Research in Astronomy and Astrophysics, 23:104005 (9pp), 2023 October https://doi.org/10.1088/1674-4527/accdc2

Pulsar Candidate Classification Using a Computer Vision Method from a Combination of Convolution and Attention

Nannan Cai[1,2], Jinlin Han[1,2,3], Weicong Jing[1,2], Zekai Zhang[4], Dejiang Zhou[1,2], and Xue Chen[1,2]
[1] National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; nncai@nao.cas.cn, hjl@nao.cas.cn
[2] School of Astronomy, University of Chinese Academy of Sciences, Beijing 100049, China
[3] CAS Key Laboratory of FAST, NAOC, Chinese Academy of Sciences, Beijing 100101, China
[4] Department of Physics, Boston College, Massachusetts 02467, USA
Received 2023 January 8; revised 2023 March 29; accepted 2023 April 11; published 2023 September 13

Abstract

Artificial intelligence methods are indispensable to identifying pulsars from large amounts of candidates. We develop a new pulsar identification system that utilizes the CoAtNet to score two-dimensional features of candidates, implements a multilayer perceptron to score one-dimensional features, and relies on logistic regression to judge the corresponding scores. In the data preprocessing stage, we perform two feature fusions separately, one for one-dimensional features and the other for two-dimensional features, which are used as inputs for the multilayer perceptron and the CoAtNet respectively. The newly developed system achieves 98.77% recall, 1.07% false positive rate (FPR) and 98.85% accuracy in our GPPS test set.

Key words: (stars:) pulsars: general – methods: data analysis – techniques: image processing

1. Introduction

Pulsars are known as rotating neutron stars whose radiation beam sweeps across the line of sight. The radio signals of these pulsars are pulsed in a wide radio band, but they are dispersed due to free electrons in the interstellar medium. Therefore, signals at a lower frequency are more delayed. The signals are received by a radio telescope and then converted to digital signals after a series of signal conversions and digital processes, and finally are stored in a digital file.

Astronomers recognize pulsar signals via searching the periodicity (P) of pulses and the best dispersion measure (DM) of many trials for the delay compensation of pulses detected at different frequencies in the radio band. After data from many frequency channels in a radio band are de-dispersed (De-DM) and added together, one can find a possible period of a pulsar after the signals are analyzed via the Fast Fourier Transform (FFT) method. The analyses can be made in many packages. The most popular pulsar searching software is PRESTO.[5] When pulsar signals are recognized as a significant detection of the periodical signal from the long de-dispersed data set, the diagnostic plot can be produced via `prepfold`, the program for folding the original signals around the most probable period P and the DM. Therefore, the folded data produced by `prepfold` are recorded in a `pfd` file with the features for diagnostics, which consist of one-dimensional (1D) features: the folded pulse profile based on the period and DM, the change of signal-to-noise ratio (i.e., χ^2) around the most probable period and the change of signal-to-noise ratio around the DM. There are also diagnostics expressed in the two-dimensional (2D) features: the detected signal in the time-versus-phase plot, or the frequency-versus-phase plot, or the signal-to-noise ratio over the P-versus-$\dot{P}$ plot, see Figure 1 for an example.

Because of radio frequency interference (RFI), many such candidates are fake. In the past, astronomers had to pick out the true pulsars from all the candidates manually. With the development of new techniques, the data volume of pulsar searches has become larger and larger, and manual selection and viewing of pulsar candidates have become increasingly difficult.

Previously, many methods were developed to speed up the selection of candidates, such as summary interfaces approaches and Semi-automated ranking approaches, which have been summarized by Lyon et al. (2016). The identification of pulsars by using artificial intelligence (AI) is very efficient, see Table 1 for a list. Eatough et al. (2010) used a 12:12:2 artificial neural network (ANN) to evaluate twelve feature numbers of candidates. Bates et al. (2012) followed the approach and increased the feature number from 12 to 22. A similar approach was followed by Morello et al. (2014) for the High Time Resolution Universe (HTRU) survey. Zhu et al. (2014) combined four methods of machine learning, including support vector machine (SVM), ANN, convolutional neural network (CNN) and logistic regression (LR) to form the Pulsar Image-based Classification System (PICS) for recognizing pulse profiles, DM-curves, time-versus-phase plots and frequency-

