Neural Processing Letters (2022) 54:677–690
https://doi.org/10.1007/s11063-021-10652-1

One-Dimensional Deep Convolutional Neural Network for Mineral Classification from Raman Spectroscopy

Xiancheng Sang[1] · Ri-gui Zhou[1] · Yaochong Li[1] · Shengjun Xiong[2]

Accepted: 30 September 2021 / Published online: 11 October 2021

Abstract

Raman spectroscopy is often used for the composition determination and rapid classification of materials because it can reflect the molecular information of materials. Its accuracy mainly depends on the performance of the classification algorithm. In addition to classic machine learning classifiers such as support vector machines and k-nearest neighbor, 1D convolutional neural networks (CNNs) have also been applied to recognize Raman spectra. However, most of the research on 1D CNNs is still in the shallow, simple structure of the network model, and its application scope is only for the classification of a few classes. Therefore, this paper proposes a spectral data classification model based on a 1D deep CNN for classifying and recognizing hundreds of classes. We used the RRUFF Raman spectrum database containing many mineral samples to construct two sub-datasets, which were used to evaluate the classification effect of the model from different perspectives without data enhancement. The experimental results show that the model has a high recognition accuracy for data sets with hundreds of classes and sufficient samples. Compared with the classic machine learning classification algorithm and the 1D CNN models proposed by other scholars, our model has higher recognition accuracy and better performance and has better applicability to datasets with thousands of classes and imbalanced class distribution.

Keywords Raman spectrum · Convolutional neural network · Deep learning · Mineral recognition

✉ Ri-gui Zhou
rgzhou@shmtu.edu.cn

Xiancheng Sang
sangxiancheng@163.com

Yaochong Li
liyaochong123@gmail.com

Shengjun Xiong
xiongshengjun@htnova.com

[1] School of Information Engineering, Shanghai Maritime University, Shanghai 201306, China

[2] HT-NOVA Co., Ltd., Beijing 101312, China

1 Introduction

Raman spectroscopy is generated by the scattering of light sources by material molecules and is often used to analyze the structural information of materials [8]. Different kinds of substances must have characteristic differences in their Raman spectra due to their different substance molecules. According to this principle, Raman spectroscopy can be used for material analysis and object recognition. Raman spectroscopy detection method has the advantages of fast analysis speed, high sensitivity, minor sample damage, wide detection range and wide application range. At present, Raman spectroscopy has been applied to many technical fields such as the petrochemical industry [4,10], environmental protection [18], geological archaeology [19], gemstone identification [2], food science [5], and medical drugs [15,22].

With the development of machine learning, many classic machine learning methods have been applied to the classification of Raman spectroscopy and achieved good results. It is a relatively common method to use Principal Component Analysis (PCA) for dimensionality reduction of Raman spectra and then use Support Vector Machine (SVM) for classification. Feng et al. used this method to detect nasopharyngeal carcinoma based on the surface growth Raman spectrum of plasma [7]. Random forest is a classifier that contains multiple decision trees. Khan et al. used the random forest to evaluate Raman spectroscopy to analyze dengue fever in infected human serum [11]. The K-nearest neighbor algorithm is a theoretically mature machine learning algorithm. Its core idea is that if a sample belongs to a specific category among the K nearest samples in the feature space, it also belongs to this category. Allen et al. used the KNN model to identify the Raman spectra of plastics and correctly distinguished six types of plastics [1]. Carey et al. used the weighted nearest neighbor algorithm to achieve 84.4% classification accuracy in the sub-data set of the RRUFF Raman spectrum database [3].

Convolution operations can extract local feature information well, and deep convolutional neural networks (CNNs) have been applied to image classification and achieved good results. As the first typical CNN network structure, LeNet-5 can recognize handwritten characters very efficiently [14]. AlexNet [12] is a model that won the championship in the 2012 ImageNet competition. Its network structure is similar to LeNet-5 on the whole. However, it is more complex, and its performance is quite excellent compared to traditional machine learning classification algorithms, so it has attracted many scholars to further study the CNN model. After this, more, more profound, and better network models such as VGG [20], GoogleNet [21], and ResNet [9] were proposed. They are all deep CNNs constructed by repeatedly stacking small two-dimensional (2D) convolution kernels and pooling layers.

