Research Article

Spectral Feature Extraction Using Partial and General Method

Bin Jiang (ID)**, Xi Fang** (ID)**, Yang Liu** (ID)**, Xingzhu Wang** (ID)**, and Jie Liu** (ID)

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, Shandong 264209, China

Correspondence should be addressed to Jie Liu; liujie_sdwh@163.com

With the rapid growth in astronomical spectra produced by large sky survey telescopes, traditional manual classification processes can no longer fulfill the requirements of precision and efficiency of spectral classification. There is an urgent need to employ machine learning approaches to conduct automated spectral classification tasks. Feature extraction is a critical step which has a great impact on any classification result. In this paper, a novel gradient-based method together with principal component analysis is proposed for the extraction of partial features of stellar spectra, that is, a feature vector indicating obvious local changes in data, which corresponds to the element line positions in the spectra. Furthermore, a general feature vector is utilized as an additional characteristic centering on the overall tendency of spectra, which can indicate stellar effective temperature. The two feature vectors and raw data are input into three neural networks, respectively, for training and each network votes for a predicted category of spectra. By selecting the class having the maximum votes, different types of spectra can be classified with high accuracy. The experimental results prove that a better performance can be achieved using the partial and general methods in this paper. The method could also be applied to other similar one-dimensional spectra, and the concepts proposed could ultimately expand the scope of machine learning application in astronomical spectral processing.

1. Introduction

Stellar spectrum analysis is an important precondition for astrophysicist and astronomers to have a better understanding of the physical and chemical properties of stars. Stellar spectra classification is based on the MK standard and plays an important role in spectra analysis. The traditional method to address the spectral classification of stars is to combine photometric and spectroscopic data of stars. This task is very difficult for O/B stars, generally embedded in the gas of HII regions. Nolan and Edward [1] presented the Atlas for morphological investigations of OB spectra based on digital data which can help distinguish O/B stars. Lennon [2] developed the criteria for B-type supergiants contained in the Small Magellanic Cloud (SMC) and Evans and Howarth [3] addressed the relationship between spectral type and physical properties for A-type supergiants in the SMC. Bresolin et al. [4] presented the first spectral catalog of supergiant stars in the Local Group dwarf irregular galaxy WLM.

Feature extraction is still regarded as a difficult and vital issue at the beginning of machine learning task processing.

One spectrum from the Sloan Digital Sky Survey (SDSS [5, 6]) or the Large Sky Area Multiobject Fiber Spectroscopic Telescope (LAMOST) [7, 8]) can be viewed as a data item where every single feature represents the flux of each optical wavelength, usually ranging from about 3,800 to 9,000 Å. Therefore, there are over 5,000 features to be processed in each spectrum. When using traditional machine learning classifiers, such a mass of features will produce a large amount of calculations. Thus, feature extraction or selection process is of great necessity to save computational time.

There are several generally used feature extraction algorithms, such as principal component analysis (PCA) [9] and Linear Discriminant Analysis (LDA) [10], which perform well in dimensionality reduction. These algorithms can be directly used for most types of data processing. However, due to their generality, the results still have room for improvement when the reduced data is classified subsequently. Therefore, taking the present application into account, it is indispensable to work out certain algorithms that could exclusively be applied to the feature extraction of spectral data.

The main idea of this paper is to use SDSS stellar spectra as experimental data and extract spectral features from two perspectives, which corresponds to two different properties of spectra, namely, stellar effective temperature and elements. The purpose of this paper is to verify whether this method has the practical application value and can be implemented into the pipeline. The paper proceeds as follows:

(1) In Section 2, a brief introduction about the designation of our algorithms and the rationality of some methods used in this paper is given. Particularly, we introduce three extracted feature vectors applied for the training of three neural networks.

(2) Section 3 illustrates the implementation of three feature extracting methods and three neural networks. Then, we draw lessons from ensemble learning to design ways of training of three neural networks simultaneously.

(3) We utilize frameworks like TensorFlow and Scikit-learn to train three networks and show the experimental results in Section 4. For demonstration purposes, this section only carries out binary classification tasks. To prove that the feature extraction methods mentioned are insensitive to signal-to-noise ratio (SNR), this section also shows experiments on different SNR spectra.

(4) Section 5 proposes some improvements for extending the function of the algorithm mentioned in this paper, for example, the method to complete multiple classifications.

(5) Finally, we make a brief summary of the core concept and achievements in this paper and propose some future works that could be adopted for astronomy.

2. Features' Extraction from Different Perspectives

There are various feature extraction or selection methods used with machine learning which have definitive advantages. For example, convolutional neural network (CNN) [11] is an artificial neural network that has specialization for being able to detect patterns and make sense of them. CNN has multiple layers, including convolutional layer, nonlinearity layer, pooling layer, and fully connected layer. Convolutional layers apply a convolution operation to the input, passing the result to the next layer. A convolution converts all the pixels in its receptive field into a single value. For example, if a convolution is applied to an image, it will decrease the image size and bring all the information in the field together into a single pixel. The final output of the CNN is a vector and can be seen as the result of feature extraction. PCA [12] is used in exploratory data analysis and for making predictive models. It is normally used for dimensionality reduction by projecting each data point onto the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible. Therefore, it is beneficial to adopt a method which combines

multiple extraction features together to fully utilize existing feature extraction algorithms. Rivard et al. [13] have proposed that, by employing preexisting techniques to extract features at different scales, the diversity of information obtained could be improved and the personal variability in signature verification might be mitigated. With regard to one-dimensional celestial spectra, in general, it can be roughly divided into partial and general features from two perspectives, which usually correspond to physical properties of spectra.

