Article 25fa pilot End User Agreement

This publication is distributed under the terms of Article 25fa of the Dutch Copyright Act (Auteurswet) with explicit consent by the author. Dutch law entitles the maker of a short scientific work funded either wholly or partially by Dutch public funds to make that work publicly available for no consideration following a reasonable period of time after the work was first published, provided that clear reference is made to the source of the first publication of the work.

This publication is distributed under The Association of Universities in the Netherlands (VSNU) ‘Article 25fa implementation’ pilot project. In this pilot research outputs of researchers employed by Dutch Universities that comply with the legal requirements of Article 25fa of the Dutch Copyright Act are distributed online and free of cost or other barriers in institutional repositories. Research outputs are distributed six months after their first online publication in the original published version and with proper attribution to the source of the original publication.

You are permitted to download and use the publication for personal purposes. All rights remain with the author(s) and/or copyrights owner(s) of this work. Any use of the publication other than authorised under this licence or copyright law is prohibited.

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please contact the Library through email: copyright@ubn.ru.nl, or send a letter to:

University Library
Radboud University
Copyright Information Point
PO Box 9100
6500 HA Nijmegen

You will be contacted as soon as possible.
Convolutional neural networks for vibrational spectroscopic data analysis

Jacopo Acquarelli a,*, Twan van Laarhoven a, Jan Gerretzen b, Thanh N. Tran b,c, Lutgarde M.C. Buydens b, Elena Marchiori a,**

a Radboud University Nijmegen, Institute for Computing and Information Science, The Netherlands
b Radboud University Nijmegen, Institute for Molecules and Materials, The Netherlands
c Center for Mathematical Sciences, Merck, Sharp & Dohme, Oss, The Netherlands

HIGHLIGHTS

- An application of convolutional neural networks (CNN) on vibrational spectroscopy data is proposed.
- The proposed CNN achieved higher accuracy with respect to Partial Least Squares (PLS) on the considered datasets.
- Data preprocessing is a critical step in chemometrics. Our method is less dependent on preprocessing than PLS.
- Using a simple method is possible to interpret the results achieved by the proposed CNN.

ABSTRACT

In this work we show that convolutional neural networks (CNNs) can be efficiently used to classify vibrational spectroscopic data and identify important spectral regions. CNNs are the current state-of-the-art in image classification and speech recognition and can learn interpretable representations of the data. These characteristics make CNNs a good candidate for reducing the need for preprocessing and for highlighting important spectral regions, both of which are crucial steps in the analysis of vibrational spectroscopic data.

Chemometric analysis of vibrational spectroscopic data often relies on preprocessing methods involving baseline correction, scatter correction and noise removal, which are applied to the spectra prior to model building. Preprocessing is a critical step because even in simple problems using ‘reasonable’ preprocessing methods may decrease the performance of the final model.

We develop a new CNN based method and provide an accompanying publicly available software. It is based on a simple CNN architecture with a single convolutional layer (a so-called shallow CNN). Our method outperforms standard classification algorithms used in chemometrics (e.g. PLS) in terms of accuracy when applied to non-preprocessed test data (86% average accuracy compared to the 62% achieved by PLS), and it achieves better performance even on preprocessed test data (96% average accuracy).
1. Introduction

Vibrational spectroscopy involves specific optical techniques of infrared (IR) and Raman spectroscopy. Classification models for vibrational spectroscopic data map input objects (spectra) to desired outputs (class assignments). Designing a classification model in this context is a challenging task performed in diverse application domains like pharmaceutical, polymers, forensic, environmental and food sciences, as well as in medicine [1–6]. Popular chemometric techniques for this task include artificial neural networks, support vector machines and linear discriminant classifiers [7]. These techniques are in general applied to preprocessed data. One of the major difficulties in this context is determining a consensus on which spectral preprocessing method and optimal settings to use.

Data preprocessing is used to improve the robustness and accuracy of subsequent multivariate analysis and to increase the interpretability of the data by correcting issues associated with spectral data acquisition. Preprocessing methods often depend on the objective of the study and on the technique (Raman or IR) used [8]. A plethora of preprocessing methods and software for vibrational spectroscopic data have been developed (see e.g. Refs. [2,9,10]). These methods are often selected based on somewhat arbitrary criteria such as “it worked well on a previous dataset”. It has been recently shown that even for relatively simple problems the majority of ‘reasonable’ preprocessing methods, with their respective parameter settings, may actually decrease the performance of the final model [7]. In general, the same preprocessing technique may work well for one dataset but does not work when applied to another dataset generated using a different machine, setting, or sample matrix.

