Analyst

PAPER

View Article Online
View Journal | View Issue



Cite this: *Analyst*, 2017, **142**, 4067

# Deep convolutional neural networks for Raman spectrum recognition: a unified solution

Jinchao Liu,†[a] Margarita Osadchy,[b] Lorna Ashton,[c] Michael Foster,†[d] Christopher J. Solomon[e] and Stuart J. Gibson *[e]

Received 17th August 2017,
Accepted 27th September 2017

DOI: 10.1039/c7an01371j

rsc.li/analyst

Machine learning methods have found many applications in Raman spectroscopy, especially for the identification of chemical species. However, almost all of these methods require non-trivial preprocessing such as baseline correction and/or PCA as an essential step. Here we describe our unified solution for the identification of chemical species in which a convolutional neural network is trained to automatically identify substances according to their Raman spectrum without the need for preprocessing. We evaluated our approach using the RRUFF spectral database, comprising mineral sample data. Superior classification performance is demonstrated compared with other frequently used machine learning algorithms including the popular support vector machine method.



## 1. Introduction

Raman spectroscopy is a ubiquitous method for characterisation of substances in a wide range of settings including industrial process control, planetary exploration, homeland security, life sciences, geological field expeditions and laboratory materials research. In all of these environments there is a requirement to identify substances from their Raman spectrum at high rates and often in high volumes. Whilst machine classification has been demonstrated to be an essential approach to achieve real time identification, it still requires preprocessing of the data. This is true regardless of whether peak detection or multivariate methods, operating on whole spectra, are used as input. A standard pipeline for a machine classification system based on Raman spectroscopy includes preprocessing in the following order: cosmic ray removal, smoothing and baseline correction. Additionally, the dimensionality of the data is often reduced using principal components analysis (PCA) prior to the classification step. To the best of our knowledge, there is no existing work describing machine classification systems that can cope directly with raw spectra such as those affected significantly by baseline distortion.

In this work we focus on multivariate methods, and introduce the application of convolutional neural networks (CNNs) in the context of Raman spectroscopy. Unlike the current Raman analysis pipelines, CNN combines preprocessing, feature extraction and classification in a single architecture which can be trained end-to-end with no manual tuning. We show that CNN not only greatly simplifies the development of a machine classification system for Raman spectroscopy, but also achieves significantly higher accuracy. In particular, we show that CNN trained on raw spectra significantly outperformed other machine learning methods such as support vector machine (SVM) with baseline corrected spectra. Our method is extremely fast with a processing rate of one sample per millisecond.‡

The baseline component of a Raman spectrum is caused primarily by fluorescence, can be more intense than the actual Raman scatter by several orders of magnitude, and adversely affects the performance of machine learning systems. Despite considerable effort in this area, baseline correction remains a challenging problem, especially for a fully automatic system.[1]

A variety of methods for automatic baseline correction have been used such as polynomial baseline modelling,[1] simulation-based methods,[2,3] penalized least squares.[4,5] Lieber *et al.*[1] proposed a modified least-squares polynomial curve fitting for fluorescence subtraction which was shown to be effective. Eilers *et al.*[6] proposed a method called *asymmetric*

[a]*VisionMetric Ltd., Canterbury, Kent, CT2 7FG, UK.*
*E-mail: liujinchao2000@gmail.com; Tel: +44 (0) 1227 811790*
[b]*Department of Computer Science, University of Haifa, Mount Carmel, Haifa 31905, Israel. E-mail: rita@cs.haifa.ac.il; Tel: +972-4-8288444*
[c]*Department of Chemistry, Lancaster University, Bailrigg, Lancaster, LA1 4YW, UK.*
*E-mail: l.ashton@lancaster.ac.uk; Tel: +44 (0)1524 593552*
[d]*IS-Instruments Ltd., 220 Vale Road, Tonbridge, Kent, TN9 1SP, UK.*
*E-mail: mfoster@is-instruments.com; Tel: +44 (0)1732 373020*
[e]*School of Physical Sciences, University of Kent, Canterbury, CT2 7NH, UK.*
*E-mail: S.J.Gibson@kent.ac.uk; Tel: +44 (0)1227 823271*
† Funded by Innovate UK, project number 132200.

‡ Software processing time only. Not including acquisition of Raman signal from spectrometer.

*least square smoothing*. In this method one first smooths a signal by a Whittaker smoother to get an initial baseline estimation, and then applies asymmetric least square fitting where positive deviations with respect to baseline estimate are weighted (much) less than negative ones. This has been shown to be a useful method, and in principle can be used for automatic baseline correction, although it may occasionally require human input. Kneen *et al.*[2] proposed a method called *rolling ball*. In this method one imagines a ball with tunable radius rolling over/under the signal. The trace of its lowest/highest point is regarded as an estimated baseline. A similar method is *rubber band*[3] where one simulates a rubber band to find the convex hull of the signal which can then be used as a baseline estimation. Zhang *et al.*[4] presented a variant of penalized least squares, called *adaptive iteratively reweighted Penalized Least Squares (airPLS)* algorithm. It iteratively adapts weights controlling the residual between the estimated baseline and the original signal. A detailed review and comparison of baseline correction methods can be found in Schulze *et al.*[7]

