The Astronomical Journal, 167:180 (9pp), 2024 April https://doi.org/10.3847/1538-3881/ad298f

Identifying Exoplanet Candidates Using WaveCeptionNet

Huiping Liao[1], Guangyue Ren[1], Xinghao Chen[1], Yuxiang Li[1], and Guangwei Li[2]

[1] Key Laboratory of In-Fiber Integrated Optics of Ministry of Education College of Physics and Optoelectronic Engineering, Harbin Engineering University, Harbin 150001, People's Republic of China; liyuxiang11@hrbeu.edu.cn

[2] Key laboratory of Space Astronomy and Technology National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, People's Republic of China; lgw@bao.ac.cn

Received 2023 May 21; revised 2024 February 12; accepted 2024 February 13; published 2024 March 27

Abstract

In this study, we propose a wavelet-transform-based light curve representation method and a CNN model based on Inception-v3 for fast classification of light curves, enabling the quick discovery of potentially interesting targets from massive data. Experimental results on real observation data from the TESS showed that our wavelet processing method achieved about a 32-fold dimension reduction, while largely removing noise. We fed the wavelet-decomposed components of light curves into our improved Inception-v3 CNN model, achieving an accuracy of about 95%. Furthermore, our model achieves F1-scores of 95.63%, 95.93%, 95.65%, and 89.60% for eclipsing binaries, planet candidates, variable stars, and instrument noise, respectively. The precision rate of planet candidates identification reaches 96.49%, and the recall rate reaches 95.38% in the test set. The results demonstrate the effectiveness of our method for light curve.

Unified Astronomy Thesaurus concepts: Exoplanet astronomy (486); Exoplanet detection methods (489)

1. Introduction

The discovery of over 5000[3] confirmed exoplanets has revealed a staggering diversity that has altered our understanding of planet formation, orbital dynamics, composition, and evolution. Transit photometry involves the periodic dimming of a star's light as a planet passes in front of it, providing information on the size, orbit, as well as the structure and evolution of the star. Tracking changes in the brightness and duration of these transits can yield invaluable insights into the properties and behavior of exoplanetary systems.

Efficient classification of the light curves obtained through transit method is essential for defining follow-up observations and analysis. As astronomical observation equipment continues to develop and observation capabilities improve, large-scale time-domain survey projects are being increasingly undertaken. Telescopes such as the Kepler Space Telescope, K2, and the Transiting Exoplanet Survey Satellite (TESS) collected vast amounts of light curve data in the search for exoplanets (Borucki et al. 2010; Howell et al. 2014; Ricker et al. 2015). Quickly and accurately searching for exoplanet candidate targets in this big data and conducting follow-up ground-based observations before the observation window ends is crucial for the search for exoplanets. However, transit observations generate light curves that are subject to instrument noise, background light, and celestial objects such as eclipsing binaries and variable stars. Manual verification of this complex massive data is inefficient, and new methods are required for automatically classifying light curves, screening candidate targets, and improving the efficiency of exoplanet searches.

In recent years, artificial neural networks and machine-learning data-mining methods have made significant strides in numerous areas of computer research, particularly in the field of time-series processing. Leveraging large training data sets, ever-increasing computing power, and innovative model development, these methods have outperformed traditional statistical methods in many practical applications, and have even surpassed human accuracy in some cases.

Processing astronomical data using artificial intelligence and deep learning is a rapidly evolving field in time-domain astronomy. By classifying data gathered from telescopes, astronomers can better plan future observation tasks, undertake preliminary research, optimize observation strategies, and reduce observation duration and expenses. With modern astronomical observation equipment and technology developing at a fast pace, telescopes and detectors are becoming increasingly more sensitive and precise, leading to the accumulation of vast amounts of astronomical data. Traditional data processing methods are insufficient to cater to the requirements of modern astronomy, particularly for high-temporal resolution of transit photometry time-domain surveys in the wide field. To automatically and swiftly identify faint exoplanets within noisy data, deep learning is an exceedingly dependable technology for resolving large-scale astronomical data processing issues.

Currently, there have been some studies on the light curves classification. (Shallue & Vanderburg 2018) developed a convolutional neural network (CNN) models to classify Kepler light curve, they use transit parameters such as period and depth to build phase-folded light curves, and then use these curves as the network inputs. The phase-folded method constructs both a global and a local view to help the model distinguish between long-term and short-term information in time-series data. Since then, the phase-folded method is widely used in subsequent studies related to light curve classification (Chaushev et al. 2019; Chintarungruangchai & Jiang 2019; Dattilo et al. 2019; Yu et al. 2019; Osborn et al. 2020; Rao et al. 2021; Valizadegan et al. 2022). Most of the previous attempts to use deep learning to classify light curves are similar to the approach introduced by Shallue & Vanderburg (2018),

[3] https://exoplanetarchive.ipac.caltech.edu/

Table 1
Dataset Labels for the Light Curves in This Paper

Type		Count
Variable star (V)	Single sinusoidal	556
	Double sinusoidal	494
	ACF	412
Planet candidate (PC)		1716
Eclipsing binary (EB)		1634
Instrumental noise (IS)		596
Total		5480

and involve making modifications to the time-series input. For example, ExoMiner introduced the use of odd/even transit views and a centroid shift test time series as well as individual vetting diagnostics.

In this paper, we attempt to propose a wavelet transform preprocessing method that does not require folding the light curves, and present the CNN model modified by Inception-v3 for classifying TESS light curves.

The rest of this paper is organized as follows. In Section 2 we describe the data set. In Section 3 we introduce a new wavelet transform method for pre-preprocessing the light curves, and our CNN architecture. Section 4 presents and discusses the model's sensitivity. Section 5 provides conclusions and discussions.

2. Data

We used 2 minute cadence Pre-search Data Conditioned Simple Aperture Photometry (Smith et al. 2012; Stumpe et al. 2012, 2014) light curves provided by the TESS Science Processing Operations Center (SPOC; Jenkins et al. 2016). SPOC generates light curves that have undergone corrections for instrumental and other non-astrophysical influences. The data are publicly available on Mikulski Archive for Space Telescopes.[4]

The TESS objects of interest (TOI) catalog assigned targets to one of six categories: confirmed planet (CPs), known planets (KPs), planet candidates (PCs), eclipsing binaries (EBs), stellar variability (V), and instrument noise/systematic (IS; Guerrero et al. 2021). The TESS follow-up observing program working group further assigned targets to Ambiguous Planetary Candidate, false positive and false alarm. We adopted the CP, PC, and KP targets labeled by TOI catalog and grouped them together as PC. The variable star data in this paper are from (Fetherolf et al. 2023), including classes: single sinusoidal, double sinusoidal, and ACF. These classes were detected using one-sine, two-sine, and autocorrelation function methods. The eclipsing binaries in our data set were sourced from the TESS-EBs catalog (Prša et al. 2022), which contains short-cadence observations of 4584 eclipsing binaries in sectors 126. The instrument noise/systematic data set is from the publicly available data set of Yu et al. (2019).[5] The data set collected above may contain the potential label noise, and we have provided a machine-readable table.[6] The types and counts of light curves in the data set are shown in Table 1. PC and EB are stitched from multiple sectors, the number of stitched sectors and target count are provided in Table 2. V and IS use observations from a single sector.