While the chemometric research community has so far mainly focused on the problem of selecting a good preprocessing method [11], the machine learning community considered the somewhat related problem of learning representations of the data that identify and disentangle the underlying explanatory factors hidden in the data [12]. In particular Convolutional Neural Networks (CNNs) were inspired by biological processes in the visual cortex in animals, where cells are sensitive to small sub-regions of the visual field.

CNNs are variations of multilayer perceptrons (MLP) (see Fig. 1), a feed-forward Artificial Neural Network (ANN) model that maps input data onto a set of appropriate outputs. ANNs for classification have been applied to vibrational spectroscopic data in the past (for an overview see the survey by Ref. [13]). However, these computational models had two main issues. First of all, they were susceptible to overfitting, leading to poor performance on new data. Secondly, it was not possible to interpret the classifiers: the trained neural network was treated as a ‘black box’.

Recent advances in machine learning allow us to address both of these issues using CNNs, which are designed to consider spatial information of the input data: they exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers [14]. CNNs have fewer parameters than traditional neural networks, and with embedded regularization techniques they are more robust to overfitting. The output of each convolutional layer of these networks is directly related to small regions of the input spectrum. Thus a CNN could be used to identify important regions of the input data from the classifier after it has been trained.

CNNs are the state of art for 2D data (image) classification [15,16] and have also been recently applied to 3D data classification of Hyperspectral Remote Sensing Scenes [17]. CNNs for image classification use pooling and fully connected layers. Here we focus on 1D data (vibrational spectroscopic data) and do not use pooling but take advantage of smoothing provided by kernels of the convolutional layer and the stride. Instead of pooling, stride makes it easier to use the final model for identifying important spectral regions. Furthermore, we employ a novel regularization term tailored in its objective, used to smooth the input signal (which in our case is a vibrational spectrum). In this way the CNN model is able to more easily adapt to different spectral inputs to and generalize beyond training data. To the best of our knowledge, the use of CNNs for classification and identifying important regions of vibrational spectroscopic data has not been investigated yet.

We demonstrate that a simple method based on shallow CNN (see Fig. 2) achieves significantly better classification accuracy than Partial Least Squares Regression - Linear Discriminant Analysis (PLS-LDA) and kNN, both without applying any preprocessing as well as by applying an optimal way of preprocessing to the input spectra. Moreover, we show that CNN is less preprocessing dependent than PLS-LDA (The software is publicly available at https://bitbucket.org/TeslaH2O/cnn_chemometrics).

The interpretation of results is important in chemometrics as much as classification and this is the reason why PLS-LDA is a standard method used in chemometrics. In out method, detection of important spectral regions is performed by applying feature selection to the output of the convolutional layer: the new representation of the spectra provided by the convolution highlights regions of the spectra considered relevant.

In summary, the main novelties of this work are:

- the employment of a non standard CNN: shallow architecture (only 1 hidden convolutional layer) and no pooling;
- the design of a custom loss function for CNN which includes a new regularization term to enforce similarity between nearby features;
- the enhancement of CNN through a procedure for finding important spectral regions.

Results of our experimental analysis show that the proposed method provides a powerful tool for classifying and interpreting vibrational spectroscopic data.

2. Methods

In this section, we will first briefly describe the general principles of ANN. Next, we will describe the implemented modifications to make the neural network suitable for data analysis of vibrational spectroscopic data. Finally, we will provide details about spectral feature selection using our CNN approach.

An ANN is composed by a set of inter-connected artificial neurons. Neurons are the basic unit of an ANN and they are characterized by a so-called ‘activation function’ that transforms the input of the neuron to its output. The artificial neurons are organized by layers and in each layer all neurons have the same activation function. An MLP is a feed-forward ANN having no connections compared to the 89% achieved by PLS). For interpretability purposes, our method includes a procedure for finding important spectral regions, thereby facilitating qualitative interpretation of results. © 2016 Elsevier B.V. All rights reserved.
between units of the same layer (feed-forward implies that there also no connections going from a neuron to a previous layer).

It is possible to divide layers into 3 sets: input, hidden and output layers. The input layer is the first layer and it generally has a linear activation function. The output layer is the last layer and it generally has a linear or softmax activation function for regression or classification respectively. An MLP has generally one or more hidden layers having the same non-linear activation function (See Fig. 1). Each unit of each layer is connected to each unit of the next layer (fully-connected) using weighted connections. These weights are generally randomly initialized and learned during the training phase in a supervised manner. For this purpose, back-propagation together with different variations of the Gradient Descent method [18] are used to find a local minimum of an objective function enclosing the prediction error of the network.

