

The Basic Shape Classification of Space Debris with Light Curves[†] ^{*}

LU Yao^{1,2,3△} ZHAO Chang-yin^{1,2△△}

(1 Purple Mountain Observatory, Chinese Academy of Sciences, Nanjing 210023)

(2 Key Laboratory for Space Object and Debris Observation, Chinese Academy of Sciences, Nanjing 210023)

(3 University of Chinese Academy of Sciences, Beijing 100049)

Abstract We study the machine learning method for classifying the basic shape of space debris in both simulated and observed data experiments, where light curves are used as the input features. In the dataset for training and testing, simulated light curves are derived from four types of debris within different shapes and materials. Observed light curves are extracted from Mini-Mega TORTORA (MMT) database which is a publicly accessible source of space object photometric records. The experiments employ the deep convolutional neural network, make comparisons with other machine learning algorithms, and the results show CNN (Convolutional Neural Network) is better. In simulational experiments, both types of cylinder can be distinguished perfectly, and two other types of satellite have around 90% probability to be classified. Rockets and defunct satellites can achieve 99% success rate in binary classification, but in further sub-classes classifications, the rate becomes relatively lower.

Key words space debris—methods: data analysis—methods: observational—methods: statistical—techniques: photometric

1. INTRODUCTION

With the improvement of the detection capabilities of the space debris observation network, more and more uncatalogued objects have been detected. For the newly detected space debris without prior information, it is necessary to understand their basic shapes, materials,

[†] Supported by the National Natural Science Foundation of China (Grant No. 11533010)

Received 2020–03–27; revised version 2020–04–21

^{*} A translation of *Acta Astron. Sin.* Vol. 61, No. 6, pp. 64.1–64.14, 2020

[△] luyao@pmo.ac.cn

^{△△} cyzhao@pmo.ac.cn

and other information, which is an important part of space situational awareness as well. Optical observations can directly obtain the angular and photometric measurements for space debris. The factors that can affect the magnitude include the observational geometry, the objects' size, shape, optical characteristics of surface material, attitude, etc.. The rotation of space debris causes the magnitude to change over time, and the continuous changes of magnitude are also called the light curve. Different types of debris will produce light curves with different characteristics, which can indicate the basic shapes and surface material types of the debris. For example, the light curve of a defunct satellite often contains specular reflection characteristics, meanwhile, the light curve of the common rocket body often shows a smooth double-peak structure, etc.. Therefore, the information about the shapes, materials, and basic classifications of unknown debris can be obtained by researching on the light curves.

Based on photometric information, a variety of space debris classification methods have been studied by scholars. Using an independent database, the statistics of the magnitude-phase characteristic curves for a batch of space debris were studied^[1]. Although the results showed that the characteristic curves for a part of spherical and cylindrical debris agreed with the theoretical predictions, it was difficult to classify the debris using the four characteristics summarized in their article. With the information such as the optical cross section of space objects, a machine learning method had achieved a recognition rate of more than 90% in space object classifications^[2], but the input features included the additional radar measurements besides the optical data. Although the above methods used photometric information, the input features are not light curves. With the development of machine learning in the astronomical field, whether the classification of 1D light curves or the feature recognition of 2D astronomical images, machine learning had realized many successful applications. Take the classifications of light curves by deep neural network as examples. A variety of machine learning methods were tested to find transit features from stellar light curves to estimate the existence of extrasolar planets^[3]. Their simulations showed that the performance of 1D convolutional neural network was better. Using the measured light curves and a more complex deep neural network, two new exoplanets were discovered from the database of the Kepler satellite^[4]. In terms of determining the space debris basic types, several types of machine learning models were trained with simulative light curves, and the results of classifications for the shape and stability of objects indicated that the random forest algorithm was the best^[5]. However, the article did not use more advanced algorithms, e.g., deep neural network, and the shape classification was mixed with a stability classification for which there are other simpler and more effective methods. Based on deep neural network, classification for the light curves of space debris was tested with both simulative and measured data^[6–8]. Although nine classes of space debris were included, some of them had certain difference from the shapes of real debris. The initial attitude and angular velocity were set as the same in the simulation. The data set did not distinguish between rotational and stable objects in

advance when classifying three types of space objects with measured data, and 500 points were directly extracted as training data for each light curve without further optimizations.

This work is also based on the light curve and the deep neural networks to explore the classification of the basic types of space debris. In order to focus on the classification of debris types, the rotating state of the debris is distinguished in advance. The classification method for each case should be developed correspondingly, because of the very difference in the light curve between rotating and stable debris. Here we focus on the classification of rotating debris in this work. The light curve as the input features should be appropriately transformed and optimized to make the characteristics of each curve comparable in the time domain. Since only rotating debris are considered, the light curve within one period, also called phase or phase curve, is aligned at the brightest point, and then used as the training data. Our simulation experiments will test the classification performance of the deep convolutional neural networks, and compare with other machine learning algorithms. The observational data experiments firstly try to dichotomize the rocket body and the defunct satellite, and further classify the sub-class of debris which have enough observational data.

Section 2 briefly introduces the deep neural networks, including the principle of the 1D convolutional layer and the fully connected layer. Section 3 presents the method to simulate light curves, the settings of the shape and material of four debris classes, and also introduces the extraction method of the observational light curves, the data distribution characteristics, etc.. Section 4 describes the simulation experiments, including the brief procedure, the model parameter tuning, the classification results, the learning curve, and the comparison with other machine learning algorithms. Section 5 presents the observational data experiments, where dichotomy and sub-class classification are tested, respectively.

2. DEEP NEURAL NETWORKS

The deep neural networks (DNNs) used in this work mainly consists of two structures: the convolutional layer and the fully connected layer. Each neuron in the convolutional layer has limited connections to enhance the learning ability for local features. Since the input features are light curves, the convolutional layer is 1D. The basic principle is shown in Figure 1 (a). The inputs make cross-correlation discretely with multiple filters (kernels), and then downsample through a pooling layer to get the output.