Classification rates have been compared for various machine learning algorithms using Raman data. The method that is frequently reported to outperform other algorithms is support vector machines (SVM).[8] An SVM is trained by searching for a hyperplane that optimally separates labelled training data with maximal margin between the training samples and the hyperplane. Binary (two class) and small scale problems in Raman spectroscopy have been previously addressed using this method. A large proportion of these related to applications in the health sciences, use a non-linear SVM with a radial basis function kernel, and an initial principal component analysis (PCA) data reduction step. In this context SVM was shown to: outperform linear discriminant analysis (LDA) and partial least squares discriminant analysis (PLS-LDA) in breast cancer diagnosis,[9] successfully sort unlabelled biological samples into one of three classes (normal, hyperplastic polyps or adeno-carcinomas)[10] and discriminate between three species of bacteria using a small number of training and test examples.[11] Although multiclass classification is possible using SVM, in practice training a non-linear SVM is infeasible for large scale problems involving thousands of classes. Random forests (RF)[12] represent a viable alternative to SVM for high dimensional data with a large number of training examples. RF is an ensemble learning method based on multiple decision trees that avoids overfitting the model to the training set. This method generated a lot of attention in the machine learning community in last decade prior the widespread popularity of CNN. However when compared with PCA-LDA and RBF SVM on Raman microspectroscopy data[13] it performed poorly. The method previously applied to spectral classification problems that is closest to our own approach is fully connected artificial neural networks (ANN). Unlike CNN, ANN is a shallow network architecture which does not have enough capacity to solve large scale problems. Maquel *et al.*[14] determined the major groupings for their data prior to a multilayered ANN analysis. Their study concluded that vibrational spectroscopic techniques are well suited to automatic classification and can therefore be used by nonexperts and at low cost.

**Table 1** Summary of Raman datasets used in classification studies

| Problems | #Classes | #Spectra | Baseline removal |
|---|---|---|---|
| Sattlecker *et al.*[9] | 2 | 1905 | N/A[a] |
| Kwiatkowski *et al.*[18] | 10 | N/A | Yes |
| Carey *et al.*[16] | 1215 | 3950 | Yes |
| Ours #1 | 1671 | 5168 | Yes |
| Ours #2 | 512 | 1676 | No |

[a] Note that in this work special filtering methods were developed to discard spectra of bad quality which account for 80% of the total amount.

A drawback associated with the methods previously used is that they require feature engineering (or preprocessing) and don't necessarily scale easily to problems involving a large number of classes. Motivated by the recent and widespread success of CNNs in large scale image classification problems we developed our network architecture for the classification of 1D spectral data. A suitable dataset to test the efficacy of the CNN is the RRUFF mineral dataset. Previous work[15,16] has focused on identifying mineral species contained in this dataset using nearest neighbour methods with different similarity metrics such as cosine similarity and correlation (also used in commercial softwares such as CrystalSleuth). Carey *et al.*[16] achieved a species classification accuracy on a subset of the RRUFF database[17] of 84.8% using a weighted neigbour (WN) classifier. Square root squashing, maximum intensity normalisation, and sigmoid transformations were applied to the data prior to classification. Accuracy was determined using cross validation with semi-randomised splits over a number of trials. The WN classifier compared favourably with the $k = 1$ nearest neighbour (82.1% accuracy) on which the CrystalSleuth matching software is believed to be based. In Table 1 we summarise the sample data used in our own work and in some previous Raman based spectral classification studies.

## 2. Materials and methods

CNNs have become the predominant tool in a number of research areas – especially in computer vision and text analysis. An extension of the artificial neural network concept,[19] CNNs are nonlinear classifiers that can identify unseen examples without the need for feature engineering. They are computational models[20] inspired by the complex arrangement of cells in the mammalian visual cortex. These cells are stimulated by small regions of the visual field, act as local filters, and encode spatially localised regions of natural signals or images.

CNNs are designed to extract features from an input signal with different levels of abstraction. A typical CNN includes convolutional layers, which learn filter maps for different types of

patterns in the input, and pooling operators which extract the most prominent structures. The combination of convolutional and pooling layers extracts features (or patterns) hierarchically. Convolutional layers share weights which allow computations to be saved and also make the classifier invariant to spatial translation. The fully connected layers (that follow the convolutional and pooling layers) and the softmax output layer can be viewed as a classifier which operates on the features (of the Raman spectra data), extracted using the convolutional and pooling layers. Since all layers are trained together, CNNs integrate feature extraction with classification. Features determined by network training are optimal in the sense of the performance of the classifier. Such end-to-end trainable systems offer a much better alternative to a pipeline in which each part is trained independently or crafted manually.

In this work, we evaluated the application of a number of prominent CNN architectures including LeNets,[21] Inception[22] and Residual Nets[23] to Raman spectral data. All three showed comparable classification results even though the latter two have been considered superior to LeNet in computer vision applications. We adopted a variant of LeNet, comprising pyramid-shaped convolutional layers for feature extraction and two fully-connected layers for classification. A graphical illustration of the network is shown in Fig. 1.