Using fully-connected layers means that a considerable amount of weights have to be trained and this amount varies depending on the number of units in each layer. When just a few samples are available to train all weights, the network can easily overfit. This is one of the major reasons why ANNs are not commonly used in chemometric data analysis, although they are very well suited to deal with highly non-linear problems (which may e.g. occur due to improper preprocessing of the data).

We tried to overcome this limitation by using a variation of an MLP, having non fully-connected layers (i.e. having fewer weights to be trained) and introducing a regularization term to the objective function [19].

![Fig. 1. Example of an MLP having two hidden layers.](image1)

![Fig. 2. Example of a CNN for a two-class classification problem with one convolutional hidden layer and two output nodes. X represents the input, k the kernel weights, w the weights between the convolutional layer and the output and Y the predicted class.](image2)
function. Regularization increases the general applicability of a neural network beyond the data on which it was trained, by enforcing small weights. These are generally preferable because, in a sense, small weights mean lower complexity and so they provide an easier explanation of the data [19].

Instead of a fully-connected layer we used a convolutional layer (hence the name CNN) in which the input is first convolved using a kernel (see Fig. 2). Generally, one or more kernels can be used on each convolutional layer to capture different properties of the input data at the cost of increasing the number of weights to be learned.

The convolution is performed by shifting a kernel \( k = \sum_{i=1}^{N} w_i \) with \( N \) elements and a certain step called ‘stride’ from the first to the last input element (See Fig. 3). Therefore a new representation of the input is generated over the whole spectrum using contributions of neighboring features: for this purpose each kernel is repeatedly applied to the input itself. After each application, the outcome of the convolution is provided as input to a rectified linear function, a different representation of the input (having size \( N \)) works in one dimension using a stride \( s \).

![Fig. 3. Example of how convolution \( conv() \) between an input vector \( x \) and a kernel \( k \) (having size \( N \)) works in one dimension using a stride \( s \).](image)

Here \( \hat{y}_n \equiv \psi(w \cdot x_n) \) is the network’s output, \( \psi(\cdot) \) is the activation function, \( x_n \) is the \( n \)-th sample, \( w \) are the weights, \( y_n \) is the target label and \( Shift(\cdot) \) is an operation that shifts the elements of an array one position to the left. In addition to the standard \( L_2 \) norm we used a ‘proximity \( L_2 \) norm’ which helps the network keeping neighboring input variables (i.e. wavenumbers for vibrational spectroscopic data) correlated by penalizing large differences between near weights. For vibrational spectroscopic data, we don’t expect those variations because the value of the spectrum at a certain wavenumber is dependent on the neighboring wavenumber values.

We used a 1-dimensional kernel because each sample (i.e. spectrum) is represented as a 1-dimensional array (vector). We also used a fully connected output layer (after the convolutional layer) with the number of units equal to the number of classes. The use of the softmax activation function on this output layer allows to obtain the class prediction of the network in response of an input sample. The Stochastic Gradient Descent (SGD) [20] updating rule was used to train our neural network since it is a standard technique for training neural networks.

Weights (i.e. values) for the convolutional layer kernels and for the output layer need to be initialized. We choose the ‘Glorot’ initialization [21] since it is possible to reproduce an initialization by keeping track of the seed which was used for randomization [22].

Therefore the parameters of our model and the range of their values are:

- number of kernels of the convolutional layer: \( \#kernels \in \{1, 2, 4\} \)
- size of kernels of the convolutional layer: \( N \in [2, 91] \)
- stride for the convolution: \( s \in [1, 39] \)
- parameters in the regularization terms: \( \lambda_1, \lambda_2 = 10^n \) where \( n \in [-3, 3] \)
- momentum in the SGD updating rule: \( momentum \in [0.1, 2.9] \)
- learning rate: \( lr = 10^n \) where \( n \in [-8, -1] \)

In order to find the best combination of parameter values we used a Random Grid Search Cross-Validation framework (RGS-CV) [23] during the training phase to select the configuration with the highest accuracy. Then the model was refitted using the whole train data and applied to the test data to obtain classification accuracies.

We call the resulting method CNNVS, Convolutional Neural Network for classification of Vibrational Spectroscopic data.