Table 2
Count of Stitched Observations Per Target

	Stitched Count		
Type	2	3	4
PC	692	650	374
EB	808	496	330

3. Methods

3.1. Wavelet Multiresolution

Wavelet transform is a multiresolution analysis method in both the time and frequency domains. It decomposes signals into different frequency components using its excellent time-frequency properties, extracting periodicities and separating noise. Wavelet analysis is widely used in various fields, including signal processing, digital communication, image compression, and speech recognition, for tasks such as denoising, compression, and segmentation of signals such as speech, images, and videos. Furthermore, wavelet analysis can be used to extract different features in light curves and analyze changes at different scales, such as periodicity, pulse, and duration. Additionally, wavelets can also be used to reduce noise and smooth signals, thereby enhancing signal resolution.

The use of wavelets for transit detection—specifically joint detection and noise characterization of photometric time series (Jenkins 2002), high time and frequency resolution (Bravo et al. 2014), and parameter estimation based on computing the likelihood in a wavelet basis (Carter & Winn 2009)—improves the estimate of the eclipse depth over white-noise analysis (Cubillos et al. 2016), wavelet-based detrending and denoising method (del Ser & Fors 2020), wavelet denoising (Saha & Sengupta 2021), wavelet pixel-ICA (Morello et al. 2016), autocorrelation with wavelet decomposition (Ceillier et al. 2017), search for periods (de Lira et al. 2019), and stellar rotation period analysis (Lu et al. 2022).

We choose Bior6.8 as the basis function, hard threshold as the threshold function, and Sqtwolog as the threshold selection method for the discrete wavelet transform (DWT). Biorthogonal wavelet pairs are a set of two wavelet filters, one for decomposition and one for reconstruction, that are designed to have different scaling and wavelet functions, but still satisfy certain orthogonality conditions. The notation biorNr.Nd specifies the properties of the biorthogonal wavelet pair, where N is the number of vanishing moments for each of the two filters, r is the degree of symmetry of the decomposition filter, and d is the degree of symmetry of the reconstruction filter.

Assuming the original sequence $x(n)$ has a length of M samples, where n represents discrete time points, and based on the Shannon sampling theorem with a sampling frequency of $2f$ where the signal's highest frequency does not exceed f. First, for the first-level decomposition of this sequence, it undergoes high-pass filtering with a high-pass filter $H(n)$ and low-pass filtering with a low-pass filter $L(n)$. After high-pass filtering, the signal's frequency range becomes $\frac{f}{2}$ to f, and after low-pass filtering, the frequency range becomes 0 to $\frac{f}{2}$. The lengths of

[4] https://archive.stsci.edu/missions-and-data/tess

[5] https://github.com/yuliang419/Astronet-Triage/blob/master/astronet/tces.csv

[6] https://github.com/HP-Liao/WaveCeptionNet

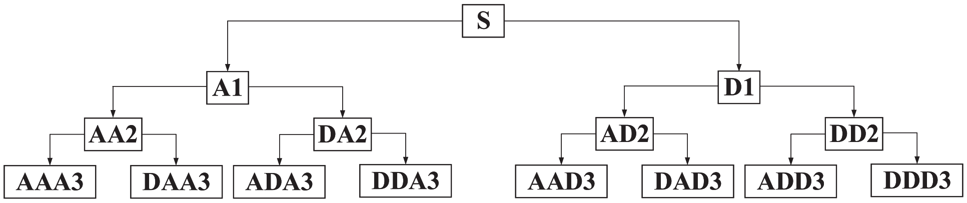

Figure 1. Wavelet packet decomposition tree with three levels. Each decomposition results in a set of coefficients. If this pattern is repeated until the sixth level, it yields AAAAAA6 and DAAAAA6, typically referred to as low-frequency approximation (CA6) and high-frequency detail (CD6).

the two filtered sequences remain M samples each. However, at this point, the signal is redundant. According to the Nyquist theorem, when the frequency range is halved, the length of the signal is correspondingly halved. Therefore, every other sample point is selected, resulting in a length of $\frac{M}{2}$ for the down-sampled sequence. The $\frac{M}{2}$ data points from the high-pass filter output (also referred to as the detail signal) in the frequency range $\frac{f}{2}$ to f represent the first-level wavelet coefficients. The $\frac{M}{2}$ data points from the low-pass filter output (also referred to as the approximation signal) are used to continue the second-level decomposition, obtaining the second-level wavelet coefficients with a frequency range of $\frac{f}{4}$ to $\frac{f}{2}$ and a length of $\frac{M}{4}$ points. In the wavelet decomposition process, simply, the data volume is halved after each decomposition.

With each decomposition, the time resolution is correspondingly halved. After the K-th decomposition, the signal length becomes 2^{-k} times that of the original signal. In high frequencies (few decomposition levels), the time resolution is relatively high, and due to the constant product of time resolution and frequency resolution, the frequency resolution is lower. In low frequencies (more decomposition levels), the time resolution is relatively low, resulting in higher-frequency resolution. Different resolutions are used to characterize the signal at different levels (corresponding to different frequencies), which is the essence of wavelet's multiresolution analysis (Harti 1993).

3.2. Wavelet Transform Preprocessing

We employ the wavelet packet decomposition tree to recursively decompose the signal into its constituent parts at different levels of resolution. Wavelet packet decomposition tree is shown in the Figure 1. We chose node AAAAAA6 (CA6) to represent the low-frequency component of original light curve, and node DAAAAA6 (CD6) to represent its high-frequency component. Figure 2 shows some examples of our wavelet transform preprocessing used in this work. The lengths of CA6 and CD6 representation after the decomposition of the DWT expressed as follows:

$$L = \mathrm{floor}\left(\frac{n-1}{2}\right) + \frac{N}{2} \tag{1}$$

where L represents the lengths of CA6 and CD6, floor represents rounding down to the nearest integer, n is the length of the original signal, and N is the length of wavelet. The length of the light curve is reduced by half with each level of wavelet decomposition. Wavelet decomposition, while representing data in terms of high and low-frequency components, has the effects of noise reduction and dimensionality reduction.

