

## Research Article

# Automated Stellar Spectra Classification with Ensemble Convolutional Neural Network

Zhuang Zhao, Jiyu Wei, and Bin Jiang 

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

Correspondence should be addressed to Bin Jiang; [jiangbin@sdu.edu.cn](mailto:jiangbin@sdu.edu.cn)

Received 22 June 2021; Revised 13 December 2021; Accepted 16 December 2021; Published 4 January 2022

Academic Editor: Kadri Yakut

Copyright © 2022 Zhuang Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Large sky survey telescopes have produced a tremendous amount of astronomical data, including spectra. Machine learning methods must be employed to automatically process the spectral data obtained by these telescopes. Classification of stellar spectra by applying deep learning is an important research direction for the automatic classification of high-dimensional celestial spectra. In this paper, a robust ensemble convolutional neural network (ECNN) was designed and applied to improve the classification accuracy of massive stellar spectra from the Sloan digital sky survey. We designed six classifiers which consist six different convolutional neural networks (CNN), respectively, to recognize the spectra in DR16. Then, according the cross-entropy testing error of the spectra at different signal-to-noise ratios, we integrate the results of different classifiers in an ensemble learning way to improve the effect of classification. The experimental result proved that our one-dimensional ECNN strategy could achieve 95.0% accuracy in the classification task of the stellar spectra, a level of accuracy that exceeds that of the classical principal component analysis and support vector machine model.

## 1. Introduction

An avalanche of astronomical data is expected with the completion of state-of-the-art survey telescopes, such as the Sloan Digital Sky Survey (SDSS) and Large Sky Area Multiobject Fiber Spectroscopy Telescope (LAMOST). SDSS [1–4] is a multiobject fiber spectroscopic telescope that is considered the most influential astronomical telescope in the world. Data Release 16 (DR16) [5] is the fourth data release of the fourth phase of SDSS. LAMOST [6, 7] is a reflecting Schmidt telescope with its optical axis fixed along the north-south meridian. Both the Schmidt mirror and the primary mirror are segmented. The focal surface is circular with a diameter of 1.75 meters ( $\sim 5^\circ$ ), with 4,000 fibers evenly distributed over it.

SDSS and LAMOST surveys have produced massive amounts of spectral data. In astronomy, stellar spectra classification is the classification of stars based on their spectral characteristics. Electromagnetic radiation from stars is analyzed by splitting it with a prism or by diffraction grating into a spectrum to exhibit the colors of the rainbow

interspersed with spectral lines. Each line indicates a particular chemical element or molecule, with the strength of the line indicating the abundance of that element. The strengths of different spectral lines vary mainly because of the temperature of the photosphere, although in some cases the differences in abundance are true. The spectral class of a star is a short code that primarily summarizes the ionization state, thereby providing an objective measure of the photosphere's temperature.

With the improvement of computing power, machine learning and deep learning methods have become popular in the field of astronomy. Gabruseva et al. [8] used an algorithm for gradient boosting of decision trees for automated classification of photometric light curves for various astronomical objects. Li et al. [9] applied principal component analysis (PCA) and random forest (RF) to classify stellar spectra; they achieved 93% accuracy in this task. Vioque et al. [10] found 693 classical Be stars from massive amounts of data. Hon et al. [11] achieved excellent results in spectral classification through 1D convolutional neural network (CNN).

However, the large sky survey projects have vast amounts of spectral data with a low signal-to-noise ratio (SNR). These methods often fail to achieve ideal results because of the low SNR spectra.

