Chemometrics and Intelligent Laboratory Systems 215 (2021) 104354

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

An automated deep learning pipeline based on advanced optimisations for leveraging spectral classification modelling

Dário Passos [a], Puneet Mishra [b,*]

[a] CEOT, Physics Department, Universidade do Algarve, Campus de Gambelas, FCT Ed.2, 8005-189, Faro, Portugal
[b] Wageningen Food and Biobased Research, Bornse Weilanden 9, P.O. Box 17, 6700AA, Wageningen, the Netherlands

ARTICLE INFO

Keywords:
artificial Intelligence
Spectroscopy
Phenotyping
Crop

ABSTRACT

In deep learning (DL) modelling for spectral data, a major challenge is related to the choice of DL network architecture and the selection of the best hyperparameters. Often, slight changes to the neural architecture or its hyperparameter can have a direct influence on the model's performance, making its robustness questionable. To deal with it, this study presents an automated deep learning modelling based on advanced optimisation techniques involving Hyperband and Bayesian optimisation, to automatically find optimal neural architecture and its hyperparameters to reach robust DL models. The optimisation requires a base neural architecture to be initialized, however, later it automatically adjusts the neural architecture and the hyperparameters to reach the optimal model. Furthermore, to support the interpretation of the DL models, a wavelength weighing schema based on gradient-weighted class activation mapping (Grad-CAM) was implemented. The potential of the approach was showed on a real case of wheat variety classification with near-infrared spectral data. The performance of the classification was compared with that previously reported on the same dataset with different DL and chemometric approaches. The results showed that with the proposed approach a classification accuracy of 94.9% was reached, which was better than the best reported accuracy on the same data set i.e., 93%. Furthermore, the better performance was obtained with a simpler neural architecture compared to what was used in earlier studies. The automated deep learning based on advanced optimisation can support DL modelling of spectral data.

1. Introduction

Classification modelling with near-infrared (NIR) spectral data is widely performed for a non-destructive and rapid identification and assignment of samples to its belonging class. For example, NIR spectroscopy has been widely explored for classification of food products such as teas [1,2], classification of bacterial pathogen strains [3], micro-plastic substrates [4], gasoline [5], pharmaceutical tablets [6], vegetable oils [7] and many more [8–11]. NIR spectroscopy allows such a classification as it can capture both the physical and chemical properties which differ between samples belonging to different classes. Although, sometimes the difference is in the physical properties, appearing as differences in the scattering information in the spectra, other times, the difference is in the chemical composition which appears as absorption peaks in the spectra. In most real-life cases, both the physical and chemical informations are mixed in the spectra as scattering and absorption. Due to such a complex mixture of information often data modelling approaches are used to model the NIR spectra.

Classification modelling for NIR spectral data can be purely chemometrics based, such as, partial least-square discriminant analysis (PLS-DA) [12] and soft independent modelling of class analogies (SIMCA) [13], while others are traditional machine learning (ML) approaches such as support vector machines (SVM) and logistic regressions [14]. A key point to note is that a main distinction between chemometrics and traditional ML approaches is the latent space modelling involved in the chemometric approaches such as PLS-DA and SIMCA. Further, based on the classification case, NIR spectral data classification can be performed either for one class [15] or multiple classes simultaneously [2].

In recent years, due to fast progress in the domains of artificial intelligence (AI) and deep neural networks (DNNs), deep learning (DL) methodologies have been slowly diffusing into the realm of chemometrics supplying a large potential to model NIR spectral data [16,17]. For predictive modelling i.e., regression, DL has already outperformed traditional chemometrics and classic machine learning approaches [18–20]. Although new algorithms are appearing in the literature every day, currently, the state of the art for spectral data modelling can be

* Corresponding author.
E-mail addresses: dmpassos@ualg.pt (D. Passos), puneet.mishra@wur.nl (P. Mishra).

https://doi.org/10.1016/j.chemolab.2021.104354
Received 21 April 2021; Received in revised form 26 May 2021; Accepted 27 May 2021
Available online 4 June 2021

subdivided into two approaches. The first is the use of deep autoencoders to extract complex features from the spectral data, which can then be combined with either a neural network or a traditional machine learning approach such as SVM for predictive modelling [18,19]. The second approach involves the use of Convolutional Neural Networks (CNNs) architectures for joint feature extraction and predictive modelling [17,21]. The application of DL for spectral classification is also increasing and DL has already shown to outperform traditional chemometric methods such as PLS-DA and ML methods such as SVM [22,23].