2.1. Partial Feature Vector. To improve classification accuracy, the feature extraction process should focus more on local spectral features, such as emission lines and absorption lines. Absorption lines generally observed in stellar spectra are Balmer series. In the field of deep learning for image recognition, the designation of the CNN proposed weight sharing [14] based on the local connectivity of images, to reduce the number of parameters in the network and also to improve calculation speed. The reason for weight sharing is that one image can include a large amount of features. However, it should be noted that, compared to image data, celestial spectra have sparser features and therefore they could be analysed without any complicated calculation.

To prove the property of feature sparsity, that is, the overall features can be expressed by just a few features, we have designed an experiment based on PCA. The variance contribution rate in PCA is used to describe the ability to express high-dimensional data with data having few features. According to the theory of PCA, $\mathbf{Z} = \mathbf{XU}$ is valid for dimension reduction. Suppose $\mathbf{X}$ is the data set before dimensionality reduction, $\mathbf{Z}$ is the data after reduction, and U is the principal component feature matrix. On the contrary, the process of data recovery requires only the execution of $\mathbf{X_{rec}} = \mathbf{ZU^T}$, where $\mathbf{X_{rec}}$ represents the data recovered through the inverse process of PCA. The variance contribution rate in PCA, also known as the data recovery rate, is shown as

$$\frac{\sum_{i=1}^{m} x^{(i)} - x_{\text{rec}}^{(i)2}}{\sum_{i=1}^{m} \|x(i)\|}, \tag{1}$$

where $x^{(i)}$ represents each data in X and $x_{\text{rec}}^{(i)}$ is each data in X_{rec}. In formula (1), the numerator part stands for the square of average projection error, and the denominator part indicates the average distance from the training sample to the origin. In other words, it shows how much the original data can be preserved after dimensionality reduction. Figure 1 shows the respective variance contribution rates of the first five principal components when PCA is applied to SDSS spectra. It can be seen from the figure that the variance contribution rate of the first principal component is 96.59%, and the variance contribution rate of the second principal component is 3.07%. The total variance contribution rate of the first five principal components has reached 99.85%, indicating that PCA can theoretically be used to reduce the A-type stellar spectra to a vector with only five features while retaining most of the information.
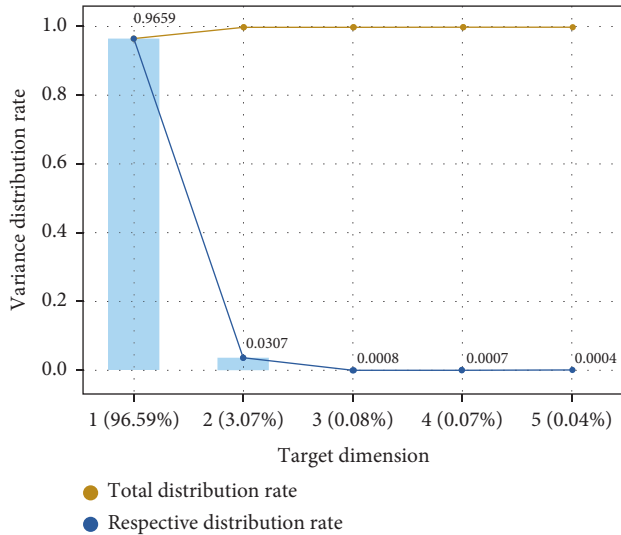
FIGURE 1: Take A-type stellar spectra as an example. The respective variance contribution rate of the first five of principal components when PCA is used in 6,000 spectra from SDSS. The value of the brown point represents the total variance distribution rate and the blue point represents the variance distribution rate of the specific principal component.

As can be seen in Figure 1, the recovery rate achieves 96.59% even when the overall features are reduced to just one, which demonstrates that A-type spectral features are suitable to be expressed sparsely. Other types of spectra are similar to class A, although there exist some numerical differences. In general, all spectral features are sparse. In the field of computer vision, the Histogram of Oriented Gradient (HOG) feature [15] is the type of feature extracting method used to gain a small number of local features from large images. Based on this method, we propose a gradient-based algorithm to extract the edge or partial features of spectra and V_{par} is a feature vector extracted by this algorithm. V_{par} is used for training one of the three neural networks.

2.2. General Feature Vector. In addition to local features, the overall characteristics of spectral data should also be taken into consideration. General characteristics of spectral data refer to the changing shape or trend of spectrum flux as the wavelength increases, which depends on the physical nature of spectra.

It only focuses on the trend of one specific spectrum, regardless of the absorption lines in a small wavelength range. To obtain the general feature vector, spectra should be smoothed to eliminate the effect of partial features, the size of spectral data should also be decreased into a small feature vector, and V_{gen} is the feature vector that only simultaneously represents the general features. Contrary to the partial feature vector, the general features' extraction process focuses on eliminating the edge features instead of maintaining them.

2.3. Learning Strategy. An obvious difference between machine learning and deep learning is that deep learning does not need to complete the feature extraction process; instead,

it applies deep neural networks to achieve the features automatically. In our model, we also take the convolutional neural network into consideration. After achieving a partial and general feature vector, we also defined V_{raw}, it is the raw spectral feature vector that having 3,522 features. To synthesize the effect of these three vectors, three neural networks are trained, respectively. This idea comes from ensemble learning [16], that is, training different classifiers to obtain the most possible results through statistical principles. For convenience, these three networks are denoted as N_{par}, N_{gen}, and N_{raw}. These networks adopt standardized V_{par}, V_{gen}, and V_{raw} as input separately and produce an output denoting the predicted category of input data. In order to display the experimental model intuitively, Figure 2 demonstrates the architecture of the whole process.

As shown in Figure 2, V_{par} is used as the input of the N_{par} network, V_{gen} is used as the input of the N_{gen} network, V_{raw} is used as the input of the N_{raw} network, and finally, the output of the three classifiers is used for voting. Although the three methods show large differences, the processing is essentially the same. All three methods firstly finish feature extraction and then train a neural network. This is consistent with the flow of classic machine learning problems. The difference is that when using convolutional networks, it can be considered that feature extraction is largely unnecessary.