To address these problems, this work aimed to improve the topological structure of traditional convolutional neural network (CNN), and an ensemble convolutional neural network (ECNN) model was proposed and applied to improve the classification accuracy. CNN is a classical deep learning method proposed by LeCun et al. [12, 13]. Instead of relying on experience to manually extract features, CNN can automatically learn hierarchical features layer-by-layer. Ensemble learning [14–19] is a machine learning method that combines multiple individual learners to obtain a better result. Most existing ensemble learning algorithms can be classified into two groups: Boosting algorithms [20–23] and Bagging algorithms [24–26]. The former increases the weight of the training sample, such as AdaBoost [20, 21], or constructs the label value (GBDT [22] and XGBoost [23]) to train each weak learner in turn. Each weak learner is related to each other, and the latter weak learner uses the information of the former weak learner. The latter trains each weak learner by randomly sampling the original training sample set and forming different training sample sets. Each weak learner can be considered to be approximately independent, and the typical representative is random forest [26]. With the development of machine learning and deep learning, more ensemble learning algorithms have been proposed and great breakthroughs have been made in many fields.

In this work, we designed an ensemble convolutional neural network (ECNN) in a bagging way. By building multiple basic classifiers and gently integrates each learning result according to certain integration strategies, the generalization ability of the model was improved and obtains better results than a single basic classifier.

## 2. Experimental Data

M-type dwarfs are stars in the main sequence stage and are the most common type of stars in the Milky Way [27]. M-type dwarfs have a low brightness, a small diameter and mass, and surface temperatures below 3500 K, and a long life because the speed of hydrogen fusion inside them is slow [28]. Such celestial bodies exist in all stages of the evolution of the Milky Way. Thus, they can reflect various pieces of information of the galaxy during the evolution period, and they are often used to track the structure and evolution of the Milky Way [29]. Aside from their importance in studying the nature of the Milky Way, M-type dwarfs are necessary for finding exoplanets that may be suitable for human habitation.

A total of 28,925 M-type stellar spectra were selected from the SDSS–DR16 with different SNR distributions in the dataset. Only the spectra of M0–M4 were selected owing to the scarcity of M5–M9 spectra. Wavelength was normalized to 4,000 Å–9,000 Å, and the flux of each original spectra was normalized in data preprocessing as follows:

$$s'_i = \frac{s_i - \text{mean}(S_i)}{\sigma(S_i)}, \quad (1)$$

where  $S_i$  represents the 1D vector formed by the spectral flux,  $\text{mean}(\bullet)$  is the mean factor, and  $b(-)$  is the standard deviation. The SNR and types of spectra are two important factors that influence the classification. Both of them were analyzed and discussed in this experiment. The distributions of both the spectral subclass types and the SNR of the experimental spectra are shown in Figure 1.

As can be seen from Figure 1, most of the spectra were low SNR ( $5 < \text{SNR} < 10$ ). In subsequent experiments, these data were divided into a training set, a validation set, and a test set at a ratio of 6:2:2. Specifically, the different SNR groups under each subclass were divided at a ratio of 6:2:2 and finally merged into three datasets. In this manner, the training set, the validation set, the test set, and the overall data could maintain better consistency.

## 3. Method

### 3.1. Basic Classifiers Based on Convolutional Neural Network

**3.1.1. Convolutional Neural Network.** Classic CNN architecture has two obvious characteristics: local connection and weight sharing. Local connections ensure that the network can extract different local features, whereas weight sharing greatly reduces the number of weights that must be trained. In this work, a CNN model with different structures was designed and an ensemble CNN classifier was applied.

The general structure of CNN typically includes an input, a convolution, pooling, and connected and output layers [30]. As the most important part of CNN, the convolutional layer is mainly used to extract features. It uses multiple convolution kernels to convolve the features of the previous layer and nonlinearly maps the results through the activation function to finally generate a feature map. The convolution operation is shown below:

$$x_j^n = f\left(\sum_{i \in M} w_{ij}^n x_i^{n-1} + b_j^l\right), \quad (2)$$

where is  $X$  the  $j$ th feature of the  $n$ th layer,  $M$  is the set of input feature maps,  $w$  represents the convolutional kernels,  $b$  is the offset value, and  $f(\cdot)$  is the activation function. The activation function used in this experiment was the rectified linear unit shown below:

$$f(x) = \max(x, 0). \quad (3)$$

The pooling layer, also known as the downsampling layer, is primarily used to compress the amount of data and parameters, thereby reducing the amount of computation. The common pooling methods are max-pooling and mean pooling. Max-pooling was adopted in this work. The fully connected layer was used at the end of CNN, and all neurons between the two layers were connected, as shown in the left panel of Figure 2. To effectively alleviate the overfitting in the training process, this work used the dropout technique in the first fully connected layer. During the training process, the

