ISSN: 2688-8645

Design of Fast Radio Burst Signal Recognition System based on Deep Learning

Haoran Yuan, Chungao Shi

Faculty of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, China

Abstract

In order to identify fast radio burst signals from the original observation data of FAST radio telescopes, this paper designs a fast radio burst signal recognition system based on deep learning object detection algorithm. The system uses the incoherent achromatization algorithm and the YOLO series target recognition algorithm to realize the recognition of fast radio burst signals, and provides users with a friendly graphical system interface. In view of the different performance of users' computers, the system has the function of selecting different algorithm models. Experiments have proved that the system achieves 86% recall and 83% accuracy in the FRB20201124A real-world data test set.

Keywords

Object Detection; Incoherent Dispersion; Fast Radio Bursts; YOLO.

1. Introduction

Fast radio bursts (FRB) are astronomical phenomena of highly dispersive, pulsed, radio radiation that lasts only milliseconds [1]. This astronomical phenomenon was first discovered in 2007 by Lorimer et al. when they analyzed historical data from pulsar surveys at the Parkes Observatory in Australia[2]. Fast radio bursts have also been observed by the Fivehundred-meter Aperture Spherical Radio Telescope (FAST) [3]. The manual processing of observation data is not efficient, so automatic identification technology needs to be used to improve work efficiency.

Deep learning object detection is an efficient automatic recognition method for FRB signal recognition. In this method, the frequency time intensity data is extracted from the observation data file, and then the incoherent dispersion algorithm is used to generate the dispersion time intensity data. Thirdly, the method utilizes the sliding window method to plot the dispersion time intensity image, where the FRB signal is in the shape of a bow. Furthermore, the method uses the deep learning object detection model to perform image classification and bounding box regression. Finally, the method uses the bounding box coordinates to calculate the position of the core of the "bowtie", and then obtains the occurrence time and dispersion value of the FRB signal.

Compared with the traditional software system, the FRB signal recognition system based on deep learning object detection has higher accuracy and broader application prospects. Traditionally, an automated, high-performance software system based on the achromatic dispersion theory has been used to search for FRB signals. Such as HEIMDALL [4], Amber [5], and Presto [6]. Due to the effects of RF interference, these software systems are challenged with false positives due to noise and RF interference. The FRB signal recognition system based on deep learning object detection can identify FRB signals more accurately and robustly, so as to obtain better performance.

ISSN: 2688-8645

In this article, we will first introduce the overall framework design of the system in Chapter 2. Subsequently, in Chapter 3, this paper will delve into the design of basic data processing algorithms, including key steps such as incoherent dispersion algorithms and sliding window drawing algorithms. Then, in Chapter 4, this article will explain the design principles and performance test results of the recognition algorithm. In Chapter 5, this article describes the methods and results of system testing. Finally, this article will summarize the system design process and look forward to possible future improvements and expansions.

2. System Framework Design

The panorama of the system is shown in Figure 1, the FRB signal recognition system integrates a complete workflow, including the preparation of training data, the training of the recognition model and the inference analysis of the model, which is mainly composed of three parts: data production module, model training module and model inference module. The data manufacturing module provides a dataset for the model training module, and the model training module provides a trained model for the model inference module. The model inference module has a friendly graphical system interface and is responsible for providing users with interactive model inference services.

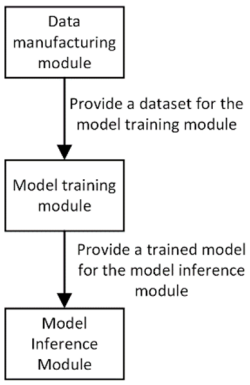

Fig 1. Panorama of the FRB signal recognition system

2.1. Data Manufacturing Module

The framework diagram of the data manufacturing module is shown in Figure 2, and making data is a key process to prepare the dataset for the model training module. Before proceeding with this process, the user needs to prepare the observation data in advance. This data is usually saved in the FITS file format, which is a file format specifically designed for astronomical data to store and exchange data. In addition, the user needs to fill in the detailed information about the FRB signal accurately in the database.