In recent years, due to the rapid development in the field of computer vision, many scholars have been attracted to the research of 2D image recognition. Therefore, a lot of 2D CNNs with excellent performance have been proposed, and good results have been achieved in the field of image recognition. Although 2D CNNs have made great achievements in the field of image recognition, these 2D CNNs models cannot be directly applied to the recognition of 1D spectral data. Although 1D CNNs have also been applied to identifying spectral data, such as microbial identification [17] and prostate cancer detection [15], etc., there are obvious shortcomings in research and application. The first is that the number of categories is small, and there are only two or a few categories. If the number of categories increases to hundreds or thousands, whether the model can still achieve better results is a question worth exploring. The second is that the convolutional network has a shallower level and a simple structure. The deeper and more complex 2D CNN model has achieved great success in image classification problems, which shows that on the premise of avoiding over-fitting

Table 1 Summary of Raman datasets used in classification studies

Problems	#Classes	#Spectra	Remarks
RRUFF	1704	8950	“excellent_oriented" and “excellent_unoriented"
Dataset_1 (ours)	192	5292	No less than 10 samples in each class
Dataset_2 (ours)	1332	8578	No less than 2 samples in each class

and improving the generalization ability of the model, deeper network can better extract the features of the target to obtain better classification results. In addition, the information of 1D spectral data is mainly contained in the local peaks or troughs. It just so happens that multiple convolution kernels in the convolutional layer of 1D CNNs can extract the local features of 1D spectral data from different angles, so that according to the difference of spectral data The local features identify the category of the spectrum. This is an advantage that other classic spectral recognition methods do not have, and it also determines that a suitable 1D CNN model can achieve better results in the recognition of 1D spectral data.Therefore, deeper and more complex 1D CNN models are also worthy of study.

Based on the network structure of the 2D CNN model VGG19 [20] successfully applied to image classification, this paper constructs a deep CNN model suitable for 1D spectral data. Great classification results have been achieved on the RRUFF mineral Raman spectrum database. Compared with the classic machine learning classification methods PCA-SVM, random forest, k nearest neighbors, etc. and the 1D CNN models proposed by other scholars, the accuracy of our constructed model has been significantly improved, and it is suitable for identifying one-dimensional spectral data with hundreds of classes and imbalanced class distribution.

2 Experimental Setup

2.1 Dataset

The data set used in this article is the RRUFF [13], which contains Raman spectra of thousands of minerals. Since most of the other data is unprocessed and the quality of the spectrum is problematic, We only used the processed spectra in the two subsets “excellent_oriented” and “excellent_unoriented” as the experimental dataset. Because there are few spectral samples in some classes, the model cannot be trained well, so we constructed two data subsets. Dataset_1 only retains the spectral data of the same category with no less than ten samples, so the model has sufficient samples to train and test the trained model. Dataset_2 only retains the spectral data of the same category with no less than two samples. There are many classes with a small number of samples in Dataset_2. For example, some classes have only two samples, so when dividing the training set and the test set, there is only one sample for training and one sample for testing. Table 1 shows the number of mineral classes and the total number of Raman spectra in datasets.

Figure 1 shows the distribution of the number of samples for each mineral in the dataset. It can be seen from Fig. 1 that most classes of samples have a small sample size, but we did not choose to perform data enhancement processing for these categories. There are two main reasons. The first is that the enhanced spectrum is uncertain. Spectral data is not like image data, which can still identify the original object after flipping or rotating the picture. After shifting the spectrum or performing other operations, the spectrum may belong to other

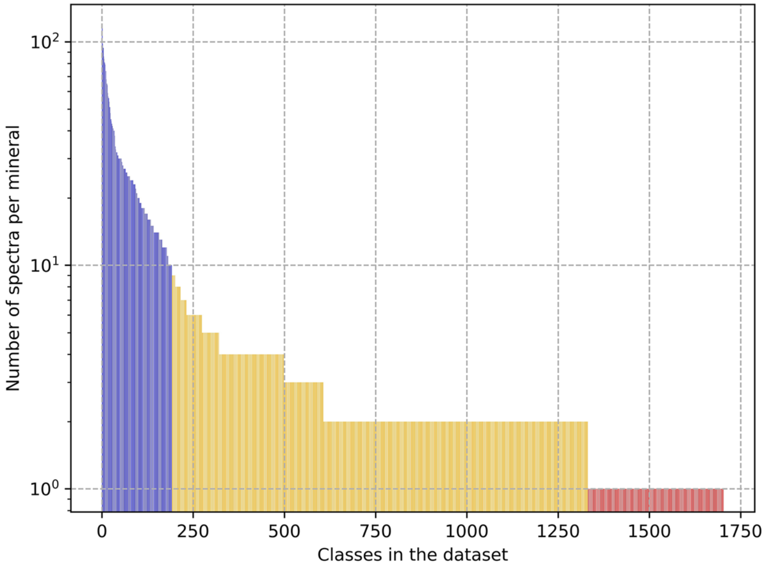


Fig. 1 The histogram shows the distribution of the number of samples for each mineral in the dataset. The blue histograms are the classes where the number of samples is not less than 10, the orange histograms are the classes where the number of samples is less than 10 and not less than 2, and the red histograms are the classes where the number of samples is equal to 1. Blue corresponds to Dataset_1. Blue and orange correspond to Dataset_2. (Color figure online)

classes. The second is the inability to select the parameters used to enhance the data. For example, use the random shift method for data enhancement on a spectrum. If the translation distance is small, the correlation between the generated spectrum and the original spectrum will be high, and the class will quickly converge, and the accuracy will be high during model training. This is similar to using multiple original spectra to train and evaluate the model. Although the accuracy rate is high, it is prone to over-fitting and causes poor model generalization ability. Similarly, if the translation distance is significant, the model will be challenging to converge, reducing accuracy. This is also the main reason why we constructed two data sets. From the data in Table 1, it can be seen that the classes and spectral data of the two datasets after screening are still affluent.