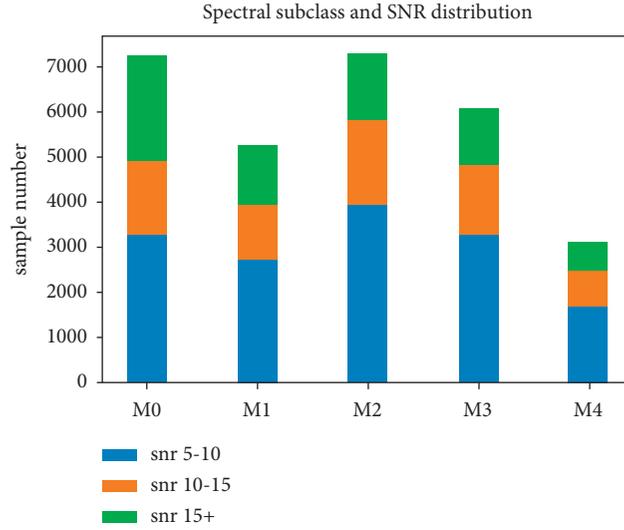


FIGURE 1: Spectral subclass and SNR distribution.

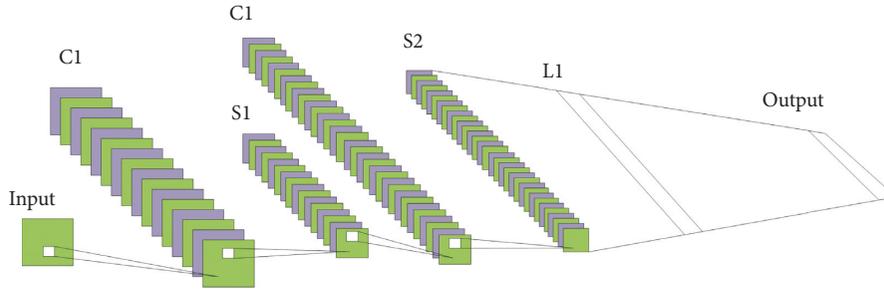


FIGURE 2: The structure of CNN2.

connections between some nodes were randomly discarded according to a certain probability, as shown in the right panel of Figure 2.

This work used the Softmax function after the original output layer, which turned the original output of the model into a probability distribution, resulting in the final output layer. Suppose the original output of the model is  $y_1, y_2, \dots, y_m$ , the Softmax function is calculated using the following equation:

$$f(y_i) = \frac{e^{y_i}}{\sum_{j=1}^n e^{y_j}}. \quad (4)$$

**3.1.2. Basic Classifiers.** In this work, we built six CNN models with different structures as the basic classifiers. In order to measure the effect of classification, we use the cross-entropy loss function to measure the difference between predict and true distribution of the classifier's predicted output and use backpropagation to optimize the parameter of the classifiers:

$$F(x) = \sum_i^n \sum_j^k y_{ij} \log F(x_{ij}), \quad (5)$$

where  $n$  indicates the number of spectra,  $k$  indicate the classes of spectra,  $y_{ij}$  indicates the class of  $x_{ij}$ , and  $F(x_{ij})$  is the probability that the classifier predicts that  $x_{ij}$  belongs to class  $j$ .