After completing the training process of the three networks, test data is entered. Finally, categories having the highest votes among the three network outputs are chosen as the final decision. For example, assuming that classes only contain 0 and 1, if the output of the three networks is 0, 0, and 1, respectively, then the test spectrum classification would be labelled 0. It should be noted that networks N_{par} and N_{gen} are normal neural networks while N_{raw} is a convolutional network.

3. Partial and General Method

3.1. Gradient-Based Algorithm. To extract partial features and ignore the general trends of spectral data, gradient is an ideal way, due to its representation of local changes of spectra. In the field of computer vision, the HOG constructs features by calculating and grading gradient direction histogram of the local region of image [17]. The essence of HOG is the statistical information of the gradient, which mainly exists at the edge of images. HOG firstly divides the image into small connected areas, which are also called cell units. A histogram of the gradient direction of each pixel in the cell unit is then acquired. Finally, these histograms can then be combined to form a feature descriptor.

However, the traditional HOG method cannot be directly applied for celestial spectra, mainly because of the noise contained in spectral data which would cause flux disturbance in any adjacent wavelength range; thus, the derivation of each feature would be meaningless. Therefore, before calculating derivatives, the data should be firstly smoothed, which enables spectral data to be derived at each feature point. Thus, we use PCA mentioned above in this paper to smooth each feature point of the spectrum so that the derivative of the point is not affected by noise
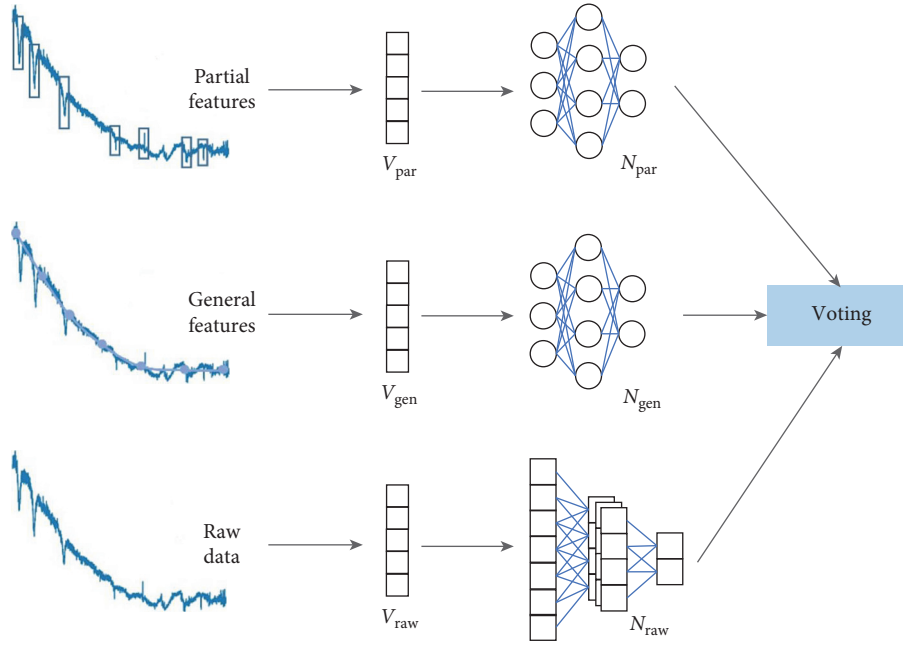
FIGURE 2: Experimental model architecture.

disturbance. The specific approach is to firstly reduce the number of the features from the data from 3,522 to 2 by PCA, thereby decreasing unimportant components. Then we aim to recover the data to the original 3,522 features using reverse PCA. In this process, noise is almost eliminated and most local or partial features are successfully preserved. For a specific classifier, we will therefore achieve higher accuracy for data reduced by PCA while the raw data is lower because of data noise.

While different from continuous functions, spectral data is discrete so that the gradient can be approximately calculated by adjacent feature points. On the other hand, both spectral data and image data are discrete data, and the difference between them is that, for spectral data, it only needs to calculate the gradient in one direction, while the gradient in image data should have multiple directions. Above all, the gradient features' map x_{der} of the spectrum can be obtained by applying the formula as follows:

$$
x_{der}^{(i)} \begin{cases} 0, & (i = 0), \\ \dfrac{x_{der}^{(i)} - x_{der}^{(i-1)}}{i - (i-1)}, & (i \neq 0). \end{cases} \tag{2}
$$

After calculating the gradient, x_{der} is also a vector having 3,522 features. However, it just preserves the partial features of spectra such as emission lines and absorption lines and it eliminates the effect of the general trend, that is, continuum. At this point, we apply PCA again to reduce the gradient vector x_{der} to only 12 features maintaining a large variance contribution rate; thus, the reduced 12 features could include almost all the partial features of spectral data. The vector consisting of these 12 features is the V_{Par} mentioned above, which will be the input of N_{Par}.

In general, the process of extracting V_{par} utilizes the principal components of the spectral data gradient map instead of the principal components of our original data. Figure 3 summarizes the entire extraction process. The spectral segments in the box represent local features.

In the process, the raw data experiences both PCA and inverse PCA processes to firstly reduce the noise in data and then calculate the derivation of each feature point to obtain a gradient map representing the partial features of raw data. The final step is to reduce the dimensionality by PCA again to form a partial feature vector V_{par}.

3.2. Pooling for General Features' Extraction.
Partial features like emission and absorption lines represent the local features of spectra. General features such as continuum of the whole spectrum describe the overall trend of the data. Both of them are crucial for improving classification accuracy and should be taken into consideration for classification.

Figure 4 illustrates the general shapes of four classes of stellar spectra.