### 2.1. CNN for Raman spectral data classification

The input to the CNN for Raman spectrum classification is one dimensional and it contains the entire spectrum (intensity fully sampled at regularly spaced wavenumbers). Hence we trained one-dimensional convolutional kernels in our CNN.

For our convolutional layers, we used LeakyReLU[24] non-linearity, defined as

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{otherwise} \end{cases}. \quad (1)$$

Formally, a convolutional layer can be expressed as follows:

$$y^j = f\left(b^j + \sum_i k^{ij} * x^i\right),$$

where $x^i$ and $y^i$ are the $i$-th input map and the $j$-th output map, respectively. $k^{ij}$ is a convolutional kernel between the maps $i$ and $j$, $*$ denotes convolution, and $b^j$ is the bias parameter of the $j$-th map.

The convolutional layer is followed by a max-pooling layer, in which each neuron in the output map $y^i$ pools over an $s \times 1$ non-overlapping region in the input map $x^i$. Formally,

$$y^i_j = \max_{0 \le m < s} \{x^i_{j \cdot s+m}\}.$$

The upper layers of the CNN are fully connected and followed by the softmax with the number of outputs equal to the number of classes considered. We used tanh as non-linearity in the fully connected layers. The softmax operates as a squashing function that re-normalizes a $K$-dimensional input vector $z$ of real values to real values in the range [0,1] that sum to 1, specifically,

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

To avoid overfitting the model to the data, we applied batch normalization[25] after each layer and dropout[26] after the first

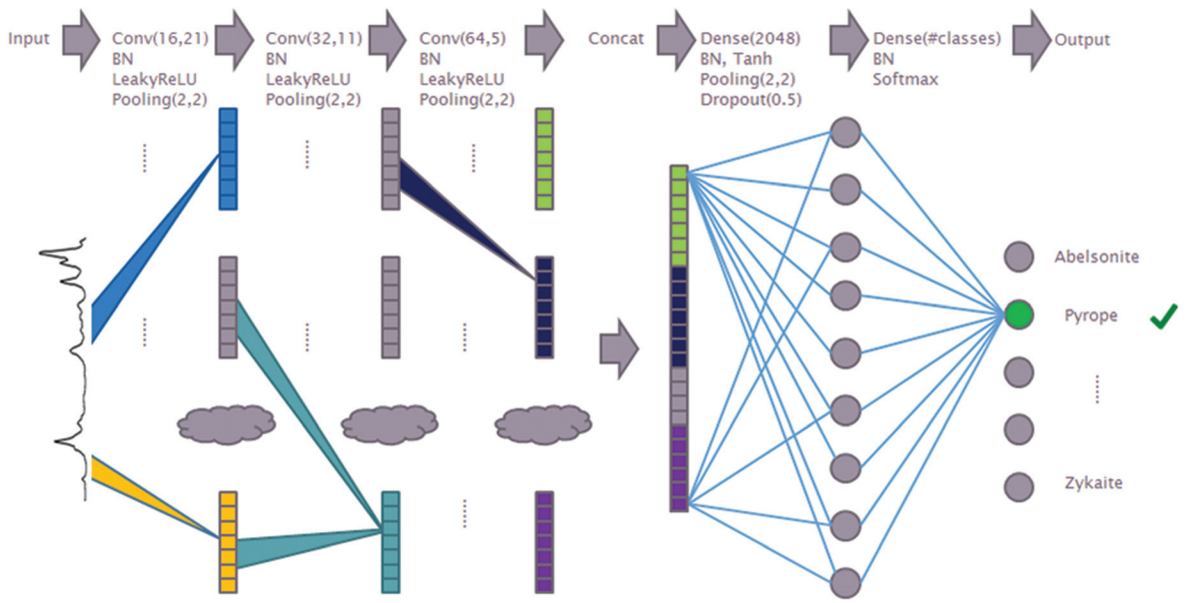



**Fig. 1** Diagram of the proposed CNN for spectrum recognition. It consists of a number of convolutional layers for feature extraction and two fully-connected layers for classification.

fully connected layer. Further details of the architecture are shown in Fig. 1.

### 2.2. CNN training

Since the classes in our experiments have very different numbers of examples, we used the following weighted loss to train the CNN:

$$\mathscr{L}(\boldsymbol{w}, x_n, y_n) = -\frac{1}{N}\sum_{n=1}^{N}\alpha_n\sum_{k=1}^{K}t_{kn}\ln y_{kn} \quad (2)$$

where $x_n$ is a training spectrum, $t_n$ is the true label of the $n^{\text{th}}$ sample in the format of one-hot encoding, $y_n$ is the network prediction for the $n^{\text{th}}$ sample, $\alpha_n \propto \frac{1}{\#C}$ and $\#C$ is the number of samples in the class $C$ that $x_n$ belongs to. $N$ is the total number of samples and $K$ is the number of the classes.

CNN is a data hungry model. To reduce the data volume requirements we use augmentation which is a very common approach for increasing the size of the training sets for CNN training. Here, we propose the following data augmentation procedure: (1) we shifted each spectrum left or right a few wavenumbers randomly. (2) We added random noise, proportional to the magnitude at each wave numbers. (3) For the substances which had more than one spectra, we took linear combinations of all spectra belonging to the same substance as augmented data. The coefficients in the linear combination were chosen at random.