In classical NIR spectral data modelling, the pre-processing of spectral data is widely performed [24–26]. The pre-processing of the spectra allows to eliminate/reduce unwanted variability from the data (noisy bands, light contamination, scattering effects, etc.) that can have detrimental effects in terms of the model performance [27,28]. For example, when the aim is to predict chemical components by predictive modelling, it is widely recommended to eliminate/reduce the scattering information from the data and use the absorption information related directly to the overtones of chemical bonds [27]. However, when the aim is to predict physical properties, the recommendation is to use the raw spectra that is rich in scattering information which is related to the physical characteristics of the samples [27]. Several pre-processing methods such as smoothing, spectral normalisation, scatter corrections and derivatives, are therefore, used to improve/enhance the model predictive power [28]. Since not all pre-processing works in the same way, i.e. they carry complementary information, that can be combined differently to improve model performance [27]. However, the positive effects of pre-processing are dependent on the case and one of the main challenges researchers face is the arduous exploration and identification of the best pre-processing from a wide set of methods [25,26]. To alleviate this burden, ensemble pre-processing approaches such as sequential [29,30] and parallel pre-processing [31] through orthogonalization are becoming increasingly popular in chemometrics. The ensemble approaches have shown that different pre-processing indeed carry complementary information and allows for improved prediction of both chemical and physical properties [27,32–35]. These works suggest that future chemometric modelling can benefit from using a pre-processing ensemble approach for spectral modelling [27]. This can be particularly helpful in the case of certain DL models because their ability to non-linearly combine distinctive features can be enhanced by the ensemble approach [17]. On the other hand, in the face of a DL model optimisation, the gains in analysis simplification by the simultaneous use of several pre-processed spectra can be diluted due to the high number of hyperparameters that this type of model requires. For DL models that require large training times, hyperparameters optimisation using grid search or random search approaches can be very computationally expensive [47,56]. Fortunately, parallelization [57] and Bayesian optimisation algorithms [43] can speed up the process of finding the correct Neural Architecture (NA) for the problem at hand and optimise its hyperparameters [42]. This automated way of implementing automated DL modelling has already given signs of being an extremely useful tool for future research [44,52] and could become a potential tool for automated DL modelling of spectral data.

One of the most appreciated features that classical chemometric techniques displays is the ability to interpret most of the chemometric models in terms of spectral responses allowing for researchers to understand better what spectral bands are more relevant to the quantity being measured. For many non-linear machine learning (ML) models such as SVM (using rbf kernels) and CNNs, the understanding of how the model behaves the way it does is reduced, i.e., in terms of interpretability, these models are not as simple to understand as classical (linear) chemometric models. Nonetheless, in the specific case of CNNs used for classification, multiple techniques that allow to understand where the network focus its attention to make a classification are already available [58–61]. These techniques can be used to expand the benefits of the improved performance of DL algorithms by supplying information about the spectral bands that are relevant the model.

In this study, an automated data analysis pipeline for spectral classification modelling that combines chemometric pre-processing ensemble and advanced DL modelling is proposed. The pipeline leverages several advances in DL model optimisation to automatically optimise the DL models to produce highly robust 1D-CNN classifier and reducing the optimisation time. Model interpretability is also included by assessing spectral bands contribution towards the classification process. The potential of the approach was showed on a real case of wheat variety classification with NIR spectral data. The performance of the classification was compared with previously reported results on the same dataset with different DL, chemometrics and ML approaches. A direct comparison with an earlier study was possible as the already pre-partitioned data i.e., training, validation, and test set, is available as open-access to the scientific community.