The formula of 1D convolutional operation is:

$$\mathbf{l}_i^j = f \left(\sum_{k=1}^K \mathbf{w}_i^{k,j} * \mathbf{l}_{i-1}^k + \mathbf{b}_i^j \right), \quad (1)$$

where K is the number of feature maps of the $(i-1)^{th}$ layer, \mathbf{l}_{i-1}^k is the input, standed for the k^{th} feature map of the $(i-1)^{th}$ layer, $\mathbf{w}_i^{k,j}$ is the k^{th} kernel to generate the j^{th} feature map of the $(i-1)^{th}$ layer. \mathbf{b}_i^j is the j^{th} bias vector of the i^{th} layer, \mathbf{l}_i^j is the j^{th} feature map of the i^{th} layer, $*$ means discrete cross-correlation operation, f is the activation function,

here we use relu function (rectified linear unit function), i.e., $f(x) = \max(0, x)$.

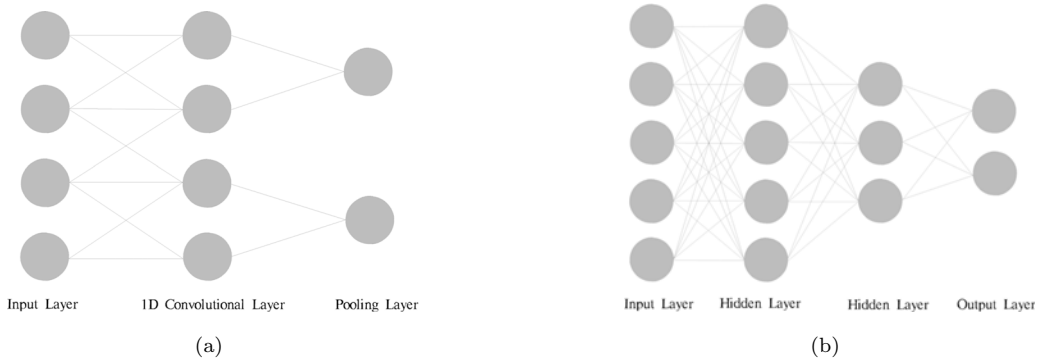


Fig. 1 Deep neural network (a) 1D convolutional layer, (b) fully connected layer

A pooling layer is used after the convolutional layer to decrease the model parameters by downsampling from original feature maps. This can reduce subsequent computation consumption, meanwhile make the feature map have a certain translation invariance, highlighting the existence rather than the exact location of a feature. The max-pooling layer is used in this work.

A fully connected neural network is typically composed of an input layer, several hidden layers, and an output layer, as shown in Figure 1 (b). Besides being used independently, it often plays the role as the back end of a convolutional network. For each neuron, the input is multiplied with weights, then the bias terms is added, and finally output through the activation function, which can be expressed by following formula:

$$\mathbf{l}_i = f(\mathbf{W}_i \mathbf{l}_{i-1} + \mathbf{b}_i), \quad (2)$$

where \mathbf{l}_{i-1} is the input, \mathbf{W}_i is the weight vector, \mathbf{b}_i is the bias, \mathbf{l}_i is the output of the i^{th} layer.

The input feature is delivered through layers, and the probability of classification problem with softmax function is calculated, i.e.,

$$S(x_m) = \frac{e^{x_m}}{\sum_{n=1}^N e^{x_n}}, \quad (3)$$

where S means the softmax function, x_m is the output of the m^{th} neuron, N is the number of neurons in output layer. The sum of each class probability would be 1 after the normalization operation.

For multi-class classification problems, cross-entropy is widely used as the loss function,

denoted as loss_{CE} :

$$\text{loss}_{\text{CE}} = - \sum_{m=1}^C t_m \lg[S(x_m)], \quad (4)$$

where C is the number of classes, $\mathbf{t} = [t_1, t_2, \dots, t_C]$ is the target vector of the classification, i.e., the true label.

The implementation of DNNs in this work is based on the Scikit-learn¹, TensorFlow², Keras³, and other dependent packages in Python. Their official websites have wealthy documentations and application examples. As deep learning tools with open source code and easy to use, they are widely used in many fields.

3. TRAINING SET

Training set should be prepared when using DNNs to solve classification problems. Good training data should have similar characteristics to the test data, and be evenly distributed in the parameter space. In this paper, simulation and observational data experiments are carried out, respectively, for which the former has the advantage of large quantity and complete coverage, but the disadvantage of limited modeling accuracy with simulation light curves. On the other hand, although the observational light curves are highly accurate, there are problems of incomplete coverage and uneven distribution due to the limited amount of the data. In the experiments, the light curve is extracted with the length of one period, and sampled 200 points uniformly.

3.1 Simulation Light Curves

In this paper, the Cook-Torrance Bidirectional Reflectance Distribution Function (BRDF)^[9] is used in the photometric model of space debris.

For fast-rotating debris or debris in high orbit, the phase angle changes within one rotational period are small, so the orbital motion of the debris is ignored in the simulation, and the flat-spinning model is used.

The process of generating the simulation light curve is as follows:

- (1) Set the shape of the simulation debris, the BRDF parameter ranges of each surface, direction of spin axis, etc.;
- (2) Set the phase to 70° , and calculate the coordinate transformation matrix on each photometric sampling point based to the rotation model;
- (3) Divide the debris surface into several structural planes, set the shape parameters of each plane, the BRDF parameters are randomly selected within the given range, then divide each structural plane into small facets to calculate the reflection flux separately, and finally accumulate the flux over all facets;

¹<https://sklearn.org>

²<https://www.tensorflow.org>

³<https://keras.io>

(4) Simulate 2500 light curves for each debris type, align the curves at the maximum flux point, and normalize the flux to 1000 km distance.