This module begins with an important step: checking the data integrity and user-submitted information for consistency, and stopping the workflow as soon as errors or mismatches are found. After verifying that the data is correct, the module continues its workflow to extract key observations from the observation data file. Next, the user is prompted to select a function: to process real data or to make simulated data. If the user decides to create analog data, the module will first exclude the FRB signal data from the observed data and create background data with only RF interference and negative samples. The module then injects FRB signal data into the background data and writes the information of the FRB signal to the database. After the user chooses to process the real data or complete the analog FRB signal injection, the module will perform incoherent dedispersion processing with a high-efficiency graphics card based on the information stored in the database to obtain the dispersion time intensity data after the dispersion is corrected. Once this data has been processed, the module performs the final task, which is to plot the dispersion temporal intensity using the sliding window technique. This

ISSN: 2688-8645

process is not only efficient, but also provides an accurate and reliable view of the data for model training.

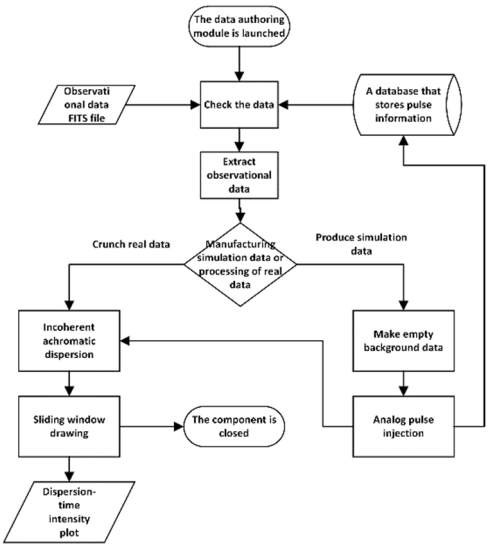

Fig 2. Framework diagram of the data manufacturing module

2.2. Model Training Module

Fig. 3 shows the framework diagram of the model training module, which is based on the MMYOLO toolbox [7], which is an open-source algorithm tool of the YOLO series based on PyTorch and MMDetection. Users only need to prepare a dataset in accordance with the COCO format and write a configuration file to complete the model training easily. The workflow of the toolbox is as follows: the user-submitted configuration file and dataset are first checked for errors, and if there are errors, the training is stopped. Then, the object detection model is trained based on the information of the configuration file. After the training is completed, the generated weight file can be converted from PyTorch format to ONNX format by the EasyDeploy tool, so that it can be deployed using the TensorRT framework later.

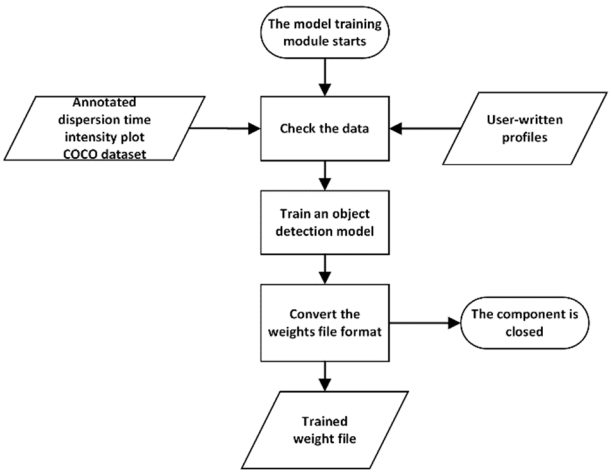

Fig 3. Framework diagram of the model training module

2.3. Model Inference Module

As shown in Figure 4, the model inference module is a powerful tool that provides a variety of model algorithms for users to choose from, including three types of YOLOv8n, YOLOv8s, and YOLOv8m. Depending on individual needs and computing resources, users can freely choose the appropriate model and its weight file to ensure optimal object detection performance. When the user selects a specific observation file or folder and triggers the inference process, the module first verifies that the selected file is indeed the observation. The uniqueness of the