2.2 Data Preprocessing

Due to the different conditions for collecting spectral data, the Raman shift, frequency, and the number of spectral data points are not uniform. Therefore, the spectral data needs to be processed. First, we normalize each spectrum individually. Separate normalization can reduce the overall impact of the Raman intensity. It can also avoid that the value of a certain point is too large during the normalization, and the normalized value of other values is too small. Then use downsampling to reacquire the spectral data. From 30 to 1599 cm^{-1}, sampling every 1 cm^{-1}, a total of 1570 spectral points. The missing points are filled with 0. The reason for performing normalization first and then filling sampling is to avoid the influence of filling value 0 on normalization. It is worth mentioning that the spectra in the data set we selected are processed data, so there is no denoising and baseline correction of the spectra. As shown in Fig. 2, the primary information of the Raman spectrum is not lost after data preprocessing.

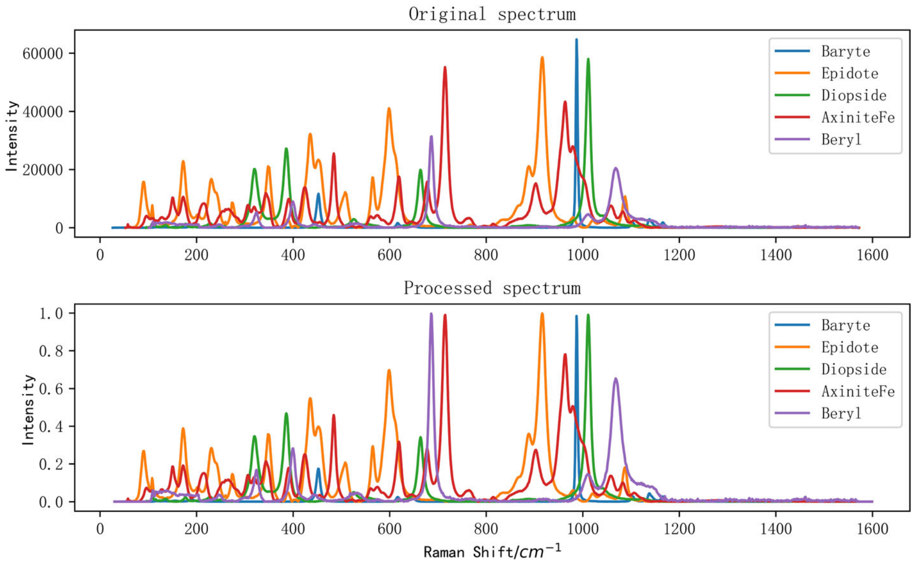


Fig. 2 Five spectra were randomly selected as examples, and the primary information is not lost after data preprocessing

Before model training, the real labels are encoded as K-dimensional vectors (K is the number of categories). Then the data set is randomly shuffled and divided into a training set and test set according to the ratio of 7:3. In order to ensure that each category has samples in the training set and test set, data is split in a stratified fashion according to category labels. In order to understand the situation of model training, we set 20% of the data in the training set as the validation set to evaluate the fit of the model being trained. In general, training set: validation set: test set = 0.56:0.14:0.30. This ratio is selected based on the size of the data set. For example, for Dataset_1, after dividing the data set using this ratio, the training set, validation set, and test set contain 2964, 740, and 1588 spectral data, respectively. Therefore, it can not only ensure that the training set has sufficient data to train the model, the validation set correctly evaluates the training situation of the model during the training process, but also the reliability of the results of the test set.

3 Proposed Methodology

3.1 1D Deep CNN Model

Raman spectrum is one-dimensional data. We refer to the network structure of VGG and adjust some parameters according to the experimental situation to construct a 1D deep CNN model. The graphic description of the network is shown in Fig. 3. The input layer of the neural network is a vector with a size of 1570. The hidden layer consists of 5 convolutional blocks, 1 flatten layer and two dense blocks. Each convolutional block contains 2–4 1D convolutional layers. The number of convolution kernels in the same block is the same, and each block contains the number of convolution kernels is 64, 128, 256, 512, 512, respectively. After each convolution block, a 1D Maxpooling layer with a step size of 2 and a filter core of 2 is connected to reduce the spectral dimension and realize feature extraction. Each dense