In simulation experiments, the space debris are roughly divided into four categories as listed in Table 1. (s_{CT} , d_{CT} , ω_{CT} , m_{CT}) are the main parameters of the Cook-Torrance model, which represent the specular coefficient, diffuse coefficient, diffuse albedo, and root-mean-square slope, respectively.

Table 1 The BRDF parameter ranges of the defined space debris

Class	Shape	Surface	s_{CT}	d_{CT}	ω_{CT}	m_{CT}
Case1	Cylinder	Specular	0.6–0.9	0.1–0.4	0.5	0.05–0.2
Case2	Cylinder	Diffuse	0.1–0.4	0.6–0.9	0.5	0.3–0.5
Case3	Box-wing	Specular	0.6–0.9	0.1–0.4	0.5	0.05–0.2
Case4	Cube	Specular	0.6–0.9	0.1–0.4	0.5	0.05–0.2

The cylindrical specular debris includes early high-orbit communication satellites. Because their surface is covered with solar panels, they are considered to be mainly specular reflection. In addition, some rocket bodies are reported that the surface is made of metal materials, and should also be mainly specular reflection. The diffuse cylinder is used to simulate most rocket bodies whose surface is coated with such as white paint, and usually considered to be mainly diffuse reflection. The box-wing satellites refer to the satellites which have roughly cube body and two symmetrical solar panels. The front and back of the solar panels should be mainly specular. Although the satellite body is difficult to model, we set it as specular reflection type, considering that some of its components are covered with thermal insulation layers, and they are thought to be mainly specular reflection. For cube satellite, some of its surface is covered with solar panels, and others are more complicated, so it is simplified to the specular reflection. Other debris are complicated and difficult to model, but some of them may be similar to one of the above types to a certain extent.

To set the shape parameters in the simulation, the size of diffuse cylinder is the real size of a type of rocket body (with length 12.38 m and diameter 3.0 m). In order to examine the classification accuracy of the model under similar settings, the same size is used for the specular cylinder. The box-wing satellite body is set to 1 m \times 1 m \times 1 m, and the solar panels point to the x axis with size of 1 m \times 4 m, and ignored the thickness. Also, to investigate the influence of the solar panels on the classification accuracy, cube satellites use the same size of the box-wing satellite body. The range of the BRDF parameter of the surface material is shown in Table 1. In the simulation, occlusion, and multiple scattering are ignored. In fact, for the rocket body, above effects have little influence. For typical satellites, occlusion generally occurs between the solar panels and the satellite body, and the simulation accuracy is often poor in such observation geometry that produces the occlusion, so the occlusion effect is ignored in the simulation. Since many components on the satellite

often have strong specular reflections, the multiple reflection effect can impact the simulation results. To speed up the simulation speed, above effects are ignored in this work. The size of the space debris is only for reference. It should be noted that the scaling of the size is only reflected in the overall up and down translation of the light curve, but with little effect on the characteristics of the light curve.

3.2 Observational Light Curves

The observational light curves are extracted from the MMT (Mini-Mega TORTORA) open database⁴. This telescope system consists of 9 independent channels installed on 5 equatorial mounts with the field of view $9^\circ \times 11^\circ$ for each channel, and uses sCMOS (scientific Complementary Metal-Oxide Semiconductor) camera as the terminal. The main scientific goal of this system is the large field of view and high temporal resolution astronomical survey observation. At the same time, the photometric information of the space debris is also recorded. As of October 2019, the database has recorded the light curves for more than 3600 space debris, which are mainly distributed in low orbits and elliptical orbits. Since MMT adopts extremely short exposure mode, e.g., the frame rate of 10 fps, the limiting magnitude is usually about 10, and only brighter debris can be observed. After filtering by the two terms of 'periodic' and 'defunct satellite or rocket body', we select several types of debris with relatively larger amount of data. If the rocket bodies are clustered by their series name, 7 categories with more observation arcs are as follows: Atlas, CZ3, Delta4, Falcon9, H2A, IUS (Inertial Upper Stage), and PSLV (Polar Satellite Launch Vehicle). defunct satellites are classified according to the type of platform (bus type), and there are more observation arcs in the following 3 categories: Globalstar, Iridium, and Tiros (a satellite platform).

The method to extract the training set from the MMT database is as follows:

(1) Download the photometric data and period information of the debris from the MMT website. The period information includes the period value of each arc calculated by the MMT team. It is generally high accurate and can be used directly. Here we recalculate the period, and focus on the possible multiple relationship between calculated and true period. Then, remove the data in penumbra and those observed with multi-color filters. Correct the magnitude to 1000 km distance, but do not correct the phase angle;

(2) Use the Lomb-Scargle method to calculate the candidate period of the light curve, remove the arcs with insignificant periodicity, and then use the light curve folding method to fine-tune the period around $\frac{1}{2}$, 1, 2, and 4 times of candidate period to find the best period. Calculate the dispersion of the folded light curve at the best period to determine whether the period extraction is successful. For the rocket body, the bimodal fitting method can also be used to check the period extraction results;

(3) Extract a phase curve every 2 or 3 periods for each acr. For the phase curves whose missing part is larger than a half, remove them if they are not from rocket bodies. After folding with the best period, use the moving median filter or the Savitzky-Golay

⁴<http://mmt.favor2.info/satellites>

filter to smooth the original curve, and interpolate to 200 uniform sampling points. The average value of the magnitude standard deviation in the window is taken as the magnitude extraction error, i.e., $E_{\text{ex}} = \text{mean}[\text{std}(M_b)], b = 1, 2, \dots, B$, where std is the function to get the standard deviation, mean is the function to get the mean, B is the total number of windows, and M_b represents the photometric data in the b^{th} window.