As can be seen from the figure, when ignoring the emission and absorption lines in the spectrum and just focusing on the entire tendency, as the wavelength increases, it could be found that the flux of the O-type spectrum decreases while the K-type spectrum experiences a rising phase followed by a gradual decrease. Furthermore, the flux of the M-type spectrum shows an upward tendency in general. However, it is difficult to classify K-type and F-type spectra because when partial characteristics are discounted, they show a strong similarity in general. This situation explains the reason for extracting various spectral features for classification, that is, using one feature vector to compensate for the defect of another feature vector, so that data information can be obtained comprehensively.
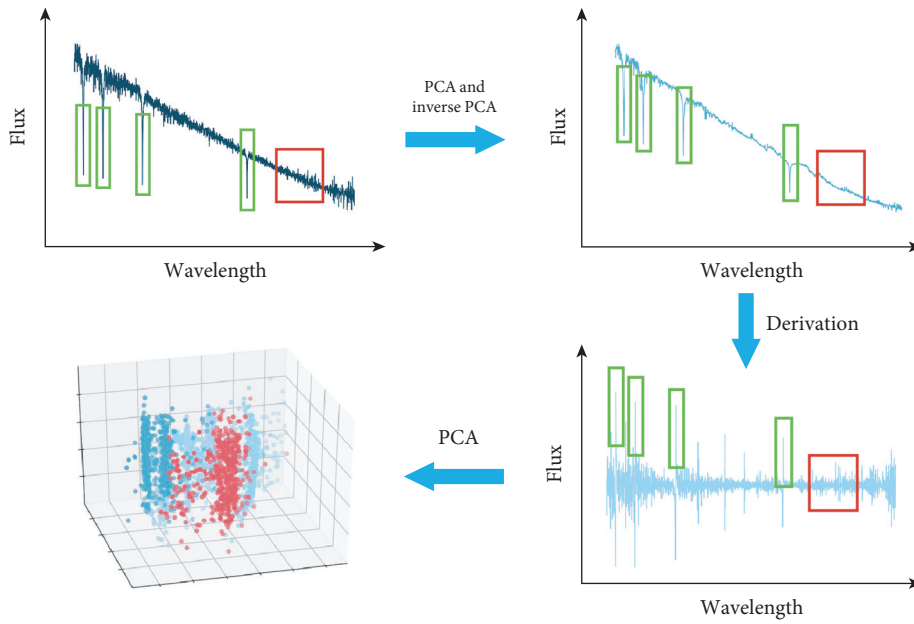
FIGURE 3: Partial features' extraction process.
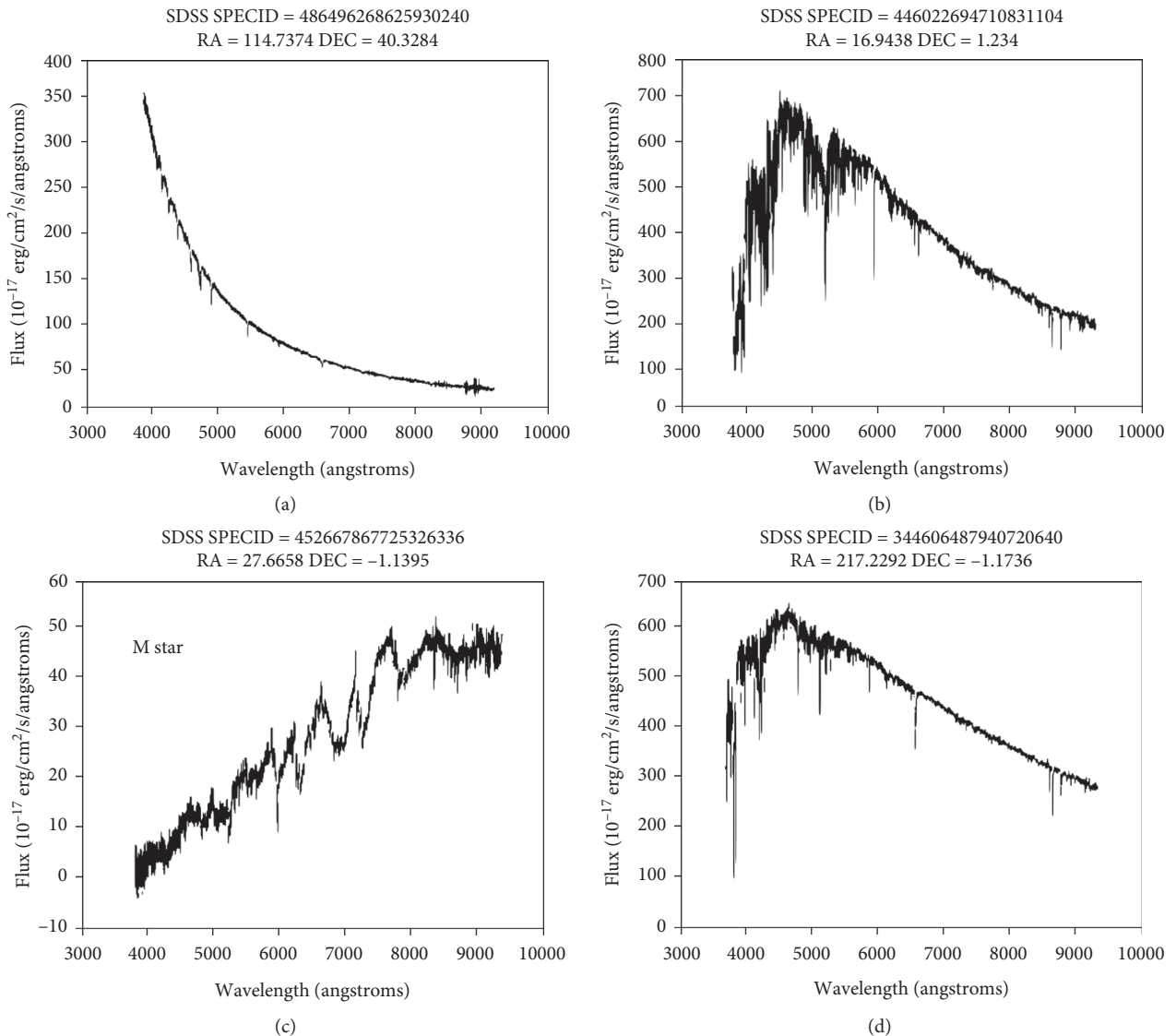

(a)

(b)

(c)

(d)

FIGURE 4: O-type, K-type, M-type, and F-type spectra.

