Chinese Journal of Information Fusion
http://dx.doi.org/10.62762/CJIF.2024.160538

RESEARCH ARTICLE

Convolutional Neural Network for Ellipse Extended Target Tracking

Binchao Bian[1] **and Hui Chen**[1,*]

[1] School of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

Abstract

In the field of extended target tracking, constrained by the sparse measurement set from radar, the target contour is commonly estimated as an elliptical shape. This paper uses convolutional neural networks to estimate the size and orientation information of extended targets. First, by establishing a systematic model for elliptical extended targets and modeling its measurement information, data normalization, and length equalization operations were conducted to provide reliable measurement data for subsequent neural network processing. Subsequently, through the construction of a convolutional neural network model, accurate estimation of the contour parameters of elliptical extended targets was achieved, and integration with Kalman filtering enabled precise tracking of the target positions. Finally, the effectiveness of the proposed method was verified through the construction of simulation scenarios, and the performance of the method was comprehensively evaluated using the Gaussian Wasserstein distance.

Academic Editor:
Ziling Wang

Submitted: 29 July 2024
Accepted: 18 September 2024
Published: 23 September 2024

Vol. 1, **No.** 2, 2024.
10.62762/CJIF.2024.160538

***Corresponding author:**
✉ Hui Chen
chenh@lut.edu.cn

Keywords: Extended target tracking, elliptical, convolutional neural network, gaussian wasserstein distance.

1 Introduction

In radar target tracking, the traditional radar target tracking algorithms assume that the target is a point without spatial extension due to the limitation of sensor resolution [17, 21]. However, with the continuous development of electronic technology, high-resolution sensors have been widely applied in the field of target tracking. These sensors can obtain multiple measurements per sampling cycle through multiple scattering points from the target. On the basis of these measurements, deep-level feature information of the target, such as contour and orientation, can be extracted, and such targets are referred to as extended targets [18, 24, 32]. In recent years, significant achievements have been made in this field through the continuous exploration, and the relevant research results have been applied in anti-missile defense, autonomous driving, etc. However, how to better extract shape and orientation information of target from the observation information remains the focus of the extended target tracking problem [25, 27]. The essence of extended target tracking is to estimate the real-time state of the target, including its position and spatial extension [11]. Currently, there are already

Citation
Bian,B. & Chen,H. (2024). Convolutional Neural Network for Ellipse Extended Target Tracking. *Chinese Journal of Information Fusion*, 1(2), 93–108.

many methods for the estimation of extended target shape using 2D sensor data. Some of these methods estimate the target contour as a basic geometrical shape like ellipse [4, 9, 10, 20, 33] or rectangle [1, 13, 38]. Additionally, in order to obtain more realistic and detailed target contour information, some methods estimate the target contour as curves [2, 3, 7, 35] or irregular shapes composed of multiple basic geometrical shapes [6, 12, 23].

In general, measurements of radar targets tend to be a sparse measurement set, and for most application requirements it is appropriate to estimate the spatial extent of the extended target as the underlying geometry. Concerning the estimation methods for elliptical contours, the main representative approaches include random matrix and random hypersurface methods. Koch et al. [20] first introduced the random matrix model to describe elliptical contour features. Unlike other measurement models for extended targets (e.g., Poisson space model [12], set clustering process [16, 31], point sets on rigid body models [5, 15]), this method approximates the measurement model of the extended target as an ellipse. By using Bayesian recursion, they derived the extended target Gaussian-inverse Wishart filter, which enabled the joint estimation of the target's motion state and shape [34]. Feldmann et al. [8] optimized the random matrix model by analyzing the influence of measurement noise on shape estimation and established a more reasonable noise measurement source model. In order to obtain the optimal state estimation in the Bayesian framework, Lan J et al. [22, 23] developed a new quantitative model and obtained the optimal state estimation equation by Bayesian recursion. Subsequently, Yang et al. [36] linked the target's shape equation and measurement equation using multiplicative noise terms and implemented joint estimation of the target's motion state and shape based on the extended Kalman filter. Govaers et al. [14] further established a more reasonable filter based on the multiplicative noise. Baum et al. [2] achieved star-convex extended target tracking by assuming that each measurement source is located on a scaled version of the actual shape of the extended target, using a specific measurement source model. However, most of these methods are limited by the measurement model. In cases where only observation data is available and it is difficult to obtain a accuracy measurement model, these methods are subject to significant limitations.

The emergence of neural network technology has provided new research directions for extended target tracking. Nezhadarya et al. [28] proposed a neural network model called BoxNet, which is based on the bounding box regression. By describing the target contour as a rectangle using observed 2D point cloud data, they achieved estimation of the target's spatial extent. Simon et al. [30] transformed the measurement data of extended targets into dual-channel image data,and then applying Gaussian blurring to aggregate historical measurement values and setting scaling factors, they realized elliptical shape estimation of extended targets. However, both the establishment of neural network models and the transformation of observation data into image data have limitations and cannot fully unleash their capabilities. In the process of extended target tracking, the most important challenge is how to extract more feature information from limited measurement data. Among numerous neural network models, convolutional neural networks (CNN) are most suitable for feature extraction. For elliptical extended targets, it is necessary to extract ellipse shape parameters, including major and minor axes and the orientation angle, from the measurement data. The operations of convolutional kernels can effectively preserve the inherent information features and provide assurances for achieving precise downstream tasks. Therefore, utilizing convolutional neural networks becomes the optimal choice for realizing extended target shape estimation.

In this paper, we propose a CNN-based method for estimating the shape of extended targets. In the second part, The process of establishing the system model for extended targets is explained. The third part utilizes a Kalman filter to achieve motion state estimation of the extended targets. In the fourth part, a complete convolutional neural network model is established to realize shape estimation of the extended targets. In the fifth part, the effectiveness of the proposed method (CNN-ETT) is validated through experimental simulations, and the experimental results are analyzed and discussed.

2 System Model

When extended target's measurements are scattered in a certain space in a disorderly manner, an ellipse can be utilized to approximate the expansion state of the target.

Assuming that the extended target's state consists of the kinematic state $x_k = [m, \dot{m}, \ddot{m}]^{\mathrm{T}}$ and shape $X_k = [\theta, l, w]^{\mathrm{T}}$, where $m, \dot{m}, \ddot{m}$ can represent the position, velocity, and acceleration. The θ, l, w represents the ellipse's rotation angle, long axis, and short axis. During the target movement, assuming that the rotation angle θ is consistent with the direction of velocity. Thus, the kinematics model of the system established can be expressed as:

$$x_k = (F_k \otimes I_d)x_{k-1} + v_k \tag{1}$$

where F_k is the state transition matrix, $\otimes$ is Kronecker product, and v_k is zero-mean Gaussian white noise with covariance $D_k \otimes X_k$, i.e. $v_k \sim \mathcal{N}(0, D_k \otimes X_k)$.

Suppose that Z_k is the measurement set of the extended target at time k, and any measurement value z_k^j in Z_k represents a position in the Cartesian coordinate system, where the measurement model is:

$$z_k^j = (H_k \otimes I_d)x_k + \omega_k^j \tag{2}$$

where H_k is the measurement matrix on each dimension,

$$H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

when the dimension is 2, and the kinematic state parameters are position, velocity and acceleration, and the sensor can only observe the position information of the target, where ω_k^j is a zero-mean Gaussian white noise with covariance $\lambda X_k + R$, and R is the error of the sensor itself, i.e. $\omega \sim \mathcal{N}(0, \lambda X_k + R)$.

Based on the research by Feldmann et al. [9], the measurement data of the target shows a closer alignment with real-world observations when λ is set to 1/4. As shown in Figure 1.

3 Kinematic state estimation

In this section, the extended target's kinematic state will be estimated. The measurement data obtained from Eq.(2) represents the Cartesian coordinate information in a 2-Dim plane, and Kalman filter is used to estimate the kinematic state.