Through this method, we extract a total of 11237 rocket body phase curves and 19204 satellite phase curves from the MMT database. Different application scenarios have different requirements for the accuracy of the light curve, e.g., for the dichotomy of basic shape (rocket body or defunct satellite), the threshold of magnitude extraction error is set as 3σ , and for the sub-class classification of the same debris type, 2σ threshold is used. Figure 2 shows the magnitude extraction error distribution of rocket body and defunct satellite, respectively. The distribution of the rocket body on the left is relatively symmetric, while the skewness of the distribution of the defunct satellite on the right is larger. Therefore, the χ distribution is used to fit the error distribution respectively as shown in Table 2. According to the fitting parameters, the magnitude extraction accuracy of rocket body is overall better than defunct satellite with average error 0.19 and 0.27, respectively, because of the simpler light curve shape of rocket body in general. Taking the dichotomy of basic shape as an example, we use the phase curve of rocket body with an extraction error less than 0.39, and the satellite with an error less than 0.66.

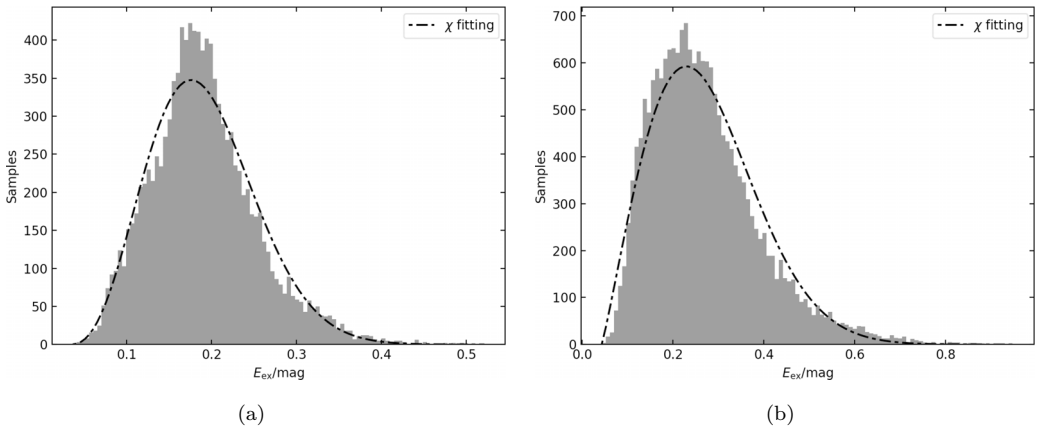


Fig. 2 The χ distribution of extracted errors for observed light curves (a) rockets, (b) satellites

Table 2 The fitting parameters of χ distribution of extracted errors for observed light curves

Parameter	Rocket	Satellite
χ degrees	3.44	2.10
mean	0.19	0.27
3σ cut-off	0.39	0.66
2σ cut-off	0.30	0.49
1σ cut-off	0.22	0.32

4. SIMULATION EXPERIMENTS

The DNNs are used to classify the basic shape of space debris by following steps:

(1) Make labeled training set and test set. The ideal data set should cover all sample types evenly. For simulation data, this is easy to achieve, and the data is divided into 2:1 for training and test set. For observational data, if the sample amount of a certain type debris is too small, we use a “data copy” method as the remedy by translating and adding noise into the original samples to generate new ones of the same type. Note that, these copy samples should only account for a smaller part of the whole, otherwise it would produce a data set with redundant features;

(2) Before inputting the data set to DNNs, standardize the input to make the mean and variance of overall features be 0 and 1, respectively. Use the one-hot method to encode the sample label, where the vector element is 1 for corresponding label and the rest are 0. This means the number of neurons in the output layer is the same with the number of debris types;

(3) Set the DNNs parameters, loss function, optimization algorithm etc., then input the training data, set the batch size and iteration times, and finally start training the network. The iteration times in this work is set to 100;

(4) Check the convergence and fitting effect of the model through the loss and accuracy curves. Input the test set into the trained network, the accuracy of the test set can represent the generalization of the model for unknown data, and the confusion matrix represents the classification ability for various types of data;

(5) Use different random number seeds to repeat the above steps 10 times, and then give the average output.

4.1 Model Parameter Tuning

When using machine learning algorithms, it is necessary to design appropriate network structure, configuration, parameters, etc., which is called model parameter tuning. In addition, it is also important to describe and transform the classification problem into a form suitable

for DNNs, including feature extraction and data preprocessing, etc.. The model parameters that are optimized in simulation experiments include the filter number, kernel size, input layer downsampling rate, and the model depth. The basic model for comparison is relatively shallower, from top to bottom including convolutional layer, convolutional layer, dropout layer, pooling layer, fully connected layer, fully connected layer, and softmax layer. For the initial parameters, convolution kernel size is $64 \times 3 \times 1$, the step size is 1, the dropout coefficient is 0.5, the pooling step size is 2, the neurons in two fully connected layers are 500 and 100 in turn. The relu activation function and adam optimizer are used, the loss function is the multi-class cross-entropy function, and batch size is 32.

Figure 3 is the statistics of the parameter tuning results. The horizontal axis in the figure means different parameter values, the vertical axis means the accuracy of the model for test set. The dotted line of the box is the median of repetitive tests under the same parameter value, the upper and lower sides of the box are respectively the 25% and 75% quantiles, the circles represent points outside the 1.5 times of the quartile, and the dashes represent extreme values within 1.5 times of the quartile. The result of the filter number tuning is shown in Figure 3 (a). Each convolutional layer has multiple filters whose values would be modified in each time of training after random initialization at very beginning. The filter makes convolution with feature map of upper layer to obtain its feature map of next layer. The more the filters, the feature extraction and recognition can be performed in a higher dimensional feature space. But more filters are not always the better, because the training time is also a considerable factor. When the other parameters of the basic model remain default, the model accuracy with filter number of 8, 16, 32, 64, and 128 are tested in simulation experiments. It can be seen that the results corresponding to 8 and 128 filters are more discrete, and the accuracy is lower. The results of 16 filters are more concentrated but the median value is slightly lower than the other cases. Overall, the filter number of 32 or 64 is optimal. Considering that the basic model uses a shallow network, and the depth and complexity of final network may increase after tuning, the number of filters is set to 64 here.