### 2.1. Finding important spectral regions

Generating a high-quality classification model and identifying potentially important spectral regions are both important aspects in vibrational spectroscopic data analysis. Methods like Partial Least Squares Regression - Linear Discriminant Analysis (PLS-LDA) may find important input features by representing them in a latent variables space. Although PLS-LDA may provide unreliable indications, especially for data with large amount of irrelevant data variation [24], this aspect contributes to their popularity in chemometric data analysis over more powerful methods, like Support Vector Machine (SVM) with non-linear kernels, for which feature

\[
OBJ(w) = \frac{1}{N} \sum_{n=1}^{N} \left[ y_n \log \hat{y}_n + (1 - y_n) \log (1 - \hat{y}_n) \right] + \lambda_1 \|w\|^2 + \lambda_2 \|w - Shift(w)\|^2
\]

Cross-entropy Error Loss

\[
\text{standard } L_2 \text{ norm } + \lambda_2 \|w - Shift(w)\|^2
\]

\[
\text{proximity } L_2 \text{ norm }
\]

Regularization term
It is straightforward to integrate a feature selection procedure into CNNVS for identifying important spectral regions as follows. Each output node of the convolutional layer of the trained CNN corresponds to a specific kernel applied to a specific region of the spectrum. Thus a feature selection algorithm can be applied to each output to find important features; these features are then back-transformed into the corresponding regions of the spectrum.

In our analysis we used stability feature selection [26]. Stability feature selection works as follows: feature selection is applied to different randomly selected subsets of the data (i.e. the output of the convolutional layer of the trained CNNVS). This feature selection is performed by retraining the last layer of CNNVS which can be seen as a Logistic Regression network having as input the output of the convolutional layer in response to different subsets of spectra, and as output the class prediction. After each retraining, features with positive coefficients are selected. The results over the subsets are then merged by considering for each feature the fraction of times it was selected, which yields a score for each feature.

The choice for stability feature selection is motivated by the effectiveness of this method in problems where the number of features is much bigger than the number of samples, which is the case for most vibrational spectroscopic datasets. We call the resulting method for identifying important spectral regions CNNVSfs, Convolutional Neural Network for classification of Vibrational Spectroscopic data with incorporated Feature Selection.

### 2.2. Interpretation of kernel elements

As mentioned in Section 2, multiple kernels can be used on a convolutional layer. The aim of using different kernels is to catch different properties of the spectra. Limitations on their number are mostly related to overfitting, thus we used the model selection method RGS-CV (described at the end of Section 2) to discard models having too many kernels for the considered dataset.

The analysis of the learned kernels may be important for the interpretation to find out artifacts on the spectra and to speculate on the behavior of each kernel in relation to well known preprocessing methods.

### 3. Evaluation study

In order to comparatively assess the performance of our method, we collected a number of publicly available vibrational spectroscopic datasets for classification. We compared CNNVS with PLS-LDA, a method often used in chemometric data analysis because it provides interpretable models with acceptable prediction accuracies. We also considered Logistic Regression, which can be seen as a very simple neural network with no hidden layers, where the input passes directly to the output layer through an activation function: \( \psi(t) = \frac{1}{1+e^{-t}} \) [27]. Comparing CNNVS with Logistic Regression therefore allows us to study the effectiveness of using a convolutional hidden layer. Finally, we used kNN, a simple method which does not require training.

We performed two series of experiments:

1. direct application of the different classification methods to the raw data (i.e. no preprocessing);
2. data preprocessing prior to the application of a classification method. For this, we preprocess the data according to the general procedure described in Refs. [7,11]. For each classification method, we used internal cross-validation to find the optimal preprocessing method (or a combination of methods) to be applied to the data, as described in Section 3.3.

### 3.1. Datasets

The following vibrational spectroscopic datasets are considered in this work:

- Beers datasets containing Rochefort 8 (class 1) and Rochefort 10 (class 2) beers, obtained using three different types of spectroscopy: Fourier Transform Infrared (FT-IR) [28], Near-Infrared (NIR) and Raman.
- Wines dataset (FT-IR) where the four classes represent different geographical areas of origin [29] (available from: \[\text{http://www.models.life.ku.dk/Wine_GCMS_FTIR}\]).
- Tablets datasets obtained using NIR and Raman, where the samples were grouped in 4 different types of tablets with different amount of active substance as described in Ref. [30] (available from: \[\text{http://www.models.life.ku.dk/Tablets}\]).
- Coffees dataset (FT-IR) containing samples of Arabica (class 1) and Robusta (class 2) coffee. This spectra in this dataset were truncated to 800–2000 cm\(^{-1}\) [31] (available from: \[\text{http://asu.ifr.ac.uk/example-datasets-for-download}\]).
- Olive Oils dataset (FT-IR) containing samples originating from 4 different countries (Greece, Italy, Portugal and Spain) corresponding to the 4 different classes. Spectra in this dataset were truncated to 799–1897 cm\(^{-1}\) [32] (available from: \[\text{http://asu.ifr.ac.uk/example-datasets-for-download}\]).
- Juices dataset (FT-IR) containing strawberry (class 1) and non strawberry juices (class 2), with spectra truncated to 899–1802 cm\(^{-1}\) [33] (available from: \[\text{http://asu.ifr.ac.uk/example-datasets-for-download}\]).
- Meats dataset (FT-IR) containing chicken (class 1), pork (class 2) and turkey meats (class 3), with spectra truncated to 1000–1800 cm\(^{-1}\) [34] (available from: \[\text{http://asu.ifr.ac.uk/example-datasets-for-download}\]).