[5] https://github.com/scottransom/presto

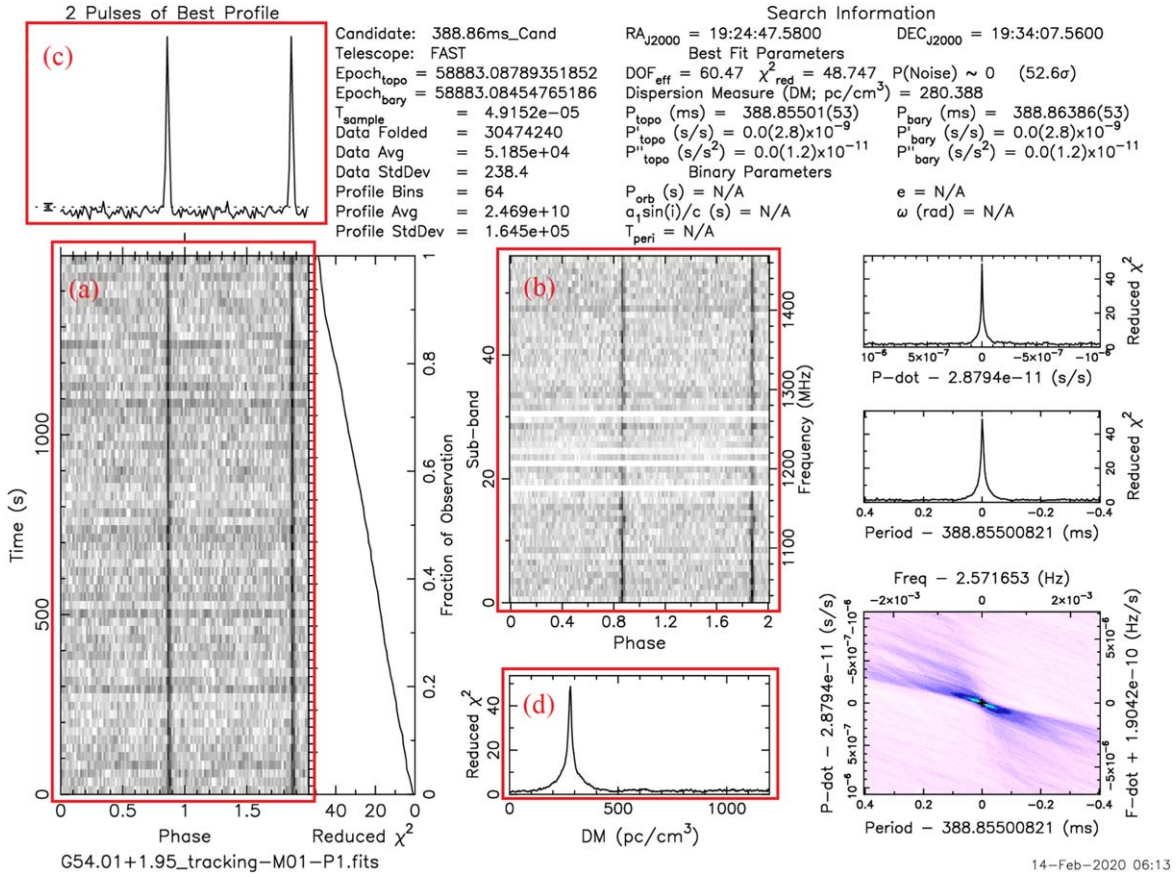

Figure 1. An example plot produced by `prepfold` for the pulsar J1924+1932g discovered by the GPPS survey. Four key data sets are taken from the `pfd` file for the AI program: (a) the time-versus-phase plot for the integrated pulse strength over all frequency channels for many subintegration durations along time; (b) the frequency-versus-phase plot for the integrated pulse strength over all time against the many subbands in the observational band; (c) the pulsar pulse profile integrated over all time and frequency channels against the pulsar rotation phase; (d) the curve for the best DM shown by the reduced χ^2 against DM.

versus-phase plots of pulsars. Wang et al. (2019a) changed the CNN module of PICS into a 15-layer residual CNN, which increased the depth of the network to improve accuracy and avoid the vanishing gradient caused by the increasing depth. Tan et al. (2018) developed an ensemble classifier consisting of five different decision trees using mean value, standard deviation, skewness and excess kurtosis extracted from the pulse profile, DM-curve, time-versus-phase plot and frequency-versus-phase plot. This approach was applied to radio pulsar search in the Low-Frequency Array (LOFAR) Tied-Array All-Sky Survey. A hybrid ensemble model combining Random Forest and XGBoost with EasyEnsemble was trained and tested on the HTRU data set (Wang et al. 2019b). Guo et al. (2019) employed a Deep Convolution Generative Adversarial Network (DCGAN) to generate more samples and learn features and adopted SVM to classify the candidates. Lin et al. (2020) designed a multi-input CNN which had a main input and an auxiliary input and utilized a Transforming Image and Adding Gaussian Noise (TIAGN) technique to augment data for solving the problem of training on a highly class-imbalanced data set. Lin et al. (2020) proposed the Grid Search and Recursive Feature Elimination for feature selection implementing the GBoost algorithm, using 18 feature numbers for candidate classification. Zeng et al. (2020) used a CNN to extract features from pulse profile, DM-curve, frequency-versus-phase plot and time-versus-phase plot, then concatenated these features horizontally or vertically for the input of the next layer of the network. Balakrishnan et al. (2021) used pulse profile, DM-curve, frequency-versus-phase plot and time-versus-phase plot to train a semi-supervised generative adversarial network. Yin et al. (2022) proposed an interesting network of a residual convolutional autoencoder (RCAE) combined with LR. It only needs to train the frequency-versus-phase plots and time-versus-phase plots of non-pulsars. Zhao et al. (2022) combined AdaBoost with the multi-input-CNN (MICNN) to form a new framework, and he added a convolutional block attention module (CBAM) to the MICNN. It also uses four features: pulse profile, DM-curve, frequency-

Table 1
Previous AI Approaches For Pulsar Identification

Method	Recall	FPR	Accuracy	F1	Reference
12:12:2 ANN	92%	⋯	⋯	⋯	Eatough et al. (2010)
22:22:2 ANN	85%	⋯	⋯	⋯	Bates et al. (2012)
6:8:2 ANN	99%	0.11%	⋯	⋯	Morello et al. (2014)
SVM+ANN+CNN+LR	92%	1%	⋯	96%	Zhu et al. (2014)
GH-VFDT	92.8%	0.5%	98.8%	⋯	Lyon et al. (2016)
	82.9%	0.8%	97.8%	⋯	
	78.9%	0.1%	99.8%	⋯	
Ensemble method with five Decision Trees	98.7%	0.5%	99.2%	⋯	Tan et al. (2018)
SVM+ANN+ResNet+LR	98%	⋯	⋯	92%	Wang et al. (2019a)
Random Forest+XGBoost	96.7%	⋯	96.9%	⋯	Wang et al. (2019b)
	92%	⋯	91.8%	⋯	
DCGAN+DeepF+SVM	100%	0%	⋯	⋯	Guo et al. (2019)
CCNN	98.16%	⋯	94.76%	⋯	Zeng et al. (2020)
AdaBoost+Multi-input CNN	95.6%	⋯	⋯	96.2%	Lin et al. (2020)
GS+RFE+GBoost	99%	0.102%	⋯	⋯	Lin et al. (2020)
Semi-supervised GAN	99.4%	1.6%	98.9%	⋯	Balakrishnan et al. (2021)
Res-conv autoencoder	99%	⋯	97.9%	⋯	Yin et al. (2022)
Ada-GBoost-MICNN	98.8%	⋯	98%	⋯	Zhao et al. (2022)
GAN+ResNeXt	100%	0%	100%	100%	Yin et al. (2022)