2. Materials and methods

2.1. Data set

The data set used in this study includes of 147,096 wheat kernels mean NIR spectra, measured on 30 varieties of wheat kernels, harvested in 2019 and stored under the same conditions after harvest i.e., dried and packed in woven plastic bags [22]. The wheat kernels come from the wheat plants grown in the same fields. More detailed description of the samples variety and growing condition can be found in [22]. To have sufficient data points for DL modelling, many kernels were measured from each variety. According to [22], the NIR data of the kernels were extracted from hyperspectral (HS) images running in spectral range of 874 to 1734 nm, and later, by selecting a region of interest at the center of the kernels, the average spectra were estimated. HS imaging was performed with a NIR-HSI system having an imaging spectrograph (ImSpector N17E; Spectral Imaging Ltd., Oulu, Finland), a high-performance camera (Xeva 992; Xenics Infrared Solutions, Leuven, Belgium) with 326 × 256 (spatial × spectral) pixels and a camera lens (OLES22; Specim, Spectral Imaging Ltd., Oulu, Finland). In the final data set, the average spectra were the same as the total number kernels. During the scan, the kernels were placed in a plate (negligible reflectance) and, using a step motor, the scan lines were recorded. To have balanced samples for training and validation for each variety, 2400 kernels were used for training, 800 kernels for validation, and the rest were used for independent testing of the model. According to [22], due to the high noise level in extreme bands the spectral range was reduced to 975–1645 nm with 200 sampling bands. The original data used in this study can be found in the supplementary material of [22]. A summary of calibration, validation and test set are shown in Fig. 1. It can be noted (in Fig. 1) that the calibration and validation sets have balanced samples for all 30 wheat varieties and do not require any pre-processing for class balancing.

2.2. Data augmentation with chemometric pre-processing ensemble

In the earlier work from where the data set was collected [22], DL modelling was performed using only the averaged reflectance spectra as input. However, recently, a new study [17] has shown that reflectance spectra stacked with several pre-processed versions to augment the input features space, can be highly beneficial for DL modelling. Therefore, following the approach described in [17], the reflectance spectra of wheat kernels were pre-processed using several pre-processing methods and stacked together as explained in Fig. 2 (see also Fig. 3).

The input vectors were therefore augmented from 200 features to 1200 features. In this study, only SNV [37] and Savitzky-Golay (SAVGOL) [38] derivatives were used as they were recommended in an earlier study for being independent of any external reference spectra or weights estimation as needed to other techniques like multiplicative scatter correction [39] or variable sorting for normalisation [40]. The window size and polynomial order for the SAVGOL filter were fixed to a 13-points window and 2nd degree, respectively. Although window size and

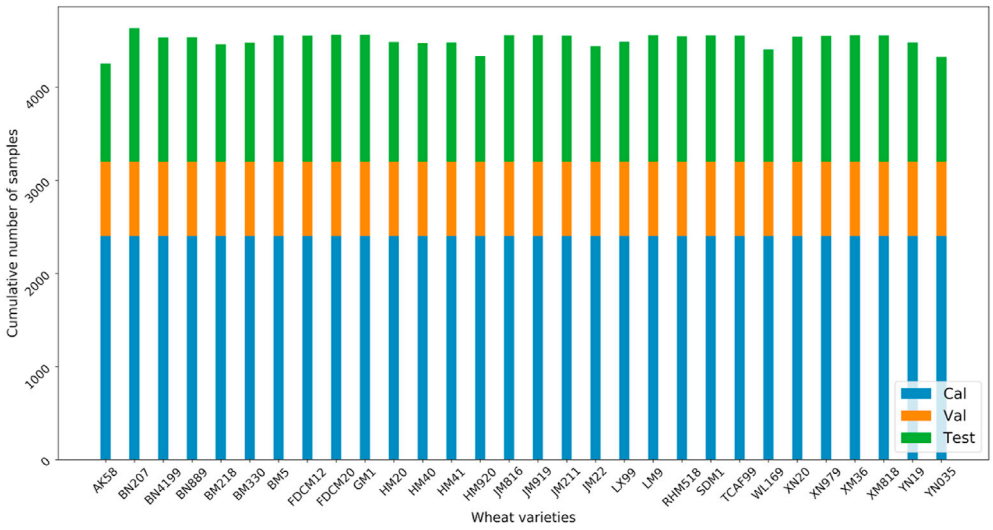

Fig. 1. A summary of samples in calibration, validation, and test set for different wheat varieties available in the data set. The classes for calibration and validation sets are balanced.