ISSN: 2688-8645

file is then checked by the MD5 encryption algorithm, and if it is found that the file has been pre-processed, the result of the processing that is already in the database is immediately available, thus avoiding duplication of effort. If the file is not inferred, the module proceeds to the next step, which is to extract key information from the observed data and perform incoherent dedispersion processing in the range of 100 pc·cm-3 to 2000 pc·cm-3 for Dispersion Measure (DM). Next, the sliding window technique is used to generate a dispersion time intensity plot for each observation file for further analysis. On this basis, with the help of selected object detection models, FRB signals can be quickly and comprehensively identified from the data. The dispersion value and signal-to-noise ratio of each FRB signal are carefully calculated, and detailed FRB signal information is recorded in a database to lay the foundation for future screening work and scientific research. In addition, the model inference module is equipped with a QT5-based graphical user interface, where the results of all back-end processing will be clearly displayed, allowing users to easily view and analyze the data. This not only makes the user's operation smooth and convenient, but also greatly improves the work efficiency.

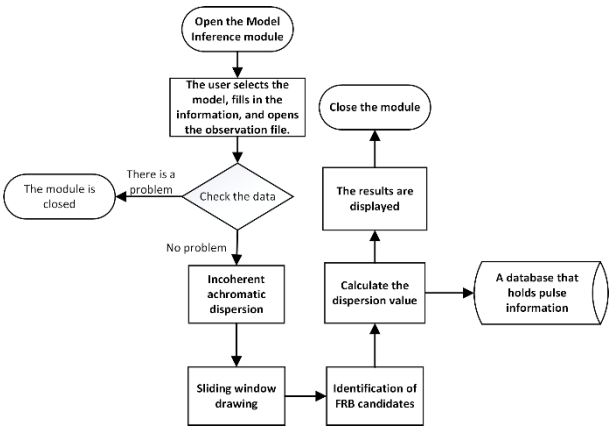

Fig 4. Model inference module framework

3. Basic Data Processing Algorithm Design

3.1. Incoherent Achromatic Dispersion Algorithm

The training data of the object detection model used in this system are derived from public datasets [8] and FRB20201124A [9]. Typically, the observation file of each FAST contains data from 4096 channels, 131072 sampling intervals, and four polarization modes. The sum of horizontally polarized and vertically polarized data is used as the frequency time intensity observation data.

When the FRB signal propagates in space, the speed is reduced due to the influence of interstellar dispersion, and the propagation speed of high-frequency radio waves is faster than that of low-frequency, so the time for high-frequency and low-frequency electromagnetic waves to reach the radio telescope is inconsistent, and the high-frequency signal arrives first and the low-frequency signal arrives later. Dispersion effects can lead to reduced frequency resolution, signal distortion, and impaired interferometric observations in radio observations. The incoherent dispersion algorithm calculates the delay time for each channel according to the dispersion formula, then adds the delay of each channel and superimposes all channels on top of each other to eliminate the dispersion effect of the data. In recent years, with the increase of dispersion, data channels, and sampling, the computational complexity of incoherent dispersion algorithms has increased dramatically, and traditional computing platforms cannot meet the demand. The programmability and parallel processing capability of the graphics processor are continuously improved, and the data processing capacity of the whole system is significantly improved after the addition of the central processing unit + graphics processor

ISSN: 2688-8645

hybrid computing system. High-performance GPU clusters can provide powerful computing resources to meet the real-time processing requirements of massive astronomical data and solve the problem that achromatic algorithms are computationally large and cannot be processed in real time [10]. The signal after adjusting the time delay is summed at all effective frequencies to obtain the dispersion time intensity data.