Note. Note that the recall, FPR, accuracy and F1 score are taken from the original references and should be interpreted with caution because they were obtained with different data sets.

versus-phase plot and time-versus-phase plot. The generative adversarial network and ResNeXt (Xie et al. 2016) are applied to pulsar candidate selection and the precision, recall and F1 score of 100% are obtained on the HTRU Medlat data set.

In the history of AI, convolutional architectures have always dominated the field of Computer Vision (CV) since LeNet (Lecun et al. 1998), AlexNet (Krizhevsky et al. 2012), VGGNet (Simonyan & Zisserman 2014) and other models made great achievements. The attention mechanism has been extremely popular in Natural Language Processing (NLP) since the Google Brain Team published their Transformer model (Vaswani et al. 2017). Recently, the attention mechanism has also become a research hotspot in CV. What is the difference between CNN and attention? A convolutional layer has a convolution kernel with fixed values for each input, and the convolution kernel is convolved with each local region of the input, and outputs a Feature Map. In other words, the convolutional layer gives each local region of each input the same weight. In contrast, the attention mechanism is designed to assign different parts of different inputs different weights inspired by a human brain's tendency to focus on certain parts and ignore others when reasoning about information. There are three types of attention mechanisms in the CV: (1) Spatial attention mechanism, such as non-local attention (Wang et al. 2017b), Spatial Transformer (Jaderberg et al. 2015); (2) Channel attention mechanism, such as Squeeze-and-Excitation Network (SENet) (Hu et al. 2017), Selective Kernel Network (SKNet) (Li et al. 2019); (3) Mixed attention mechanism, such as CBAM (Woo et al. 2018), Residual Attention Network (ResAttNet) (Wang et al. 2017a) and CoAtNet (Dai et al. 2021). So far, the attention mechanisms have not yet been widely used to identify pulsar candidates.

The Five-hundred-meter Aperture Spherical radio Telescope (FAST) is the largest single-dish radio telescope in the world (Nan et al. 2011). We are carrying out a big pulsar searching project, namely the FAST Galactic Plane Pulsar Snapshot (GPPS) survey (Han et al. 2021), and up to now, this project has discovered more than 500 pulsars.[6] In the effort, millions of pulsar candidates have been accumulated during the data processing. We take these as training and testing of new AI methods for pulsar selection. We compare the effectiveness of three methods involving the attention mechanism in identifying 2D features of pulsar candidates: Vision Transformer (Dosovitskiy et al. 2020), ResAttNet and CoAtNet. We find that CoAtNet proposed by the Google Brain Team tends to perform better than others at discriminating the 2D features. Therefore CoAtNet is chosen in this paper as the classifier for 2D features of candidates.

This paper combines CoAtNet, MLP and LR to build our new pulsar identification system. The remaining part of the paper proceeds as follows: The overall structure of the pulsar identification system is introduced in Section 2. The setting and results of experiments about data pre-processing, selection of data set and the option of model parameters are presented in Section 3. Finally, conclusions are given in Section 4.