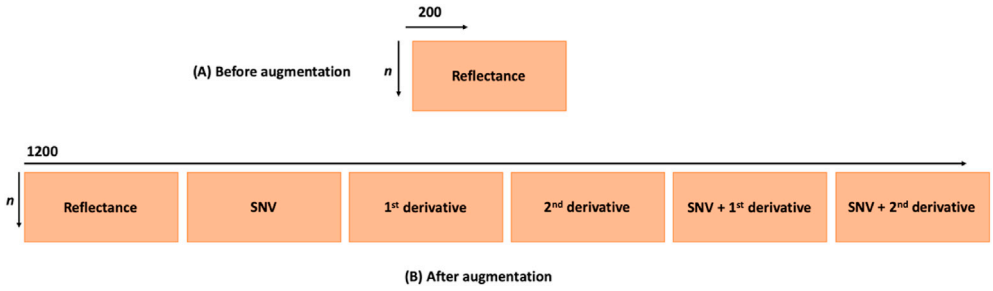

Fig. 2. The spectral data augmentation approach. (A) Reflectance, and (B) reflectance data augmented with different pre-processing methods.

polynomial order can also be optimised, in this study, to decrease the computational load, only the previously mentioned parameters were used. Prior to the DL modelling the spectra were standardized column-wise to remove differences between the signal amplitudes obtained after different pre-processing's and put them in a value range required by the DL model. The labels of the wheat classes where one-hot encoded from their original numerical format {0, 1, 2, …, 29} prior to use in the DL model.

2.3. Initial architecture of the deep learning model

DL model implementation and optimizations were done using the Python (3.6) language and the open-source deep learning framework TensorFlow/Keras (2.4.0) [65], running on a desktop workstation equipped with a NVidia GPU (GeForce RTX 2080 Ti), an Intel® Core i7 4770k @3.5 GHz and 16 GB RAM, running Microsoft Windows 10 OS.

To classify the same data set, in earlier studies [22], the authors adopted a CNN architecture composed of an attention block, 3 conv. layers and 3 fully connected (FC) layers for classification of the wheat data. In this study, a simpler neural architecture is chosen as base model and then automatically evolved through Neural Architecture (NA) search. The base model adopted is a 1D convolutional neural network (1D-CNN) presented in [21] and slightly adapted for this classification problem. There were two main reasons for using the already available model as the starting point in the present analysis. The first reason was that NAS optimisation if more efficient if it starts from a template model, compared to randomly probe subsets of combinations of multiple types of layers. The template model serves as a base structure for the NAS to organize the layers. That being said, it would be possible to create a pool of different types of layers (conv1d [with 1 or multiple kernels], max-pooling, mean-pooling, etc.) and build a "novel" network architecture from scratch. However, that would require a lot of more computational resources than the ones presently at our disposal. The second reason to choose the neural architecture presented in [21] as template model is because such a simple 1D-CNN architecture has shown promising results for spectral data modelling [16,17,21,36]. The template model was composed of just 1 conv. layer with 1 filter (size = 5) and stride 1, followed by 3 FC layers (with 512, 256 and 128 units) and a final output layer with 30 units, to account for the 30 classes in the data. In the last layer a "softmax" activation function was used, and for all the other layers, exponential linear units "elu" were used. The weights in all the layers were initialized using the "He normal" initialization procedure [49], setting the random seed parameter to 42 to ensure better model reproducibility. Weight regularization in all layers is also implemented in the form of L2 regularization. Since this is as multi class problem, the categorical cross-entropy was used as the loss function and the model was trained using the adaptive moment optimiser algorithm (Adam) [50]. Table 1 supplies further details on the base CNN architecture which was automatically adjusted based on the optimisation as mentioned in the following sections.