The training of the CNN was performed using the Adam algorithm,[27] which is a variant of stochastic gradient descent, for 50 epochs with learning rate equal to $1 \times 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1 \times 10^{-8}$. The layers were initialised from a Gaussian distribution with a zero mean and variance equal to 0.05. We applied early stopping to prevent overfitting. Training was performed on a single NVIDIA GTX-1080 GPU. The training time was around seven hours. While for inference, it took less than one millisecond to process a spectrum.

### 2.3. Evaluation protocol

We tested the proposed CNN method for mineral species recognition on the largest publicly available mineral database RRUFF[17] and compared it with a number of alternative, well known, machine learning methods. As there are usually only a handful of spectra available for each mineral, we use a leave-one-out scheme to split a dataset into training and test sets. To be specific, for minerals which have more than one spectra, we randomly select a spectrum for testing and use the rest for training. We compared our method to cosine similarity[16]/correction[18] (which has been used in commercial software such as CrystalSleuth and Spectral-ID), and to other methods that have been shown to be successful in classification tasks including applications based on Raman spectroscopy: nearest neighbor, gradient boosting machine, random forest, and support vector machine.[28]

The proposed CNN was implemented using Keras[29] and Tensorflow.[30] The gradient boosting machine method was implemented based on lightGBM released by Microsoft. All other methods were implemented using Scikit-learn.[31]

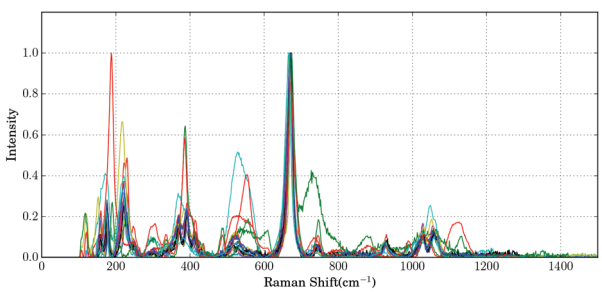
## 3. Results and discussion

### 3.1. Classifying baseline-corrected spectra

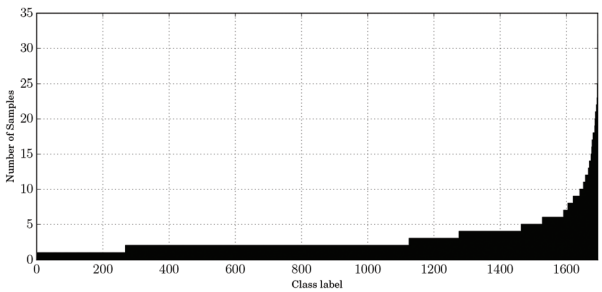
We first evaluated our CNN method on a processed mineral dataset from the RRUFF database. These spectra have been baseline corrected and cosmic rays have also been removed. The dataset contains 1671 different kinds of minerals, 5168 spectra in total. Spectra for the mineral *Actinolite* are shown in Fig. 2(a), illustrating the typical within-class variance. The number of spectra per mineral ranges from 1 to 40. The distribution of sample numbers per a mineral species is shown in Fig. 2(b). We followed the protocol as described in section 2.3 to generate training and test sets randomly using the leave-one-out scheme.

In a large scale classification, some classes could be quite similar and differentiating between them could be very difficult or even impossible. Hence, it is common to report top-1 and top-$k$ accuracy. In the former, the class that the classifier assigns the highest probability to is compared to the true label. The latter reports whether the true label appears among the $k$ classes with the highest probability (assigned by the classifier).

We report in Table 2, the top 1, 3 and 5 accuracies of the compared methods, averaged over 50 independent runs. One can see that CNN outperformed all other methods and




(a) Spectra of *Actinolite*[17].




(b) Number of spectra per mineral of the whole dataset.

**Fig. 2** (a) Spectra of an example mineral species (*Actinolite*) indicating the within class spectrum variation and (b) a frequency plot showing the imbalance regarding spectra per species.

Table 2 Test accuracy of the compared machine learning methods on the baseline corrected dataset

| Methods | KNN($k$ = 1) | Gradient boosting | Random forest§ | SVM(linear) | SVM(rbf) | Correlation | CNN§ |
|---|---|---|---|---|---|---|---|
| Top-1 accuracy | 0.779 ± 0.011 | 0.617 ± 0.008 | 0.645 ± 0.007 | 0.819 ± 0.004 | 0.746 ± 0.003 | 0.717 ± 0.006 | 0.884 ± 0.005 |
| Top-3 accuracy | 0.780 ± 0.011 | 0.763 ± 0.011 | 0.753 ± 0.010 | 0.903 ± 0.006 | 0.864 ± 0.006 | 0.829 ± 0.005 | 0.953 ± 0.002 |
| Top-5 accuracy | 0.780 ± 0.011 | 0.812 ± 0.010 | 0.789 ± 0.009 | 0.920 ± 0.003 | 0.890 ± 0.007 | 0.857 ± 0.005 | 0.963 ± 0.002 |

achieved top-1 accuracy of 88.4% and top-3 accuracy of 96.3%. The difference in classification accuracy between CNN and the second best method is statistically significant, $t(50) = 71.78$, $p < 0.001$.