The result of the kernel size tuning is shown in Figure 3 (b). The kernel size determines the scale of feature extraction. Small sizes are sensitive to tiny features, but they can not directly perceive features with a larger span. When stack more layers and add pooling layers, a small size kernel can also learn large-scale features in the original input. Therefore, in the field of image recognition, normally a deep network with small size convolution kernel is used. Note that, the kernel size parameter and the input layer downsampling parameter are coupled in a sense, which can influence each other in tuning. This time the feature vector of input layer is not downsampled, i.e. the vector size is 1×200 . Based on above setting, we test the model accuracy at the kernel size of 2, 3, 5, 7, and 11. The results show that when the kernel size is 5, the median and the distribution of model accuracy are the best.

The result of the input layer downsampling rate is shown in Figure 3 (c). We hope to

keep a high classification accuracy with as less training time as possible. Downsampling the light curve can reduce the input dimension, greatly cut down the amount of calculation. The disadvantage of this operation is that it may overlook some local tiny features. Therefore, the experiments test the model accuracy when the light curve is binned from 1 to 5. The result shows that the performance of the model under each parameter is not much different, especially for the whole distribution. Considering that the final network depth and complexity may increase, the input data dimension is too low would limit the model depth, so we do not make binning on the input light curve, and the previous optimization results of the kernel size are still valid.

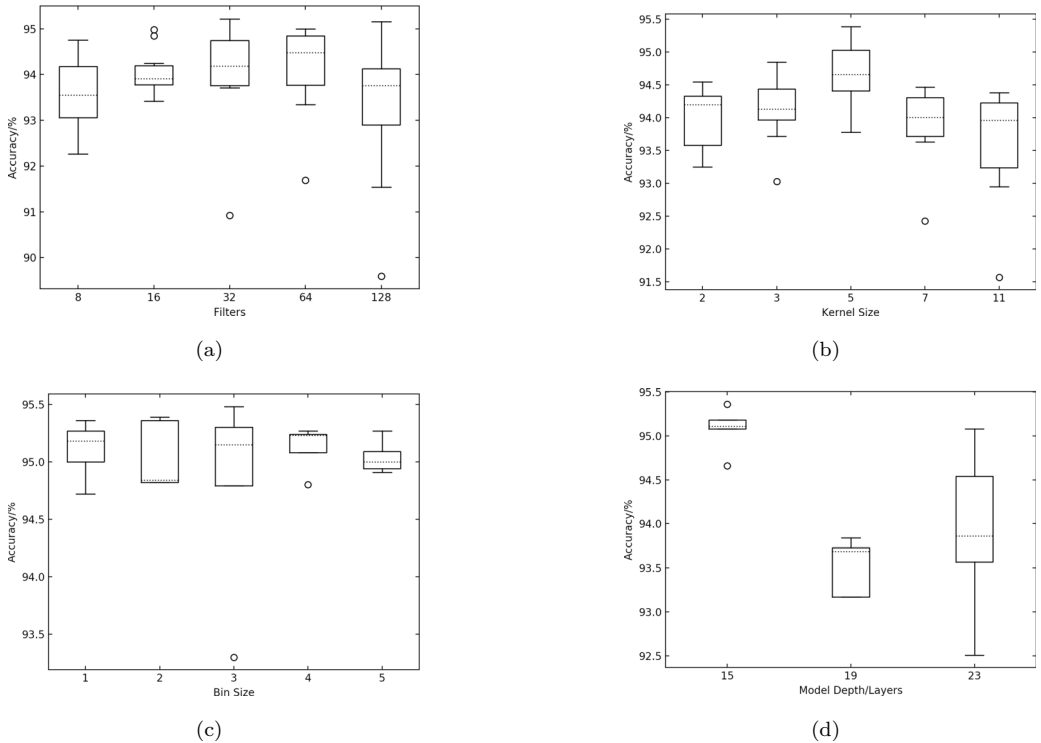


Fig. 3 Hyperparameter tuning for Deep Convolutional Network (a) filter number, (b) kernel size, (c) bins, and (d) model depth.

Finally, the effect of the model depth is tested, as shown in Figure 3 (d). If the combination of two convolutional layers, one dropout layer, and one pooling layer is seen as a group of convolutional layers, the 3, 4, and 5 groups of convolutional layers correspond to the network structure with depth of 15, 19, and 23, respectively. The parameters of each group of convolutional layers are set as the same. The test results show that using 3 groups of convolutional layers is significantly better than other deeper networks, and its classification accuracy is roughly above 95%. Based on the conclusions of all the above

parameter optimizations, we finally use the networks with a number of filters of 64, a kernel size of 5, an input layer feature dimension of 200, and a depth of 15 layers.

4.2 Classification Accuracy

The classification accuracy of different categories can be intuitively represented by the confusion matrix, as shown in Figure 4, where cyl_s, cyl_d, bow-w, and cube represent the specular cylinder, diffuse cylinder, box-wing satellite, and cube satellite, respectively. The value in each square represents the probability that the true category of the horizontal axis is recognized as the predicted category of the vertical axis. The total classification accuracy is 95.5%. The figure shows that the recognition rates for specular and diffuse cylinder both are 100%, although the shape and size of them are exactly the same, the difference in the optical characteristics of the material can distinguish them. This means that the typical rocket body can be easily classified by the light curve, no matter whether its surface is coated with paint or smooth metal materials. The recognition rates for box-wing and cube satellite are 89% and 93%, respectively, in which 10% of box-wing are misclassified as cube, and 6% of cube are misclassified as box-wing. The reason may be that the size of cube satellite is enlarged to be equivalent to the box-wing satellite body in the simulation, and the surfaces are all set to specular reflection characteristics, resulting in the high similarity of the characteristics between them.