In fact, the depth of the network had a great impact on the learning effect, and CNN1–CNN4 were 1D convolutional networks that varied in depth from small to large. The difference in the dimensions of the processed data also affected the learning effect, and CNN5–CNN6 were 2D convolutional networks [13]. To obtain the 2D data input to CNN5 and CNN6, we changed the original  $1 \times 5,000$  1D data into two different  $50 \times 100$  2D data by using the following equations:

$$x'(i, j) = x(i \times 100 + j), \quad (6)$$

$$x'(i, j) = x(i + (j - 1) \times 50). \quad (7)$$

The structural parameters of CNN1–CNN6 are given in Table 1, where “ $1 \times 5-16$ ” indicates that the convolution kernel had a size of  $1 \times 5$  and the number of channels was 16. Figure 2 shows the structure of CNN2.

**3.2. Ensemble CNN.** A classical Bagging algorithms aims to train  $k$  independent base learners and integrate the results of each base learner to obtain a strong learner. Therefore, there are three problems to be solved: (a) how to train base learners and (b) how to integrate the results of each base learner. In this paper, according the Bagging algorithms, we apply Bootstrap method and weighted voting as the training

TABLE 1: The structural parameters of the classifier.

Model	Input	CNN	Pooling	CNN	Pooling	CNN'	Pooling	FCN	Output
CNN1	1 × 3500	1 × 5-16	1 × 3	—	—	—	—	1 × 128	1 × 5
CNN2	1 × 3500	1 × 5-16	1 × 3	1 × 3-32	1 × 2	—	—	1 × 128	1 × 5
CNN3	1 × 3500	1 × 5-16 1 × 3-16	1 × 3	1 × 3-32 1 × 2-32	1 × 2	—	—	1 × 128	1 × 5
CNN4	1 × 3500	1 × 5-16 1 × 3-16	1 × 3	1 × 3-32 1 × 2-32	1 × 2	1 × 2-64 1 × 1-64	1 × 2	1 × 128	1 × 5
CNN5	50 × 70	5 × 5-32 5 × 3-32	2 × 2	1 × 3-32 1 × 2-32	5 × 5	—	—	1 × 128	1 × 5
CNN6	50 × 70	5 × 5-32 5 × 3-32	2 × 2	3 × 3-64 3 × 3-64	5 × 5	—	—	1 × 128	1 × 5