To understand the trained model of CNN better, we also closely examined typical predictions, especially where these did not agree with the correct labelling. In Fig. 3 the top spectrum in each set is the test sample (shown in red) which is followed by the top-3 predictions given by the CNN. The correct prediction is highlighted in green. We also show scores in each plot which reflect the confidence level of predictions. Fig. 3(a) shows the examples where the CNN made the correct prediction. Fig. 3(b) shows the examples in which the correct prediction is scored second. In Fig. 3(c), the top-3 predictions do not include the correct label.

As shown in Fig. 3(a), the CNN successfully predicted the correct mineral, *actinolite*, and also ranked *Ferroactinolite* and *Tremolite* as the second and third probable candidates. In fact, these three minerals are all members of the same mineral group. This is not uncommon. For instance, in Fig. 3(b), the

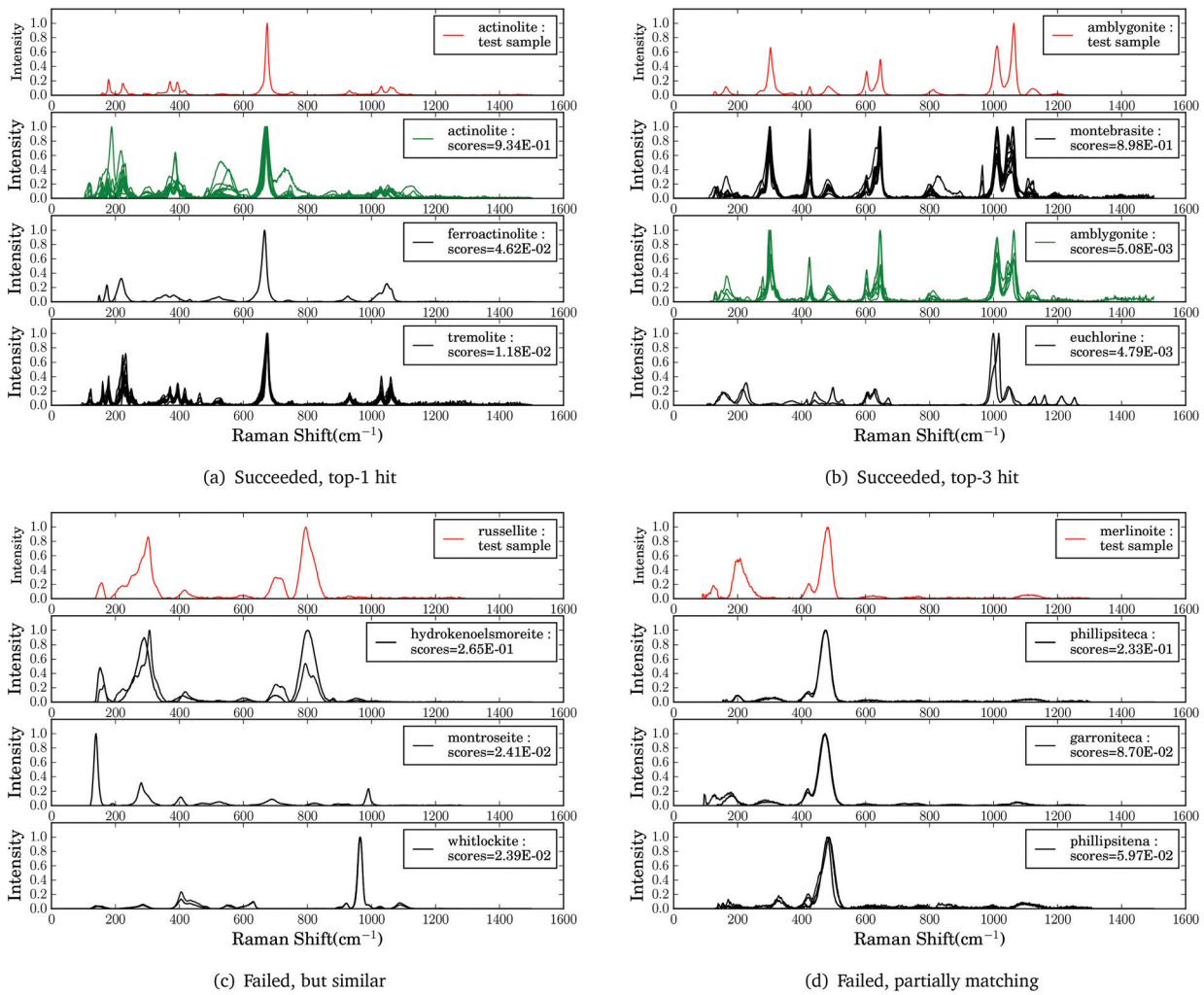


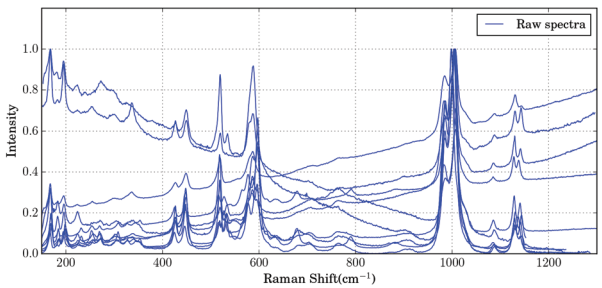

Fig. 3 Examples of successful and unsuccessful mineral species classifications. In each plot, the top spectrum which is marked in red is a test sample. The three spectra below were the top-3 predictions given by the CNN among which the correct one was highlighted in green. The prediction scores were also shown in each plot which reflect the confidence level of predictions.