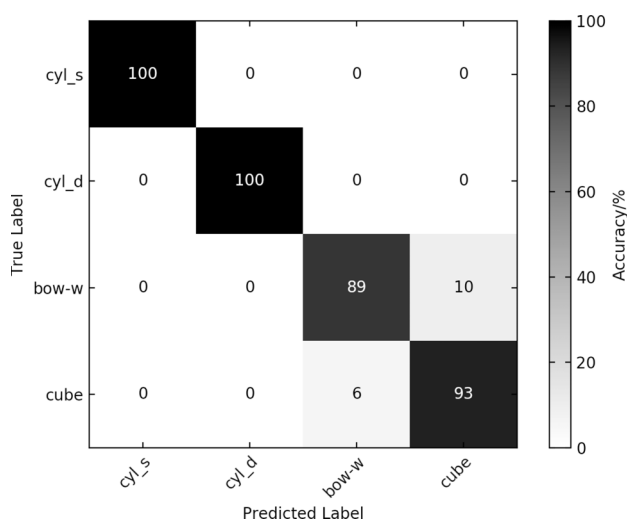


Fig. 4 The confusion matrix for the results of 4 types of debris in simulation experiments

4.3 Learning Curves

During the training of the network, according to the classification accuracy of the training set (actually a part of the training set) and the test set after each iteration, we can evaluate the effect of model updating. The curves that the classification accuracy and the loss varied with the model iteration are called learning curves. The learning curve of the training set indicates the change of model's learning degree, while the learning curve of the test set represents the change of model's generalization ability, i.e., the model's predictive ability on unknown data. According to the convergence speed of the learning curve, we can reasonably estimate the number of training iterations. When the accuracy curve almost does not change, it means that the model has been unable to learn new knowledge from training set, and the training can be stopped right now. When the training is stopped, but the accuracy curve is still in rising trend or loss curve in falling trend, indicating that the model is underfitting, and the training should continue.

The difference between the learning curves of the training and test set reflects the overfitting degree of the model. If the accuracy of the test set has basically converged to a certain value or even becomes drop while the accuracy of the training set is still rising, it indicates that the model becomes overfitting. When the loss curves of the training and test set decrease together but always with a large interval between them, the possible reason is that the training set is not representative enough to the test set, i.e., the training set can not provide enough information for the model to correctly recognize the features in the test set. This often occurs when the characteristics of several categories are very similar. If the learning curve fluctuates significantly, the problem may be related to the classification data set itself.

The learning curve of the simulation experiments is shown in Figure 5, the solid and dotted lines represent the training set and test set, respectively. The result shows that the accuracy of the training set continues to rise by exceeding 98% after 100 iterations, but the increase of the test set accuracy is not obvious, staying at about 95%. This indicates that although the model continues to learn new knowledges from the training set, but these knowledges have no effect on the classification of the test set, i.e., the model is overfitted after 100 iterations, and has poor generalization ability. The same conclusion can be drawn from the loss curve. In addition, the obvious fluctuation of the curve also indicates that the data set is not ideal in the design of our experiments, some features may be less distinguishable, and the classification results have the occasionality to some extent.

4.4 Comparison With Other Machine Learning Algorithms

As the comparison, we test the classification accuracy using other machine learning algorithms with the same data set, including K-Nearest Neighbors (KNN), Decision Tree (DT), Support Vector Machines (SVM), and Random Forest (RF). It should be noted that these machine learning algorithms usually use ordered features as input, i.e., they would do not support for the translation invariance of the input features. However, using light curve to

classify the basic shape of space debris needs to consider the translation invariance of the features. Two possible solutions: (1) align the light curve in time domain, such as aligning at the brightest point, and then use the Principal Component Analysis (PCA) method to further reduce the feature dimension; (2) convert the temporal signal to the frequency domain, and the parameters other than the initial phase are the translation invariance parameters of the original signal. We use the first method by extracting the first 12 main components of the aligned light curve from the original 200 dimension features, which can represent the data characteristics of 97.64%.

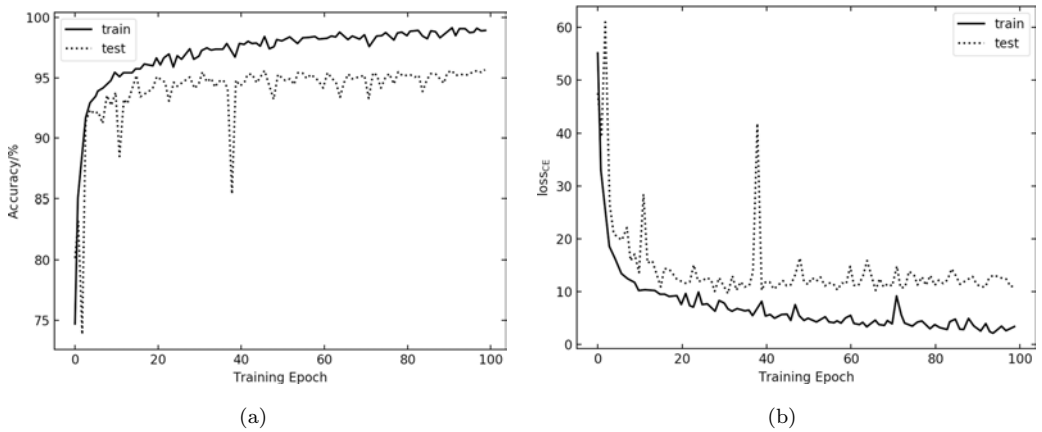


Fig. 5 The learning curves of simulation experiments (a) accuracy curves, (b) loss curves

This work uses the Scikit-learn package⁵ to implement the above machine learning algorithms. Basic tuning of the key parameters of each algorithm gives that: the K value of KNN is 5, the criterion function in DT is gini, the C value in SVM is 1, and the kernel is the radial basis function (rbf), the number of estimators in RF is 100. Finally, the optimized models use the bootstrap method to divide the training set and the test set. The statistics of accuracy is shown in Figure 6, where the dotted line represents the median value, the upper and lower sides of the box are respectively the 25% and 75% quantiles, and the dashes represent the upper and lower limits of the data. The accuracy of the KNN is about 93%, and the DT is about 91%, which is the worst among all tested algorithms. The accuracy of SVM and RF are between 94% and 95%, indicating that these two algorithms are also better choices for our classification problem besides the CNN. This is why we choose the RF algorithm in the subsequent platform classification experiment. CNN has the highest accuracy rate by exceeding 95%, so we use the DNNs to classify the basic shape of space debris.