Training DL models on large data sets can be computationally expensive and, therefore, one should implement strategies that optimise this process. In this work, Early Stopping was adopted during training to reduce overfitting problems and decrease training time (whenever possible). This algorithm watches the progress of the accuracy on a validation dataset during the training phase and interrupts training if this metric shows no improvement after a certain number of epochs. Another useful strategy to improve training speed, is to start the training with a

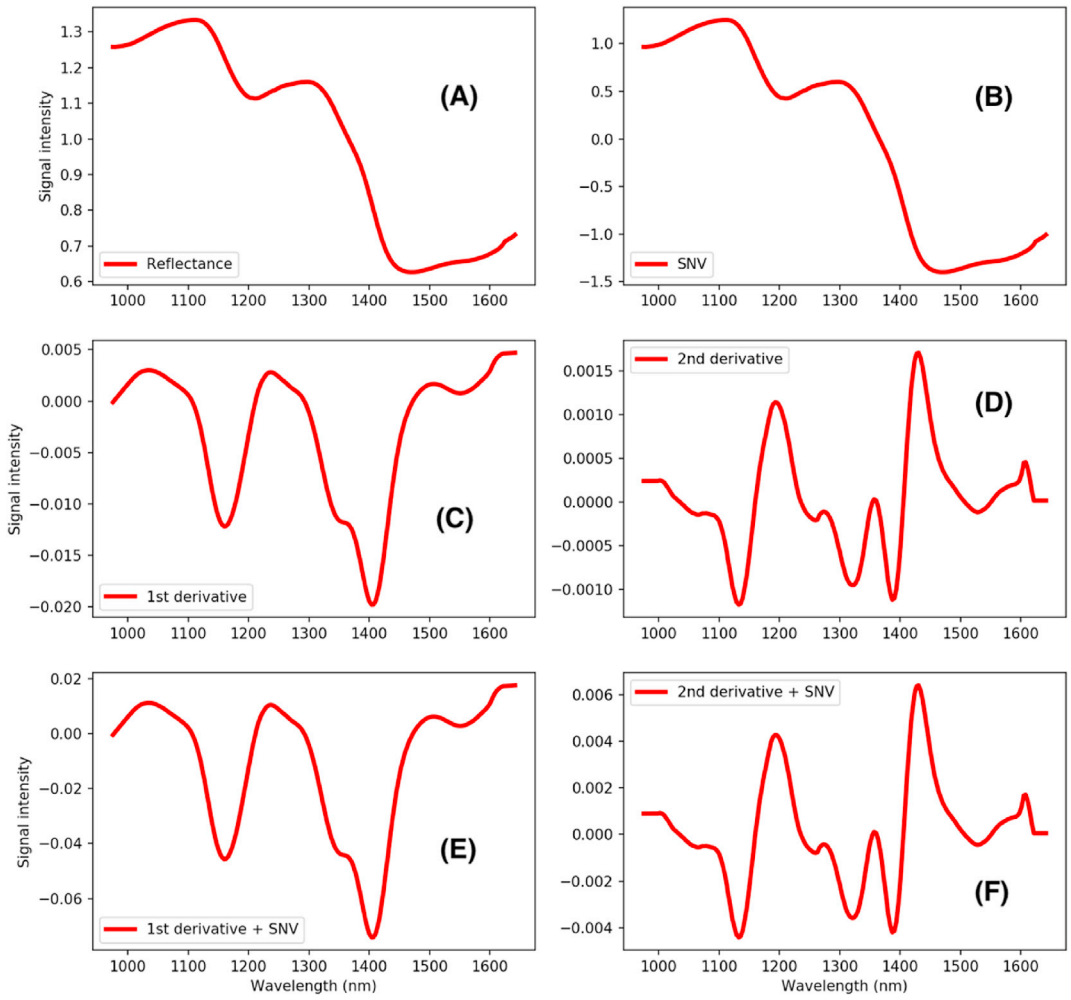

Fig. 3. A spectra of wheat kernel pre-processed with several pre-processing methods used for data augmentation.

Table 1
Intervals used for neural architecture (NA) and hyperparameters (HP) optimisation. Learning rate was the only hyperparameter optimised separately.