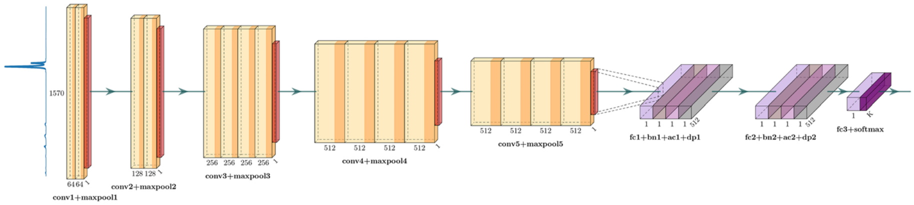


Fig. 3 Diagram of the proposed 1D CNN for spectrum recognition. It consists of 5 convolution blocks for feature extraction and two fully connected blocks for classification

Table 2 Detailed information about the parameters of the 1D CNN model

Layers	Type	Numbers	Output shape	Parameters
Input	Input	1	(Batch, 1570, 1)	–
Block 1	Conv1D	2	(Batch, 1566, 64)	(64, 3, 1)
	MaxPooling1D	1	(Batch, 783, 64)	(2, 2)
Block 2	Conv1D	2	(Batch, 779, 128)	(128, 3, 1)
	MaxPooling1D	1	(Batch, 389, 128)	(2, 2)
Block 3	Conv1D	4	(Batch, 381, 256)	(256, 3, 1)
	MaxPooling1D	1	(Batch, 190, 256)	(2, 2)
Block 4	Conv1D	4	(Batch, 182, 512)	(512, 3, 1)
	MaxPooling1D	1	(Batch, 91, 512)	(2, 2)
Block 5	Conv1D	4	(Batch, 83, 512)	(512, 3, 1)
	MaxPooling1D	1	(Batch, 41, 512)	(2, 2)
Flatten layer	Flatten	1	(Batch, 20992)	–
Dense block	Dense	2	(Batch, 512)	(512)
	BatchNormalization			–
	Activation			–
	Dropout			(0.5)
Output	Dense	1	(Batch, #classes)	(#classes)

The parameters of the Conv1D represent (filters, kernel_size, strides). The parameters of the Conv1D represent (filters, kernel_size, strides), the MaxPooling1D represent (pool_size, strides), the Dense represent (units), and the Dropout represent (rate). Except that the activation function of the output layer is "softmax", the activation functions of all Conv1D and Activation are "relu"

block consists of a dense layer, a batch normalization, an activation layer, and a dropout layer. The dense layer contains 512 neurons. The activation function of the activation layer is Relu. The dropout layer randomly sets the input unit to 0 at a ratio of 0.5 to prevent the model from overfitting. The final output layer uses the softmax activation function to output a K-dimensional vector, where K is the number of categories. Table 2 shows the detailed parameters of the CNN. The CNN is constructed by TensorFlow (version: 2.3.1) and runs under the Python (version: 3.8) environment. In addition, we used the NVIDIA 2080Ti graphics card to speed up model training.

3.2 Model Training

The training and optimization of the deep convolutional neural network depend on the optimizer, which can guide the various parameters of the loss function in the correct direction

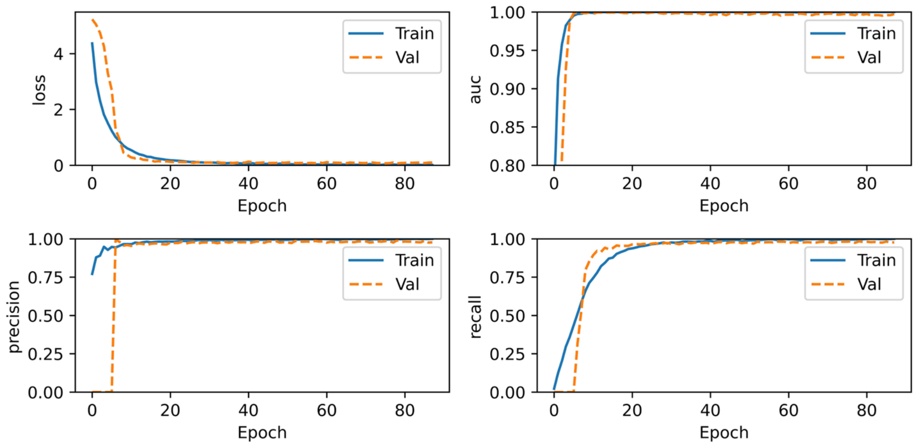


Fig. 4 The loss, AUC, precision, and recall curves during model training. The blue curve is the value of the training set, and the orange curve is the value of the validation set. (Color figure online)