An overview of the characteristics of the different datasets is given in Table 1. For the Tablets, Coffees, Meats and Olive Oils datasets, in absence of a pre-defined test set, we built a training set containing the 67% of randomly selected samples and the remaining 33% of the samples were used as test set.

### 3.2. Other classification techniques for comparison

We considered the standard Partial Least Squares Linear Discriminant Analysis (PLS-LDA) method used in chemometric data classification, Logistic Regression (LogReg) and k-Nearest Neighbor (kNN).

#### 3.2.1. PLS-LDA

Partial Least Squares is a regression method that aims to represent a dataset with \( n \) samples and \( p \) variables into a space of \( a \) latent variables \( T = [t_1, t_2, \ldots, t_a] \), with directions that have both

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Scan</th>
<th># Train samples</th>
<th># Test samples</th>
<th># Features</th>
<th># Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beers</td>
<td>FTIR</td>
<td>44</td>
<td>30</td>
<td>1660</td>
<td>2</td>
</tr>
<tr>
<td>Beers</td>
<td>NIR</td>
<td>30</td>
<td>14</td>
<td>700</td>
<td>2</td>
</tr>
<tr>
<td>Beers</td>
<td>Raman</td>
<td>30</td>
<td>15</td>
<td>3453</td>
<td>2</td>
</tr>
<tr>
<td>Tablets</td>
<td>NIR</td>
<td>211</td>
<td>99</td>
<td>407</td>
<td>4</td>
</tr>
<tr>
<td>Tablets</td>
<td>Raman</td>
<td>82</td>
<td>38</td>
<td>3402</td>
<td>4</td>
</tr>
<tr>
<td>Wines</td>
<td>FTIR</td>
<td>30</td>
<td>14</td>
<td>842</td>
<td>4</td>
</tr>
<tr>
<td>Coffees</td>
<td>FTIR</td>
<td>38</td>
<td>18</td>
<td>278</td>
<td>2</td>
</tr>
<tr>
<td>Olive Oils</td>
<td>FTIR</td>
<td>82</td>
<td>38</td>
<td>562</td>
<td>4</td>
</tr>
<tr>
<td>Juices</td>
<td>FTIR</td>
<td>666</td>
<td>317</td>
<td>235</td>
<td>2</td>
</tr>
<tr>
<td>Meats</td>
<td>FTIR</td>
<td>90</td>
<td>30</td>
<td>448</td>
<td>3</td>
</tr>
</tbody>
</table>
high variance and high correlation with the response. This representation is then used to fit a set of \( n \) responses to all samples. This contrasts with Principal Component Regression (PCR), which only seeks for latent variables that maximize the variance criterion. As stated in Ref. [35] PLS is helpful and effective when the within-group variability of the samples dominates the between-group variability and when variable reduction is needed. This is the case for many chemometric datasets, since they often contain a lot of variables and a relatively small number of samples.

3.2. Logistic regression

Logistic Regression can be seen as the last layer of a CNN: in other words, it is a very simple neural network where there are no hidden layers, therefore the input goes directly to the output layer [27]. Thus, comparing CNN against Logistic Regression with L2 regularization allows us to directly investigate the importance of the convolutional layer. For classification the objective function consists of a Cross-entropy Error Loss and an L2 regularization term for the weights:

\[
OBJ(w) = \frac{1}{N} \sum_{n=1}^{N} \left[ y_n \log \hat{y}_n + (1 - y_n) \log (1 - \hat{y}_n) \right] + \lambda_1 \cdot ||w||^2
\]

where \( \hat{y}_n = \Psi(w \cdot x_n) \), \( \Psi(\cdot) \) is the activation function, \( x_n \) is the \( n \)-th sample, \( w \) are the output weights and \( y_n \) is the target label. In other words, this is a simple network to train (due to the few parameters to tune) and so it gives a baseline which CNN can be compared to. The hyperparameters of the network to learn with RGS-CV are:

- \( \lambda_1 \) for the regularization term: \( 10^9 \) where \( n \in [-3.3] \),
- learning rate: \( 10^9 \) where \( n \in [-8,-1] \).