⁵<https://sklearn.org>

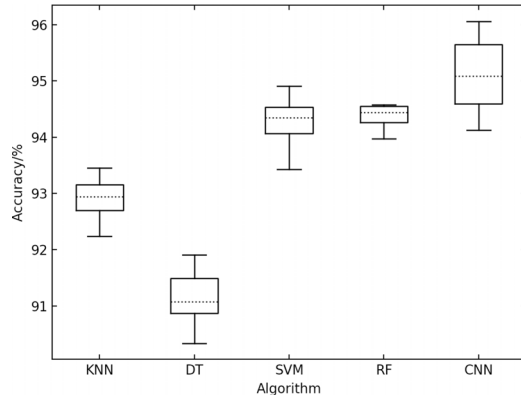


Fig. 6 Accuracy comparison of different machine learning algorithms

5. OBSERVATIONAL DATA EXPERIMENTS

5.1 Dichotomy Experiments

Due to the small amount and uneven distribution of the observational data for each debris sub-class, we first discuss the basic dichotomy problem, i.e., the debris is rocket body or defunct satellite. Randomly select 2500 phase curves from the observational data set, then adopt the 7 layer basic DNN model in the simulation experiment, finally use the bootstrap method to train 100 times. The classification results are shown in Table 3. The average classification accuracy is 99.31%, the median is 99.35%, and the confidence interval of 3σ is (97.60%, 99.83%). The recognition rate of the rocket body is 99.24%, slightly lower than the satellite of 99.50%. The result indicates that for the observational light curves extracted in this work, the DNNs can distinguish between rocket bodies and defunct satellites with high accuracy. Note that this experiment is based on the observational data of a small number of debris types, which would have obvious selection effects. The shapes of many space debris are very different from the samples we used here, and the results of larger-scale shape classification may be different. Limited by the amount and distribution of available observational data, it is hard to cover more types of debris within the scope of this paper. Although there are many limitations to further distinguish the sub-class of the debris based on these data, we still select several types of rocket bodies and satellites with relatively more data, and try to classify the series of rocket body or the bus type of defunct satellite.

Table 3 The accuracy of binary classification for satellites and rockets

Statistical	Results
mean	99.31%
median	99.35%
3σ interval	(97.60%, 99.83%)
accuracy for rocket	99.24%
accuracy for satellite	99.50%

5.2 Sub-class Classification Experiments

The number of phase curves of each debris type is shown in Table 4. Due to the small data set, the experiments disuse the DNNs, and adopt the RF algorithm. We require that each sub-class data set contains 500 samples, and make classification among the rocket body class and satellite class, respectively. The phase curves amount of CZ3, Atlas, H2A, and Falcon9 in the rocket bodies is more than 500, and the samples are randomly selected among each rocket series, respectively. There are 352 curves for Delta4, and we use the “data copy” method by adding Gaussian noise with zero mean and 0.5 magnitude standard deviation to the phase curve, and then translating 0.5 phase to generate new samples to obtain the total 500 samples. For IUS and PSLV, their data is such little that the above method is more likely to generate data set with redundant features, and should no longer be used in the classification experiments. Therefore, for the identification experiments of the rocket body series, 5 categories are included: Atlas, CZ3, Delta4, Falcon9, and H2A.

Table 4 The population distribution for extracted observed light curves

Rocket	Atlas	CZ3	Delta4	Falcon9	H2A	IUS	PSLV
Count	2700	6189	352	626	1115	103	152
Satellite	Globalstar	Iridium	Tiros				
Count	18275	300	629				

For defunct satellite, most of the phase curves are from 37 Globalstar satellites, which are all in the rotating state with the periods from a few seconds to more than 100 seconds. The Tiros platform contains a series of satellites. The shape of different satellites may be different in detail, but they should be roughly the same. Since the number of phase curves of the Iridium series satellites is about 300, we also use the “data copy” method to expand the sample set. Therefore, the satellite platform classification experiments include 3 categories: Globalstar, Tiros, and Iridium.

The training set and the test set are divided into 3:2, and the experiment is repeated 100 times using the bootstrap method. Table 5 is the statistics of the classification accuracy. Overall, the accuracy of the rocket body is about 85%, and the confidence interval of 3σ is (82.60%, 88.73%), which is significantly lower than the result of the satellite whose accuracy is about 93%, and the confidence interval of 3σ is (88.98%, 96.20%). The classification result is coincident with our expectation, i.e., for rocket bodies it is more difficult to distinguish among them because of their relatively smaller difference in their typical light curves of double-peak structure caused by the cylinder-like shape, but for the 3 types of satellites they have obvious differences in shape and more complicated light curves which can provide more features for classification.

Table 5 The classification accuracy of space debris types

	Rocket	Satellite
mean	85.32%	93.18%
median	85.18%	93.33%
3σ interval	(82.60%, 88.73%)	(88.98%, 96.20%)

The confusion matrix of the classification result is shown in Figure 7. Except Atlas, the recognition rate of other rocket bodies is higher than 80%. The highest is Delta4 and H2A, whose actual pictures show that they are not completely similar to cylinders, but composed of a cylinder on the top and several ellipsoidal fuel tanks on the bottom. This complex structure could make more features in the light curve, which can improve the recognition rate. Meanwhile, due to the similar shape between Delta4 and H2A, the probability of confusion is relatively high. The sum of the probability of mutual misclassification between them is about 17%, which is the highest of all combinations. Another couple with high probability of mutual misclassification is CZ3 and Falcon9, which are all typical long cylindrical shapes, and usually coated with white paint on the surface.