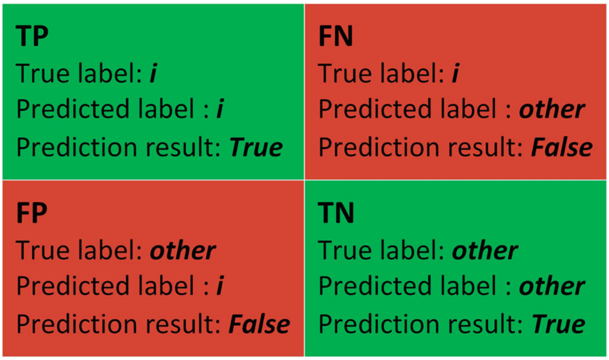
to update the appropriate size during the model backpropagation process, and finally adjust the weight of the model to the optimal value. The optimizer in this article uses the Adam optimization algorithm. Adam is a gradient-based optimization algorithm. The method is simple to implement, computationally efficient, and has a small memory footprint. It is suitable for non-stationary objective functions. The hyperparameters have intuitive explanations and do not require complex parameter adjustment procedures. The parameters of the Adam optimizer are set to learning_rate = 1e−4, beta_1 = 0.9, beta_2 = 0.999.

In the model training process, another critical parameter is the loss function, which is used to evaluate the degree to which the model's predicted value is different from the true value. The classification of mineral Raman spectra studied in this paper is a multi-classification problem and the true label has been coded by one-hot, hence the categorical_crossentropy is employed as the loss function in this paper.

In order to prevent the model from overfitting and increase the generalization ability of the model, a extra Dropout layer is added to the fully connected block and the early stopping strategy is also adopted. We first set Epoch to 200 and then monitor the accuracy of the validation set. When the accuracy of the validation set does not improve within 30 Epochs, the training is terminated early, and the model weight is restored from the period with the best value of the number of monitoring. Figure 4 shows the loss, AUC, precision, and recall curves of the model's training process using Dataset_1. In Fig. 4, as the number of training increases, the four curves all rise (or fall) first and eventually tend to be stable. And the actual number of training iterations of the model is less than 100, so the model does not have the problem of underfitting.

3.3 Model Evaluation

After the model is trained with the training set, use the model to predict the data in the test set. The most intuitive indicator in model evaluation is the accuracy rate, that is, the proportion of the correct data in the test set to the total data. Assuming that the number of samples in



TP True label: ***i*** Predicted label : ***i*** Prediction result: ***True***	**FN** True label: ***i*** Predicted label : ***other*** Prediction result: ***False***
FP True label: ***other*** Predicted label : ***i*** Prediction result: ***False***	**TN** True label: ***other*** Predicted label : ***other*** Prediction result: ***True***

Fig. 5 In the multi-class problem, the confusion matrix of class *i*

the test set is N_{total}, the number of predicted labels that are the same as the true labels is recorded as TP_{total}, and then the accuracy can be defined as:

$$Accuracy = \frac{TP_{total}}{N_{total}} \tag{1}$$

In addition, precision, recall, and F1 score are also often used to evaluate the quality of a model. There are three calculation methods for the precision rate, recall rate, and F1-score in the multi-classification problem. The micro-average method: calculates the index globally by calculating the total true positives, false negatives, and false positives. Macro-average method: Calculate the indicators of each label and calculate their arithmetic average. Weighted-average method: Calculate the index of each label, and calculate the weighted average according to the sample weight.

In Dataset_1 and Dataset_2, the number of samples in each class is imbalanced. The data in the training set and the test set are randomly divided according to the ratio of 7:3. Therefore, the number of samples in the test set is also imbalanced. It is more appropriate to use the weighted average method. The precision, recall, and F1-score mentioned in the result part of this article are all weighted calculated values. In order of brevity, we have omitted the weighted flag.

Assuming class *i* in the test set, the number of samples is $\mathrm{N_i}$, then its weight is $\mathrm{W_i}$,

$$W_i = \frac{N_i}{N_{total}} \tag{2}$$

Use P (Position) and N (Negative) to represent positive samples and negative samples, and use T (True) and F (False) to represent correct and incorrect predictions, so that we can get four evaluation indicators of TP, FN, FP, and TN:

TP: predicting positive samples as positive samples;
FN: predicting positive samples as negative samples;
FP: predicting negative samples as positive samples;
TN: predicting negative samples as negative samples.

The confusion matrix of class *i* can be obtained as shown in the Fig. 5:

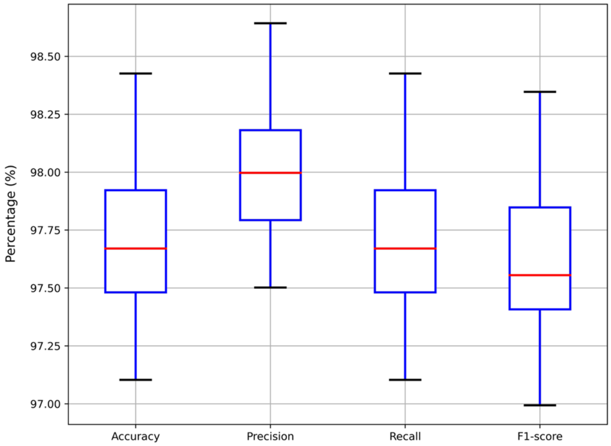


Fig. 6 This box plot presents the evaluation results of the test set on the model through four evaluation indicators: accuracy, precision, recall, and F1-score