SGD was used to learn the weights of the network.

3.2.3. k-Nearest Neighbor

k-Nearest Neighbor (kNN) is a very simple and well-known classification method. The way new samples are classified is by selecting the most frequent class belonging to their \( k \) closest neighbors in the feature space. The metrics we used for evaluating the distance between samples are the Manhattan and the Euclidean distance. We considered a number of neighbors \( k \in [3,10] \). As before, the best value for \( k \) in combination with the best metric are selected using RGS-CV.

3.3. Spectral preprocessing

Spectral data often contains multiple data artifacts. These data artifacts represent variation in the data that is unrelated to the sample under study. For vibrational spectroscopic data, the most common data artifacts are a baseline, light scatter effects and instrumental noise [7]. Since this variation is unrelated to the response variable, it is often removed from the data by data preprocessing before the actual data analysis. In this work, we apply a sequential way of data preprocessing, similar to [7], in which a selection of preprocessing methods is consecutively applied to the data. This selection of consecutively applied preprocessing methods is called a preprocessing strategy. Our preprocessing strategy consists of four preprocessing steps, each associated with a number of possible methods:

1. **Baseline correction** to deal with spectra with sloped or varying baselines:
   - Polynomial detrending (PolDetr) [36], (polynomial order \( \in [2,4] \)),
   - Asymmetric Least Squares baseline estimation (AsLS) [37],
   - 1st/2nd derivative (deriv1/deriv2).

2. **Scatter correction** to deal with different light scatter effects:
   - Mean/Median/Max Scaling (Mean/Median/MaxScaling)
   - L2 Norm (L2),
   - Standard Normal Variate transform (SNV) [36],
   - Robust Normal Variate transform (RNV) [38],
   - MSC [39].

3. **Noise removal** to smooth the spectra: Savitsky-Golay (SavGol) [40] (polynomial order polynomial order \( \in [2,4] \) and window width in \( \in [5,9,11] \) px).

4. **Scaling** [41]:
   - Mean centering (MeanCent),
   - Auto/Level/Logarithm/Range/Pareto/Poisson Scaling (Auto/Lev/Range/Pareto/PoissonScaling),
   - Logarithm Transformation (LogTrans).

These methods are briefly described in the Supplementary Material. Note that the order of methods is fixed to the order presented here, since this is the most commonly applied sequence of preprocessing methods [7]. Meancentering is always applied to the data.

The optimal combination of preprocessing methods for a dataset may depend on the type of classification model considered. Therefore, for each dataset and method, we obtained the best preprocessing strategy as previously done e.g. in Ref. [7]. All combinations of the methods from the above mentioned families were considered (e.g. 7 \( \times \) 10 \( \times \) 10 \( \times \) 7 = 4900 combinations) and the one achieving the highest 10-fold cross-validation accuracy on the training data was selected. The resulting preprocessing strategies for CNN are summarized in Table 2. The preprocessing strategies for PLS-LDA and Logistic Regression are reported in the Supplementary Material.

4. Results

Table 3 contains the accuracies of the classification methods on the test set: for each method, the model that achieved the highest accuracy using cross-validation was selected. A description of the model used for each dataset is included in the Supplementary Material.

The reported results refer to application of classification methods on preprocessed data (pre) and on the original data (raw).
The preprocessing strategy adopted for each combination of method and dataset was obtained using the approach described in 3.3.

### 4.1. Results for raw data

On the non-preprocessed data, CNNVS achieves a significantly better accuracy than PLS-LDA (p-value < 0.001) with a 24% average improvement. It also improves over Logistic Regression (p-value < 0.001) with a 6% average improvement. CNNVS always outperform Logistic Regression except on the Raman Tablets dataset. Looking at differences of this dataset with respect to others, no big discrepancies were noticed in the learning curve, learned convolutional kernels or original spectra (in this case compared with other Raman spectra).

### 4.2. Results for preprocessed data

Results on the preprocessed data show a boost in performance of PLS-LDA with a 27% average increase in accuracy compared to non-preprocessed data. Also in this case CNNVS achieves significantly better accuracy than PLS-LDA (p-value < 0.001) and Logistic Regression (p-value < 0.001). The average accuracy of PLS-LDA remains lower than that of CNNVS (7% lower). In most of the datasets that we considered, there is a preprocessing strategy that makes CNNVS better than PLS-LDA. In fact, any method that uses an optimized preprocessing strategy significantly outperforms all methods that do not use preprocessing. In other words, preprocessing is really a valuable addition for data analysis, irrespective of the data analysis method used.