The principle of incoherent achromatic dispersion is shown in Equation (1), Equation (2) and Figure 5. Figure 5 illustrates the process of incoherent achromatization, Figure 5(a) is not achromatic, Figure 5(b) is the result of the achromatic process, and the horizontal and vertical axes of Figure 5 are the time and frequency axes, respectively.

$$t_2 - t_1 = D \times \left(\frac{1}{v_2^2} - \frac{1}{v_1^2}\right) \times DM \tag{1}$$

$$t_{end} = t_{start} + D \times \left(\frac{1}{v_{start}^2} - \frac{1}{v_{end}^2}\right) \times DM \tag{2}$$

Equation (1) is the dispersion delay formula, D is the dispersion constant, D = 4.15times103 MHz2 ·pc-1 ·cm3 ·s, DM is the measured dispersion amount, v1 and v2 are the two center frequencies, and (t2 - t1) are the delay of the two center frequencies. As shown in equation (2), the end time of the FRB signal can be found by using the dispersion delay formula under the premise that the occurrence time, frequency range and dispersion value of the FRB signal are known.

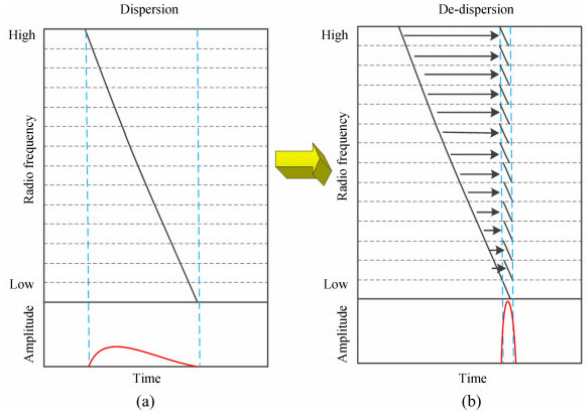

Fig 5. Schematic diagram of the incoherent dispersion algorithm

3.2. Sliding Window Drawing Algorithm

An example of a dispersion time intensity plot is shown in Figure 6. In the process of testing, this paper finds that the original recognition model is not ideal for the recognition of the "bow" pattern in the edge area of the image. In order to improve the accuracy, this paper decided to use the sliding window method for drawing. First, the width of the sliding window is set to 7200 sampling intervals and the height is 360 pc·cm-3, and the sliding step is set to 6200 sampling intervals horizontally and 310 pc·cm-3 vertically. Then, in the direction of increasing time and the amount of dispersion, the intensity of each dispersion time is plotted step by step.

Additional processing of image data is required before and after drawing. Before image drawing, the image data is decentralized and standardized, and different features are ensured to have the same scale in deep learning, which accelerates the convergence speed of gradient descent and improves the stability and efficiency of model training. After the image is drawn, add the name of the source observation file, the start time of the image data, and the start value of the dispersion measurement of the image data in the metadata partition of the PNG image, so that the occurrence time of the FRB pulse signal and the DM value of the FRB signal can be calculated during the image inference process.

ISSN: 2688-8645

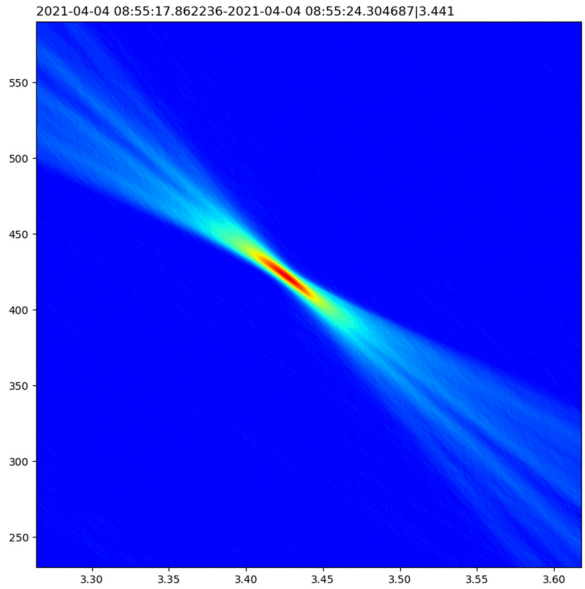

Fig 6. Dispersion time intensity plot of a real FRB20201124A pulse

3.3. Make Analog Data Algorithms

Currently, the number of FRB data samples available for deep learning model training is still limited. Although there are similarities between pulsar single pulse signals and FRB signals in the Milky Way, directly using such pulsar signals as training data may lead to the problem of overfitting the model to pulsar characteristics [11]. In this study, based on the simulation method proposed by Connor et al., the convolution operation of Gaussian function and scattering profile is used to construct FRB pulse profiles on different frequency channels [11]. Subsequently, it is embedded in real astronomical observation data that contains only noise signals and interference signals.