After performing six levels of decomposition with Sqtwolog hard method, we used the resulting low-frequency component CA6 and high-frequency component CD6 as the representation of the original data, each with a length of $\frac{1}{64}$ of the original signal. The frequency ranges for CA6 and CD6 are $[0, \frac{f}{64}]$ and $[\frac{f}{64}, \frac{f}{32}]$, respectively. The temporal durations represented by CA6 and CD6 remain the same as the original signal, only with reduced temporal resolution and a decrease in signal sampling points. This process eliminates many unnecessary high-frequency noise. The decimation of the wavelet coefficients and resampling may also reduce the energy in the transit signatures and thereby reduce the effective signal-to-noise ratio (S/N) and detection sensitivity even though the decimation also reduces the noise by some degree.

We removed all NaN values from the time series without shifting or interpolating them.For targets with multiple collections of observations, we took the average of adjacent elements in their light curves to obtain a new array with half the length of the original array. Subsequently, we applied wavelet decomposition. Due to quality masks and NaN values, the length of each light curve sequence varies. However, neural networks require input data in a matrix form with consistent dimensions. For training, we resampled the components obtained after wavelet decomposition using the cubic spline interpolation method to 500 bins. Finally, we conducted Min-Max normalization on the CA6 and CD6 components. The final form of CA6 and CD6 in this step, is shown in Figure 3. The complete data preprocessing pipeline is shown in the Figure 4.

For training, we remove all NaN values of original light curves, and resampled the components obtained after wavelet decomposition using the cubic spline interpolation method to 500 bins. Then, we conducted Min-Max normalization on the CA6 and CD6 components. The final form of CA6 and CD6 in this step, is shown in Figure 3. The complete data preprocessing pipeline is shown in the Figure 4.

3.3. Network Architecture

We use TensorFlow (Abadi et al. 2016) and the functional API in Keras to construct a dual-channel model. Our deep-learning models modified by Inception-v3 (Szegedy et al. 2016), a schematic diagram of our neural network in this work is shown in Figure 5. The Inception networks use convolutional kernels of varying sizes to create receptive fields of different scales, allowing for the fusion of features of different resolutions through concatenation. Factorized convolutions separate the spatial and depthwise convolution operations, and the incorporation of batch normalization, helps to improve the stability of the network during training.

Furthermore, we add a spatial dropout layer after each Inception block, which randomly drops out neurons with a certain probability. Spatial dropout is applied during training by randomly setting some neuron outputs to zero, which helps the network learn more robust features, reduces overfitting, and

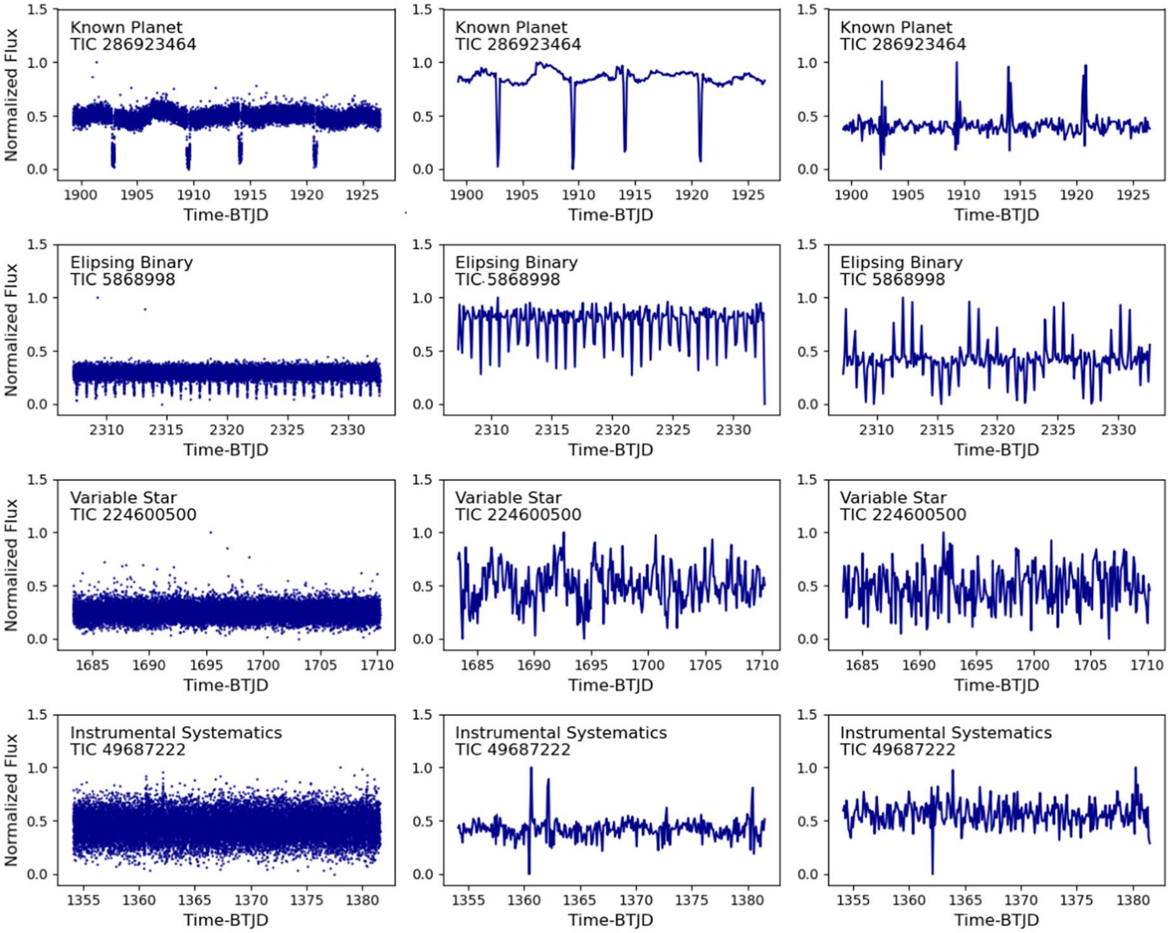

Figure 2. Wavelet transform preprocessing. The classes, in order from top to bottom, are KP, EB, V, and IS. The left column is normalized flux. The middle column is a CA6 representation of original light curve. The right column is a CD6 representation of original light curve.

improves generalization. Unlike standard dropout, spatial dropout does not apply dropout independently to each neuron, but sets some channels to zero for each time step. This type of zeroing not only considers individual time steps but also takes into account the correlation between time steps.