### 4.3. Discussion

Overall, the results indicate that CNNVS is less dependent on preprocessing than PLS-LDA. In order to further substantiate this result, we compared the accuracies of many different preprocessing strategies for both CNN and PLS-LDA using a histogram plot. The results of this on the Beers FTIR dataset are shown in Fig. 4. The histogram plot shows that the accuracy on the raw data of PLS-LDA is quite distant from the best possible accuracy achieved with preprocessing, while this difference is smaller for CNN. Moreover, the range of accuracy values achieved by CNN is smaller than PLS-LDA. Thus this indicates that CNN is less dependent on preprocessing than PLS-LDA. Appropriate preprocessing, however, still is able to improve the accuracy of a CNN model. For Logistic Regression and CNN, preprocessing is most of the time counterproductive (i.e. leading to a lower accuracy than the raw data). This indicates that selection of appropriate preprocessing is an important step in data analysis [11]. Similar trends are observed in the other datasets (see Supplementary Material), kNN performs reasonably well on raw data while it does not seem to benefit from ad-hoc preprocessing.

We also looked at the learned kernels by CNNVS to possibly relate them with well-known preprocessing methods. We did not perform a comprehensive investigation, but it is interesting to give at least an example of interpretation of learned kernels. For instance, those learned by CNNVS for the FTIR beer non-preprocessed dataset (see Fig. 5) have a large number of non-zero elements, showing that they are performing a kind of smoothing, while their linear trend indicates that they are performing a forward and a backward-derivative kernel.

Our model selection method chose two kernels for the convolutional layer to achieve the highest accuracy based on cross-validation. Note that these two kernels are not redundant, because of the rectifying non-linearity after the convolutional layer. Hence, the first kernel outputs non-zero values when the intensity goes up, while the second kernel outputs non-zero values when the intensity goes down.

If we consider the kernels learned using preprocessed data, the situation is not as clear, and CNNVS seems to have learned kernels corresponding to higher derivatives as well. Our model selection method chose four kernels for the convolutional layer instead of two that have been chosen for the non preprocessed data. This suggests that the use of optimally designed preprocessing allows CNNVS to use more kernels without overfitting and consequently achieve higher accuracy.

The other dataset kernels learned by CNNVS show a similar behavior with few exceptions, indicating that a smoothing-derivative kernel is needed for the examined non preprocessed data. Plots of these kernels for all the datasets are included in the Supplementary Material.

### 5. Important spectral regions

In chemometric data analysis, model interpretation may be a very important aspect. Therefore, it is crucial to identify important spectral regions corresponding to discriminatory chemical properties of the sample, like the presence of a substance or its concentration. In the following sections, we will compare important regions detected by CNNVSs with and without optimal preprocessing of the input spectra.

For the FTIR Beers and the Tablets datasets information about some important spectral regions was reported in Refs. [28] and [30], respectively. Therefore we focus on these datasets and investigate the use of CNNVSs to find important spectral regions.

---

**Table 3**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Scan</th>
<th>Raw data</th>
<th>Preprocessed data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CNNVS</td>
<td>PLS-LDA</td>
</tr>
<tr>
<td>Beers</td>
<td>FTIR</td>
<td>77</td>
<td>55</td>
</tr>
<tr>
<td>Beers</td>
<td>NIR</td>
<td>79</td>
<td>44</td>
</tr>
<tr>
<td>Beers</td>
<td>Raman</td>
<td>91</td>
<td>67</td>
</tr>
<tr>
<td>Tablets</td>
<td>NIR</td>
<td>94</td>
<td>61</td>
</tr>
<tr>
<td>Tablets</td>
<td>Raman</td>
<td>76</td>
<td>39</td>
</tr>
<tr>
<td>Wines</td>
<td>FTIR</td>
<td>50</td>
<td>21</td>
</tr>
<tr>
<td>Coffees</td>
<td>FTIR</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Olive Oils</td>
<td>FTIR</td>
<td>100</td>
<td>83</td>
</tr>
<tr>
<td>Juices</td>
<td>FTIR</td>
<td>97</td>
<td>69</td>
</tr>
<tr>
<td>Meats</td>
<td>FTIR</td>
<td>100</td>
<td>77</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td>86</td>
<td>62</td>
</tr>
</tbody>
</table>
5.1. FTIR beers dataset

In Ref. [28] was reported that the most important spectral regions are between 1000 and 1200 cm⁻¹.

