marine area of interest corresponds to the labeled dataset. We focused on the same marine area and extracted a subset of snapshots numbered from 6200 to 7200 from a BTR image consisting of 27,394 snapshots. After applying open-peak extraction, we obtained the peak BTR image. Subsequently, we constructed the base and utilized automatic selection to derive the BTR image results for target detection. However, these results contained visible false alarms, which prompted us to conduct manual false alarm screening based on human visual inspection. Following the manual screening process, we marked the visible false alarms in green, resulting in annotated results. Among the marked circles, 107 were green, while 5461 were red, leading to a false alarm rate of 1.9593% and a target detection accuracy of 98.0407%, these images are illustrated in Fig. 20.

In the second scenario, which involved a different marine area B but the same frequency, we selected a subset of snapshots numbered from 6600 to 8600 from a BTR image consisting of 12,543 snapshots. As with the previous case, we performed open-peak extraction to obtain the peak BTR image. After base construction and automatic selection, we derived the BTR image results for target detection. However, similar to the previous scenario, these results included visible false alarms. To address this issue, we conducted manual false alarm screening based on human visual inspection. Subsequently, we marked the visible false alarms in green, resulting in annotated results. Among the marked circles, 199 were green, while 7196 were red, leading to a false alarm rate of 2.7654% and a target detection accuracy of 97.2346%, these images are illustrated in Fig. 21.

5.4. Real-time performance

After conducting a thorough analysis of the 1000 snapshots contained within Dataset A (equivalent to 1000 s of data), this study compared the temporal efficiency of image processing technology against several neural network architectures (ANN, DNN, CNN) in performing target detection tasks. We set each interval at 60 s, with a step size of 1 s. Using image processing technology for automated primitive classification to achieve the goal of target detection, the total time