most probable mineral *Montebrasite* (as predicted by the CNN) belongs to the same group as the correct one, *Amblygonite*, and they share similar spectral structure.

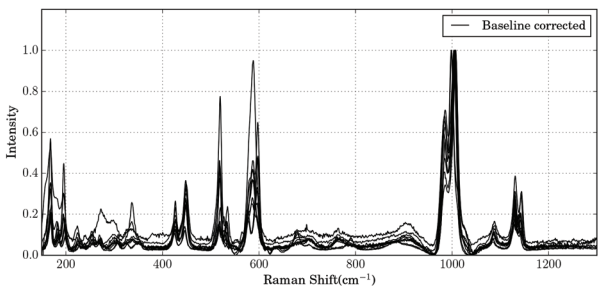
If we examine the peak similarity, for instance in Fig. 3(c), the peak locations of the top-1 prediction, *Hydrokenoelsmoreite*, are almost identical to those of the test sample *Russellite*. In Fig. 3(d), only the main peaks were matched correctly. These plots demonstrate that the CNN was capable of matching the peaks characteristic of a particular species even when the prediction did not agree with the correct label.

### 3.2. Unified Raman analysis using CNN

The results in section 3.1 have shown that CNN was able to achieve significantly better accuracy compared to other conventional machine learning methods on the *baseline-corrected* spectra. Recall that conventional machine learning methods such as SVM and Random Forest are not capable of handling Raman signals which are not properly baseline corrected, and therefore require explicit baseline correction in their processing pipelines. However, robust baseline correction is a challenging problem, especially for a fully automatic system.[1] On the other hand, in a variety of applications, CNN has been shown to be very successful as an end-to-end learning tool since it can process the data and learn features automatically, avoiding hand crafted feature engineering.[32] Therefore, we evaluated the proposed CNN and the other classification methods using both raw and baseline corrected spectra. Specifically, we were interested in the performance of CNN on raw data compared to the previous state-of-the-art spectral classification methods for which baseline correction *was included*.




(a) Ten raw spectra where baselines can be clearly observed.




(b) Baseline corrected by asymmetric least squares

**Fig. 4** Spectra of a mineral, *hydroxylherderite*, from RRUFF raw database and corresponding baseline corrected ones by asymmetric least squares.

For this set of experiments, we selected another dataset from the RRUFF database which contains raw (uncorrected) spectra for 512 minerals and six widely-used baseline correction methods: *modified polynomial fitting*,[1] *rubber band*,[3] *robust local regression estimation*,[33] *iterative restricted least squares, asymmetric least square smoothing*,[6] *rolling ball.*[2] We used implementations of these methods in the R packages *baseline*[34] and *hyperSpec.*[35] An example of raw spectra and corresponding baseline corrected ones by asymmetric least squares is shown in Fig. 4. We followed the training and evaluation protocol as described in section 2.3. The results are reported in Table 3.

For the conventional classification methods, used as a comparison in our work, PCA was adopted to reduce dimensionality and extract features, except for Random Forest where we found that PCA decreased the performance. This is indicated in the table by §. The number of principal components were determined such that 99.9% of total variance was retained. One can see that CNN on the raw spectra achieved an accuracy of 93.3% which is significantly better, $t(50) = 77.14$, $p < 0.001$, than the second best method, KNN with rubber band baseline correction, that achieved an accuracy of 82.5%.

There are a few remarks which are worth highlighting. Firstly, it is not a surprise that baseline correction greatly improved the performance of all the conventional methods by 20%–40%. On other hand, CNN's performance dropped by about 0.5%–2.5% when combined with baseline correction methods. This may indicate that CNN was able to learn more efficient way of handling the interference of the baselines and to retain more discriminant information than using an explicit