In this study, 9000 simulated FRB samples with DM values ranging from 100 pc·cm-3 to 2000 pc·cm-3 were generated and stored in a database, as shown in Table 1.

Table 1. Parameters for creating simulated FRB signals

Parameter	Distribution	Range
Fluence (Jy ms)	Log-normal	μ=3.5, σ=1
DM (pc cm-3)	Uniform	100, 2000
Width (ms)	Uniform	0.5, 50
Spectral Index	Uniform	-4,4
Scattering Timescale	Uniform	0, Width

4. Design and Implementation of FRB Recognition Algorithm

The FRB signal recognition model is derived from the YOLOv8 series. YOLO means that you only need to browse once to identify the category and location of the object in the picture. YOLOv8 is an efficient object detection algorithm that uses a single deep learning model to achieve end-to-end prediction from image input to object bounding boxes and category labels. Through a neural network, it segments the image into grids and simultaneously performs confidence scoring and categorical probability calculations on multiple bounding boxes within the grid. YOLOv8 uses multi-scale prediction and improved loss function to optimize detection accuracy, and applies non-maximum suppression to remove redundant detection to ensure a balance between real-time performance and high accuracy, which is suitable for object detection tasks in various real-world scenarios.

ISSN: 2688-8645

In this study, 9000 simulation data were used as the training and validation sets. In addition, 1000 positive samples with signal-to-noise ratios greater than 8 were selected from the FRB 20201124A observation data of FAST in April and May 2021 as the test set. Table 2 lists the details.

Table 2. Design of training, validation, and test sets

type	quantity	remark
Training set	7000	Simulated data
Validation set	2000	Simulated data
Test set	1000	Real data

Table 3 shows the performance test results of the three algorithms in the FRB20201124A real data test set, and the accuracy-recall curve of YOLOv8n is shown in Figure 7. When selecting the object detection model, accuracy (P) and recall (R) are often used as evaluation indexes, and are calculated by equations (3) and (4). Among them, TP stands for the true example, that is, the number of targets correctly detected by the model; FP stands for false positive, i.e., the number of non-targets that are incorrectly predicted as targets; FN stands for false negative examples, i.e., the number of misjudged targets as non-targets.

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP+FN} \tag{4}$$

Table 3. Performance test results of the three algorithms

model	P	R
YOLOv8n	83.83%	86.71%
YOLOv8s	83.81%	87.10%
YOLOv8m	84.63%	87.13%

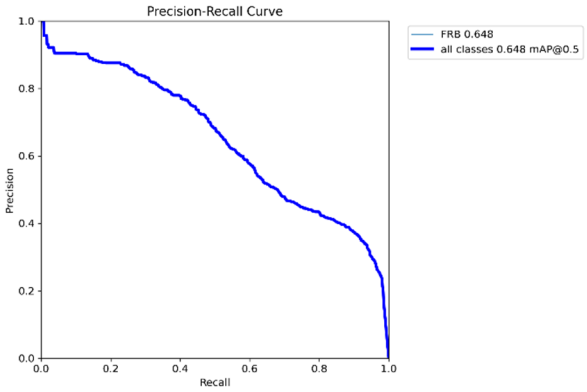

Fig 7. Accuracy-recall curves of YOLOv8n

The occurrence time and dispersion value of the FRB signal are obtained by coordinate calculation. Firstly, the object detection model is used to infer the dispersion time intensity image, and the coordinates (x, y) of the four corners of the FRB pulse and the FRB pulse bounding box are obtained. Again, the exact location of the center point of the FRB pulse is determined by the average calculation of the coordinates of the upper left and lower right corners. Next, the PNG image metadata was extracted to obtain the start time of the image data (t0) and the starting value of the dispersion measurement of the image data (dm0). Finally, as shown in Equations (5) and (6), the occurrence time and dispersion values of the FRB signal are calculated.