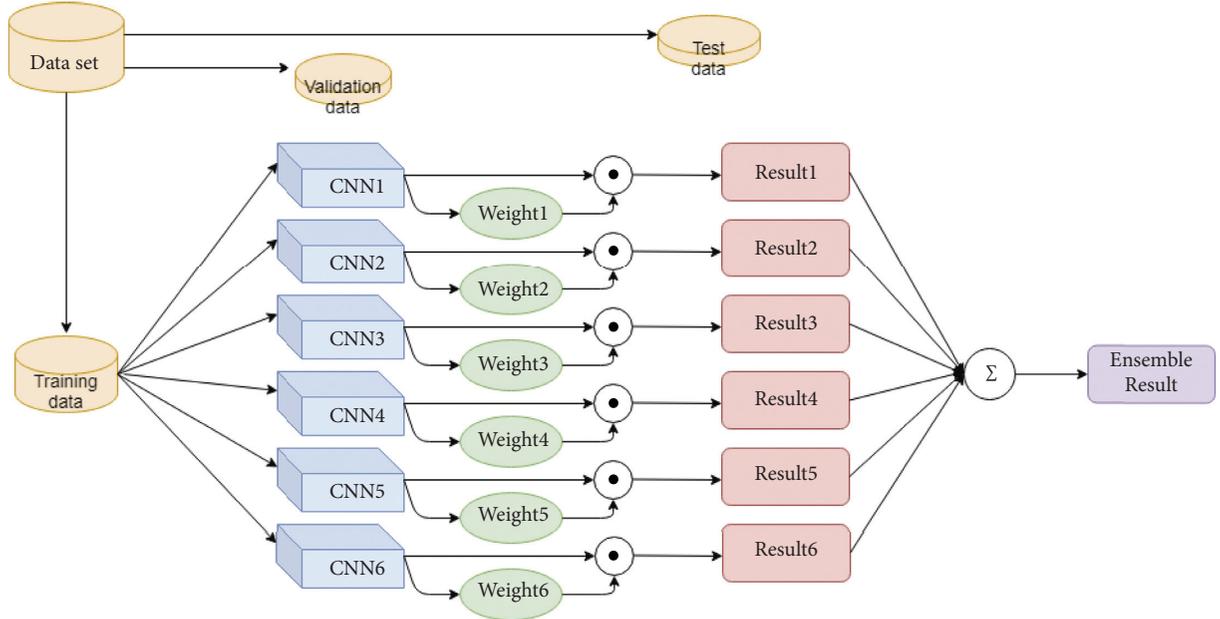


FIGURE 3: Spectral subclass and SNR distribution.

method and integration strategy, respectively. The overall structure of the ensemble model is shown in Figure 3.

For problem *a*, in order to achieve a strong generalization integration, the differences between base learners should be as large as possible, and the ability of each individual learner could not be too bad, so we apply the Bootstrap method [15]. In this experiment, 60% of all spectra were randomly selected as the overall training data, whereas 20% of all spectra were randomly selected as the validation data. For each classifier, the training data came from 50% of the overall training set that was randomly sampled. The classifiers are designed differently, as shown in Section 3.1.2.

For problem *b*, one of the most popular integration strategy is voting, which includes majority voting, plurality voting, and weighted voting. With the basic classifiers of the six CNN models, the model used the weighted voting as the integrated strategy to obtain the final classification result. Assuming that the classification result of the *i*th base classifier is  $f_i(x)$ , the classification result  $F(x)$  of the ensemble classifier can be calculated using the following equation:

$$F(x) = \sum_{i=1}^6 w_i f_i(x), \quad (8)$$

where  $W_i$  is calculated as

$$W_i = \frac{r_i}{\sum_{i=1}^6 r_i}, \quad (9)$$

where  $r$  represents the order of the accuracy of each basic classifier on the spectra from low to high.

Set the expectation and variance of single model as  $\mu$  and  $\sigma$ ; then, the mathematical expectation and the variance of the prediction of Bagging are

$$E\left(\frac{1}{n} \sum_{i=1}^n X_i\right) = \frac{1}{n} E\left(\sum_{i=1}^n X_i\right) = E(X_i) \approx \mu, \quad (10)$$

$$\text{Var}\left(\frac{1}{n} \sum_{i=1}^n X_i\right) = \frac{\sigma^2}{n} + \frac{n-1}{n} \rho\sigma, \quad (11)$$

TABLE 2: Comparison with other methods.

Methods	Train accuracy (%)	Valid accuracy (%)
SVM	88.84	88.70
Random Forest	91.20	90.51
CNN	94.20	93.88