[6] http://zmtt.bao.ac.cn/GPPS/GPPSnewPSR.html

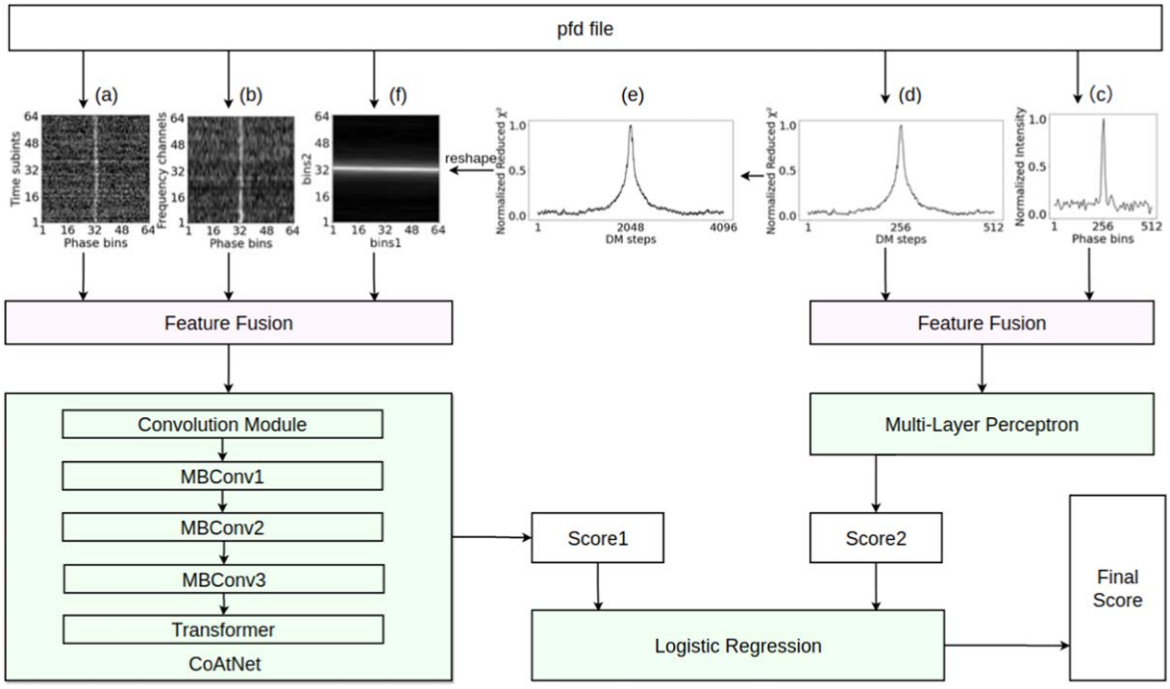

Figure 2. The flow chart for the pulsar candidate selection and evaluation system. The inputs come from the `pfd` file, including (a) the 64 × 64 frequency-versus-phase plot; (b) the 64 × 64 time-versus-phase plot; (c) the normalized pulsar pulse profile over 512 bins; (d) the normalized reduced χ^2 array for 512 DM steps. These data sets can have different sizes. Others are the converted data sets, such as (e) the normalized reduced χ^2 array calculated for 4096 DM steps; (f) the plot is reshaped by the normalized reduced χ^2 array which has length 4096 bins but is in two-dimensions. The features are fused and fed to the CoAtNet and an multilayer perceptron (MLP), and the outputs are further evaluated in terms of the LR for the final score.

2. New Approach For Pulsar Candidate Identification

The flow chart of our newly designed system for pulsar candidate identification using the machine learning method is presented in Figure 2.

We extract 2D features and 1D features from a `pfd` file of pulsar candidates for pulsar candidate identification. Feature fusion is used to combine multiple feature information into a feature. Two feature fusion ways have been made in our method. First, the feature fusion is carried out for three inputs of 2D image features (a, b, f), and the combined feature is fed into CoAtNet; second, feature fusion is carried out for two inputs of 1D features (d, c), and the combined feature is fed into an MLP.

Then, we take the outputs of the CoAtNet and MLP as the inputs of the model implemented using the LR method. Our method is named "CoAtNet-MLP-LR." The specific details of data processing by using "CoAtNet-MLP-LR" are introduced in Section 3. In this section, we want to describe the network architecture and theory of CoAtNet, MLP and LR.

2.1. CoAtNet

In the field of machine learning, the capacity and generalization ability of a model are critical factors for its success. The model capacity of a machine learning model refers to its ability to learn complex patterns from data, while the generalization ability refers to how well the model performs on unseen data. A critical issue for a model to have high generalization capacity is to avoid overfitting, a phenomenon in which a model memorizes the training data, leading to poor performance on new, unseen data. A Google Brain study (Dai et al. 2021) concluded that the convolutional module has better generalization, while the attention module has higher model capacity. To address this trade-off between model capacity and generalization, they proposed a novel model called CoAtNet, which marries the depthwise convolution and attention mechanism. By combining the generalization capacity of depthwise convolution with the model capacity of Transformer, CoAtNet is expected to achieve better performance on both capacity and generalization.

The depthwise Convolution block proposed by MobileNetV2 (Sandler et al. 2018)—MBConv is illustrated in Figure 3, which includes three steps: (1) expanding the channels of input using convolution with kernel size 1 × 1; (2) applying a convolution with kernel size 3 × 3 and pad 2; (3) compressing output channels using a convolution with kernel size 1 × 1. Each step is followed by an activation function, where GeLU (Hendrycks & Gimpel 2016) is used in our work. Depthwise convolution has a lower computational cost and

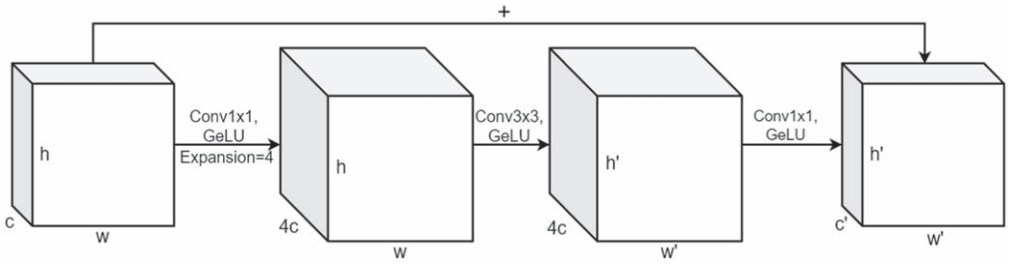

Figure 3. Detailed operation in the MBConv block.

smaller parameter size, and its inverted residual bottleneck design helps improve recognition accuracy.