ISSN: 2688-8645

$$t = \left| t_0 + \frac{(x-64)\times 7200}{720} \right| \times tsamp \tag{5}$$

$$dm = dm_0 + \frac{(y-40)\times 360}{720} \tag{6}$$

The tsamp in Eq. (5) is the sampling interval, which is typically $4.9152 \times 10^{-5} s$.

5. System Testing

The user can identify the FRB signal in a few steps. Click on the "Select Model" drop-down menu to pick the model you need. In the file selection screen triggered by the Select Observation FITS File" button, select the observation file or the folder that includes the observation file. If the dispersion value of the target is known, please fill in the dispersion value input box (100 pc·cm-3 - 2000 pc·cm-3 by default), which can help speed up the process of incoherent dispersion and inference. After clicking the "Inference" button, the model inference module will lock the buttons, drop-down menus, and input boxes mentioned above, and start model inference. At this time, the System Status Display will update the inference progress of the file in real time. The File List display box will list the names of all observed files that have been inferred (each file is recorded only once); The "FRB Signal List" will add information about the identified FRB signals. The main interface is shown in Figure 8.

Once the inference is complete, the user can view information about the fast radio burst pulsed signal. Users can double-click on any row in the FRB Signal List to view the labeled dispersion time intensity image, as shown in Figure 9. The user can double-click on any row in the File File List to review the details of all FRB signals in the corresponding observation file, as shown in Figure 10. All FRB signal information will be stored in the SQLite database, and the information in the "File File List" and "FRB Signal List" will still be retained after the model inference module is restarted.