After Inception blocks, two fully connected layers (FC) are utilized to address nonlinear problems and act as the classifier for the entire convolutional neural network. These layers map multiple feature maps extracted from the convolutional layer to output categories. The softmax function is applied for multiclass prediction output. L1 regularization is employed in the output layer to prevent overfitting and improve generalization. During model training, L1 regularization penalizes the weights of the network and sets some values of the parameters to zero. This results in feature selection and dimensionality reduction, further decreasing the complexity of the model. The network is biased toward selecting smaller weights for prediction, as smaller weights lead to smaller regularization terms and therefore smaller total loss values. L1 regularization is computed as:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(f(x_i, \theta), y_i) + \lambda \sum_{j=1}^{m} |\theta_j| \tag{2}$$

where *L(θ)* denotes a loss function with L1 regularization, and *L(*)* is the prediction error of the model on the ith light curve. θ_j represents the jth parameter, n is the number of training samples, and m is the number of model parameters. λ is the strength of L1 regularization that can be optimized through network training.

In classification tasks, cross entropy (CE) is a commonly used loss function to guide model training

$$\mathrm{CE}(Y, \hat{Y}) = -\sum_{i}(y_i \cdot \ln(\hat{y}_i)),\ i \in \{0, 1, 2, 3\} \tag{3}$$

where $y_i \in \{0, 1, 2, 3\}$ is the label of each sample, and $\hat{y}_i \in \{0, 1, 2, 3\}$ is the prediction of each sample. As $\hat{y}_i$ approaches y_i , the value of CE decreases.

The time-domain survey targets exhibit an uneven distribution of categories, and the learning difficulty varies for each category. When the sample distribution is imbalanced, the loss function distribution becomes skewed, with minority samples dominating the function, causing the model to prioritize the learning of majority class samples and resulting in misclassification of minority class samples. To address class imbalance issue, there are two main approaches: (1) at the data level, methods such as oversampling, undersampling, and data augmentation are used; and (2) at the algorithm level, techniques such as weighted loss functions and ensemble learning are employed. In this work, we introduce focal loss from the field of computer vision to improve cross entropy (Lin et al. 2017). The specific formula is as follows:

$$\mathrm{Loss}_{\mathrm{FL}}(Y, \hat{Y}) = \begin{cases} -\alpha \cdot (1 - \hat{y}_i)^{\gamma} \cdot \log \hat{y}_i, & y_i = 1 \\ -(1 - \alpha) \cdot \hat{y}_i^{\gamma} \cdot \log(1 - \hat{y}_i), & y_i \neq 1 \end{cases} \tag{4}$$

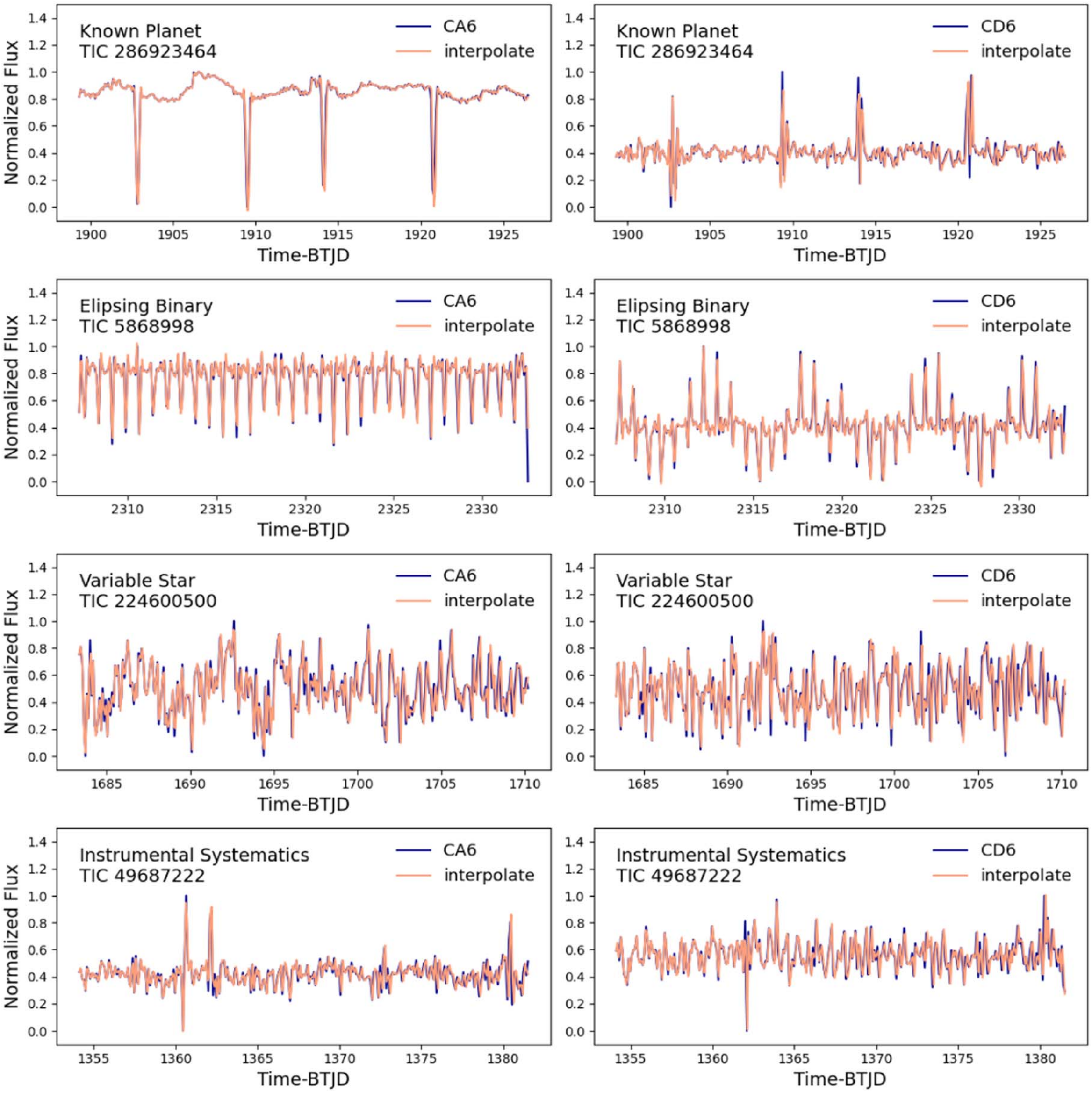

Figure 3. Wavelet representation normalization and interpolation. The classes, in order from top to bottom, are KP, EB, V, and IS. The left column is normalized and interpolated CA6. The right column is normalized and interpolated CD6. The orange curves represent the interpolation of the original blue curve.