In the Transformer block, CoAtNet replaces the self-attention mechanism with the relative-attention mechanism, which is a natural mixture of depthwise convolution and attention with minimum additional cost. The self-attention mechanism is the core of the famous Transformer model. It trains matrix W_Q, W_K, W_V and multiplies the input X to get matrix Q, K, V. Then it realizes "dynamic weight" for different input X by the following formula

$$Y = \text{softmax}(\frac{QK^T}{\sqrt{d_K}})V. \tag{1}$$

CoAtNet is a deep neural network architecture comprising five vertically arranged stages, denoted by $S0$, $S1$, $S2$, $S3$ and $S4$. While $S0$ uses a standard convolution block, $S1$ to $S4$ can adopt either an MBConv block or a Transformer block, with the former always preceding the latter. As such, CoAtNet offers five possible variants: (1) C-C-C-C; (2) C-C-C-T; (3) C-C-T-T; (4) C-T-T-T; (5) T-T-T-T (i.e., ViT_{REL}). According to a Google Brain experiment, the variants can be ranked as follows by model capacity: C-C-T-T $\approx$ C-T-T-T $>$ ViT_{REL} $>$ C-C-C-T $>$ C-C-C-C; for the generalization, the ranking is: C-C-C-C $\approx$ C-C-C-T $\geqslant$ C-C-T-T $>$ C-T-T-T $\gg$ ViT_{REL}. To achieve a balance between model capacity and generalization capacity, we chose to experiment with the C-C-C-T, C-C-T-T and C-T-T-T variants on our data set. Our experimental results indicate that the C-C-C-T variant is the most effective, as further detailed in Section 3.

2.2. Multi-layer Perceptron

A multi-layer perceptron is a type of fully connected feedforward ANN that consists of an input layer, an output layer, and one or more hidden layers. The input layer receives the input features, and each feature is connected to each artificial neuron node in the first hidden layer, resulting in a fully connected network. If we assume that the input feature is $X = [x_1, x_2, \ldots, x_n]$ and the first hidden layer has m artificial neuron nodes $(a_1, a_2, \ldots, a_m)$, the output of the input feature X after passing through the jth neural node is

$$h_j = g\left(\sum_{i=1}^{n} w_{i\,j} \cdot x_i + b_j\right). \tag{2}$$

Here, function g is the activation function, which introduces nonlinearity and enhances the learning ability of the neural network. The parameters w_{ij}, b_j are learned by the back-propagation algorithm during the training process. In our work, we use the Log Loss Function, i.e., Cross-Entropy Loss Function, as the loss function. Then, the gradient of the loss with respect to each parameter can be calculated by the chain rule. For instance, in a simple MLP with one hidden layer, the outputs of the hidden layer H and the output layer O can be computed as follows:

$$H = g^{(1)}(W^{(1)} \cdot X + B^{(1)}), \tag{3}$$

$$O = g^{(2)}(W^{(2)} \cdot H + B^{(2)}). \tag{4}$$

If the loss between the predicted output O and true labels Y is denoted by $L(Y, O)$, the gradient of W can be computed as follows:

$$\Delta W^{(2)} = \frac{\partial L}{\partial O} \cdot \dot{g}^{(2)} \cdot H, \tag{5}$$

$$\Delta W^{(1)} = \frac{\partial L}{\partial O} \cdot \dot{g}^{(2)} \cdot W^{(2)} \cdot \dot{g}^{(1)} \cdot X. \tag{6}$$

The neural network parameters are iteratively updated based on their gradients, allowing for fast and nonlinear learning. While the prediction accuracy of neural networks is generally high, it may be slightly lower than that of more advanced methods such as SVMs, which require more computationally expensive matrix calculations. In our pulsar identification task, we found that the frequency-versus-phase plot was the most important 1D feature. To identify this feature, we employed the simple and fast MLP algorithm. However, for 2D features, we used a relatively more complex algorithm. Specifically, the MLP block in our pulsar candidate identification system consisted of four hidden layers, with 2048, 4096, 1365 and 455 neuron nodes, respectively.

2.3. Logistic Regression

LR is a classification algorithm that predicts the probability of an event taking place. For the case of one or more independent variables and a single binary dependent variable, the model predicts the probability of a positive class as

$$p(y_i = 1|X_i) = \frac{1}{1 + e^{-(W\dot{X}_i + W_0)}}. \quad (7)$$

The parameter is estimated using maximum likelihood estimation, and it is not possible to find a closed-form solution that maximizes the likelihood function, so it must be searched using an iterative process. The goal of maximizing the log-likelihood $\ln L(W) = \sum[y_i \ln p_i + (1 - y_i)\ln(1 - p_i)]$ is equivalent to minimizing the cost function with regularization term $r(W)$

$$C\sum_{i=0}^{n}[-y_i \ln p_i - (1 - y_i)\ln(1 - p_i)] + r(W). \quad (8)$$

3. Experiments and Results

3.1. Data Set

The FAST GPPS survey project has accumulated tens of thousands of candidates including pulsars, their harmonics and fake candidates. We select some of them to form a data set, which is divided into a training data set and a testing data set. The training data set determines the upper limit of how well the model can learn, and the test data set is the standard to judge the performance of the model. The ratio of positive and negative samples (i.e., true pulsars and RFI) in all our training sets is approximately 1:1.

Experiments show that the data sets for the model are very important, which must be large enough and include different types of real pulsars including drifting, nulling, scattering, binary, intrinsic wide pulse and so on. The detected pulsar harmonics with multiple peaks are excluded from the data set. We include some pulsars with multiple peaks in the data set to ensure this type of pulsar can be identified.

A model with higher model capacity can capture more details of features in training data, but it is easier to overfit in the case of an insufficient amount of training data set. As a result, the generalization ability of the model is worse (i.e., the model performs well on the training data set but not on the testing data set). We experimented with training data of different sizes (2000, 6000, 8000, 12,000, 16,000), in order to find out how the data volume size affects the model capacity of our model. The experimental results are shown in Table 3.

The inputs of our pulsar candidate identification system are the pulse profile, reduced χ^2 values of DMs, frequency-versus-phase plot and time-versus-phase plot. Human experts identify pulsars based on the pulsar diagnostic plots and the above features are mainly taken into account. These features are extracted from a pfd file, and the pulse profiles are normalized and resized to 512 bins, the reduced χ^2 values of DMs are calculated to 512 bins and normalized, and the frequency-versus-phase plots and time-versus-phase plots are reshaped to 64×64 2D arrays.