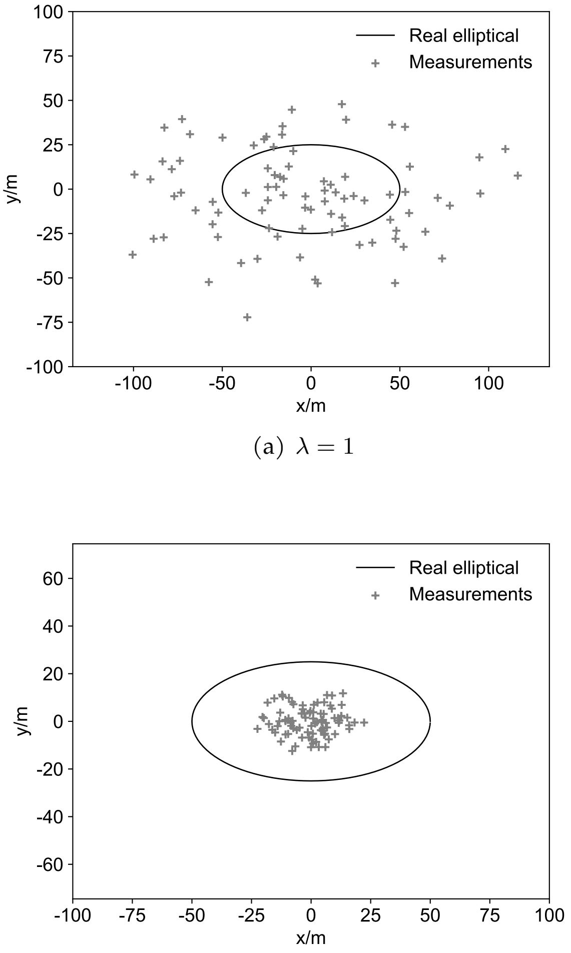
Assuming the measurement data set is:

$$Z_k = \{z_k^j\}_{j=1}^{n_k} \tag{3}$$

where n_k is the number of measurements included in the measurement set, and follows a Poisson distribution.

Firstly, the covariance of the measurement noise can be obtained from the system's measurement model:

$$\tilde{R}_k = \lambda X_k + R \tag{4}$$



(a) $\lambda = 1$

(b) $\lambda = \frac{1}{4}$

Figure 1. The measurement distribution with different values of noise covariance

where X_k contains the shape information of the extended target.

The Kalman filter is frequently used for tracking and predicting single point targets. However, in the case of extended targets, there are multiple measurement values n_k available at each time instance. To enhance compatibility with the Kalman filter, it is beneficial to incorporate the mean value of the measurement set at time k. This approach allows for better integration of the measurement data and improves the accuracy of tracking and prediction. The mean value can be calculated as follows:

$$\bar{z}_k = \frac{1}{n_k}\sum_{j=1}^{n_k} z_k^j \tag{5}$$

After obtaining the covariance of the measurement noise and the mean value of the measurement, the subsequent stage involves tracking the kinematic state of the extended target by combining Kalman filter.

When the measurement at time k is not obtained, a prediction of the kinematic state at time k is made beforehand. The predicted values of the target kinematic state obey the following probability density distribution:

$$p(x_k|\overline{z}_{k-1}) = \int p(x_k|x_{k-1}, \overline{z}_{k-1})p(x_{k-1}|\overline{z}_{k-1})d(x_{k-1}) \tag{6}$$

Further, it can be obtained:

$$\hat{x}_{k|k-1} = (F_k \otimes I_d)\hat{x}_{k-1|k-1} \tag{7}$$

$$P_{k|k-1} = (F_k \otimes I_d)P_{k-1|k-1} + Q_k \tag{8}$$

After obtaining the measurement value at time k, the kinematic state is estimated. The estimated kinematic state values of the target obey the following probability density distribution:

$$p(x_k|\overline{z}_k) = \frac{p(\overline{z}_k|\overline{z}_{k-1}, x_k)p(x_k|\overline{z}_{k-1})}{p(\overline{z}_k|\overline{z}_{k-1})} \tag{9}$$

Further more the following equation can be obtained:

$$\begin{aligned} K_k =& P_{k|k-1}(H_k \otimes I_d)^{\mathrm{T}} \\ & ((H_k \otimes I_d)P_{k|k-1}(H_k \otimes I_d)^{\mathrm{T}} + \tilde{R}_k)^{-1} \end{aligned} \tag{10}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(\overline{z}_k - (H_k \otimes I_d)\hat{x}_{k|k-1}) \tag{11}$$

$$P_{k|k} = (I_d - K_k(H_k \otimes I_d))P_{k|k-1} \tag{12}$$

where Q_k, K_k and P_k respectively denotes the covariance of the system process noise, the gain matrix and the covariance of the kinematic estimation.

After obtaining the measurement set of the extended target, the kinematic state estimation of the extended target can be achieved by Eq.(6) to Eq.(12).

4 CNN Model

4.1 Measurement information processing

4.1.1 Data preprocessing

Data processing plays a crucial role in building a neural network, whether it is during the initial stage of model training or the subsequent stage of model application. With that in mind, this section will provide an overview of the data processing procedures.

In the previous section, the extended target's kinematic state was estimated through Kalman filtering, and all the n_k measurement data included in the measurement set contained information about the target position. Generally, the target's motion position changes over time, while the convolutional neural network model is mainly used to estimate the specific information of the target shape at each moment. Therefore, when the data is sent into the model for estimation, it is necessary to remove the position information contained in it. The measurement set $\tilde{Z}_k$ constructed from this will no longer include position information, i.e.:

$$\tilde{z}_k^j = z_k^j - \hat{x}_{k|k} \tag{13}$$

$$\tilde{Z}_k = \left\{ \tilde{z}_k^j \right\}_{j=1}^{n_k} \tag{14}$$

where $\hat{x}_{k|k}$ is the updated value at time k obtained from the Kalman filter.

In practice, the dimensions (length and width) of the target are not restricted to a specific range, and excessively large or small values are not conducive to model training. Hence, it becomes necessary to normalize the data.

$$\widetilde{s}_{k,(x,y)} = \frac{s_{k,(x,y)} - s_{k,(x,y),min}}{s_{k,(x,y),max} - s_{k,(x,y),min}} \tag{15}$$

where, $s_{k,(x,y)} = \tilde{z}_k^j$.

The numerical range of the data changes after normalization, but its position distribution remains the same, preserving the shape information effectively contained in the original measurement.

Generally, the measurement data of extended targets has a rotation angle in its spatial distribution, which is often consistent with the velocity direction. According to the measurement equation, the measurement data is distributed around the measurement source. Therefore, the rotation angle of the target can be obtained by defining the scattering matrix.

Firstly, the scattering matrix is defined as:

$$Z^k = \sum_{j=1}^{n_k} (z_k^j - \bar{z}_k)(z_k^j - \bar{z}_k)^{\mathrm{T}} \tag{16}$$

(Note:Z_k is measurement set,Z^k is scattering matrix) The rotation matrix of the measurement data can be obtained by calculating the eigenvectors of the scattering matrix.

$$r_k = \mathrm{eig}(Z^k) \tag{17}$$

Assuming the rotation matrix obtained at time k is:

$$r_k = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{18}$$

The rotation angle of the extended target at this moment is:

$$\theta_k = \arctan(\frac{c}{a}) \tag{19}$$

Therefore, the rotation matrix is also equivalent to the following equation:

$$r_k = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \tag{20}$$

The design of this model is to obtain the parameters of the long and short axes of the ellipse. Before feeding the data into the model for prediction, it is necessary to fix the data at a certain rotation angle at each moment. In the method proposed in this paper, the rotation angle of all data is fixed at $0°$.