Name	Type	Interval/step	Base CNN
Number of FC layers	NA	[1–5]/1	3
Number of units p/FC layer	HP	[128–512]/2	[512, 256, 128]
Conv. filter size	HP	[3–20]/1	5
Number of Dropout layers	NA	[1–5]/1	0
Dropout rate p/Dropout layer	HP	[0–1]/0.005	0
L2 regularization β	HP	[0–0.003]/0.00001	0.003
Batch size	HP	[128–1024]/64	512
Learning rate*	HP	$[1 \times 10^{-8} - 0.1]$	

large learning rate (LR) in the gradient descent optimiser algorithm (Adam in this case) and progressively decrease it, allowing gradient descent to smoothly approach a minimum. This is done using the LR scheduler, ReduceLROnPleateau that reduces the LR by half (until a predefined minimum LR) whenever validation loss stops improving. To find the largest and minimum LR to explored in the optimisation, a LR range test first proposed by [53] was implemented. This test allows visualizing how the validation loss varies as a function of LR and helps find the range of LR that conduces to stable solutions. For this reason, the LR hyperparameter is optimised first and independently from the other hyperparameters.

In terms of NA search, the automatic optimisation pipeline explores the number of FC layers (not counting the last output layer, as it was fixed to 30 class output), the use of Dropout layers [62] between FC layers and the use of L2 regularization on all layers. On the hyperparameters side, optimisation accounts for: the convolution filter size, the number of units for each FC layer, the Dropout Rate (DR) for each Dropout layer, the strength of the global L2 regularization β, the training batch size and the Learning rate (LR). Table 1 shows the range of values probed during the optimisation process.

Dropout randomly zeros the connections weights between units in different layers during training enforcing network sparsity. Dropout and L2 regularization provide layer/weights regularization, stabilizing gradient descent algorithms and decreasing overfitting. The L2 regularization considered in this study is globally applied to all layers (Conv. and FC), i.e. there is no optimisation of β for each individual layer. In contrast, Dropout Rate (DR, the fraction of the units in a layer that are stochastically "dropped out") is optimised individually for each dropout layer. The number of dropout layers is optimised implicitly by allowing the search interval for the DR to include zero. For example, if the optimisation process finds a solution were DR in one of the dropout layers is zero then, effectively, that would be equivalent to not including that dropout layer in the first place.

2.4. Bayesian Optimisation strategy to automate neural architecture and hyperparameter optimisation: Tree-structured Parzen Estimators (TPE) and hyperband

For DL models to show their full potential, it is necessary to first choose/engineer the correct Neural Architecture (NA) that best adapts to the problem at hand and after that, carefully optimise its corresponding

hyperparameters **θ**. Commonly this is done in two separate steps, with the first one involving a considerable degree of experience and/or experimentation to find the right NA that best suits the user needs and after that, optimising its hyperparameters. In [42], the authors showed that performing joint NA and hyperparameter search can lead to better model performance when compared to the case where NA and hyperparameter optimisations are performed separately. Based on their recommendations and in [43,44], in this work, a combination of Bayesian Optimisation and Hyperband is used to perform joint automated optimisation of the DL model architecture and its hyperparameters. The idea is to start from a general base architecture (as explained in earlier section) and optimise it and its hyperparameters in an automated way.

Hyperband [45] is a bandit algorithm for hyperparameter optimisation that extends the capability of the Successive-Halving methodology [46]. In broad lines, Hyperband optimises hyperparameter search by considering a certain predefined computational budget B (e.g. CPU time, number of training epochs, number of iterations, etc.) and by dynamically allocating more resources to the most promising candidates by watching a predefined objective function f. For a deep learning classifier, such as the one presented in this study, the computational budget can be the maximum number of training epochs and the target objective function, the model's accuracy. In this case, this algorithm works like a variable Early Stopping strategy that interrupts training if the model with the hyperparameters being tested converges to a bad solution (low accuracy) and extends training if the model performance improves (high accuracy). In the context of this study, Hyperband executes the following steps: 1) given a predefined maximum number of training epochs ($B = 450$), dynamically allocates resources for several random hyperparameters configurations θ_k using Successive-Halving; 2) for each trial, select many configurations for Successive-Halving, run several of those for a small number of epochs and compute their accuracy; 3) increase the training epochs for the most promising ones and 4) complete a predefined fraction (1/4) of the best configurations. One of the downsides of Hyperband is that, since it uses something like a random search to select the hyperparameters configurations, there is no information passed from trial to trial that could be used to focus the optimisation process on areas of the hyperparameter space that show the best results. To overcome this drawback, a Bayesian Optimisation (BO) technique can be implemented in substitution of the random process that Hyperband uses for picking the next hyperparameter configuration.