The classification result of satellite platform shows that Tiros has the highest recognition rate, followed by Iridium. It should be noted that Globalstar and Iridium are highly distinguishable from each other, while the probability of mutual misclassification between Tiros and the others is significantly greater. The shape of the Globalstar is similar to the box-wing satellite, while the shapes of Iridium and Tiros are very irregular. The mutual misclassification can reflect the correlation of the shape among space debris.

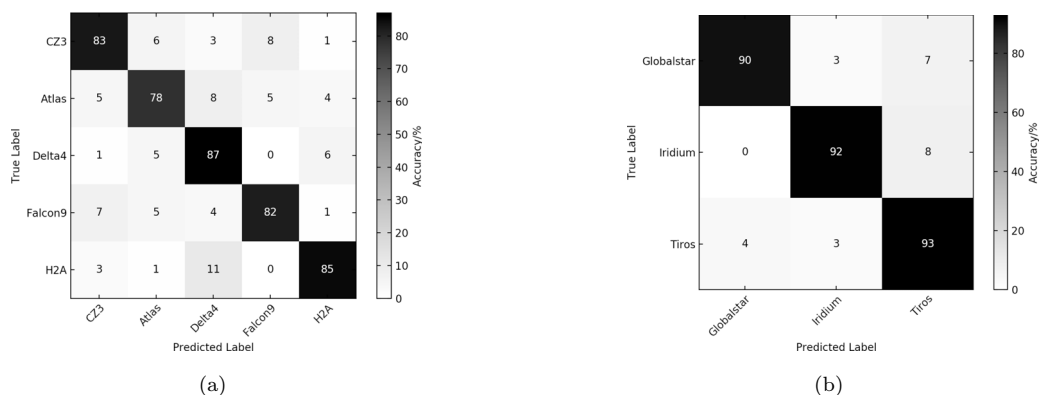


Fig. 7 The confusion matrix for the classification results of space debris types/bus-types (a) 5 types of rockets, (b) 3 bus-types of satellites.

6. SUMMARY AND OUTLOOK

This work discusses the classification method for the basic shape of space debris based on the light curves. The simulation and observational data classification experiments of the debris shape and series or platform have been done using the DNNs etc..

The simulation experiments include 4 categories: diffuse cylinder, specular cylinder, box-wing satellite, and cube satellite. The classification accuracy of the DNNs is 95.5%, in which the recognition rate of diffuse and specular cylinders is significantly higher than that of box-wing and cube satellites. The performance of KNN, SVM, RF, and DT algorithms is also tested on this problem, and the results show that DNNs is the best, followed by RF and SVM. In the observational data experiments, the light curves of Atlas, CZ3, Delta4, Falcon9, and H2A rocket bodies, and Globalstar, Iridium, and Tiros satellites are extracted from the MMT database. In the dichotomy test of rocket body and defunct satellite, the classification accuracy confidence interval of 3σ is (97.60%, 99.83%) with the DNNs and the bootstrap method. In addition, we use the RF algorithm to classify the debris series or platform, and the 3σ confidence interval of the classification accuracy is (82.60%, 88.73%) for rocket body series. In detail, the sum of the mutual misrecognition probability of H2A and Delta4 is 17%, and that of Falcon9 and CZ3 is 15%, indicating that there is a high similarity within the couple. The confidence interval of 3σ of the classification accuracy for the three satellite platforms is (88.98%, 96.20%). The results show that the difference between Globalstar and Iridium is more significant, and the misclassification probability of Tiros to the others is greater.

The simulation experiments in this paper only covers 4 types of debris, which are simplified in the shape and optical characteristic parameters. In fact, there are many types of space debris with more complex structures. The limitations of simulation experiments are obvious in such a context. For actual debris, giving the rough estimation of the shape is also worth studying. It is really a challenge to model the complex debris and generate the simulation data, due to the difficulty of obtaining the precise shape and material information, and describing the rotation with a simple model. In the next step, we plan to enlarge the types of debris, and test the recognition rate for randomly simulated debris, looking forward to give a larger range of debris shape classification conclusions. In addition, the ground photometric simulation laboratory for space debris can provide a large number of space debris simulation light curves with high accuracy. We plan to combine the observational data and the high fidelity simulation data as the training data set to achieve the correct classification of unknown debris found in observation.

References

- 1 Zhang Y. P., Zhao C. Y., Zhang X. X., et al., ChA&A, 2015, 39, 100
- 2 Wang W. Z., Li Z., Lai J. Z., et al., Space objects intelligent identification based on the data-driven method. Journal of Command and Control, 2019, 5, 25-30, in chinese
- 3 Pearson K. A., Palafox L., Griffith C. A., MNRAS, 2018, 474, 478
- 4 Shallue C. J., Vanderburg A., AJ, 2018, 155, 94
- 5 Howard M., Klem B., Gorman J., RSO Characterization with Photometric Data Using Machine Learning. Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Wailea, September 15-18, 2015
- 6 Linares R., Furfaro R., Space Object Classification Using Deep Convolutional Neural Networks. Proceedings of the 19th International Conference on Information Fusion (FUSION), Heidelberg, July 5-8, 2016
- 7 Furfaro R., Linares R., Reddy V., Space Objects Classification via Light-Curve Measurements: Deep Convolutional Neural Networks and Model-Based Transfer Learning. Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference, Maui, 2018
- 8 Furfaro R., Campbell T., Linares R., et al., Space Debris Identification and Characterization via Deep Meta-Learning. Proceedings of the 1st International Orbital Debris Conference, Sugar Land, 2019
- 9 Cook R. L., Torrance K. E., ACM Transactions on Graphics, 1982, 1, 7