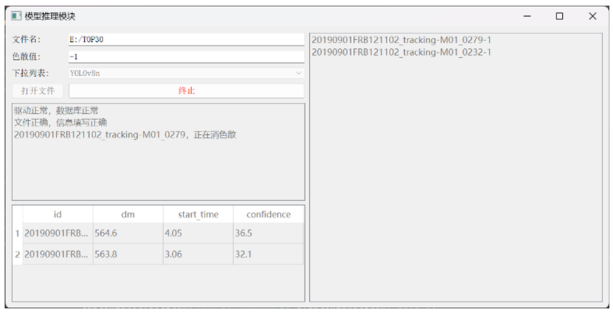

Fig 8. The main page of the model inference module

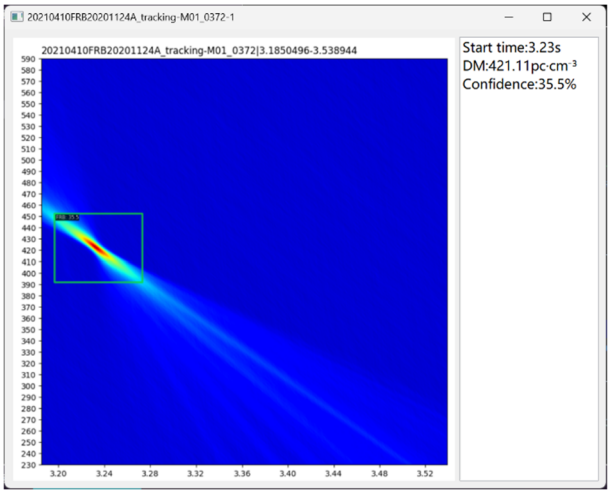

Fig 9. Viewing the labeled dispersion time intensity image

ISSN: 2688-8645

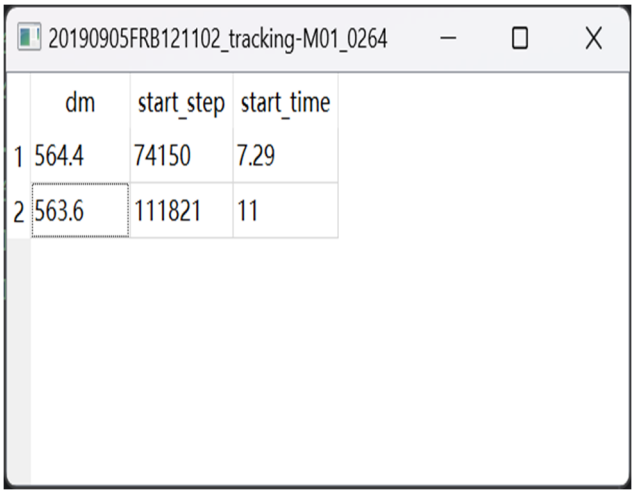

Fig 10. View the details of all FRB signals in the observation file

6. Conclusion

In this paper, a radio burst signal recognition system based on deep learning object detection algorithm is proposed, which aims to improve the accuracy of fast radio burst signal recognition in the original observation data of FAST radio telescope. The system adopts the incoherent achromatization algorithm and the YOLO series target recognition algorithm to effectively identify the FRB signal and provide a friendly graphical user interface, which is easy to operate. The system has the ability to select different algorithm models to adapt to different computer performance. The system was verified to exhibit 86% recall and 83% accuracy on FRB20201124A real-world data test set.

However, it was found that the performance of the algorithmic model needs to be improved, and the next step will focus on optimizing the dataset and improving the algorithmic model to improve the recall and accuracy. For datasets, richer and more diverse observational data will be collected, and data preprocessing techniques will be updated to improve data quality. For the algorithm model, the network architecture, parameter settings, and ensemble learning methods will be considered to improve the recognition effect. Through continuous efforts, it is believed that with the dual efforts of optimizing the dataset and improving the algorithm model, the recognition algorithm will be significantly improved, which will provide more reliable support for the research of radio burst signals.

References

[1] Liu Yanling, Chen Maozheng, Yuan Jianping. A review of fast radio burst search methods based on Machine learning [J]. Astronomical Research and Technology, 202, 19(05) : 509-517.

[2] Lorimer D R, Bailes M, Mclaughlin M A, et al. A bright millisecond radio burst of extragalactic origin [J]. Science, 2007, 318(5851): 777-780.

[3] Niu C H, Li D, Luo R, et al. CRAFTS for fast radio bursts extending the dispersion-fluence relation with new FRBs detected by FAST [J]. The Astrophysical Journal Letters, 2021, 909 (1) : L8.

[4] Barsdell B R, Bailes M, Barnes D G, et al. Accelerating incoherent dedispersion [J]. Monthly Notices of the Royal Astronomical Society, 2012, 422(1) : 379-392.

[5] Sclocco A, Leeuwen J V, Bal H E, et al. Real-time dedispersion for fast radio transientsurveys, using auto tuning on many-core accelerators[J]. Astronomy and Computing, 2016, 14 : 1-7.

[6] Ransom S M. New search techniques for binary pulsars[D]. Cambridge : Harvard University, 2001.

[7] MMYOLO Contributors. (2022). MMYOLO: OpenMMLab YOLO series toolbox and benchmark. Retrieved from https://github.com/open-mmlab/mmyolo.

ISSN: 2688-8645

[8] Xu Zhijun, An Tao, Guo Shaoguang et al. A fast radio burst data set for raw data search [J]. Science in China: Physics, Mechanics and Astronomy,2023,53(02):48-59. (in Chinese)

[9] Xu, H., Niu, J.R., Chen, P. et al. A fast radio burst source at a complex magnetized site in a barred galaxy. Nature 609, 685–688 (2022).

[10] Tohti Tinur, Zhang Hailong, Wang Jie, Ye Xinchen. Incoherent dispersion algorithm based on GPU. Astronomical Research and Technology, 21, 18(1) : 60-68.

[11] Connor L, Van Leeuwen J. AJ, 2018, 156: 256.