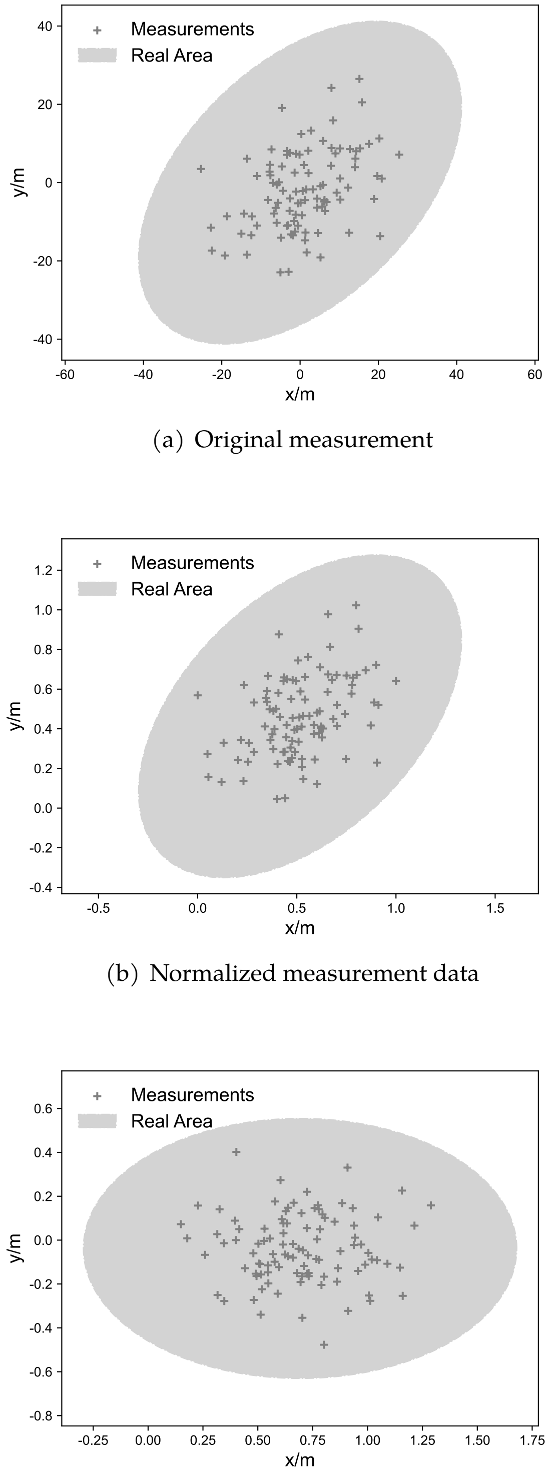
After obtaining the rotation angle of the spatial distribution of the measured data, the rotation matrix of the measured data can be obtained. By taking the inverse matrix, the rotation angle of the measured data can be fixed at $0°$, that is:

$$z'^{j}_{k} = r_k^{-1} z_k^j \tag{21}$$

The distribution of the measurement data after rotation is shown in Figure 2.

According to Eq.(3), the quantity of measuring points at each moment follows a Poisson distribution, resulting in varying numbers of obtained measuring points. To effectively utilize the neural network model's performance, it is necessary to process the data into equal lengths. In this paper, the measurement data is represented as 2-Dim Cartesian coordinates, with position information in 2 rows and n_k columns. After equal-length processing, the data is transformed into position coordinates with 2 rows and 256 columns.

In the data processing step, the following method was employed to maximize the retention of information within the measurement data. For $n_k \leq 256$, the missing columns are filled with the average value to reach 256 columns. For $n_k > 256$, an outer circle is formed using the maximum coordinates of the measurement values. Then, an inner circle is created at $\frac{1}{4}$ of the maximum of the outer circle's coordinates. Within this elliptical contour area, 128 measurements are randomly selected from both inside and outside. If the number of measurements is less than 128, the average of the measurements is used to fill in the remaining values. This process is illustrated in Figure 3.



(a) Original measurement

(b) Normalized measurement data

(c) Set rotating the angle to 0°

Figure 2. Data preprocessing process

4.1.2 *Measurement Information Transmission*

In order to effectively utilize historical measurement information for tracking extended targets, different sampling instants during the target motion process yield different measurement points. Therefore, it is

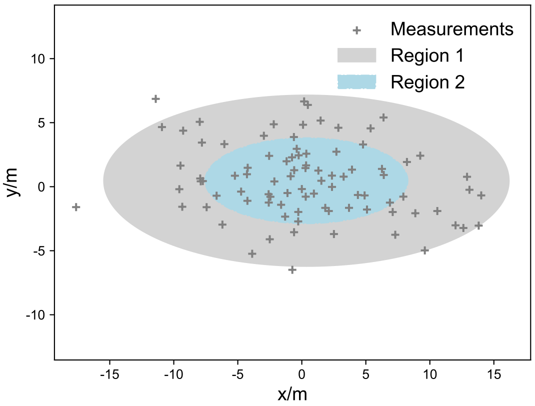


Figure 3. Resampling

necessary to appropriately process these data.

When measurement information at time k has not yet been obtained, the measurements from time $k-1$ and earlier are first spatially aggregated, and all measurements are fixed near the origin of the Cartesian coordinate system. This approach enables a more effective utilization of historical measurement information. Subsequently, the aggregated historical information is resampled according to the method shown in Figure 3, which reduces computational load.

After the measurement information of time k is obtained, the measurement information of time k can be formed by combining it with the resampled historical measurement information. This allows for a more accurate estimation of the shape. The entire process is illustrated in Figure 4.

4.2 Construction of Neural Networks Model

There are numerous neural network models available, and in the elliptical extended target tracking covered in this paper, neural networks need to be utilized to achieve the estimation of the spatial range of the extended target. Convolutional neural network itself is designed for feature information extraction, so CNN is chosen to realize the feature extraction of the measurement information.

The construction of the network model is the basis for realizing the extended target shape range estimation, and the design of the network model will be described in detail in this subsection.

4.2.1 Design of Convolutional Layers

Convolutional layer is the key part of information extraction in the convolutional neural network model, in order to better accomplish the information extraction, the convolutional layer structure designed is shown in Figure 5 shown.

The most critical part of the convolutional layer is the convolution operation, by which the information can be initially extracted, and the extraction process is:

$$h \cdot g(n) = \sum_{\tau=-\infty}^{+\infty} h(\tau)g(n-\tau) \tag{22}$$

After that the extracted information is then processed by normalization, this operation is to remove the outliers in the obtained information, by normalization all the information can be controlled within a reasonable range for subsequent processing, the process is processed as:

$$\hat{Y}_i = \frac{Y_i - \mu_i}{\sigma_i} \tag{23}$$

Through the processing of these two steps, the basic processing of the information is basically completed, while the next by adding the ReLU activation function can be effective in retaining the learned function relationship, the ReLU function mapping is as follows:

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

In order to prevent the network from overfitting and at the same time to improve the information representation ability of the network, a maximum pooling layer is also added to the convolutional layer, which can be processed by the maximum pooling layer to improve the information fitting ability of the network even further. The construction of the network model is the foundation for achieving accurate shape estimation of the extended target.

4.2.2 Design of the full connectivity layer

After the obtained measurement data are processed by the convolutional layer, the information obtained is able to preliminarily characterize the spatial extent of the target, and then the extracted information needs to be further fitted to complete the final parameter fitting. In order to realize this purpose, a fully connected layer is designed to fit the parameters.

There are three layers in the fully-connected layer to realize the parameter fitting, which is calculated in each layer:

$$y_{out} = W \times x_{in} + b \tag{25}$$

where the $Tanh(\cdot)$ activation function is calculated as:

$$tanh(x_{in}) = \frac{e^{x_{in}} - e^{-x_{in}}}{e^{x_{in}} + e^{-x_{in}}} \tag{26}$$

The structure of FC-Layer is shown in Figure 6.