For the entire test set, the formulas for calculating Weighted-Precision, Weighted-Recall, and Wighted-F1-score are:

$$Precision = \sum_{i=1}^{n} W_i * \frac{TP_i}{TP_i + FP_i} \tag{3}$$

$$Recall = \sum_{i=1}^{n} W_i * \frac{TP_i}{TP_i + FN_i} \tag{4}$$

$$F1 = \sum_{i=1}^{n} W_i * \frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i} = \sum_{i=1}^{n} W_i * \frac{2 * TP_i}{2 * TP_i + FN_i + FP_i} \tag{5}$$

4 Results and Discussion

4.1 1D CNN Classification Results

In order to reduce the randomness of the results and increase the reliability of the results, we controlled the model's parameters unchanged and performed 100 repeated training experiments on the model by randomly changing the Dataset_1 division. The result not only records the accuracy rate of the test set but also records the weighted precision, recall, and F1-score. The experimental results are shown in Fig. 6.

It can be seen from Fig. 6 that the average accuracy of the test set is as high as 97.72%, which shows that the 1D deep CNN can classify mineral Raman spectra very well. The precision of 97.99% indicates that the model has high accuracy in predicting positive samples. The 97.72% recall rate shows that the model is correctly identified with a high percentage of positive samples. The F1-score of 97.63% shows that the model achieves a good balance between accuracy and recall. These results show that the 1D deep CNN model has high accuracy and reliability.

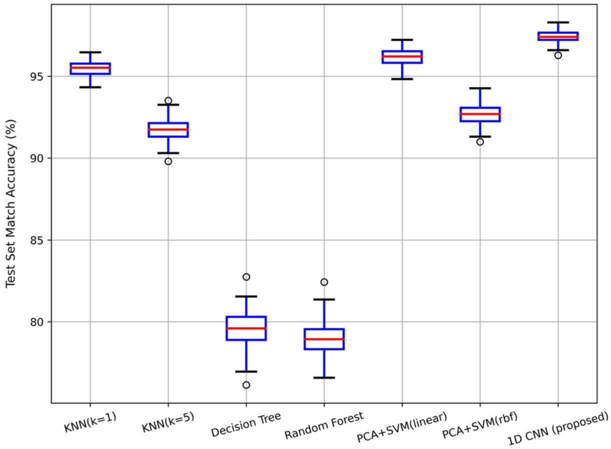


Fig. 7 The accuracy of our proposed model is compared with the classic machine learning classifier on the test set in Dataset_1

4.2 Comparison with Other Classification Methods

In order to verify the Raman spectrum classification performance of our proposed model, this paper also uses five classic machine learning algorithms (KNN (k = 1 and k = 5), Decision Tree, Random Forest, PCA + SVM (linear), PCA + SVM (rbf)) to conduct experiments on Dataset_1, which are compared with the 1D deep CNN model proposed in this paper. In the above five classification algorithms, the control model parameters are unchanged, and the Dataset_1 is changed randomly. One hundred repeated experiments have been carried out, and the Grid Search algorithm has been used for parameter debugging and optimization of each classifier. The experimental results are shown in Fig. 7.

The box plot in Fig. 7 shows the classification results of a variety of classic machine learning classifiers and our proposed 1D deep CNN model after 100 random experiments. The 1D deep CNN model we proposed has the best classification effect. The max accuracy of the test set is 98.43%, and the average is 97.72%. Among the classic machine learning classifiers, using principal component analysis to extract features and then using linear support vector machines for classification has the highest accuracy. In the examples in this article, the linear kernel has a better classification effect than the Gaussian kernel. The main reason is that when the number of examples in the training set is large, the Gaussian kernel is difficult to normalize, and the linear kernel can also perform normalization well in this case.

In order to compare the 1D deep CNN model we proposed and the CNN model proposed by other scholars. We rebuilt the models proposed by Liu et al. [16], Fan et al. [6], and Zhang et al. [23] according to the model structures in papers, and conducted experiments on Dataset_1 and Dataset_2. The experimental results are shown in Table 3. In order to reduce the randomness of the results, the data in Table 3 is the average of the results of 100 repeated random experiments. As can be seen from the data in Table 3, the model we proposed is better in all aspects.