We have now completed this method to extract partial feature vector. For general features, we draw on the idea of average pooling [17], that is, for the CNN to reduce the resolution of data and reduce spectral data of 3,522 features to only 7 features which are smooth enough to represent the overall trend of spectra. Firstly, 22 random sample points are eliminated for the convenience of calculation. For a partial feature analysis, it is unreasonable to drop feature points randomly. However, it has little impact on the general trend of spectral data. Here, the pooling unit slides by a certain step length until reaching the end. This step is repeated four times and finally, seven-feature data set can be gained. Figure 5 demonstrates the whole procedure and effect of generating the general feature vector. It should be noted that both emission lines and absorption lines together with noise in each spectrum can be completely eliminated; thus, average pooling is used instead of max pooling or other pooling models.

Convolutional neural network [18] is a feed-forward network used for image classification and recognition. Although images are two-dimensional data, while spectra are data with one dimension, both of them have the property of local continuity. Because of the similarity between these two data sets, when just focusing on one dimension, meaning that the convolutional kernels only slip horizontally in spectral data, the network can also achieve high classification accuracy by extracting convolutional features from original data. Therefore, with the third network, the original data is directly input without the process of feature extraction.

According to the interpretability and visualization theory of the convolutional neural network [19], some partial features can be recognized in shallow network layers and more thorough characteristics can be extracted in a deep network convolutional kernel. This illustrates that the more the network layers, the more complex the extracted features. Moreover, it does not need complex network architecture because it may achieve overfitting [20] and much noise in spectra could be extracted. Finally, we decided to apply a structure similar to LeNet-5 [21], where the version is used for two-dimensional digits' recognition. Figure 6 shows the distinctions between LeNet-5 and the network used in this paper.

4. Results and Discussion

A complete machine learning process is implemented after the architecture of the three networks is designed. In this experiment, the task flow includes data preprocessing, feature extraction, model training and testing, model evaluation, and final optimization. Because of the process of feature extracting, the amount of data has increased. Therefore, the three feature vectors can be stored as three separate files and be trained simultaneously by a multicore processor or a graphic processing unit (GPU) to improve training speed. It is a simple way for achieving parallelization in this experiment.

4.1. Data Collection. The experiment uses stellar spectra from A-type to M-type to complete a binary classification task. Each group has 5,000 spectra from the SDSS database

[22], respectively. These stars are in the Milky Way and have already been applied to the reddening correction and subtraction of sky lines. SDSS data is stored in the form of Flexible Image Transport System files (FITS). These files can contain both images and binary data tables in a well-defined format. Stellar spectral data is stored in a FITS file in a one-dimensional file format. This experiment intercepted the 3,800 to 9,000 Å part of each spectrum and converted it into 3,522 feature points to form the experimental data set. SNR (signal-to-noise ratio) is a primary indicator to test the quality of spectra. The SNR of "15" is an empirical value for qualified spectra. We conducted a query in the database of SDSS with the criterion of SNR over 15 and can achieve 5,000 spectra for each group. Thus, in the first phase of the experiment, in order to achieve higher accuracy and guarantee the experimental number, only spectra with an SNR over 15 are selected for the experiment and low-quality spectra are eliminated to guarantee the reliability of the model.

4.2. Data Preprocessing. Flux of the same type of spectra will be of great difference under the influence of various factors. In other words, the same type of spectra has the same pattern, but there is a gap in their order of magnitude. Therefore, the spectra need to be normalized to a certain scale. For example, in an image processing field, each pixel value ranges from 0 to 255 while all pixels can be normalized from 0 to 1. In this experiment, each spectrum is mapped with an [0–1] interval through min-max normalization [23] to ensure that all data has the same order of magnitude:

$$x_{\mathrm{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{3}$$

x in this formula represents the flux of the single feature in the stellar spectrum, $x_{\min}$ and $x_{\max}$ represent the minimum and maximum flux in the stellar spectrum, respectively, and x_{norm} is the normalized result. Besides, in addition to raw data, we also obtain partial feature vectors and general feature vectors by using the methods mentioned above, and these newly generated data should also be standardized in the same way.