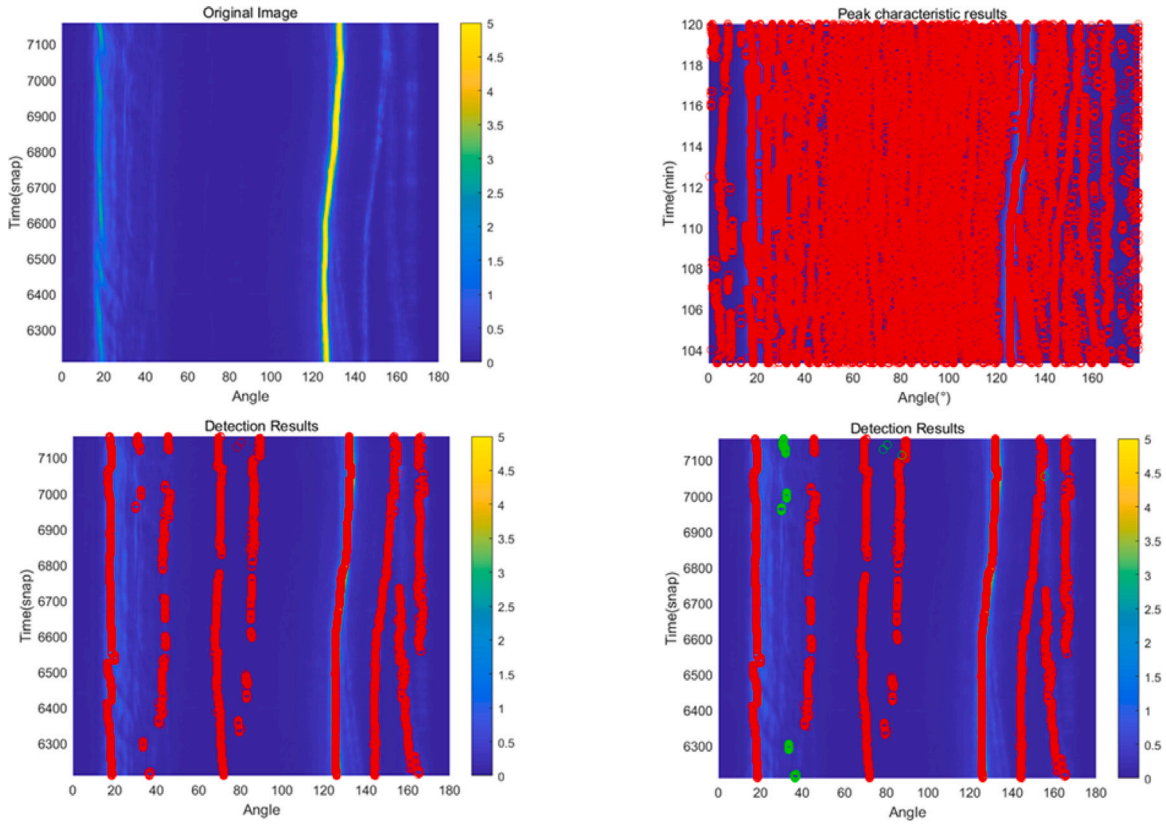

Fig. 20. BTR in the same marine area A.

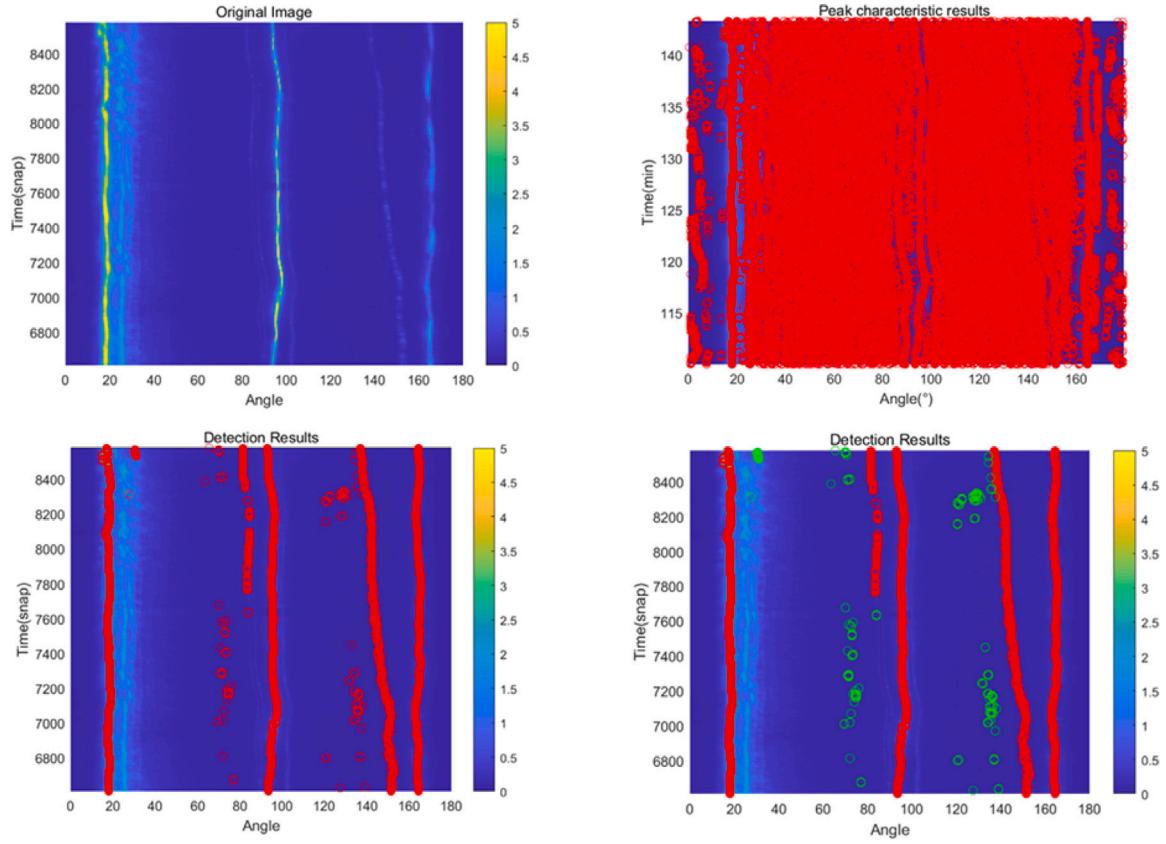

Fig. 21. BTR in the different marine area B.

Table 4
Processing time comparison for 1000 shots.