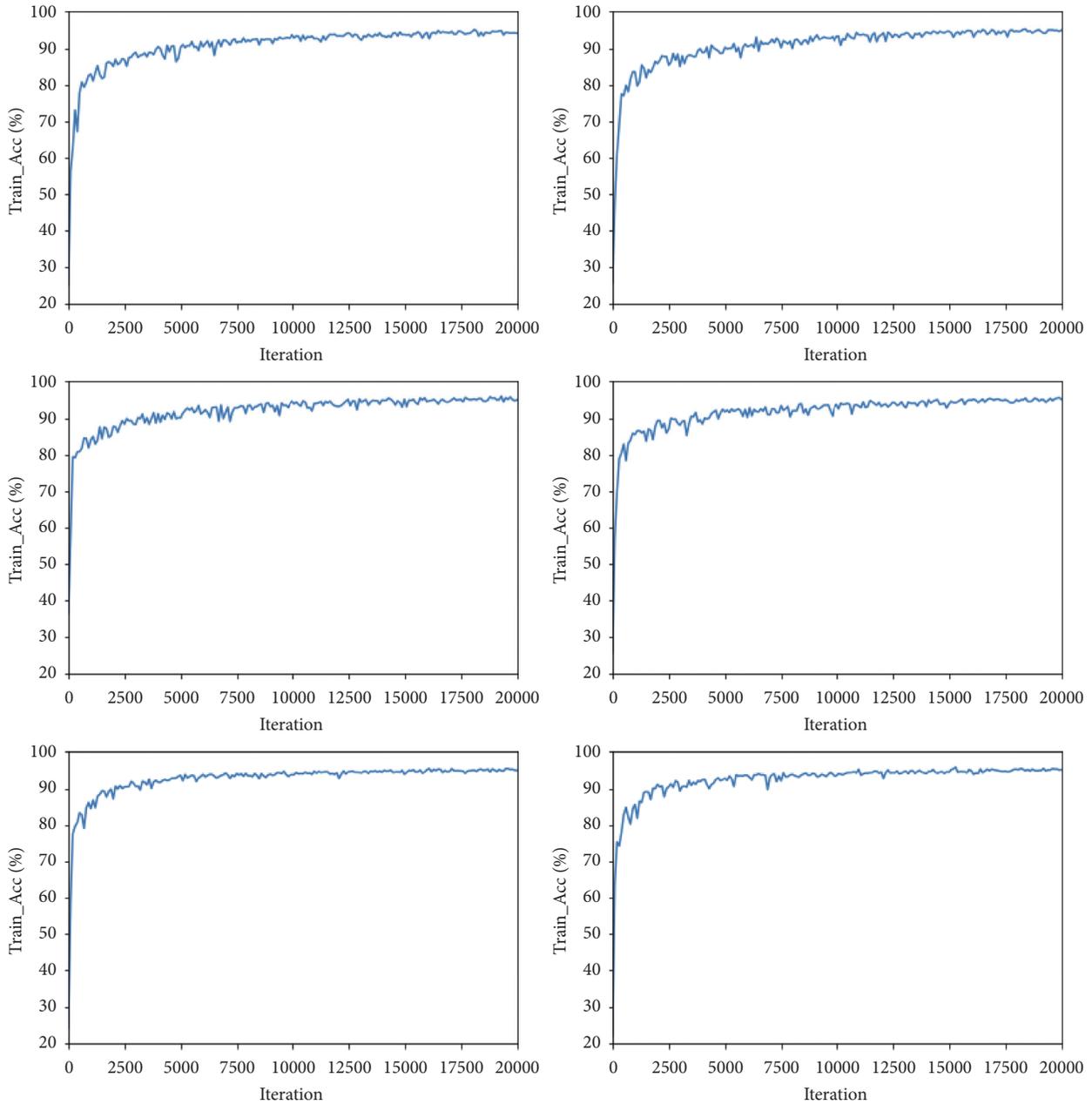


FIGURE 4: Training process of each base classifier (left to right and top to bottom CNN1-CNN6).

where  $X_i$  indicates the expectation prediction of classifier  $i$ ,  $n$  indicates the number of classifiers, and  $\rho$  indicates the correlation coefficient of single model. With  $n$  increases, equation (11) tends to  $((n - 1)/n)\rho\sigma$ ; this suggests that Bagging can reduce variance while keeping the expectation.

#### 4. Experiments and Analysis

4.1. Comparison of Methods. The performance of CNN was tested by comparing it with PCA + support vector machine (SVM) and PCA + RF, which are classic methods for spectral

TABLE 3: Accuracy and weight of the classifier.

Method	Train accuracy (%)	Weight (%)
CNN1	93.17	4.8
CNN2	93.37	9.5
CNN3	94.40	28.6
CNN4	93.93	23.8
CNN5	93.53	14.3
CNN6	93.73	19.0