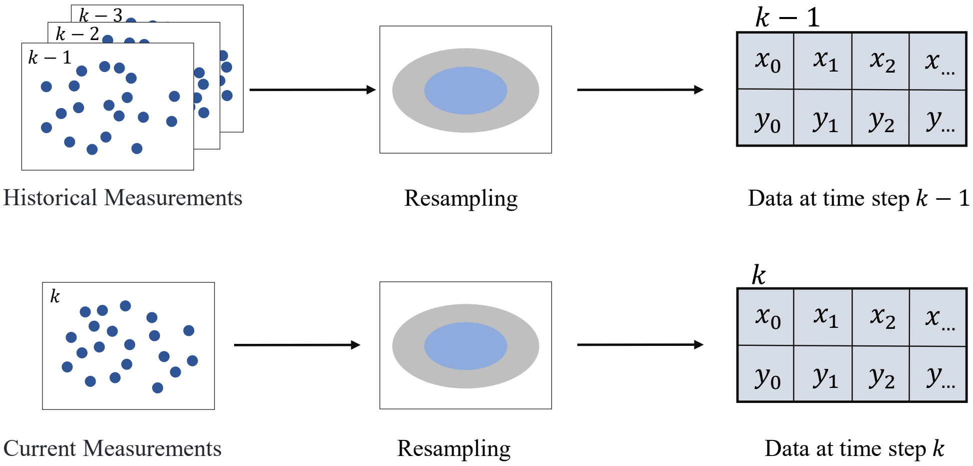


Figure 4. Measurement Information Transmission

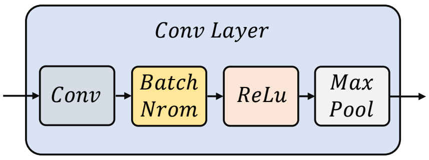


Figure 5. The structure of Convolutional layer

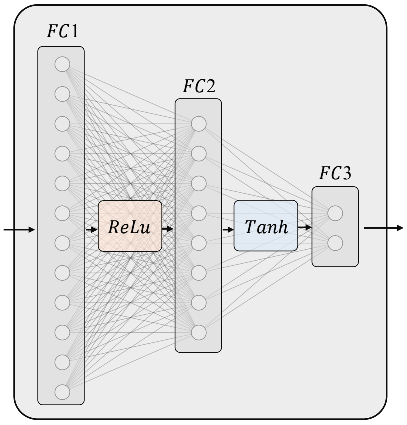


Figure 6. The structure of fully connected layer

4.2.3 Design of the overall network model

The excellent performance of a neural network model depends significantly on its structure. The model consists of three convolutional layers followed by three fully connected layers. The convolutional layers extract feature channels with sizes of 64, 256, and 512, respectively. The convolutional kernels have a size of 2×2, with strides set to 2, 2, and 1 for each layer. By utilizing the feature extraction capabilities of the convolutional layers, a wide range of feature information can be derived from the original measurement data. To stabilize the feature values, a Batch Normalization (BN) layer is applied after each convolutional layer, normalizing the extracted features before they are passed to the activation layer [4]. The ReLU activation function is used in each convolutional layer to enhance the model's non-linear fitting ability.

During the data processing stage, the measurement data is transformed into coordinate information with a shape of 2 rows and 256 columns, enabling it to be input into the convolutional neural network. However, it is important to note that the input measurement information may not contain sufficient feature information. To maximize the utilization and extraction of valuable features, a MaxPooling layer is incorporated after each convolutional layer. This layer effectively reduces computational workload and mitigates the risk of overfitting.

After feature extraction through the convolutional layers, the extracted feature information is effectively fitted through three fully connected layers. As the shape parameters and measurement information of the extended target may not satisfy a linear relationship, ReLU and Tanh activation functions are used in the fully connected layers. The ReLU activation function can avoid the problem of gradient disappearance in the backpropagation process and can shield negative values, which plays an important role in the subsequent estimation of shape parameters. The Tanh activation function is used between the second layer and the output layer. Since improper learning rate setting of the ReLU activation function may cause the neuron parameter value to be 0 during the training process, the Tanh activation function is used to update the data input to the output layer [19, 26, 29] to achieve the fitting of the nonlinear relationship and output the estimated values of the major and minor axes of the ellipse.

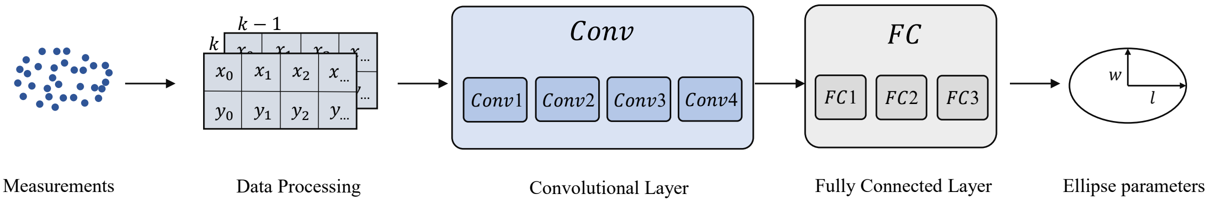


Figure 7. The process of estimating the extended target shape using CNN

The overall processing process is illustrated in Figure 7. The final elliptical profile parameters of the extended target are obtained as:

$$C_k(w, l) = \mathbf{CNN}(Z_k) \tag{27}$$

4.3 Model Training

The final performance of the neural network model depends largely on the quality of the dataset during the training process. A total of 100,000 measurements from 5000 time series at different moments in time were used to train the model, and all time series data were generated from specific targets in random scenarios.

In particular, each dataset of time series measurement data contains m measurements at each time point, and each measurement information at a time point contains n_k location information, where m is usually less than 200, and n_k follows a Poisson distribution. During the training process, 20 time points are randomly selected from each time series, resulting in the generation of 20 measurement datasets for each time series. For example, at moment k, the dataset corresponding to that time point will contain measurement data from k and all previous time points, which is then preprocessed with data standardization and equal−length processing.

During the training process, the sample label is the length and width corresponding to the real target. The loss function used is the mean squared error function (MSE). The learning rate is 5e-4, and the training process lasts for a total of 50 epochs. The batch size is set to 200. MSE is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{b}(Y_i - \hat{Y}_i)^2 \tag{28}$$

where Y_i is the actual data value and $\hat{Y}_i$ is the estimated value by the neural network model. The variation of loss values during training is shown in Figure 8.

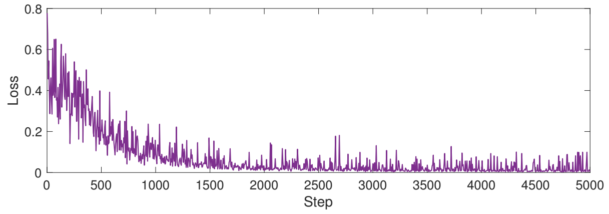


Figure 8. Change in value of losses

The computer configurations used during model training are as follows:

- Deep Learning Framework: Pytorch2.0.1
- CUDA: V11.2
- Operating System: Windows 10
- CPU: Intel i9-12900K@3.19GH
- GPU: NVIDIA GeForce GTX3090Ti

4.4 Designing Extended Target Tracking Methods

The estimation of kinematic state and spatial range of the extended target is accomplished in the previous section, and the tracking of the elliptical extended target can be finally realized by the effective integration of the previous sections. Eq. (11) and Eq. (27) together constitute the estimation of the elliptical spatial range and kinematic state of the extended target, and the final shape of the target can be estimated as:

$$m_k = r_k(\hat{x}_{k|k} + C_k(w, l)) \tag{29}$$

where r_k is rotation matrix, $\hat{x}_{k|k}$ is estimates of kinematic state, w is short axis of ellipse, l is long axis of ellipse.

With the above calculations, then the elliptical spatial extent of the extended target is finally obtained.

5 Experiment

5.1 Simulation Experiment Test

In this section, we will conduct a comprehensive simulation test on the proposed method CNN-ETT to verify the effectiveness of the proposed algorithm. During the test, the CNN-ETT algorithm will be comprehensively compared with two other excellent algorithms (Feldmann et al. 's method [9] and Simon et al.'s method [30]).

5.1.1 Feasibility testing

A. Scenario Construction

To validate the effectiveness of the proposed CNN model in this paper, we selected Feldmann's elliptical extended target tracking method based on random matrix and Simon's elliptical extended target tracking method based on neural network as the comparison methods. In this simulation scenario, we assume that there is a target moving nearly constant accelerating (CA) in a two-dimensional plane, and the actual target has a major axis of $120m$ and a minor axis of $60m$. The evolution model of the target's motion state at time k can be described as:

$$x_k = (F_k \otimes I_d)x_{k-1} + v_k, v_k \sim \mathcal{N}(0, \sigma_k) \tag{30}$$

where

$$F_k = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix},$$

and sampling time is $T = 200s$, sampling period is $\Delta t = 2s$.