Application of CNNVSfs to the non-preprocessed dataset identifies two adjacent spectral regions between 1000 and 1200 cm⁻¹ (see Fig. 6). Application of CNNVSfs to the preprocessed dataset instead identifies two spectral regions, including that between 1000 and 1200 cm⁻¹. Therefore both CNNVSfs trained using preprocessed or non-preprocessed input spectra highlight a part of the important region.

5.2. Tablets dataset

On the non-preprocessed NIR Tablets dataset CNNVSfs identifies different spectral regions at wavenumbers around 7700, 8850, 1000 and 1200 cm⁻¹. Therefore both CNNVSfs trained using preprocessed or non-preprocessed input spectra highlight a part of the important region.

Fig. 4. % Accuracy achieved by CNNVS (top), PLS-LDA (center) and Logistic Regression (bottom) using different combinations of preprocessing methods. For each accuracy value (x-axis) the percentage of preprocessing combinations yielding that value is shown (y-axis). The vertical red line (dashed) represents the accuracy achieved by each method using raw data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 5. CNN learned kernels for the Beers FTIR dataset. The plots show the value (y-axis) for each element of the learned kernels (x-axis).

Fig. 6. A sample spectrum of the FTIR Beers dataset and important spectral regions detected by CNNVSfs on this dataset (shaded areas) using optimal preprocessing (left) or not using preprocessing (right). The known important regions are found around the vertical dashed lines.
9500 and 10500 cm\(^{-1}\) (see Fig. 7). The Tablets data was artificially generated using a known active substance which characterizes important spectral regions of the data\[^{30}\]. In particular the region around wavenumber 8830 cm\(^{-1}\) contains a peak for the known active substance. This peak is also selected by CNNVSfs on the preprocessed input spectra. Therefore even for this dataset, both CNNVSfs trained using preprocessed or non-preprocessed input spectra highlight the important region.

In the Raman Tablets dataset important regions stemming from the presence of the active substance include a peak at wavenumber 2233 cm\(^{-1}\) which originates from the cyanide (\(\text{C} \equiv \text{N}\)) group in the active substance, and several other peaks from the active substance can be seen in the tablet spectrum (e.g. at wavenumbers 1614 and 3075 cm\(^{-1}\)). On this dataset the PLS classification method used by Ref.\[^{30}\] achieved a rather low accuracy. Therefore the authors did not further investigate the important spectral regions identified by the method.

On the non-preprocessed dataset CNNVSfs selects spectral regions between wavenumbers around 350, 1700 and 3100 cm\(^{-1}\) (see Fig. 8). Thus some peaks from the active substance are identified, but the main peak at wavenumber 2233 cm\(^{-1}\) is not considered important by CNNVSfs. On the preprocessed dataset CNNVSfs instead selects spectral regions between wavenumbers around 2233 (main peak of the active substance) and 1900 cm\(^{-1}\). So in this case, the optimal preprocessing of the input spectra is responsible for a change of the identified important regions.

6. Conclusion

We designed a simple CNN-based classification method for vibrational spectroscopic data, called CNNVS, and showed how it could be easily enhanced in order to select important spectral regions. Application of CNNVS to data which was not preprocessed as well as to preprocessed data yielded superior accuracy performance compared to PLS-LDA and Logistic Regression. Results indicated that CNNVS is less dependent on preprocessing than the standard PLS-LDA method for vibrational spectroscopic data classification and achieves excellent performance on preprocessed data. More vibrational spectroscopy classification datasets are needed to see if these strong results hold also in different settings.

In general, vibrational spectroscopic data have relatively few samples and many features. A technique to increase the size of the data for improving accuracy performance of the model is data augmentation, which amounts to add perturbed samples. This technique is widely used in image analysis using CNN. However, on vibrational spectroscopic data, it did not have a beneficial effect. Preliminary experiments indicate that this may be due to difficulty of modeling the noise in this context as shown by the different combinations of preprocessing strategies needed for different data.

The kernels that are learned by CNN perform a kind of smoothing and derivative filtering. This would mean that such filters are not be needed as separate preprocessing steps, simplifying the selection of an optimal preprocessing method. It would be interesting to see if CNNVS can indeed always achieve similar accuracy with such a simpler preprocessing selection.

The present investigation considered the classification of vibrational spectral data. It would be interesting to extend the investigation to compositional analysis of vibrational spectroscopic data, which could be performed using regression.

Appendix A. Supplementary data

Supplementary data related to this article can be found at http://dx.doi.org/10.1016/j.aca.2016.12.010.

References

transform IR spectroscopy to analyze biological materials, Nat. Protoc. 9 (2014) 1771–1791.


B. Hanson, Chemospec: an r package for chemometric analysis of spectroscopic data and chromatograms, Package Version 2 (2014).