The distribution of class data in Dataset_2 is very imbalanced. Most classes contain only a small number of samples. In order to evaluate the model's ability to recognize small sample classes, we use the area value under the PRC (Precision–Recall Curve) to measure the model's classification effect on unbalanced data. The larger the PRC value, the better

Table 3 The results of comparative experiments between the 1D deep CNN model we proposed and the models proposed by other scholars on Dataset_1 and Dataset_2

	Precision	Recall	F1-score	AUC	PRC	Top_1 acc.	Top_3 acc.	Top_5 acc.
Dataset_1								
Liu et al.	0.9660	0.9601	0.9587	0.9967	0.9846	0.9601	0.9927	0.9944
Fan et al.	0.9718	0.9679	0.9667	0.9948	0.9804	0.9679	0.9938	0.9955
Zhang et al.	0.9653	0.9583	0.9563	0.9961	0.9830	0.9583	0.9940	0.9962
Ours	0.9799	0.9772	0.9763	0.9979	0.9909	0.9772	0.9976	0.9984
Dataset_2								
Liu et al.	0.8846	0.8309	0.7950	0.9622	0.8648	0.8309	0.9096	0.9199
Fan et al.	0.8893	0.8368	0.8070	0.9541	0.8602	0.8368	0.8981	0.9094
Zhang et al.	0.8990	0.8466	0.8192	0.9615	0.8754	0.8466	0.9140	0.9233
Ours	0.9127	0.8791	0.8511	0.9691	0.9073	0.8791	0.9279	0.9339

The precision, recall, and F1-score are all weighted calculated values. The AUC represents the area under the ROC curve. The PRC represents the area under the precision–recall curve the model's classification effect on unbalanced data sets. It can be seen from Table 3 that although the AUC value of each model for Dataset_2 classification is very close, the PRC value is quite different.

4.3 Analysis of CNN Classification Results

The experimental results show that the 1D deep CNN model can well realize the classification of mineral Raman spectra. The reason is that the convolution operation can effectively extract the local features of the spectrum, and the information of the Raman spectrum mainly exists in the local peaks. The convolutional layer in CNN implicitly learns from the training data during the training process. A convolutional layer often contains many kernels, and each convolution kernel can extract specific features, and different convolution kernels can extract different features. The maximum pooling layer behind the convolutional layer is to select representative features from the features extracted by the convolutional layer. Therefore, even the small differences between different spectra can obtain different feature information after a series of convolution and pooling operations. This is the ability that traditional Raman spectral feature extraction methods such as feature peak selection and principal component analysis do not have. The final fully connected layer identifies the spectrum based on the spectral features extracted by the convolutional layer. In this way, different spectra can be identified according to the subtle differences of the spectra.

CNN requires a large amount of data to train the model. The purpose is to make the parameters in the model better extract features and distinguish the samples of each category. It can be seen from Table 3 that the recognition accuracy of Dataset_2 is lower than that of Dataset_1. One of the reasons is that there are more classes and sample sizes in Dataset_2. However, the main reason is that the sample size in some classes in Dataset_2 is too small, the model cannot learn the characteristic information of the spectrum well. The establishment of a complete spectral database is of great significance to the study of spectral classification.

The reason for setting up the comparative experiment of Dataset_1 and Dataset_2 is to verify the scalability of the model application. The proposed model can not only have high recognition accuracy for one data set. When a new class is added to the data set, the model should still identify the samples in the data set better. The experimental results show that when the number of classes increases from 192 to 1332 and the number of samples increases from 5292 to 8578, the accuracy of Top1 can still reach 87.91%, and the accuracy of Top5 can reach 93.39%.

Dataset_2 is an imbalanced data set. It contains many small sample classes. The data set can be used to measure the model's ability to classify unbalanced data sets, as well as the model's ability to recognize small sample categories. Compared with the models proposed by other scholars, the PRC value of our model on Dataset_2 has increased by about 3%.

5 Conclusion and Future Work

In this article, we proposed a 1D deep CNN model to realize the identification of mineral Raman spectra in the RRUFF dataset. Compared with the classic machine learning methods and the other 1D CNN model proposed by other scholars, the model has better performance in terms of accuracy, precision, and recall. In addition, the model still has a high accuracy rate in the case of thousands of classes and imbalanced sample distribution. Although our experiment used Raman spectral data, the 1D deep CNN model can also be applied to the classification and recognition of other 1D spectral data. In the future, while we expect a 1D CNN model with higher accuracy and better performance to be proposed, we also hope that the outstanding classification algorithm can be combined with portable Raman equipment to be applied to the real-time detection of objects.

Acknowledgements This work was supported by the National Key R&D Plan under Grant No. 2018YFC1200203, the National Natural Science Foundation of China under Grant No. 6217070290 and Shanghai Science and Technology Project in 2020 under Grant No. 20040501500.

Author Contributions XS: Project administration, software, validation, visualization, writing—original draft. RZ: Conceptualization, formal analysis, funding acquisition, writing—review and editing. YL: Validation, visualization, writing—review and editing. SX: Data curation, investigation, methodology.

Availability of Data and Codes Data and codes are available at Github: https://github.com/sangxiancheng/1D_CNN_For_Mineral_Classification.

Declaration

Conflict of interest There are no conflicts to declare.