The measurement and evolution models of the extended target at time k are as follows:

$$z_k^j = (H_k \otimes I_d)x_k + \omega_k^j \tag{31}$$

where the observation matrix and observation noise are defined as:

$$H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\omega_k^j \sim \mathcal{N}(0, \frac{1}{4}X_k + R)$$

$$X_k = \begin{bmatrix} 120 & 0 \\ 0 & 60 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

At time k, the number of observation data n_k follows a Poisson distribution with $p = 50$. The initial motion state of the target is:

$$\begin{aligned} x_0 &= \left[\ s_{x,0}, s_{y,0}, v_{x,0}, v_{y,0}, a_{x,0}, a_{y,0}\ \right]^{\mathrm{T}} \\ &= \left[\ 0, 0, 10, -10, 0, 0.01\ \right]^{\mathrm{T}} \end{aligned} \tag{32}$$

The motion direction of the target is consistent with the velocity direction, and all three methods have the same initial estimated value. Here the setting of the initial value of the target is accomplished through a priori knowledge:

$$\begin{aligned} m_0 &= \left[\ \hat{s}_{x,0}, \hat{s}_{y,0}, \hat{\theta}_0, \hat{l}_{1,0}, \hat{l}_{2,0}\ \right]^{\mathrm{T}} \\ &= \left[\ 0.5, 0.5, \pi/10, 100, 70\ \right]^{\mathrm{T}} \end{aligned} \tag{33}$$

B. Analysis of test results

In order to better evaluate the performance of the proposed method, a unified evaluation metric is necessary to evaluate different methods. For an ellipse, its shape parameters are typically composed of centroid coordinates, major and minor axes, and angle of rotation. In reference [37], a method was proposed to evaluate the similarity between two ellipses, called Gaussian Wasserstein distance:

$$\begin{aligned} &d_{GW}(m_x, m_{\hat{x}})^2 = \left\|s_{x,y} - s_{\hat{x},\hat{y}}\right\|^2 \\ &+ \mathrm{Tr}\left(\Sigma_e + \Sigma_{\hat{e}} - 2\sqrt{\sqrt{\Sigma_e}\Sigma_{\hat{e}}\sqrt{\Sigma_e}}\right) \end{aligned} \tag{34}$$

where

$$m_x = \left[\ s_x,\ \ s_y,\ \ \theta,\ \ l,\ \ w,\ \right]^T$$

$$\Gamma = \mathrm{rot}(\theta)diag(l^2, w^2)$$

rot refers to the rotation matrix and $\Sigma_e = \frac{1}{2}(\Gamma + \Gamma^{\mathrm{T}})$, m is the true value, and $\hat{m}$ is the estimation result.

Based on the set simulation scenario, the motion state and shape of the target were tracked over 100 sampling cycles. The results are shown in Figure 9 and Figure 10 (In the figure, to clearly illustrate the target's motion process, a plot was generated every 5 sampling periods.).

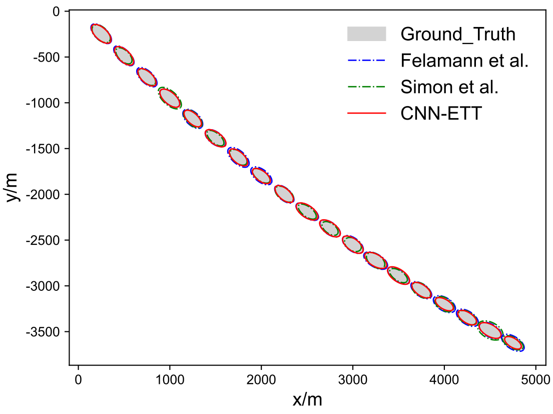
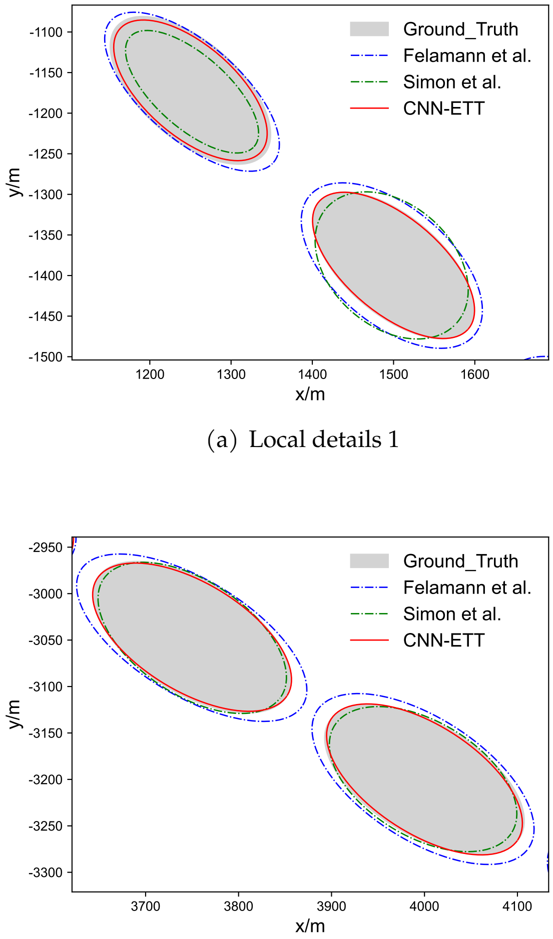


Figure 9. Tracking results under constant acceleration motion

Firstly, it can be very intuitively seen in Figure 9 that the three algorithms are able to accurately and efficiently



(a) Local details 1

(b) Local details 2

Figure 10. Partial detail

achieve tracking the kinematic state of the target, and their direction estimation is also consistent with the real state when the target makes a turn. So from the results in Figure 9, the CNN-ETT is able to achieve basic elliptical extended target tracking. Immediately after that, it is also obvious in the two local details shown in Figure 10 that the spatial range estimated by CNN-ETT is more accurate compared to the other two methods.

During the target motion, the performance metrics are evaluated for each sampling cycle according to Eq.(31), and Figure 11 shows the comparison of the performance computation results in this scenario, in which it is obvious that all three algorithms have more obvious convergence results during the tracking process. The method designed by Feldmann et al. can achieve convergence after about 20 sampling cycles, while the method designed by Simon et al. and CNN-ETT can achieve convergence after about 3 sampling cycles. Moreover, in this scenario, it can be clearly found that the CNN-ETT algorithm outperforms the remaining two methods. Therefore, after analyzing Figure 11, it can be finally determined that the method CNN-ETT proposed in this paper can be used for elliptical extended target tracking, and has a faster convergence speed as well as a good performance.

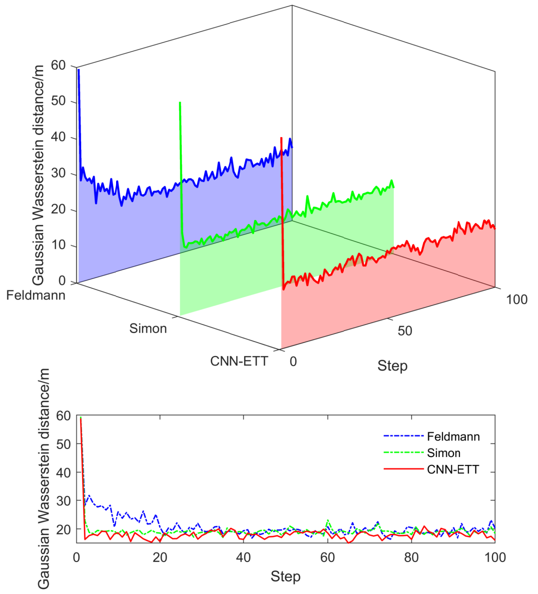


Figure 11. Gaussian Wasserstein distance for the 100 sampling periods

5.1.2 *Adaptability Test*

A. Scenario Construction

Tracking estimation of elliptically extended targets can usually be widely applied in various types of target tracking, so in order to validate the performance of testing the CNN-ETT algorithm under different target types, Monte Carlo method is used to test the performance of the algorithm in this section.