Past research using AI methods to identify pulsars usually used separate models to judge each feature individually. We experimented with feature fusions in the stage of data preprocessing and obtained better results than the results of judgment with independent features. The purpose of feature fusion is to combine multiple feature information into a feature that is easier to distinguish than the original feature. The feature fusion work as follows: When pulsar profile data are extracted, normalized and resized to 512 bins, it will combine with the normalized reduced χ^2 array length 512 to form a 1D merged array with length 1024. We also calculate the reduced χ^2 values to 4096 bins and reshape the 1D array to a 64×64 2D array. Then the frequency-versus-phase array, the time-versus-phase array and the 2D array of reduced χ^2 values are combined to a "multi-channel image," and each 2D feature is regarded as a "single channel image," which, like an RGB image, consists of three channels of red, green and blue.

Without feature fusion, multiple features are predicted separately, and then the predicted results are synthesized to form the final predicted score. But in our system, the ANN model can capture the relationship between profile and reduced χ^2 values, and the CoAtNet model can capture the relationship between the frequency-versus-phase plot, the time-versus-phase plot and the 2D array of reduced χ^2 values. In fact, it is easy to imagine that judgment with feature fusion is closer to human judgment thinking. Human experts often combine details of the frequency-versus-phase plot, the time-versus-phase plot and the DM-curve on the diagnostic plots to determine whether a candidate is a pulsar. In order to verify the effectiveness of feature fusion, comparative experiments are conducted, and the experimental results are shown in Table 4.

3.2. Performance Evaluation

Recall, FPR and accuracy are usually used to evaluate the performance of models for machine learning classification tasks. Recall is the ratio of true positives (TP) to the sum of TP and false negatives (FN)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (9)$$

while FPR is the ratio of false positives (FP) to the sum of FP and true negatives (TN)

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (10)$$

Table 2
The Performance on Recall, FPR and Accuracy Rate for Three CoAtNet Variants, Trained Using the Same Data Set

	Recall	False Positive Rate	Accuracy
C-C-C-T	98.75%	1.31%	98.72%
C-C-T-T	98.15%	2.99%	97.61%
C-T-T-T	94.84%	4.29%	95.26%

Note. The CoAtNet variant C-C-C-T is the best.

Accuracy is the ratio of the sum of TP and TN to the sum of TP, TN, FP, and FN

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (11)$$

In our experiments, we define recall as the ratio of correctly identified pulsars to the total number of true pulsars in the data set. The FPR refers to the ratio of the number of false candidates identified as pulsars to the total number of false candidates in the data set. Accuracy is the proportion of all candidates correctly identified in the data set to the total number of candidates. The training goal of our AI model is to maximize recall while minimizing the FPR.

It is worth noting that our recognition system produces a score between 0 and 1, with a candidate being classified as a pulsar if its score exceeds a threshold value. In all experiments of this paper, we set the threshold value at 0.5. However, to minimize the risk of missing any potential pulsars, a more conservative threshold of 0.1 is adopted in the practical application of the GPPS project. Any candidate with a score greater than 0.1 is classified as a pulsar and subsequently undergoes manual inspection.

3.2.1. Finding the Most Suitable Variant

From Section 2, we know that the MBConv block or Transformer block can be used for the four stages of CoAtNet, so there are five CoAtNet variants: C-C-C-C, C-C-C-T, C-C-T-T, C-T-T-T and T-T-T-T. Considering the balance of the model ability and generalization ability, we choose the C-C-C-T variant, C-C-T-T variant and C-T-T-T variant to experiment on the fused 2D feature in our data set. These models were trained on a training set that consisted of 6333 pulsar samples and 6148 RFI samples, and their parameters were saved for testing. We evaluated these models on a test data set containing diverse types of pulsar and RFI samples, and their performance metrics are displayed in Table 2; each performance metric of the C-C-C-T CoAtNet performed better than the other variants. This result indicated that the CoAtNet with C-C-C-T architecture is the most suitable variant for our data, so we use it to build our pulsar identification system.

Table 3
Performance Metrics For Different Sizes of Training Samples

Size of Training Sample	2000	6000	8000	12,000	16,000
Recall in training(%)	99.36	99.97	99.92	100	100
Recall in testing(%)	87.28	96.82	98.05	98.81	99.24
FPR in training(%)	2.96	0.0	0.12	0.0	0.0
FPR in testing(%)	3.08	1.45	0.96	1.12	1.61
Accuracy in training(%)	98.19	99.98	99.9	100	100
Accuracy in testing(%)	91.86	97.64	98.52	98.85	98.84

3.2.2. The Model Capacity of Our Model

We trained five models using five training data sets with sample sizes of 2000, 6000, 8000, 12,000 and 16,000, and then used the same test data set to evaluate their performance. To assess the performance of these models, we utilized a test data set consisting of 6320 true pulsar samples and 5717 RFI samples, and measured their recall, FPR and accuracy. As affirmed in Table 3, increasing the sample size in the training data set improved the model's performance, with the best accuracy reaching about 98.85% on the GPPS test data set. However, we observed that the model's accuracy tends to plateau when the training data set size exceeds 12,000, indicating that the proposed model requires a substantial amount of input data, with a minimum of 12,000 samples necessary to achieve optimal performance.