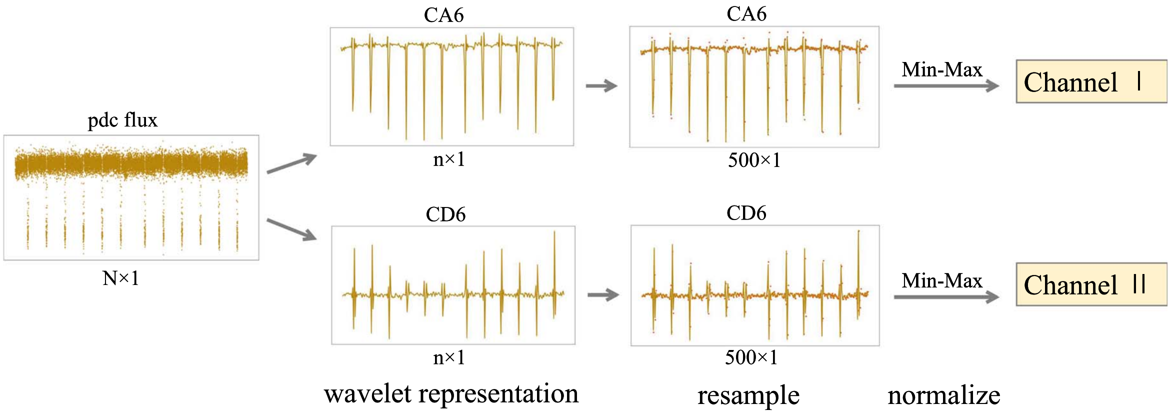

Figure 4. General flowchart of the wavelet representation preprocessing. The orange scatterplots represent a resampling of the original orange series. Finally, each light curve is transformed into a representation of dimensions (500, 2).

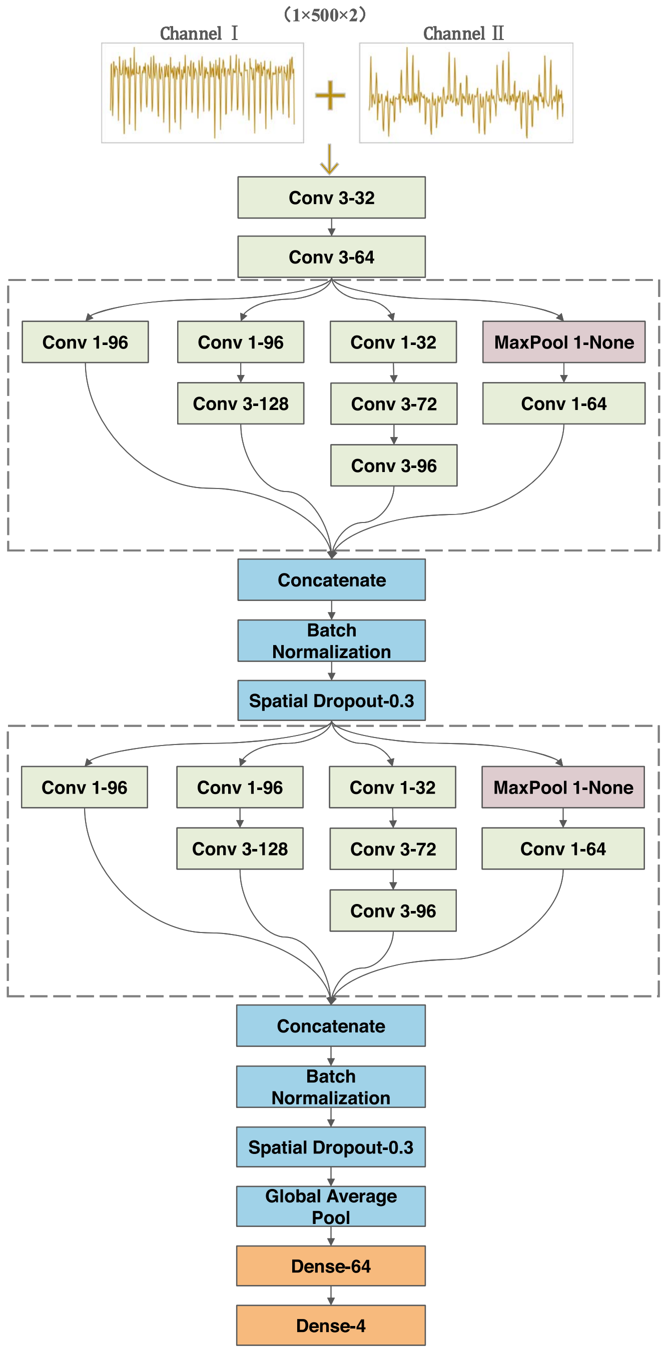

Figure 5. Architecture of our CNN model. Network inputs are passed through two repeated blocks of Inception. Convolutional layers are denoted Conv-kernel size-number of feature maps, max pooling layers are denoted MAXPOOL-window length-stride length, spatial dropout layers are denoted spatial dropout probability, and the fully connected layers are denoted FC-number of units. The output of the final, softmax layer is the predicted probability of the class for each light curve.

where $\alpha \in \{0, 1\}$ is a weighting factor to address class imbalance, $\gamma \in [0, +\infty]$ is a modulating factor that adjusts the rate at which easy examples are downweighted.

The loss function is minimized by the ADAM optimizer (Kingma & Ba 2014), where the learning rate is set to 0.0001. The batch size is set to 40. During the training process, keep track of the loss value of the validation set. If the loss does not decrease for 30 epochs, reduce the learning rate by a factor of 0.5. To reduce the risk of overfitting, we enabled early stopping the training when there is no improvement in validation accuracy.

4. Results

4.1. Evaluation

We follow the customary practice in machine learning by randomly dividing the labeled data into three separate sets, with the ratio of the training set to the validation set to the test set being 8:1:1. We conducted a four-class experiment on PCs, V, EBs, and IS. The overall accuracy rate of our model in classifying the test data into four types is 95.0368%. The performance metrics we used are following:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1}-\text{score} = \frac{2\text{precison} \cdot \text{recall}}{\text{precison} + \text{recall}} \tag{8}$$

$$\text{Macro}-\text{average} = \frac{1}{C}\sum_{i=1}^{C} \text{Metric}_i \tag{9}$$

$$\text{Weighted}-\text{average} = \sum_{i=1}^{C} \frac{\text{Metric}_i \cdot \text{Samples}_i}{\text{TotalSamples}}. \tag{10}$$

Here, C is the number of classes, Metric$_i$ is the metric value for the ith class, Samples$_i$ is the number of samples for the$i$th class, and Total Samples is the total number of samples. Accuracy represents the proportion of correctly classified events. Precision represents the proportion of true positives among all positive predictions, and recall represents the proportion of true positives among all actual positive instance. The F1-score is designed as the harmonic mean of precision and recall, to address their one sidedness. The macroaverage is the arithmetic average of metrics across all classes, and weighted average is an improvement upon the macroaverage, taking into account the proportion of samples in each class in the total sample.