4.3. Networks Designing and Training. The complexity of data for classification is usually relevant to the number of network layers, which should be considered when designing a network structure. In large-scale image recognition tasks, some neural networks can even consist of hundreds of layers. Nevertheless, after our experiment for spectral feature extraction, V_{par} and V_{gen} are small feature vectors, while a network with three or four layers is sufficient for their classification task. Deeper network structures can easily lead to overfitting. As mentioned earlier, the 1D convolutional network is similar to the LeNet-5. According to the above, we designed three networks as shown in Tables 1–3, respectively. "Net" indicates the name of the network, "Layer" indicates a specific layer in the network, "Neuron" indicates the number of neurons in this layer, and "Activation function" indicates the activation function used in this layer, while "Type" indicates the type of this layer.
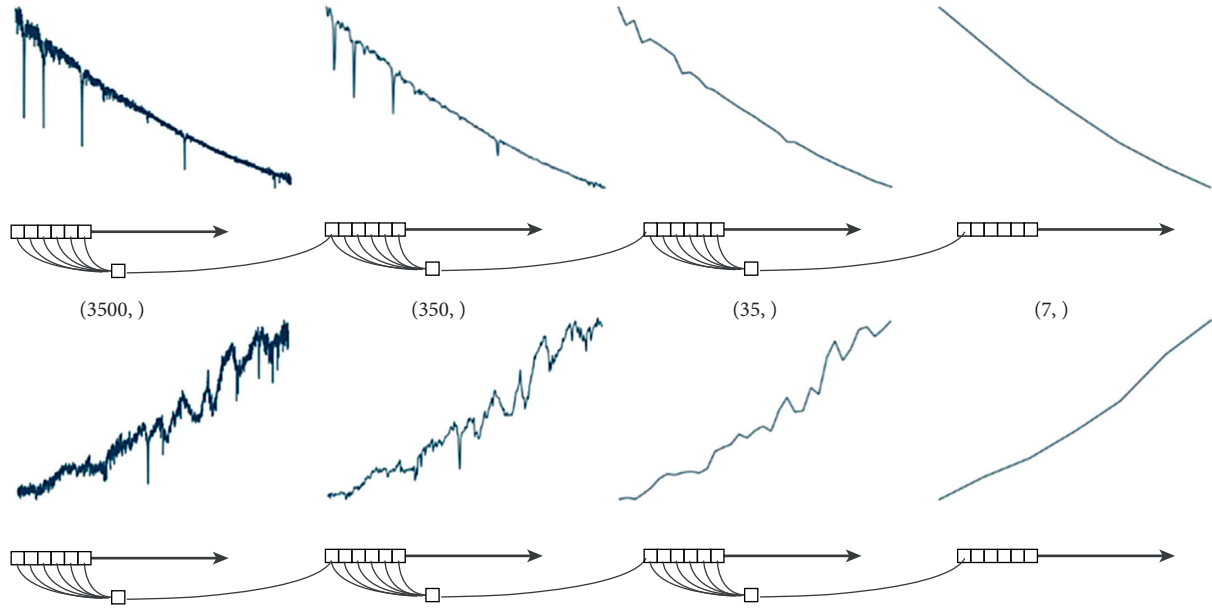
FIGURE 5: The procedure of average pooling for extracting general features of spectral data.1D convolutional neural network for spectral classification.
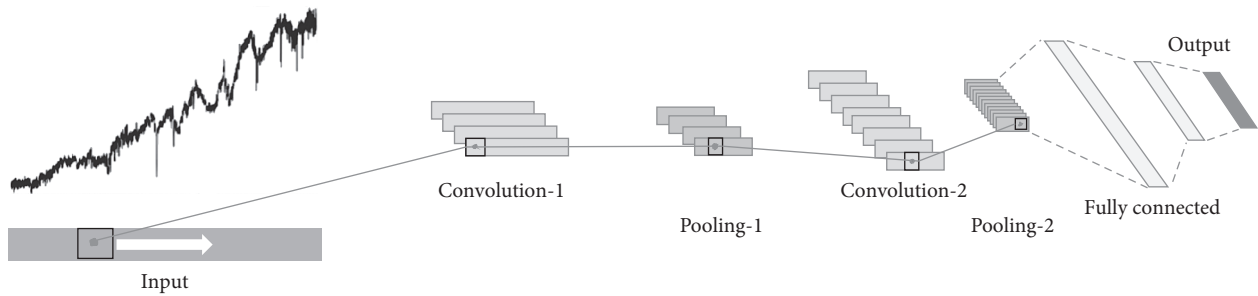

FIGURE 6: The network architecture based on LeNet-5 with two convolutional layers, two pooling layers, and two fully connected layers in total.

TABLE 1: Network details of N_{Par}.

Net	Layer	Neuron	Activation function
N_{Par}	1	(12)	ReLU
N_{Par}	2	(8)	ReLU
N_{Par}	3	(4)	ReLU
N_{Par}	4	(2)	Softmax

TABLE 2: Network details of N_{gen}.

Net	Layer	Neuron	Activation function
N_{gen}	1	(7)	ReLU
N_{gen}	2	(4)	ReLU
N_{gen}	3	(2)	Softmax

TABLE 3: Network details of N_{raw}.

Net	Layer	Neuron	Activation function	Type
N_{raw}	1	(705, 32)	ReLU	Convolution
N_{raw}	2	(88, 32)	—	Pooling
N_{raw}	3	(18, 64)	ReLU	Convolution
N_{raw}	4	(2, 64)	—	Pooling
N_{raw}	5	(64)	ReLU	Full-connect
N_{raw}	6	(2)	Softmax	Full-connect

4.4. Splitting a Table into Multiple Horizontal Components. We used TensorFlow to construct the network architecture and apply the minibatch gradient descent method to train them for 20 epochs, of which the whole data set is used for training once. For the sake of consistency, a common loss function cross-entropy is chosen. Moreover, in order to test the accuracy, we divided the data set into a training set (75%) and a testing set (25%).

4.5. Classification Results. The loss functions of the three networks have decreased to a small value after training and their training times (epochs) also vary for different complexities of the three networks. Simultaneously, the accuracy of the final vote increases continually. Figure 7 illustrates the training process of the three networks.

The structure and input vector of the three networks are different, and the loss function curve during the training process is different too. It can be seen from Figure 7 that as the training progresses, the loss values of the three networks all decrease quickly during the first period of time, and the accuracy of the model increases quickly simultaneously. The classification accuracy in the testing data reaches 99.86%

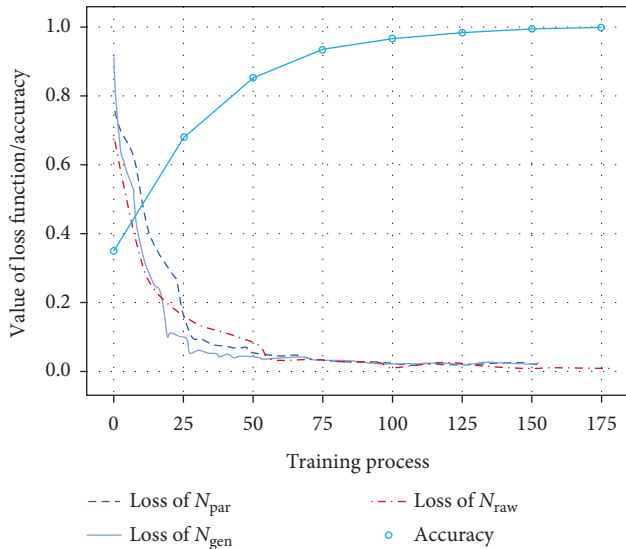| Ensemble networks		Individual network	
Stellar type	Accuracy (%)	Stellar type	Accuracy (%)
A&M	99.86	A&M	99.71
F&O	99.89	F&O	99.88
K&F	99.27	K&F	98.56
G&M	99.60	G&M	99.52

& equals and.