Method	First attempt	Second attempt	Third attempt	Average time
Image	44.327	46.682	43.296	44.77
ANN	136.9824	138.3642	134.3866	136.58
DNN	134.5512	131.8871	132.2696	132.9
CNN	162.2788	165.3527	163.9876	163.87

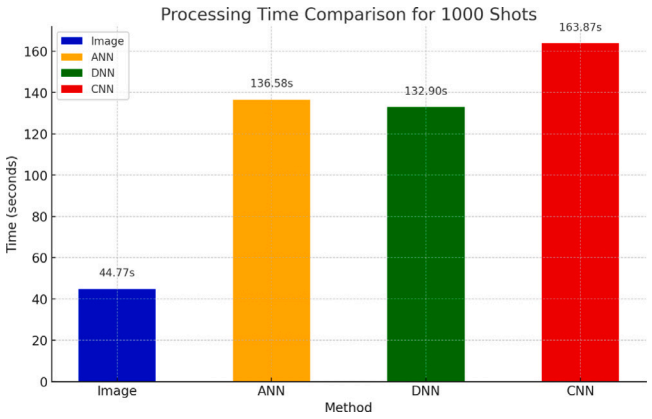

Fig. 22. Processing Time Comparison for 1000 Snapshots.

required to process 1 snapshot was merely 0.04477 s. In contrast, for the same task, the processing times for the ANN, DNN, and CNN architectures were 0.13658 s, 0.1329 s, and 0.16387 s, respectively. Each of the experiments was repeated three times to ensure data accuracy and calculate the average values. The related data has been summarized in Table 4, and visualized in the bar chart shown in Fig. 22. Clearly, the image processing method not only meets the requirements for real-time detection but also processes each instance in far less than a second, providing sufficient time margin to ensure the continuity and stability of real-time responses, even when considering the addition of some extra preliminary signal processing steps.

6. Conclusion

This paper researches an automated target detection method for BTR images. We propose a computer vision approach inspired by human vision for target detection. Initially, we introduce an unsupervised learning method to extract bases with well-defined physical meanings. Subsequently, we present two different automated base selection approaches: one driven on data and the other driven by algorithms. In the evaluations using South China Sea trial data, both of these automated selection methods achieved a 100% accuracy rate. Furthermore, we compare the automatically detected target results with manually observed target detection results on BTR images. The results show that the target detection method with computer vision techniques exhibits a false alarm rate of less than 3% compared to human visual observation, which shows the potential of this method in enhancing the automation efficiency of target detection in underwater unmanned devices.

6.1. Error analysis

In this study, we employed unsupervised learning methods for base construction and used data-driven and algorithm-driven methods for base automatic classification. Although theoretically, automatic classification of base can achieve 100% accuracy, in practical application, compared to human eye recognition, we observed an approximate 3% false alarm rate. After careful analysis, we believe that this 3% false alarm rate mainly originates from the unsupervised learning process in the base construction phase. During the base construction process, unsupervised learning can identify meaningful structures from a large amount of unlabeled data, thereby facilitating the automatic classification of bases. However, since this method does not rely on prior label information, it may not be able to fully distinguish between very similar categories in certain situations, leading to false alarms.

6.2. Existing problems

One persistent challenge is the performance variability of the MVDR method across different SNR conditions. While the MVDR method demonstrates strong performance in varying SNR conditions, we have observed that its effectiveness is closely tied to specific SNR conditions. Consequently, optimizing the MVDR method's effectiveness necessitates fine-tuning the BTR images it generates for different SNR environments. Additionally, during the process of element construction, we encountered difficulties related to size matching. Although the fuzzy recognition method has partially mitigated this issue, when dealing with marine data of diverse sizes, the fixed-size neural network dataset may prove inadequate in addressing this particular challenge.

6.3. Future work

We plan to employ multiple methods to reduce the false alarm rate. Firstly, considering alternative clustering methods may be crucial in improving model performance. By adjusting the parameters and iteration counts of clustering algorithms, we aim to capture data characteristics more accurately, thus effectively reducing false alarms. Additionally, we will explore how to make the MVDR method more adaptable to different SNR conditions, enhancing its robustness under varying conditions. Regarding neural networks and image processing methods, they have demonstrated outstanding applicability in automatic base selection, maintaining a 100% accuracy rate. To address the size issue, we are considering the development of adaptive network architectures capable of handling inputs of different sizes or the use of data standardization techniques to align inputs with fixed-size network requirements. These efforts are expected to significantly enhance the flexibility and accuracy of network processing for data of varying sizes, laying a solid foundation for future applications.