To compare the proposed method with existing methods, we trained two other methods, namely PICS and PICS_res, using the same training data sets with different sample sizes and evaluated their performance on the test data set. Figure 4 shows that the proposed method outperformed the other two methods in terms of identification accuracy, especially when trained on a sufficient amount of data. Our experimental results demonstrate that as the size of the training data set increases, the proposed method outperforms the other two methods, indicating its stronger model capacity. Specifically, when the amount of training data is increased by the same amount, our method exhibits a faster increase in accuracy compared to the other two methods. These findings suggest that the proposed method can effectively benefit from a larger training data set and has a higher potential for improving performance with additional training data. In summary, our results indicate that sufficient training data are crucial for achieving high identification accuracy, and the proposed method has a stronger capacity to utilize larger training data sets than the other two methods.

3.2.3. The Usefulness of Feature Fusion

We conducted a pair of control experiments to demonstrate the effectiveness of feature fusion. Experiment I involved training five models using the pulse profile 1D array with length 512 bins, the 1D array of reduced χ^2 values with length 512 bins, the 64×64 frequency-versus-phase 2D array, the

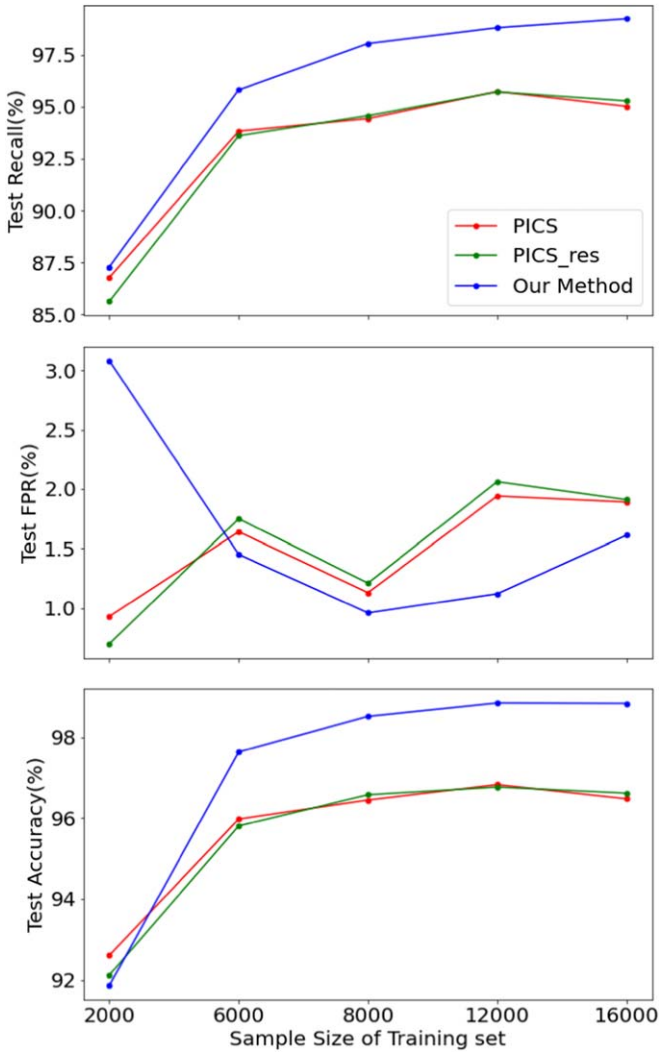

Figure 4. The recall, FPR and accuracy in test data set of PICS, PICS_res and our method using five training sets of different sizes.

64×64 time-versus-phase 2D array and the 64×64 reduced χ^2 values 2D array. The predicted results of these models were then synthesized using LR. In experiment II, we trained ANN models using the combined 1D feature, trained the CoAtNet model using the combined 2D feature and synthesized the predicted results of the two models using LR. The identification results with and without feature fusion are presented in Table 4, demonstrating an increase in the final test accuracy from 97.68% to 98.85% with feature fusion.

3.3. Comparison with Other Methods

We have searched all the papers in Table 1. Among them, the methods proposed by Eatough et al. (2010); Bates et al. (2012) and Morello et al. (2014) were found to be easily reproducible but outdated. In the remaining papers, the methods PICS (Zhu et al. 2014), PICS_res (Wang et al. 2019a), CCNN (Zeng et al. 2020) and SGAN (Balakrishnan et al. 2021) are provided with codes. The code of PICS AI and its upgrade PICS_res can be downloaded from https://github.com/zhuww/ubc_AI, and they are easy to train using our own training set. The code of CCNN and SGAN can be downloaded from https://github.com/xrli/CCNN/ and https://github.com/vishnubk/sgan respectively, but we met some difficulties when re-training our training data set.

Table 4
The Accuracy For Pulsar Candidate Identification by Using Fused Features or Individual Features Via Methods in "CoAtNet-MLP-LR"

Input Feature	Method	Accuracy
(c): 1D profile	MLP	88.62%
(d): 1D DM curve	MLP	92.96%
(c)+(d): fused 1D feature	MLP	95.33%
(b): 2D frequency-phase plot	CoAtNet	98.31%
(a): 2D time-phase plot	CoAtNet	91.90%
(f): 2D DM plot	CoAtNet	92.81%
(a)+(b)+(f): fused 2D feature	CoAtNet	98.47%
(a) → CoAtNet; (b) → CoAtNet; (f) → CoAtNet; (c) → MLP; (d) → MLP	LR	97.68%
(c)+(d) → MLP ; (a)+(b)+(f) → CoAtNet	LR	98.85%

Note. Feature fusion can give a higher accuracy.