The precision, recall, and F1-score for each class in the test set, along with the macroaverage and weighted average of precision and recall, are presented in Table 3. Figure 6 shows the confusion matrix generated from the model's predictions on the test set. The model achieves a precision rate exceeding 92% for all classes, with a recall rate exceeding 95% for EB, PC, and V. The F1 scores for EB, PC, V, and IS are calculated as 95.63%, 95.93%, 95.65%, and 89.60%, respectively. For PC, the model achieves an precision of 96.49%, a recall of 95.38%. Compared to other classes, the model exhibits a relatively lower recall rate for IS, at 83.58%. Given the variety of noise, more accurate predictions would require additional data on noise.

Figure 7 presents the precision-recall (PR) curve from the models predictions on the test set. In highly imbalanced data sets, the PR curve is more effective than ROC in evaluating classifier performance, as precision is influenced by both true-positive and true-negative samples. Choosing the model threshold depends on the specific use case, whether prioritizing accurate target identification or maximizing target recall. Average precision (AP) is the area under the PR curve,

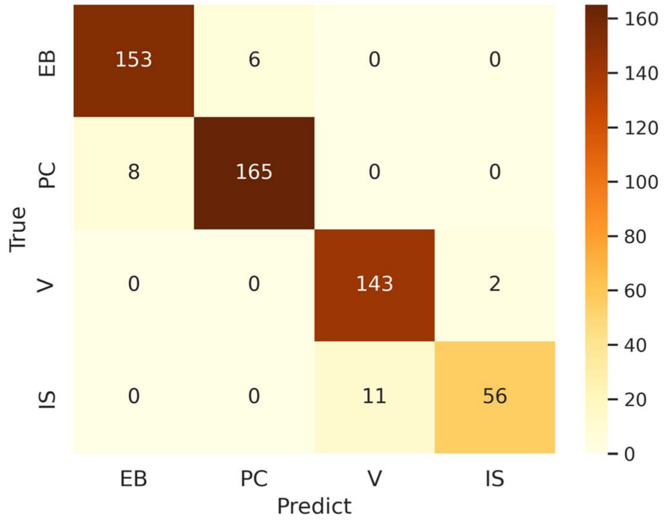

Figure 6. Confusion matrix from the models predictions on the test set, where the vertical axis represents the actual category and the horizontal axis represents the predicted category.

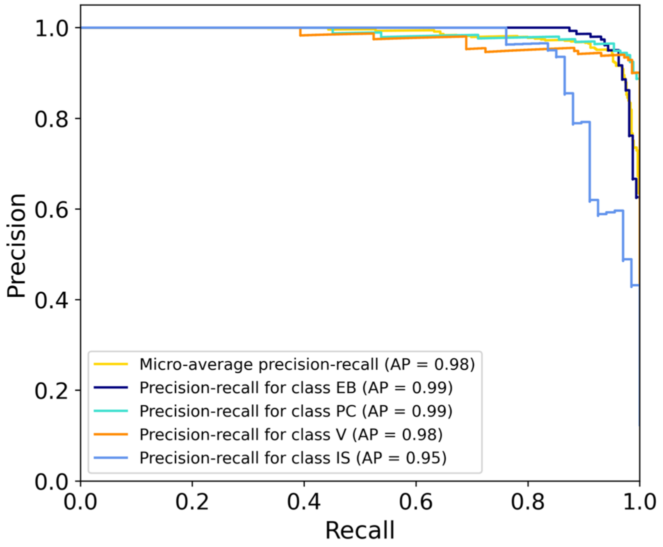

Figure 7. The PR curve has recall on the *x*-axis and precision on the *y*-axis for the test set, showing the recall and precision at varying thresholds. TCEs above the threshold are classified as the corresponding class. Average precision is the weighted mean of precision at each threshold.

Table 3
Our Model Performance on Test Set

	Precision	Recall	F1-score
V	0.9286	0.9862	0.9565
PC	0.9649	0.9538	0.9593
EB	0.9503	0.9623	0.9563
IS	0.9655	0.8358	0.8960
Macroaverage	0.9523	0.9345	0.9420
Weighted average	0.9510	0.9504	0.9499

reflecting the average precision of the model across different recall levels. Our model achieved the highest AP on PC and EB, with AP = 0.99.

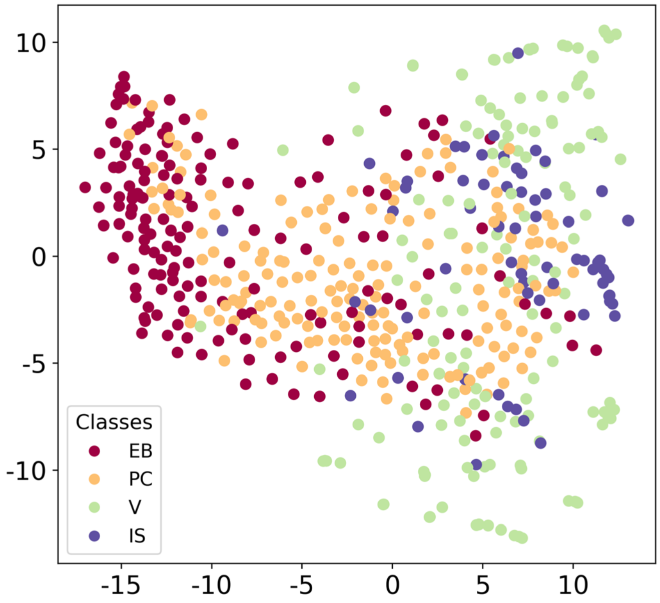

Figure 8. The t-SNE embedding results from CA6 and CD6 of the input data on the test set. PC (yellow) is close to EB (red), and V (green) is close to IS (purple).

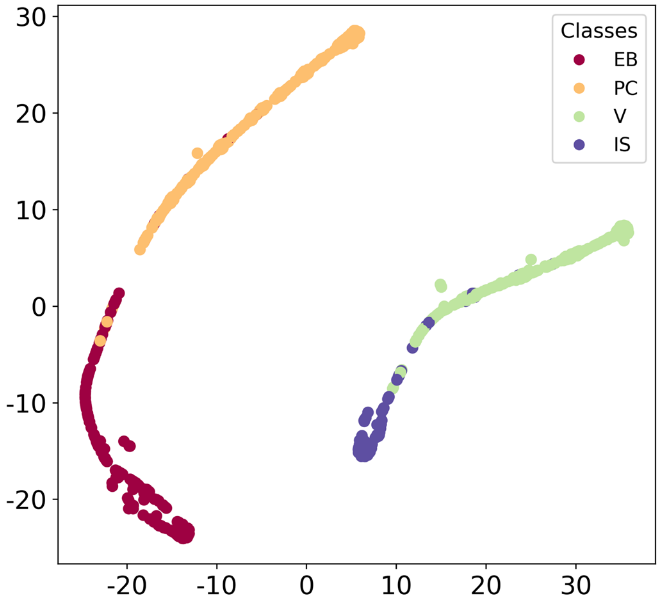

Figure 9. After model classification, the t-SNE embedding results from the final layer activations on the test set show a significant improvement in distinguishing between light curves.