In the testing process, the number of Monte Carlo simulations is set to 100 times, and the target kinematic state parameters other than shape parameters are set as in Section 5.1.1. Because it is necessary to verify the performance under different target sizes in this section, a randomized way is used to generate the long axis, which takes the value range of $(1m, 150m)$, and the short axis takes the value of $0.4 - 0.8$ times of the long axis. In each simulation, a target range is randomly generated to finalize 100 Monte Carlo simulations.

B. Analysis of test results

The final test results are shown in Figure 12. The most obvious thing in the figure is the randomness in the testing process, which is in line with the testing requirements of Monte Carlo simulation.

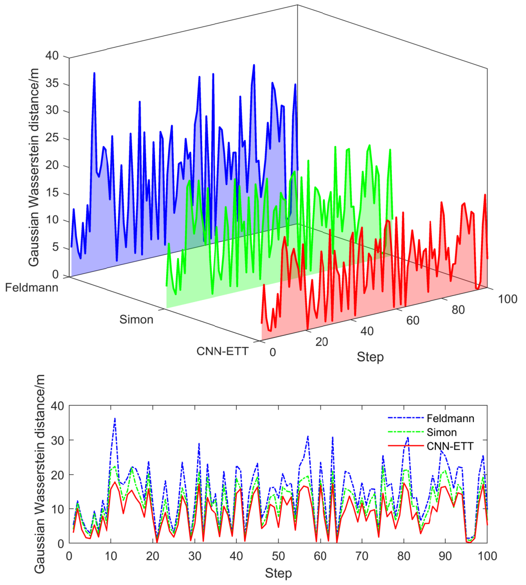


Figure 12. Performance comparison of the three methods in 100 Monte Carlo simulation experiments

It is immediately apparent from the figure that the method designed by Feldmann et al. does not perform as well as the other two methods, this is because the method designed by Feldmann et al. converges slowly when tracking the target, whereas the CNN-ETT method converges faster due to the use of the neural network in conjunction with the Kalman filter. The method designed by Simon et al. outperforms Feldmann's method but is slightly weaker than the CNN-ETT method, mainly because Simon's network converts the measurement data into an image, and some of the data is lost in the process, which affects the final performance.

5.1.3 Real Time Testing

A. Scenario Construction

In radar extended target tracking, because it is necessary to track the target in real time and accurately, and this puts strict requirements on the running speed of the algorithm, therefore, in this subsection the running speed of the algorithm will be tested.

Among the many factors that affect the speed of an algorithm, the number of radar measurements is one of the most significant. It is obvious that when the number of measurements increases, the amount of information to be processed will also increase, and the processing time required will also increase. Therefore, in order to obtain a more realistic and effective single-step running time, a total of 50 Monte Carlo simulations were performed. In each run, the motion parameters used are the same as described before, where the number of measurement data is a randomly generated number in each simulation with a fixed range of $(10, 70)$, and all three algorithms are run in the same computer equipment with the following parameters:

- Programming language: Python 3.9
- RAM: 16.0GB
- CPU: Intel(R) Core(TM) i5-8250U@1.60GHz
- GPU: NVIDIA GeForce MX150

B. Analysis of test results

The single-step average running time for 50 Monte Carlo simulations is shown in Figure 13.

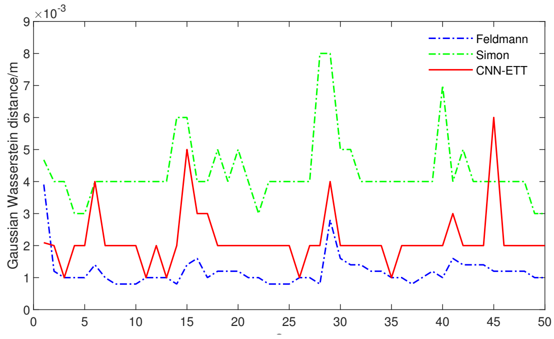
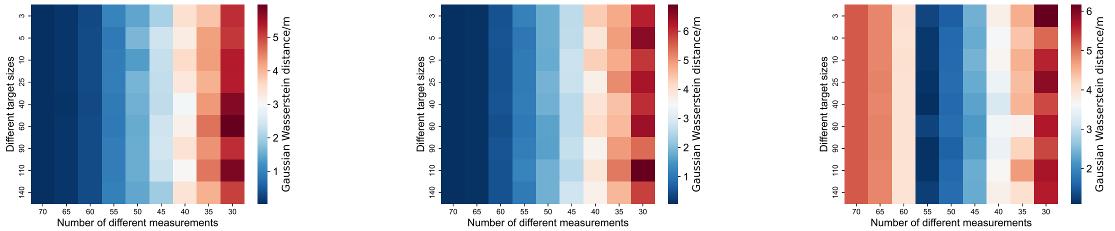


Figure 13. Single-step running time of the three methods

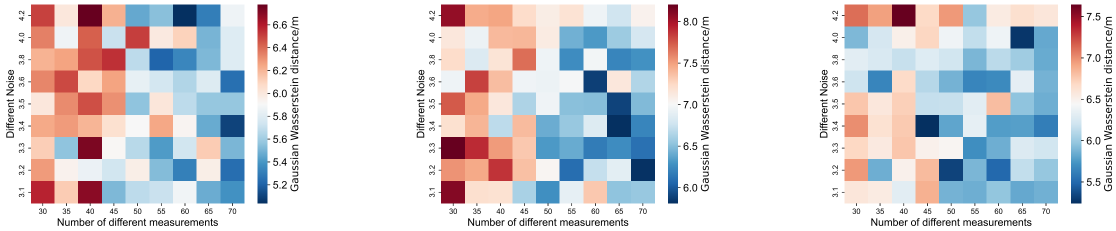
In Figure 13, the maximum single-step running time of the three methods does not exceed $8 \times 10^{-3}s$, which also satisfies the basic requirements of extended target tracking. Simon's method is slower than the other two methods because the overall running time averages $0.006s$ due to the fact that it is performed by converting the target measurements into a two-channel image during the tracking process. Feldmann's method is faster than the other two methods using neural networks because it uses an iterative estimation method, so its overall running time is around $0.001s$. CNN-ETT is faster than Simon's method because it is a direct data-driven method and does not have too much other processing.

After validation, it is proved that the method designed in this paper can also meet the requirements of target tracking in terms of operation speed, and the speed of CNN-ETT is improved compared with other neural networks.