References

1. Allen V, Kalivas JH, Rodriguez RG (1999) Post-consumer plastic identification using Raman spectroscopy. Appl Spectrosc 53(6):672–681. https://doi.org/10.1366/0003702991947324
2. Barone G, Bersani D, Lottici PP, Mazzoleni P, Raneri S, Longobardo U (2016) Red gemstone characterization by micro-Raman spectroscopy: the case of rubies and their imitations. J Raman Spectrosc 47(12):1534–1539. https://doi.org/10.1002/jrs.4919
3. Carey C, Boucher T, Mahadevan S, Bartholomew P, Dyar MD (2015) Machine learning tools for mineral recognition and classification from Raman spectroscopy. J Raman Spectrosc 46(10):894–903. https://doi.org/10.1002/jrs.4757
4. Chung H, Ku MS (2000) Comparison of near-infrared infrared, and Raman spectroscopy for the analysis of heavy petroleum products. Appl Spectrosc 54(2):239–245. https://doi.org/10.1366/0003702001949168
5. Fan C, Hu Z, Riley LK, Purdy GA, Mustapha A, Lin M (2010) Detecting food- and waterborne viruses by surface-enhanced Raman spectroscopy. J Food Sci. https://doi.org/10.1111/j.1750-3841.2010.01619.x
6. Fan X, Ming W, Zeng H, Zhang Z, Lu H (2019) Deep learning-based component identification for the Raman spectra of mixtures. Analyst 144(5):1789–1798. https://doi.org/10.1039/c8an02212g
7. Feng S, Chen R, Lin J, Pan J, Chen G, Li Y, Cheng M, Huang Z, Chen J, Zeng H (2010) Nasopharyngeal cancer detection based on blood plasma surface-enhanced Raman spectroscopy and multivariate analysis. Biosens Bioelectron 25:2414–9
8. Ferrari AC, Meyer JC, Scardaci V, Casiraghi C, Lazzeri M, Mauri F, Piscanec S, Jiang D, Novoselov KS, Roth S, Geim AK (2006) Raman spectrum of graphene and graphene layers. Phys Rev Lett 97:187401. https://doi.org/10.1103/PhysRevLett.97.187401
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2016.90
10. He X, Liu X, Nie B, Song D (2017) FTIR and Raman spectroscopy characterization of functional groups in various rank coals. Fuel 206:555–563. https://doi.org/10.1016/j.fuel.2017.05.101
11. Khan S, Ullah R, Khan A, Sohail A, Wahab N, Bilal M, Ahmed M (2017) Random forest-based evaluation of Raman spectroscopy for dengue fever analysis. Appl Spectrosc 71:2111–2117
12. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
13. Lafuente B, Downs RT, Yang H, Stone N (2015) 1. The power of databases: the RRUFF project. In: Highlights in mineralogical crystallography. De Gruyter (O), pp 1–30. https://doi.org/10.1515/9783110417104-003

14. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791
15. Lee W, Lenferink AT, Otto C, Offerhaus HL (2019) Classifying Raman spectra of extracellular vesicles based on convolutional neural networks for prostate cancer detection. J Raman Spectrosc 51(2):293–300. https://doi.org/10.1002/jrs.5770
16. Liu J, Osadchy M, Ashton L, Foster M, Solomon C, Gibson S (2017) Deep convolutional neural networks for Raman spectrum recognition: a unified solution. Analyst 142:4067–4074
17. Maruthamuthu M, Raffiee A, De OD, Ardekani A, Verma M (2020) Raman spectra-based deep learning: a tool to identify microbial contamination. Microbiologyopen 9:e1122
18. Piorek BD, Lee SJ, Santiago JG, Moskovits M, Banerjee S, Meinhart CD (2007) Free-surface microfluidic control of surface-enhanced Raman spectroscopy for the optimized detection of airborne molecules. Proc Natl Acad Sci 104(48):18898–18901. https://doi.org/10.1073/pnas.0708596104
19. Rantitsch G, Lämmerer W, Fisslthaler E, Mitsche S, Kaltenböck H (2016) On the discrimination of semi-graphite and graphite by Raman spectroscopy. Int J Coal Geol 159:48–56. https://doi.org/10.1016/j.coal.2016.04.001
20. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. In: Computer science. arXiv:1409.1556v6. Accessed on Tue, 06 July 2021
21. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2015.7298594
22. Toma W, Guimarães LL, Brito AR, Santos AR, Cortez FS, Pusceddu FH, Cesar A, Júnior LS, Pacheco MT, Pereira CD (2014) Safflower oil: an integrated assessment of phytochemistry antiulcerogenic activity, and rodent and environmental toxicity. Rev Bras Farmacogn 24(5):538–544. https://doi.org/10.1016/j.bjp.2014.09.004
23. Zhang R, Xie H, Cai S, Hu Y, Liu GK, Hong W, Tian ZQ (2019) Transfer-learning-based Raman spectra identification. J Raman Spectrosc 51(1):176–186. https://doi.org/10.1002/jrs.5750

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.