4.2. *t-SNE Visualization*

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for visually representing the similarity between points in high-dimensional space through nonlinear dimensionality reduction. We present the t-SNE embedding results for CA6 and CD6 of the input data, along with the final layer activations from the test set. Despite the initial data chaos, after classification by the model, data points of the same category cluster together in the t-SNE mapping, revealing clear aggregation patterns and indicating the location of classification decision boundaries, as shown in Figures 8 and 9.

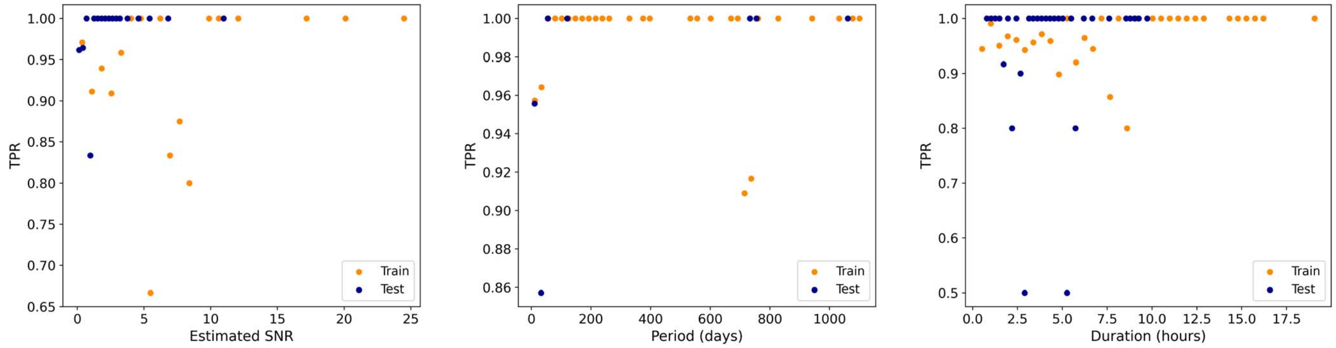

Figure 10. In the left-to-right sequence, the columns show the influence of S/N, orbital period, and transit duration on TPR. Both training (yellow) and test (purple) sets contain only PC on the test set.

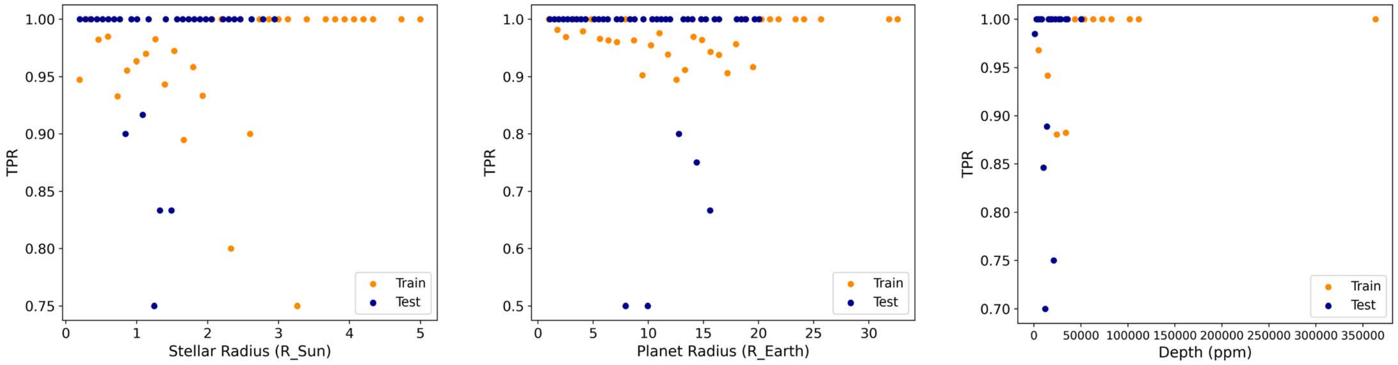

Figure 11. In the left-to-right sequence, the columns show the influence of stellar radius, planet radius, and transit depth on TPR. Both training (yellow) and test (purple) sets contain only PC on the test set.

Table 4
Sensitivity to Number of Stitched Sectors

TPR		Sector Count	
	2	3	4
Train	0.9471	0.9703	0.9617
Val	0.9460	1.0	0.9375
Test	0.9306	1.0	0.9297

4.3. Sensitivity

We extracted modelFitSnr, observedTransitCount, and orbitalPeriodDays for each PC target crossing event (TCE) from the DV XML files. Since the S/N scales with the square root of the number of transits, we divided the modelFitSnr by the square root of the observedTransitCount, and then scaled it by the square root of the number of transits that occur in the sectors analyzed by waveCeptionNet. With this information, we estimated the S/N of PC, even though the range of sectors in the data set differs from that in which the TCE was found. In Figure 10, the first column shows the model's sensitivity to S/N of PC, with the true-positive rate (TPR) oscillating for $S/N < 10$ and stabilizing at 1 for $S/N > 10$.

We obtained the orbital period, transit duration, stellar radius, planet radius, and transit depth for PC from TOI catalog. The model's sensitivity stabilizes when the orbital period exceeds 20 (days), transit duration exceeds 10 (hr), stellar radius exceeds 3 (R Sun), planet radius exceeds 17 (R Earth), and transit depth exceeds 50,000 (ppm), as shown in Figures 10 and 11. Our classifier shows no clear preference among S/N, orbital period, transit duration, stellar radius, planet radius, and transit depth. However, performance tends to improve with higher values of these parameters. While analyzing sensitivity to the number of stitched sectors, the model's peak TPR occurred at 3, but differences were not significant, as shown in Table 4 across training, validation, and test data sets.