(a) CNN-ETT (b) Feldmann (c) Simon

Figure 14. Performance at different sparsity levels



(a) CNN-ETT (b) Feldmann (c) Simon

Figure 15. Performance under different noise levels

Table 1. Comprehensive Performance Comparison

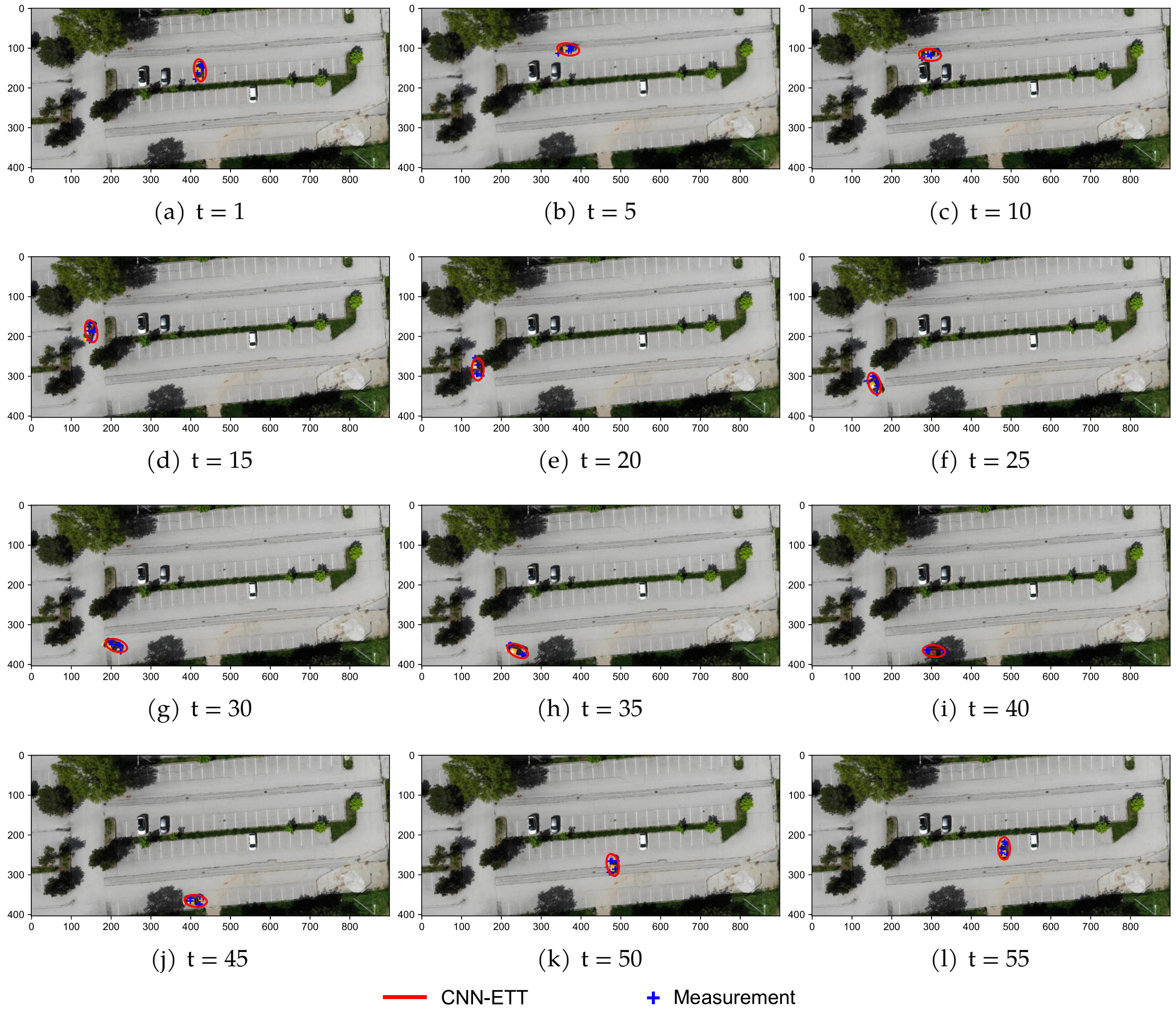
	CNN-ETT	Feldmann et al. [8]	Simon et al. [30]
Convergence period	1.5	20	2.5
(CNN-ETT vs.)	–	↑ 92.50%	↑ 40.00%
Average single-step running time	0.0039	0.0021	0.0047
(CNN-ETT vs.)	–	↓ -85.71%	↑ 17.02%
Performance under sparse measurements	5.5	6.5	6
(CNN-ETT vs.)	–	↑ 15.38%	↑ 8.33%
Noise Resistance	6.6	8	7.5
(CNN-ETT vs.)	–	↑ 17.50%	↑ 12.00%
Performance	18.12	20.85	19.58
(CNN-ETT vs.)	–	↑ 13.09%	↑ 7.46%

5.1.4 Robustness testing

A. Scenario Construction

In extended target tracking, the number of measurements, the range of target sizes, the sparseness of the measurements, and the amount of noise collectively affect the performance of the algorithm. Therefore, in order to further verify the stability of the algorithm under various conditions, the performance of the algorithm is comprehensively tested under different influencing factors.

The performance of the algorithm under different sparsity levels is firstly tested by setting different numbers of target measurements and different target size ranges. Firstly, the long axis of the target spatial range is varied from $3m$ to $140m$, and the short axis is $\frac{1}{2}$ of the long axis, the number of measurement points is increased from 30 to 70 in turn, and the motion parameters of the target as well as observation parameters are the same as the parameter settings in Section 5.1.1. On the other hand,



(a) t = 1 (b) t = 5 (c) t = 10
(d) t = 15 (e) t = 20 (f) t = 25
(g) t = 30 (h) t = 35 (i) t = 40
(j) t = 45 (k) t = 50 (l) t = 55

Figure 16. Tracking of moving vehicles

the resistance of the algorithm to noise is verified by setting different observation noise and different number of measurements. In this process, the number of measurement points is increased from 30 to 70, the noise covariance is varied from 3.1 to 4.2, and the motion parameters of the target as well as the observation parameters are the same as described before.

B. Analysis of test results

Figure 14 shows the test results of the algorithm under different sparsity levels. Figure 15 shows the test results of the algorithm at different noise intensities.

In Figure 14, the three subfigures correspond to the test results of the three different algorithms. First, it is clear from the figure that the performance of the algorithms decreases as the number of measurements decreases, but the effect of target size is not significant. Overall, the algorithms perform worse when the number of measures is sparser. Comparing the performance representation values on the rightmost side of the three subfigures, it is clear that the performance values range from $(0.0 - 6.0)$ for CNN-ETT, $(0.0 - 7.0)$ for Feldmann's method, and $(1.0 - 6.0)$ for Simon's method. In Figure 14(c), the performance of Simon's method decreases when the number of measurements is greater than 55, which is because the method is limited by the number of measurements and the image resolution during the image conversion process, and too many measurements will lead to a decrease in performance instead. CNN-ETT is directly data-driven and therefore does not have such a problem. In summary, the CNN-ETT algorithm can also have good performance for sparse measurement data.

In Figure 15, the most intuitive feeling is that the three methods are more sensitive to noise, and the overall performance is not as intuitive as that shown in Figure 14, which precisely proves the impact of noise on the performance of the algorithms in the process of extended target tracking. The performance test results of the three algorithms basically show that the performance is gradually getting better from the top left to the bottom right. On the other hand, in the corresponding numerical columns of the three algorithms, the CNN-ETT method has a numerical range of (5.0-6.8), the method designed by Feldmann et al. has a numerical range of (6.0-8.0), the method designed by Simon et al. has a numerical range of (5.5-8.0), and the performance of (5.5-8.0) is in the range of (5.5-8.0). The performance is in the range of (5.5-7.5), so even when more extensive performance tests are performed under different noises, the CNN-ETT algorithm still outperforms both outside methods.

The above stability test also proves that the CNN-ETT has excellent resistance to sparse measurements and noise, and also proves that the algorithm designed in this paper can be stably applied in various complex environments.

After the above validation finally the performance results can be obtained as shown in Table 1.

5.2 Tests in real scenarios

In the previous section, the effectiveness of the designed algorithm was verified through various experiments and was fully demonstrated. In this subsection, the CNN-ETT will be used for tracking tests in real scenarios to verify that it can be applied in real scenarios.

In addition, since the performance of CNN-ETT has been fully verified in the previous section, only CNN-ETT is tested alone in this subsection and is not compared with other methods.

5.2.1 Operating parameter

The test scenario performed has a vehicle moving in a parking lot at a constant turn rate. In this scenario, the tracking time is $T = 60s$, the sampling period is $\Delta = 1s$, the number of measurements obtained by the radar in each sampling moment during the movement of the vehicle is $p = 5$, and the initialized motion parameters are set to:

$$\begin{aligned} m_0 &= \left[\begin{array}{c} \hat{s}_{x,0}, \hat{s}_{y,0}, \hat{\theta}_0, \hat{l}_{1,0}, \hat{l}_{2,0} \end{array} \right]^{\mathrm{T}} \\ &= \left[\begin{array}{c} 450, 245, \pi/2, 3, 1.5 \end{array} \right]^{\mathrm{T}} \end{aligned} \tag{35}$$

5.2.2 Analysis of tracking results

Through the tracking results shown in Figure 16.

In figure, it can be verified that the proposed algorithm CNN-ETT is able to portray the spatial range of the target during the elliptical extended target tracking process, and when the vehicle is occluded during the vehicle movement, the measurement obtained is not accurate ($t = 40$), in this case the designed algorithm can still estimate the range of the target more accurately. The testing of the scenarios fully verifies that the algorithm can be used in real-life scenarios, which is useful in the fields of navigation and guidance, and predictive driving.

6 Conclusion

The innovation of this paper is to propose a new tracking method to solve the problem of method complexity when tracking elliptical extended targets. The method innovatively designs a convolutional neural network for estimating the spatial range of the extended target, and together with Kalman filtering achieves the joint estimation of the kinematic state and the approximate spatial range of the extended target. The effectiveness of the proposed algorithm is verified through sufficient experimental simulations.