FIGURE 7: The values of loss functions and accuracy vary with the training process of three neural networks.

overall. Besides, the results of the three individual networks reach 98.92%, 98.60%, and 99.76%, respectively.

The primary result indicates that the ensemble method shows better performance than individual results. It can be explained that the ensemble method considers features of spectra more comprehensively than individual methods. In order to prove that methods mentioned in this paper can be widely applied in all types of spectral data, we used various types of spectral data in the SDSS database to conduct experiments for comparison and the results are shown in Table 4. For proving that extracted features can replace raw data with 3,522 features, a comparison with an individual neural network is made in addition.

The accuracy of each classification task exceeds 99.00%. Even K and F class data can achieve an accuracy of 99.27%, which means that although one feature extraction method is unstable, the other two methods can make up for the shortcomings of this method. It demonstrates that the feature extraction methods and classifier designation are robust enough to be directly used for stellar spectral classification.

The SNR is also an important factor for the classification result. In the above experiment, data with an SNR larger than 15 are chosen. However, low SNR data are also plentiful. If our methods can achieve a high accuracy rate for these data, it would be of great significance.

Accordingly, we designed another experiment to compare the classification accuracy on diverse data having a different SNR range, which is illustrated in Figure 8.

It can be seen from the figure that as the SNR of the stellar spectra increases, the accuracy of the binary classification of each group of stars does not rise steadily. Even in the K-type and F-type binary classification experiments, the spectra of stars with SNR less than 15 can get better classification accuracy than the spectra of stars with SNR greater than 15. This may be due to the fact that stars with a higher SNR stellar spectrum in the data set are relatively small. The experimental results show the insensitivity for SNR of partial and general feature extraction methods, indicating that the SNR cannot influence the effect of the algorithms in this paper to a large extent. Because spectra with a high SNR are limited, which affects the training of neural networks, the classification effect of higher SNR data is not as good as that of the data having a lower SNR. The robustness of our model is proven again according to the results.

4.6. Improvements and Future Works. When class voting is performed using the methods described above, only two-class tasks can be completed because there are just three classifiers, and if the number of classes increases, it would produce contradictory classification results. When considering multiclassification tasks, it is certainly possible to use the One-versus-One (OvO) or One-versus-Rest (OvR) methods as is in traditional machine learning, but it is also feasible to achieve multiclassification by increasing the number of classifiers. In the simplest case, if there are N target categories, we can set $N + 1$ classifiers for classification and then select the category with the highest number of votes. Furthermore, if we increase the number of networks to several hundred such as those used for random forest, the model's generalization capability would increase remarkably.

Another method is to set up multiple neural network classifiers while giving each classifier a weight. For instance, in this paper, it is workable to obtain the classification accuracy when three networks are used for classification, respectively. Assume that they obtained a classification accuracy of 100%, 90%, or 80%, respectively, then give higher weight to the network with higher classification accuracy, for example, the weights of these three networks are set as 0.371, 0.333, and 0.296, respectively. Specific weight values could then be set according to the experimental results.

Above all, the improved methods can be integrated as shown in Figure 9. For any future improvement for multiclassification, the experiment set M feature extraction methods and N classifiers ($N > M$) could be utilized. One extracted feature vector could be adopted as inputs of multiple classifiers while each classifier is assigned a weight, which is calculated in the final ensemble process for voting.

In future work, we could attempt to search for the spectra of special and rare stars such as carbon stars and cataclysmic variables (CVs). Two of the feature vectors proposed could also be improved. For example, the partial
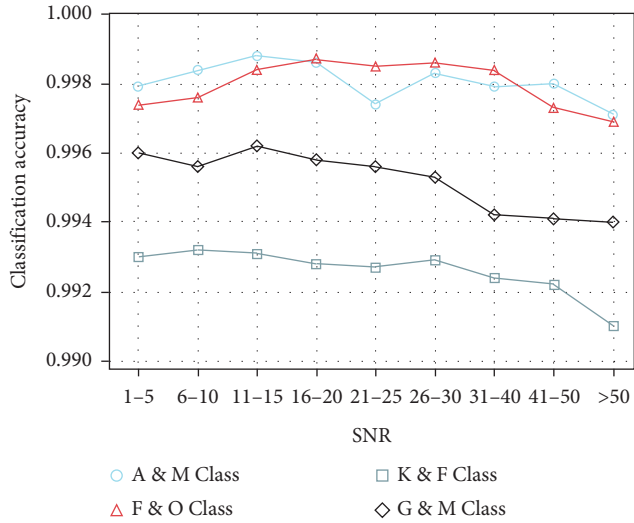
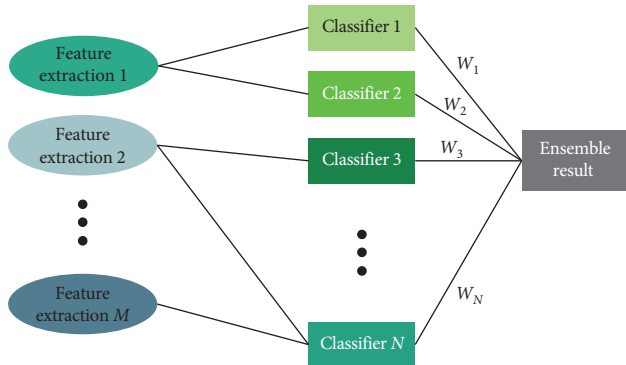FIGURE 8: The accuracy of binary classification of each group of stars under different SNR stellar spectra.