4.4. Testing on Multiplanet Candidate Systems

To identify PC in multiple transiting planet systems and assess how WaveCeptionNet's performance is affected by too few transits. We compared the TPR for the PC in single planet systems with that of the multiple transiting planet systems in the confirmed planet system table at NExScI.[7] The data is confirmed, published, and involve only one star system. The results are shown in Table 5, the support column indicates number of planets. The column of single transiting planet systems includes PC in both the test and validation sets. The PC of multiple transiting planet systems were additionally collected from NExScI, and are not included in the statistics in Table 1. WaveCeptionNet's performance decreases with two planets, hits its worst with six, and is optimal with four or five ($TPR = 1$). In cases of misclassification, the model tends to classify PC in multiplanet systems as EB. Confirming WaveCeptionNet's sensitivity to systems with varying numbers of planets requires a larger data set.

[7] https://nexsci.caltech.edu/

Table 5
Sensitivity to Multiple Transiting Planet Systems

Number of Planets	1	2	3	4	5	6
TPR	0.94	0.75	0.88	1	1	0.57
Support	343	77	32	19	9	14
Misidentified as EB	19	13	4	0	0	6
Misidentified as IS	0	6	0	0	0	0
Misidentified as V	0	0	0	0	0	0

4.5. Comparision

(Yu et al. 2019) and Rao et al. (2021) data set include non-SPOC data, while our data set comprises solely SPOC data with no sector restrictions. Consequently, making a direct comparison is not applicable. However, the recall for (Yu et al. 2019) and Rao et al. (2021) are 61% and 74.3%, respectively, compared to WaveCeptionNet's 95.38% in the PC. To some extent, these comparisons suggest that WaveCeptionNet achieves better performance.

According to the NExScI Exoplanet Archive's planetary systems table to date, we provide the performance of WaveCeptionNet on these objects with SPOC data on our GitHub.

5. Conclusions and Discussions

Currently, there have been significant advancements in the research on exoplanet research and light curves classification, researchers have developed various algorithms for automated analysis and classification of photometric sequences. As an additional endeavor, in this paper, we explored the use of wavelet-transformed data set for neural networks, and CNN model based on Inception-v3 for classifying TESS light curves. We proposed a wavelet-transform-based light curve representation method to create preprocessed data sets. As dual-channel data input to the neural network for processing. The high-dimensional data from multiple observations of exoplanet research, after being processed by our method, has been reduced in length by a factor of 32 compared to the original data. Applying wavelet transform to the data allows us to significantly reduce the noise and anomalies present in the raw light curve data, thereby enhancing the neural network's efficiency and dependability. Compared with the traditional preprocessing method of phase folding, our method does not require period parameters, only the light curve. Low-frequency components typically contain trend information of the time series, while high-frequency components typically contain detailed information of the time series. By using this preprocessing method, the dimensionality of the original time series is greatly reduced, which reduces the complexity of the input data and speeds up network computation. With this model, we conducted experiments on the TESS light curve data set and found that the network can extract subtle features from the light curve and perform effective light curve classification and detection of exoplanets.

Exoplanet research and light curve classification of the transit method still face challenges, such as the sensitivity of algorithms to measurement errors and noise in stellar brightness variations, as well as sensitivity to observation sector and multitransiting planet systems. More observational data needed to determine algorithm tendencies and enhance accuracy. Unlike research fields such as computer vision and natural language processing, astronomy has its own highly specialized domain knowledge, which is crucial for problem analysis. Therefore, in addition to data-driven models, it is also necessary to incorporate astronomical expertize into the models to develop knowledge-driven models, which has been proven effective (Yu et al. 2019; Osborn et al. 2020; Rao et al. 2021). In future work, we will provide additional transit parameters to improve both the data processing pipeline and the model, and run them on all TESS light curves to identify new previously unidentified transiting planet candidates.

ORCID iDs

Huiping Liao https://orcid.org/0009-0005-7048-2476
Yuxiang Li https://orcid.org/0000-0003-3897-2270
Guangwei Li https://orcid.org/0000-0001-7515-6307

References

Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467
Borucki, W. J., Koch, D., Basri, G., et al. 2010, Sci, 327, 977
Bravo, J., Roque, S., Estrela, R., Leão, I., & De Medeiros, J. 2014, A&A, 568, A34
Carter, J. A., & Winn, J. N. 2009, ApJ, 704, 51
Ceillier, T., Tayar, J., Mathur, S., et al. 2017, A&A, 605, A111
Chaushev, A., Raynard, L., Goad, M. R., et al. 2019, MNRAS, 488, 5232
Chintarungruangchai, P., & Jiang, G. 2019, PASP, 131, 064502
Cubillos, P., Harrington, J., Loredo, T. J., et al. 2016, AJ, 153, 3
Dattilo, A., Vanderburg, A., Shallue, C. J., et al. 2019, AJ, 157, 169
de Lira, S., Bravo, J., Leão, I., et al. 2019, MNRAS, 484, 3935
del Ser, D., & Fors, O. 2020, MNRAS, 498, 2778
Fetherolf, T., Pepper, J., Simpson, E., et al. 2023, ApJS, 268, 4
Guerrero, N. M., Seager, S., Huang, C. X., et al. 2021, ApJS, 254, 39
Harti, A. 1993, ApNM, 12, 153
Howell, S. B., Sobeck, C., Haas, M., et al. 2014, PASP, 126, 398
Jenkins, J. M. 2002, ApJ, 575, 493
Jenkins, J. M., Twicken, J. D., McCauliff, S., et al. 2016, Proc. SPIE, 9913, 99133E
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. 2017, in Proc. of the IEEE Int. Conf. on computer vision, 2980
Lu, Y., Benomar, O., Kamiaka, S., & Suto, Y. 2022, ApJ, 941, 175
Morello, G., Waldmann, I. P., & Tinetti, G. 2016, ApJ, 820, 86
Osborn, H. P., Ansdell, M., Ioannou, Y., et al. 2020, A&A, 633, A53
Prša, A., Kochoska, A., Conroy, K. E., et al. 2022, ApJS, 258, 16
Rao, S., Mahabal, A., Rao, N., & Raghavendra, C. 2021, MNRAS, 502, 2845
Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2015, ATIS, 1, 014003
Saha, S., & Sengupta, S. 2021, AJ, 162, 221
Shallue, C. J., & Vanderburg, A. 2018, AJ, 155, 94
Smith, J. C., Stumpe, M. C., Van Cleve, J. E., et al. 2012, PASP, 124, 1000
Stumpe, M. C., Smith, J. C., Catanzarite, J. H., et al. 2014, PASP, 126, 100
Stumpe, M. C., Smith, J. C., Van Cleve, J. E., et al. 2012, PASP, 124, 985
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016, in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (New York: IEEE), 2818
Valizadegan, H., Martinho, M. J., Wilkens, L. S., et al. 2022, ApJ, 926, 120
Yu, L., Vanderburg, A., Huang, C., et al. 2019, AJ, 158, 25