To ensure a valid and meaningful comparison between two or more methods, it is imperative to train them using the same training set and evaluate their performance on the same test set. In our comparative analysis, the re-trained PICS and re-trained PICS_res have undergone this type of rigorous evaluation. In addition, we use the GPPS test data set to test the re-trained PICS model and the re-trained PICS_res model, and also test the re-trained PICS model and the re-trained PICS_res model. The assessment result of our method and other methods are shown in Table 5. We can see that the comparison between these retrained models and our approach is fair and valid, and our method achieves 98.77% recall, 1.07% FPR and 98.85% accuracy; it shows improvement on each performance evaluation metric. The models without re-training here are for reference only.

4. Conclusions

We applied CoAtNet, a new deep learning algorithm for image classification, and developed "CoAtNet-MLP-LR" for pulsar candidate identification, combined with the feature fusion method. Then, larger training and testing sets are selected and a series of experiments are conducted, and it is found that the C-C-C-T CoAtNet is more suitable for pulsar identification. Our method "CoAtNet-MLP-LR" has a stronger model capacity and requires a larger amount of training data to fully utilize its model capacity; the training data set should

Table 5
Performance Metrics of Our Method Compared with Those of Other Methods, Which are Evaluated by Using the Same Test Data Set

	Recall	False Positive Rate	Accuracy	Reference
CoAtNet-MLP-LR	98.77%	1.07%	98.85%	this work
PICS	86.77%	0.93%	92.61%	Zhu et al. (2014)
PICS*	95.73%	1.94%	96.83%	Zhu et al. (2014)
PICS_res	72.17%	4.78%	83.12%	Wang et al. (2019a)
PICS_res*	95.71%	2.06%	96.77%	Wang et al. (2019a)
CCNN	96.28%	7.17%	94.65%	Zeng et al. (2020)
SGAN	67.43%	4.90%	80.58%	Balakrishnan et al. (2021)

Note. "*" indicates a re-trained model using our data.

contain about 10,000 samples. Based on a large amount of training data, we get a well-trained result with about 98% accuracy. Our feature fusion plan is useful in pulsar identification, which increases the accuracy from 97.68% to 98.85%. Through the results of the comparative experiments in Table 5, our model performs best on the test set of GPPS project, and the identification accuracy can reach 98.85%, demonstrating that it is superior to other methods.

5. Code Available

The code of our pulsar identification system "CoAtNet-MLP-LR" is available in a GitHub repository.[7]

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC, Nos. 11988101 and 11833009) and the Key Research Program of the Chinese Academy of Sciences (grant No. QYZDJ-SSW-SLH021).

ORCID iDs

Nannan N. Cai https://orcid.org/0000-0002-5915-5539
JinLin Han https://orcid.org/0000-0002-9274-3092
WeiCong Jing https://orcid.org/0000-0002-1056-5895
DeJiang Zhou https://orcid.org/0000-0002-6423-6106

References

Balakrishnan, V., Champion, D., Barr, E., et al. 2021, MNRAS, 505, 1180
Bates, S. D., Bailes, M., Barsdell, B. R., et al. 2012, MNRAS, 427, 1052
Dai, Z., Liu, H., Le, Q. V., & Tan, M. 2021, arXiv:2106.04803
Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2020, arXiv:2010.11929
Eatough, R. P., Molkenthin, N., Kramer, M., et al. 2010, MNRAS, 407, 2443
Guo, P., Duan, F., Wang, P., et al. 2019, MNRAS, 490, 5424
Han, J. L., Wang, C., Wang, P. F., et al. 2021, RAA, 21, 107
Hendrycks, D., & Gimpel, K. 2016, arXiv:1606.08415
Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. 2017, arXiv:1709.01507
Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. 2015, arXiv:1506.02025
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, Commun. ACM, 60, 84
Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, Proceedings of the IEEE, 86, 2278
Li, X., Wang, W., Hu, X., & Yang, J. 2019, arXiv:1903.06586
Lin, H., Li, X., & Luo, Z. 2020, MNRAS, 493, 1842
Lin, H., Li, X., & Zeng, Q. 2020, ApJ, 899, 104
Lyon, R. J., Stappers, B. W., Cooper, S., Brooke, J. M., & Knowles, J. D. 2016, MNRAS, 459, 1104
Morello, V., Barr, E. D., Bailes, M., et al. 2014, MNRAS, 443, 1651
Nan, R., Li, D., Jin, C., et al. 2011, IJMPD, 20, 989
Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. 2018, arXiv:1801.04381
Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556
Tan, C. M., Lyon, R. J., Stappers, B. W., et al. 2018, MNRAS, 474, 4571
Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, arXiv:1706.03762
Wang, F., Jiang, M., Qian, C., et al. 2017a, arXiv:1704.06904
Wang, H., Zhu, W., Guo, P., et al. 2019a, SCPMA, 62, 959507
Wang, X., Girshick, R., Gupta, A., & He, K. 2017b, arXiv:1711.07971
Wang, Y., Pan, Z., Zheng, J., Qian, L., & Li, M. 2019b, Astrophysics and Space Science, 364, 139
Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. 2018, CBAM: Convolutional Block Attention Module in European Conference on Compter Vision
Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. 2016, arXiv:1611.05431
Yin, Q., Li, J., Zheng, X., et al. 2022, MNRAS, 516, 3082
Yin, Q., Li, Y., Li, J., Zheng, X., & Guo, P. 2022, ApJS, 264, 2
Zeng, Q., Li, X., & Lin, H. 2020, MNRAS, 494, 3110
Zhao, H., Jin, J., Liu, Y., Shen, Y., & Jiang, Y. 2022, MNRAS, 513, 2869
Zhu, W. W., Berndsen, A., Madsen, E. C., et al. 2014, ApJ, 781, 117

[7] https://github.com/cnn-OvO/pulsar-identification.git