**Table 3** Test accuracy of the compared machine learning methods on raw dataset with or without baseline correction methods

| Methods | KNN($k = 1$) | Gradient boosting | Random forest§ | SVM(linear) | SVM(rbf) | Correlation | CNN§ |
|---|---|---|---|---|---|---|---|
| Raw | 0.429 ± 0.011 | 0.373 ± 0.019 | 0.394 ± 0.016 | 0.522 ± 0.011 | 0.434 ± 0.012 | 0.310 ± 0.007 | 0.933 ± 0.007 |
| Asymmetric least squares | 0.817 ± 0.010 | 0.773 ± 0.009 | 0.731 ± 0.019 | 0.821 ± 0.012 | 0.629 ± 0.016 | 0.777 ± 0.013 | 0.927 ± 0.008 |
| Modified polynomial | 0.778 ± 0.007 | 0.740 ± 0.016 | 0.650 ± 0.016 | 0.785 ± 0.014 | 0.629 ± 0.016 | 0.734 ± 0.013 | 0.920 ± 0.008 |
| Rolling ball | 0.775 ± 0.009 | 0.737 ± 0.008 | 0.689 ± 0.018 | 0.795 ± 0.011 | 0.624 ± 0.013 | 0.730 ± 0.010 | 0.918 ± 0.008 |
| Rubber band | 0.825 ± 0.007 | 0.792 ± 0.015 | 0.741 ± 0.009 | 0.806 ± 0.015 | 0.620 ± 0.010 | 0.789 ± 0.010 | 0.911 ± 0.008 |
| IRLS | 0.772 ± 0.010 | 0.710 ± 0.008 | 0.675 ± 0.007 | 0.781 ± 0.011 | 0.614 ± 0.010 | 0.711 ± 0.011 | 0.911 ± 0.008 |
| Robust local regression | 0.741 ± 0.009 | 0.694 ± 0.008 | 0.667 ± 0.012 | 0.759 ± 0.013 | 0.600 ± 0.013 | 0.696 ± 0.011 | 0.909 ± 0.007 |

Table 4 Test accuracy of the compared machine learning methods on Unipr-mineral dataset with or without baseline correction methods

| Methods | KNN($k$ = 1) | Gradient boosting | Random forest§ | SVM(linear) | SVM(rbf) | Correlation | CNN§ |
|---|---|---|---|---|---|---|---|
| Raw | 0.893 ± 0.023 | 0.723 ± 0.069 | 0.695 ± 0.044 | 0.880 ± 0.037 | 0.874 ± 0.031 | 0.823 ± 0.022 | 0.947 ± 0.020 |
| Asymmetric least squares | 0.905 ± 0.020 | 0.787 ± 0.030 | 0.737 ± 0.051 | 0.913 ± 0.030 | 0.926 ± 0.018 | 0.903 ± 0.014 | 0.952 ± 0.018 |
| Modified polynomial | 0.882 ± 0.024 | 0.768 ± 0.052 | 0.722 ± 0.061 | 0.904 ± 0.026 | 0.893 ± 0.020 | 0.855 ± 0.023 | 0.952 ± 0.016 |
| Rolling ball | 0.912 ± 0.015 | 0.774 ± 0.047 | 0.751 ± 0.044 | 0.918 ± 0.019 | 0.924 ± 0.024 | 0.885 ± 0.021 | 0.949 ± 0.018 |
| Rubber band | 0.864 ± 0.030 | 0.723 ± 0.085 | 0.701 ± 0.049 | 0.873 ± 0.030 | 0.894 ± 0.032 | 0.912 ± 0.014 | 0.942 ± 0.022 |
| IRLS | 0.876 ± 0.014 | 0.794 ± 0.046 | 0.731 ± 0.046 | 0.912 ± 0.026 | 0.928 ± 0.025 | 0.873 ± 0.016 | 0.930 ± 0.019 |
| Robust local regression | 0.866 ± 0.017 | 0.770 ± 0.046 | 0.719 ± 0.045 | 0.878 ± 0.026 | 0.909 ± 0.024 | 0.850 ± 0.022 | 0.923 ± 0.017 |

baseline correction method. The advantage of CNNs in achieving high accuracy of classification while requiring minimal preprocessing of spectra opens new possibilities for developing highly accurate fully automatic spectrum recognition systems.

Besides the two datasets from the RRUFF database, we also validated the proposed method on another (independent) dataset which includes both baseline corrected and uncorrected spectra. The additional dataset, which we refer to as Unipr-mineral,[36] comprises spectra for 107 minerals, 163 spectra in total. The same training, evaluation protocol, and network architecture were used. Results for this dataset are presented in Table 4 and show that CNN is again significantly more accurate than the second best method ($t(50) = 4.20$, $p < 0.001$).



## 4. Conclusion and future work

In this paper, we have presented a deep convolutional neural network solution for Raman spectrum classification which not only exhibits outstanding performance, but also avoids the need for spectrum preprocessing of any kind. Our method has been validated on a large scale mineral database and was shown to outperform other state-of-the-art machine learning methods by a large margin. Although we focused our study on Raman data we believe the method is also applicable to other spectroscopy and spectrometry methods. We speculate that this may be achieved very efficiently by exploiting basic similarities in the shape of spectra originating from different techniques and fine tuning our network to address new classification problems. This process is known as transfer learning and has been demonstrated previously in many object recognition applications.

## Conflicts of interest

There are no conflicts to declare.