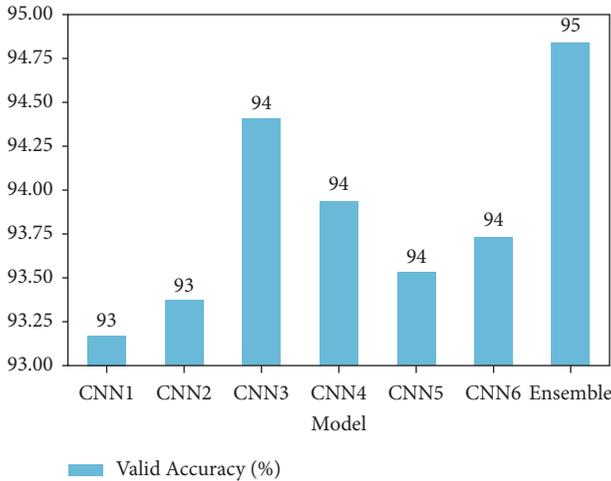


FIGURE 5: Effort of each classifier.

classification. In both methods, PCA was first used to perform dimensionality reduction and feature extraction to 50 dimensions, and then, the parameters of SVM and RF were adjusted until the model was optimal. For the CNN model, a reasonable CNN network structure was first designed, and then, grid search was performed to find the optimal parameters for training and finally output the trained model. In the comparative experiments, CNN showed a stronger fitting ability than the other methods. Table 2 summarizes the results of the comparative experiments. The CNN model exhibited excellent performance in spectral classification tasks with low SNR.

**4.2. Training Process of Base Classifiers.** For the training of base classifiers, the optimal parameter input for each network model was explored via the grid tuning method. An early stop strategy was added during the training process to prevent overfitting and to allow the base classifier to display excellent performance and excellent generalization ability. The training process of some of the base classifiers is shown in Figure 4. During the training process, the accuracy rate of classifiers first converges rapidly and then steadily increases indicating that the training of our model is stable and reliable.

**4.3. Effects of the Ensemble Classifier.** The six basic classifiers trained here were verified using the validation data, and the weight of each basic classifier was generated according to

equation (8). Table 3 shows the accuracy in the validation data and the corresponding weight of each classifier. The depth of the network and the dimensions of the input data both had an impact on the classification results.

In Table 3, the shallow convolutional networks are found to be more suitable for spectral classification tasks. By comparison, deep networks suffer from gradient disappearance and gradient explosion owing to their own deep structure. Moreover, parameter tuning is difficult and a high training overhead is required. By contrast, shallow convolutional networks have the advantages of strong generalization ability and low training overhead. After training, the shallow networks can achieve the same performance as deep structure networks, and they can also have a better generalization performance in the evaluation set.

And since the spectral data are one-dimensional, we can obviously see that the effect of one-dimensional convolution is significantly better than that of two-position convolution.

Then, the ensemble CNN classifier was constructed according to the integrated strategy proposed herein. The accuracy of the six basic classifiers and the ensemble classifier in the test data are shown in Figure 5.

After adjustment and training, the base classifiers achieved a relatively ideal performance. As shown in Figure 5, the ensemble classifier had the highest accuracy, close to 95%, demonstrating the advantage of the integrated strategy. Given that the training data of each classifier had a certain variance, and the integration of these classifiers with a certain degree of independence thus reduced the variance and kept the mathematical expectation, thereby improving the performance of the integrated model. However, the cost of this method to improve the model's performance method was a high computational overhead.

## 5. Discussion and Conclusions

This work focused on the extraction of the convolution features of stellar spectra through CNN and evaluated the validity of its analysis in stellar spectrum classification. To improve sampling robustness during the training process for the spectra, this work proposed an ensemble CNN learning model to replace the traditional model. Experimental results proved that the 1D CNN could extract more effective spectral features to achieve stellar type classification compared with the traditional methods. And the proposed ensemble model could significantly improved classifier performance by integrating multiple base classifiers which proved the effectiveness of ensemble learning in astronomical spectral classification. The method described herein not only is applicable to SDSS but also is to classify other specific spectra, such as quasi-stellar objects and galaxies.