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 statement

All data that support the findings of this study are included within the paper.

Acknowledgement

This work is supprted by the National Natural Science Foundation of China (62163023, 61873116, 62173266, 62366031)

References

[1] Abdallah, F., Gning, A., and Bonnifait, P. (2008). Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3):807–815. [CrossRef]

[2] Baum, M. and Hanebeck, U. D. (2009). Random hypersurface models for extended object tracking. In *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183. IEEE. [CrossRef]

[3] Baum, M. and Hanebeck, U. D. (2011). Shape tracking of extended objects and group targets with star-convex rhms. In *14th International Conference on Information Fusion*, pages 1–8. IEEE. [CrossRef]

[4] Baur, T., Reuter, J., Zea, A., and Hanebeck, U. D. (2022a). Extent estimation of sailing boats applying elliptic cones to 3d lidar data. In *2022 25th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE. [CrossRef]

[5] Buhren, M. and Yang, B. (2006). Simulation of automotive radar target lists using a novel approach of object representation. In *2006 IEEE Intelligent Vehicles Symposium*, pages 314–319. IEEE. [CrossRef]

[6] Cao, X., Lan, J., and Li, X. R. (2016). Extension-deformation approach to extended object tracking. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1185–1192. IEEE. [CrossRef]

[7] CHEN, H., ZENG, W., LIAN, F., and HAN, C. (2023). Non-star-convex extended target tracking algorithm for level-set gaussian process. *Journal of Electronics and Information Technology*, 45:1–10. [CrossRef]

[8] Feldmann, M. and Franken, D. (2008). Tracking of extended objects and group targets using random matrices—a new approach. In *2008 11th International Conference on Information Fusion*, pages 1–8. IEEE. [CrossRef]

[9] Feldmann, M., Fränken, D., and Koch, W. (2010). Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420. [CrossRef]

[10] Fowdur, J. S., Baum, M., and Heymann, F. (2021). An elliptical principal axes-based model for extended target tracking with marine radar data. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE. [CrossRef]

[11] Gao, L., Battistelli, G., Chisci, L., and Tesori, M. (2023). Superquadric-based 3d extended object tracking. *Available at SSRN 4385730*. [CrossRef]

[12] Gilholm, K. and Salmond, D. (2005). Spatial distribution model for tracking extended objects. *IEE Proceedings-Radar, Sonar and Navigation*, 152(5):364–371. [CrossRef]

[13] Gning, A., Ristic, B., and Mihaylova, L. (2011). A box particle filter for stochastic and set-theoretic measurements with association uncertainty. In *14th International Conference on Information Fusion*, pages 1–8. IEEE. [CrossRef]

[14] Govaers, F. (2019). On independent axes estimation for extended target tracking. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE. [CrossRef]

[15] Hammarstrand, L., Svensson, L., Sandblom, F., and Sorstedt, J. (2012). Extended object tracking using a radar resolution model. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2371–2386. [CrossRef]

[16] Honer, J. and Kaulbersch, H. (2020). Bayesian extended target tracking with automotive radar using learned spatial distribution models. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 316–322. IEEE. [CrossRef]

[17] Hui, C., JinRui, D., and ChongZhao, H. (2018). An adaptive tracking algorithm for irregular shape extended target. *Control Theory and Applications/Kongzhi Lilun Yu Yinyong*, 35(8):1–12. [CrossRef]

[18] Hui, C., Li, W., and ChongZhao, H. (2022). Student's t inverse wishart filter based on random matrix modeling. *Control Theory and Applications/Kongzhi Lilun Yu Yinyong*, 39(6):1–11. [CrossRef]

[19] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in neural information processing systems*, 30. [CrossRef]

[20] Koch, J. W. (2008). Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059. [CrossRef]

[21] Lan, J. (2023). Extended object tracking using random matrix with extension-dependent measurement numbers. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–16. [CrossRef]

[22] Lan, J. and Li, X. R. (2012). Tracking of extended object or target group using random matrix—part i: New model and approach. In *2012 15th International Conference on Information Fusion*, pages 2177–2184. IEEE. [CrossRef]

[23] Lan, J. and Li, X. R. (2014a). Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix. *IEEE Transactions on Signal Processing*, 62(9):2450–2463. [CrossRef]

[24] Li, M., Lan, J., and Li, X. R. (2023). Tracking of elliptical object with unknown but fixed lengths of axes. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–14. [CrossRef]

[25] Li, Z., Wang, H., Yan, S., Zou, H., and Du, M. (2022). Distributed extended object tracking filter through embedded admm technique. In *2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–6. IEEE. [CrossRef]

[26] Lu, L., Shin, Y., Su, Y., and Karniadakis, G. E. (2019). Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*. [CrossRef]

[27] Lyu, X., Wu, S., Ca, R., Zheng, X., and Xie, Y. (2023). Spawning extended target tracking based on ggiw-pmb. *Computer Systems and Applications*, 32(5):220–226.

[28] Nezhadarya, E., Liu, Y., and Liu, B. (2019). Boxnet: A deep learning method for 2d bounding box estimation from bird's-eye view point cloud. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1557–1564. IEEE. [CrossRef]

[29] Povey, D., Zhang, X., and Khudanpur, S. (2014). Parallel training of dnns with natural gradient and

parameter averaging. *arXiv preprint arXiv:1410.7455*. [CrossRef]

[30] Steuernagel, S., Thormann, K., and Baum, M. (2022). Cnn-based shape estimation for extended object tracking using point cloud measurements. In *2022 25th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE. [CrossRef]

[31] Swain, A. and Clark, D. (2010). Extended object filtering using spatial independent cluster processes. In *2010 13th International Conference on Information Fusion*, pages 1–8. IEEE. [CrossRef]

[32] Tan, J.-T., Qi, G.-Q., Qi, J.-J., Yang, Y.-J., Li, Y.-Y., and Sheng, A.-D. (2022). Model parameter adaptive approach of extended object tracking using random matrix and identification. In *2022 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, pages 91–97. IEEE. [CrossRef]

[33] Thormann, K., Yang, S., and Baum, M. (2020). A comparison of kalman filter-based approaches for elliptic extended object tracking. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE. [CrossRef]

[34] Xia, Y., García-Fernández, Á. F., Meyer, F., Williams, J. L., Granström, K., and Svensson, L. (2022). Trajectory pmb filters for extended object tracking using belief propagation. *arXiv preprint arXiv:2207.10164*. [CrossRef]

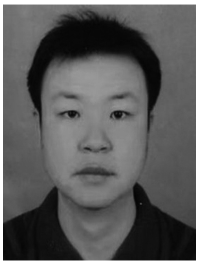
[35] Yang, J. L., Li, P., & Ge, H. W. (2014). Extended target shape estimation by fitting b-spline curve. *Journal of Applied Mathematics*, 2014(1), 741892. [CrossRef]

[36] Yang, S. and Baum, M. (2019). Tracking the orientation and axes lengths of an elliptical extended object. *IEEE Transactions on Signal Processing*, 67(18):4720–4729. [CrossRef]

[37] Yang, S., Baum, M., and Granström, K. (2016). Metrics for performance evaluation of elliptic extended object tracking methods. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 523–528. IEEE. [CrossRef]

[38] Zhang, Y., Ji, H., and Hu, Q. (2017). A box-particle implementation of standard phd filter for extended target tracking. *Information Fusion*, 34:55–69. [CrossRef]

Binchao Bian received his bachelor's degree from Lanzhou University of Technology. He is currently a graduate student at the School of Electrical Engineering and Information Engineering, Lanzhou University of Technology. His research interests include radar target tracking and neural networks. (Email: bianbinchao@lut.edu.cn)



Hui Chen received the M.S. and the Ph.D. degrees in control science and engineering from Xi'an Jiaotong University, Xi'an, China. He is now a Professor in the School of Electrical and Information Engineering, Lanzhou University of technology, Lanzhou, China. During 2018-2020, he did research on data fusion at the Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, USA. His current research interests include data fusion, statistical signal processing, machine learning and intelligent decision-making. (Email: chenh@lut.edu.cn)