## References

1 C. A. Lieber and A. Mahadevan-Jansen, Automated Method for Subtraction of Fluorescence from Biological Raman Spectra, *Appl. Spectrosc.*, 2003, **57**, 1363–1367.
2 M. Kneen and H. Annegarn, Algorithm for fitting XRF, SEM and PIXE X-ray spectra backgrounds, *Nucl. Instrum. Methods Phys. Res., Sect. B*, 1996, **109**, 209–213.
3 S. Wartewig, *IR and Raman Spectroscopy: Fundamental Processing*, Wiley-VCH Verlag GmbH & Co. KGaA, 2005, pp. 75–124.
4 Z.-M. Zhang, S. Chen and Y.-Z. Liang, Baseline correction using adaptive iteratively reweighted penalized least squares, *Analyst*, 2010, **135**(5), 1138–1146.
5 S.-J. Baek, A. Park, Y.-J. Ahn and J. Choo, Baseline correction using asymmetrically reweighted penalized least squares smoothing, *Analyst*, 2015, **140**, 250–257.
6 P. H. C. Eilers and H. F. Boelens, Baseline Correction with Asymmetric Least Squares Smoothing, *Leiden university medical centre technical report*, 2005.
7 G. Schulze, A. Jirasek, M. Marcia, A. Lim, R. F. Turner and M. W. Blades, Investigation of selected baseline removal techniques as candidates for automated implementation, *Appl. Spectrosc.*, 2005, **59**(5), 545–574.
8 V. Vapnik, The Nature of Statistical Learning Theory, in *Data mining and knowledge discovery*, 1995.
9 M. Sattlecker, C. Bessant, J. Smith and N. Stone, Investigation of support vector machines and Raman spectroscopy for lymph node diagnostics, *Analyst*, 2010, **135**, 895–901.
10 E. Widjaja, W. Zheng and Z. Huang, Classification of colonic tissues using near-infrared Raman spectroscopy and support vector machines, *Int. J. Oncol.*, 2008, **32**, 653–662.
11 A. Kyriakides, E. Kastanos and C. Pitris, Classification of Raman Spectra using Support Vector Machines, in *2009 9th International Conference on Information Technology and Applications in Biomedicine*, 2009, pp. 1–4.
12 T. K. Ho, The random subspace method for constructing decision forests, *IEEE Trans. Pattern Analysis Machine Intelligence*, 1998, **20**(8), 832–844.
13 A. Maguire, I. Vega-Carrascal, J. Bryant, L. White, O. Howe, F. Lyng and A. Meade, Competitive evaluation of data mining algorithms for use in classification of leukocyte subtypes with Raman microspectroscopy, *Analyst*, 2015, **140**(7), 2473–2481.
14 K. Maquelin, C. Kirschner, L. Choo-Smith, N. Ngo-Thi, T. van Vreeswijk, M. Stammler, H. Endtz, H. Bruining, D. Naumann and G. Puppels, Prospective study of the performance of vibrational spectroscopies for rapid identifi-

cation of bacterial and fungal pathogens recovered from blood cultures, *J. Clin. Microbiol.*, 2003, **41**, 324–329.
15 S. T. Ishikawa and V. C. Gulick, An Automated Mineral Classifier Using Raman Spectra, *Comput. Geosci.*, 2013, **54**, 259–268.
16 C. Carey, T. Boucher, S. Mahadevan, P. Bartholomew and M. Dyar, Machine learning tools for mineral recognition and classification from Raman spectroscopy, *J. Raman Spectrosc.*, 2015, **46**(10), 894–903.
17 B. Lafuente, R. T. Downs, H. Yang and N. Stone, The power of databases: the RRUFF project, in *Highlights in Mineralogical Crystallography*, 2015, pp. 1–30.
18 A. Kwiatkowski, M. Gnyba, J. Smulko and P. Wierzba, Algorithms of chemicals detection using Raman spectra, *Metrol. Meas. Syst.*, 2010, **17**(4), 549–559.
19 D. H. Hubel and T. N. Wiesel, Receptive Fields and Functional Architecture of Monkey Striate Cortex, *J. Physiol.*, 1968, **195**, 215–243.
20 Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*, 1998, **86**(11), 2278–2324.
21 Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, *Proc. IEEE*, 1998, 2278–2324.
22 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, *Computer Vision and Pattern Recognition (CVPR)*, 2015.
23 K. He, X. Zhang, S. Ren and J. Sun, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
24 A. L. Maas, A. Y. Hannun and A. Y. Ng, *Proc. ICML*, 2013.
25 S. Ioffe and C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, ArXiv e-prints, 2015.
26 N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.*, 2014, **15**, 1929–1958.
27 D. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.
28 C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 2006.
29 F. Chollet, *et al.*, Keras, 2015, https://github.com/fchollet/keras.
30 M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015, Software available from tensorflow.org.
31 L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt and G. Varoquaux, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.
32 A. Krizhevsky, I. Sutskever and G. E. Hinton, *Advances in neural information processing systems*, 2012, pp. 1097–1105.
33 A. F. Ruckstuhl, M. P. Jacobson, R. W. Field and J. A. Dodd, Baseline subtraction using robust local regression estimation, *J. Quant. Spectrosc. Radiat. Transfer*, 2001, **68**(2), 179–193.
34 K. H. Liland and B.-H. Mevik, *baseline: Baseline Correction of Spectra*.
35 C. Beleites and V. Sergo, *hyperSpec: a package to handle hyperspectral data sets in R*, 2016.
36 DiFeST, http://www.fis.unipr.it/phevix/ramandb.php.