## 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.

## Acknowledgments

This work was supported by the Shandong Provincial Natural Science Foundation, China (no. ZR2020MA064). Funding for the Sloan Digital Sky Survey IV has been provided by the Alfred P. Sloan Foundation, the U.S. Department of Energy Office of Science, and the participating institutions. SDSS acknowledges support and resources from the Center for High-Performance Computing at the University of Utah. The authors acknowledge the use of spectra from LAMOST and SDSS.

## References

- [1] K. Abazajian, J. K. Adelman-McCarthy, M. A. Agueros et al., "The first data release of the sloan digital sky survey," *The Astronomical Journal*, vol. 126, p. 2081, 2003.
- [2] C. Stoughton, R. H. Lupton, M. Bernardi et al., "Sloan digital sky survey: early data release," *The Astronomical Journal*, vol. 123, p. 485, 2002.
- [3] B. Yanny, C. Rockosi, H. J. Newberg et al., "SEGUE: a spectroscopic survey of 240,000 stars with  $g = 14-20$ ," *The Astronomical Journal*, vol. 137, pp. 4377–4399, 2009.
- [4] 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," *The Astronomical Journal*, vol. 154, p. 28, 2017.
- [5] R. Ahumada, C. A. Prieto, A. Almeida et al., "The 16th data release of the sloan digital sky surveys: first release from the APOGEE-2 southern survey and full release of eBOSS spectra," *The Astrophysical Journal-Supplement Series*, vol. 249, no. 3, 2020.
- [6] 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, p. 1197, 2012.
- [7] 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.
- [8] T. Gabruseva, S. Zlobin, and P. Wang, "Photometric light curves classification with machine learning," *Journal of Astronomical Instrumentation*, vol. 9, no. 1, Article ID 2050005, 2020.
- [9] X.-R. Li, Y.-T. Lin, and K.-B. Qiu, "Stellar spectral classification and feature evaluation based on a random forest," *Research in Astronomy and Astrophysics*, vol. 19, no. 8, p. 111, 2019.
- [10] M. Vioque, R. D. Oudmaijer, M. Schreiner et al., "Catalogue of new Herbig Ae/Be and classical Be stars," *Astronomy & Astrophysics*, vol. 638, p. A21, 2020.
- [11] M. Hon, D. Stello, and J. Yu, "Deep learning classification in asteroseismology using an improved neural network: results on 15 000 Kepler red giants and applications to K2 and TESS data," *Monthly Notices of the Royal Astronomical Society*, vol. 476, no. 3, pp. 3233–3244, 2018.
- [12] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Intelligent Signal Processing*, pp. 306–351, Wiley, Hoboken, NJ, USA, 2001.
- [14] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: concepts and methodology," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 708–713, 1979.
- [15] B. Efron, "Bootstrap methods: another look at the jackknife," *Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979.
- [16] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [17] X. Dong, Z. Yu, and W. Cao, "A survey on ensemble learning," *WIREs Data Mining Knowledge Discovery*, vol. 14, pp. 241–258, 2020.
- [18] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
- [19] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.
- [20] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [21] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the International Conference on Machine Learning*, vol. 96, pp. 148–156, Bari, Italy, July 1996.
- [22] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, 2001.
- [23] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, San Francisco, CA, USA, August 2016.
- [24] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.
- [25] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [26] T. K. Ho, "Random decision forests (PDF)," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278–282, Montreal, QC, Canada, August 1995.
- [27] D. A. Fischer and G. W. Marcy, "Multiplicity among M Dwarfs," *The Astrophysical Journal*, vol. 396, p. 178, 1992.
- [28] M. S. Bessell, "The late-M dwarfs," *The Astronomical Journal*, vol. 101, p. 662, 1991.
- [29] T. Lejeune, F. Cuisinier, and R. Buser, "A standard stellar library for evolutionary synthesis," *Astronomy & Astrophysics Supplement Series*, vol. 130, no. 1, pp. 65–75, 1998.
- [30] I. Hadji and R. P. Wildes, "What do we understand about convolutional networks?," 2018, <https://arxiv.org/abs/1803.08834>.