FIGURE 9: Improved feature extraction and classification process for achieving multiclassification task.

feature vector can be modified to record locations of each emission line and absorption line, instead of using PCA to obtain the feature vector.

5. Conclusions

Stellar spectrum analysis is crucial for the in-depth study and further development of astrophysics. There is a great need for automatic classification of specified objects in massive spectra. This paper designs three feature extraction methods and three neural networks for spectral data classification. The input of each neural network comes from different feature vectors, and the classification of each spectrum is carried out by the number of class votes. The experiment classifies A and M class spectra taken from the SDSS database and achieves high classification accuracy. Finally, we propose some practical recommendations and methods to expand the scope of algorithmic applications. The idea forming this paper has been tested through the appropriate experiments and could be widely applied in completing all kinds of classification tasks. In the future, we hope to develop and implement our method for the LAMOST data processing pipeline.

Data Availability

The SDSS data used to support the findings of this study are available from the SDSS website (http://www.sdss.org).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] N. R. Walborn and E. L. Fitzpatrick, "Contemporary optical spectral classification of the OB stars: a digital atlas," *Publications of the Astronomical Society of the Pacific*, vol. 102, p. 379, 1990.

[2] D. J. Lennon, "Revised spectral types for 64 B-supergiants in the Small Magellanic Cloud: metallicity effects[J]," *Astronomy and Astrophysics*, vol. 317, pp. 871–882, 1997.

[3] C. J. Evans and I. D. Howarth, "Characteristics and classification of A-type supergiants in the small magellanic Cloud," *Monthly Notices of the Royal Astronomical Society*, vol. 345, no. 4, pp. 1223–1235, 2003.

[4] F. Bresolin, G. Pietrzyński, M. A. Urbaneja, W. Gieren, R. P. Kudritzki, and K. A. Venn, "The Araucaria project: VLT spectra of blue supergiants in WLM- classification and first abundances," *The Astrophysical Journal*, vol. 648, no. 2, pp. 1007–1019, 2006.

[5] B. Yanny, C. Rockosi, H. J. Newberg et al., "A spectroscopic survey of 240,000 stars with $g = 14$–20," *The Astronomical Journal*, vol. 137, no. 5, pp. 4377–4399, 2009.

[6] M. R. Blanton, M. A. Bershady, B. Abolfathi et al., "Sloan digital sky survey IV: mapping the milky way," *Nearby Galaxies, and the Distant Universe*", vol. 154, no. 1, p. 28, 2017.

[7] X.-Q. Cui, Y.-H. Zhao, Y.-Q. Chu et al., "The large sky area multi-object fiber spectroscopic telescope (LAMOST)," *Research in Astronomy and Astrophysics*, vol. 12, no. 9, pp. 1197–1242, 2012.

[8] G. Zhao, Y.-H. Zhao, Y.-Q. Chu, Y.-P. Jing, and L.-C. Deng, "LAMOST spectral survey - an overview," *Research in Astronomy and Astrophysics*, vol. 12, no. 7, pp. 723–734, 2012.

[9] B. R. Bakshi, "Multiscale pca with application to multivariate statistical process monitoring," *AIChE Journal*, vol. 44, no. 7, pp. 1596–1610, 1998.

[10] J. Yang and J.-y. Yang, "Why can lda be performed in pca transformed space?" *Pattern Recognition*, vol. 36, no. 2, pp. 563–566, 2003.

[11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Antalya, Turkey, May 2017.

[12] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[13] D. Rivard, E. Granger, R. Sabourin, E. Granger, R. Sabourin, and R. Sabourin, "Multi-feature extraction and selection in writer-independent off-line signature verificationfication," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 1, pp. 83–103, 2013.

[14] J. Garland and D. Gregg, "Low complexity multiply-accumulate units for convolutional neural networks with weight-sharing," *ACM Transactions on Architecture and Code Optimization*, vol. 15, no. 3, pp. 1–24, 2018.

[15] A. Miermont, J. Uhlendorf, M. McClean, and P. Hersen, "The dynamical systems properties of the hog signalling cascade," *Journal of Signal Transduction*, vol. 2011, pp. 930940–931012, 2011.

[16] W. Li, T. Wang, M. Zhang, C. Wang, G. Shan, and H. Snoussi, "Region based ensemble learning network for fine-grained classification," *IEEE*, vol. 2019, pp. 4173–4177, 2018.

[17] H. G. Jung, "Analysis of reduced-set construction using image reconstruction from a HOG feature vector," *IET Computer Vision*, vol. 11, no. 8, pp. 725–732, 2017.

[18] P. Ballester and R. M. Araujo, "On the Performance of Googlenet and Alexnet Applied to Sketches," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pp. 1124–1128, Phoenix, AZ, USA, February 2016.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Computer Vision - ECCV 2014*, vol. 8689, pp. 818–833, 2014.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012.

[21] Z.-H. Zhao, S.-P. Yang, and Z.-Q. Ma, "License plate character recognition based on convolutional neural network lenet-5," *Xitong Fangzhen Xuebao/Journal of System Simulation*, vol. 22, no. 3, pp. 638–641, 2010.

[22] M. R. Blanton, M. A. Bershady, B. Abolfathi et al., "Sloan digital sky survey iv: mapping the milky way, nearby galaxies, and the distant universe," *Astronomical Journal*, vol. 154, no. 1, p. 28, 2017.

[23] S. Patro and K. K. Sahu, "Normalization: a preprocessing stage," 2015, http://arxiv.org/abs/1503.06462.