CRediT authorship contribution statement

Hao Yin: Writing – original draft, Validation, Software, Methodology, Formal analysis. **Chao Li:** Writing – review & editing, Project administration, Investigation, Funding acquisition, Data curation. **Haibin Wang:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization. **Fan Yin:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization. **Fan Yang:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Data availability

The data that has been used is confidential.

Acknowledgments

This article is funded by the General Program of National Natural Science Foundation of China (ID: 62171440), the project of the Chinese Academy of Sciences (ID: XDB0700403), and supported by the China Scholarship Council.

References

Carbone, C.P., Kay, S.M., 2012. A novel normalization algorithm based on the three-dimensional minimum variance spectral estimator. IEEE Trans. Aerosp. Electron. Syst. 48, 430–448. http://dx.doi.org/10.1109/TAES.2012.6129646.

Chenhui, Y., 2003. Peak energy detection with application to passive sonar display. Appl. Acoust..

Department of the navy, 2021a. Science and technology strategy for intelligent autonomous systems.

Department of the navy, 2021b. Unmanned campaign framework.

King, D.B., Wertheimer, M., 2005. Max Wertheimer & Gestalt Theory.

Lei, Z., Yang, K., Zhang, Q., Xia, H., 2016. Two dimensional tv-l1 regularization for underwater acoustic source tracking. In: OCEANS 2016 - Shanghai. pp. 1–4. http://dx.doi.org/10.1109/OCEANSAP.2016.7485353.

Milan, S., Vaclav, H., Roger, B., 2015. Image processing, analysis, and machine vision. In: Cengage Learning.

Palmer, S.E., 1999. Vision Science: Photons to Phenomenology. The MIT Press, Cambridge, MA, USA.

Saucan, A.A., Sintes, C., Chonavel, T., Le Caillec, J.M., 2014. Robust, track before detect particle filter for bathymetric sonar application. In: 17th International Conference on Information Fusion. FUSION, pp. 1–7.

Shapiro, J., Green, T., 2000. Performance of split-window multipass-mean noise spectral estimators. IEEE Trans. Aerosp. Electron. Syst. 36, 1360–1370. http://dx.doi.org/10.1109/7.892683.

Struzinski, W.A., Lowe, E.D., 1984. Performance comparison of four noise background normalization schemes proposed for signal detection systems. J. Acoust. Soc. Am. 76, 1738–1742.

Xin, J., Le, B., Bo, L., Luo, L., 2015. Track before detect of weak trajectory using hidden markov model. In: 2015 4th International Conference on Computer Science and Network Technology. ICCSNT, pp. 1473–1477. http://dx.doi.org/10.1109/ICCSNT.2015.7491007.

Xin, J.R., Luo, L.Y., 2016. Bearing-only trajectory detector based on hidden Markov model. Syst. Eng. Electron..

Yang, T.C., 2018a. Deconvolved conventional beamforming applied to the swellex96 data. J. Acoust. Soc. Am. 144, 1768.

Yang, T.C., 2018b. Deconvolved conventional beamforming for a horizontal line array. IEEE J. Ocean. Eng. 43, 160–172. http://dx.doi.org/10.1109/JOE.2017.2680818.

Yin, F., Li, C., Wang, H., Nie, L., Zhang, Y., Liu, C., Yang, F., 2023. Weak underwater acoustic target detection and enhancement with bm-seed algorithm. J. Mar. Sci. Eng. 11, http://dx.doi.org/10.3390/jmse11020357, URL: https://www.mdpi.com/2077-1312/11/2/357.

Yin, F., Li, C., Wang, H., Yang, F., 2019. Automatic acoustic target detecting and tracking on the azimuth recording diagram with image processing methods. Sensors 19.

Yin, Fan, Li, Chao, Wang, Haibin, Yang, Fan, 2022. Automatic tracking of weak acoustic targets within jamming environment by using image processing methods. Appl. Sci. 12, http://dx.doi.org/10.3390/app12136698, URL: https://www.mdpi.com/2076-3417/12/13/6698.

Zhao, S., et al., 2011. Multi-modal feature extraction and clustering for multi-view learning. In: Proceedings of the IEEE International Conference on Computer Vision.

Zheng, Y., Hu, C., Li, Q., Sun, C., 2005. A method to extract multi-target's bearing time tracks on real time. Acta Acust..