In general terms, BO uses a probabilistic model $p(f|D)$ to model the objective function f given a vector of observed points $\boldsymbol{H} = \{(\boldsymbol{\theta}_0, y_0), \ldots, (\boldsymbol{\theta}_{i-1}, y_{i-1})\}$, composed by pairs of hyperparameters $\boldsymbol{\theta}$ and objective evaluations y. It is assumed that the evaluation of $f(\boldsymbol{\theta})$ is subjected to some noise/uncertainty ε and therefore, what is seen is $y(\boldsymbol{\theta}) = f(\boldsymbol{\theta}) + \varepsilon$. At each iteration, BO tries to maximize/minimize an acquisition function $a(\boldsymbol{\theta})$, a function that balances exploration and exploitation of the hyperparameter space to choose what points will be selected/"acquired" next. For example, a common choice for $a(\boldsymbol{\theta})$ is the expected improvement (EI) over the currently best observation $\alpha = \min\{y_0, \ldots, y_n\}$ After selecting a new point θ_{new}, based on the optimised acquisition function, BO evaluates $y_{new} = f(\boldsymbol{\theta}_{new}) + \varepsilon$ and updates this point to the vector of observed points $\boldsymbol{H}$.

In [43,44] the authors propose the use of a Tree-structured Parzen Estimator (TPE) [47], a BO method that uses Parzen window estimators (a.k.a kernel density estimators) to estimate the probability densities associated with good and bad hyperparameter configurations

$$l(\boldsymbol{\theta}) = p(y < \alpha | \boldsymbol{\theta}, D)$$

$$g(\boldsymbol{\theta}) = p(y > \alpha | \boldsymbol{\theta}, D)$$

over the input configuration space instead of modelling the objective function f directly by implementing $p(f|D)$. To select a new candidate $\boldsymbol{\theta}_{new}$ to evaluate, it maximizes/minimizes the ratio $l(\boldsymbol{\theta})/g(\boldsymbol{\theta})$ depending on the type of objective function. For example, considering that the objective function is the model's accuracy, for each trial, TPE looks for the hyperparameters that provide an evaluated accuracy y, higher than the best previously found value α, i.e., it works towards maximizing $g(\boldsymbol{\theta})$ and towards minimizing $l(\boldsymbol{\theta})$. Due to computational resources limitations, in this study, NA search is restricted to optimise the number of FC layers, the number of units per FC layer and the inclusion (or not) of weights regularization and dropout layers. The NA and hyperparameter optimisation pipeline (using TPE and Hyperband) was implemented using Optuna v.2.6 [48], an automatic hyperparameter optimisation software framework, designed for machine learning.

3. Results and discussion

3.1. Spectral profiles

A summary of reflectance and pre-processed spectra are shown in Fig. 3. Although the spectra were presented in separated plots in Fig. 1, however for deep learning, all signals were concatenated and used as single input vector. In the reflectance spectra peaks related to overtones of OH, CH and NH bonds can be identified [41]. For example, at 1200 nm the peak can be related to 2nd overtones of CH bond [41], at 1450 nm bands can be related to OH [41] overtones related to moisture in wheat kernels, at ~1500 nm can be related to the 2nd overtone of NH bond [41] which are abundant in food molecules such as protein. In relation to the effect of different pre-processing, a key point to note is that SNV (Fig. 3B) retained the same shape of the spectra as the reflectance, however, the intensity was normalised. Such an intensity normalisation with SNV allows to remove the additive and multiplicative effects from the NIR data which otherwise effects chemometric modelling [27,37]. The 1st derivative pre-processing (Fig. 3C) allowed revealing underlying peaks at several spectral locations such as 1150 nm, 1250 nm, 1400 nm and 1500 nm. The 2nd derivative pre-processing (Fig. 3D) further revealed peaks previously unresolved by 1st derivative pre-processing such as 1380 nm and 1600 nm. The SNV pre-processing over the 1st and 2nd derivative (Fig. 3E and F), did not revealed any new peaks, however, normalised the signal intensity. The SNV over derivatives was used as it is a common combination of pre-processing to correct the spectra of food for additive and multiplicative effects [29]. In this study, several underlying peaks revealed by derivatives, and the spectral normalisation attained with SNV, is expected to complement the reflectance data while DL modelling.

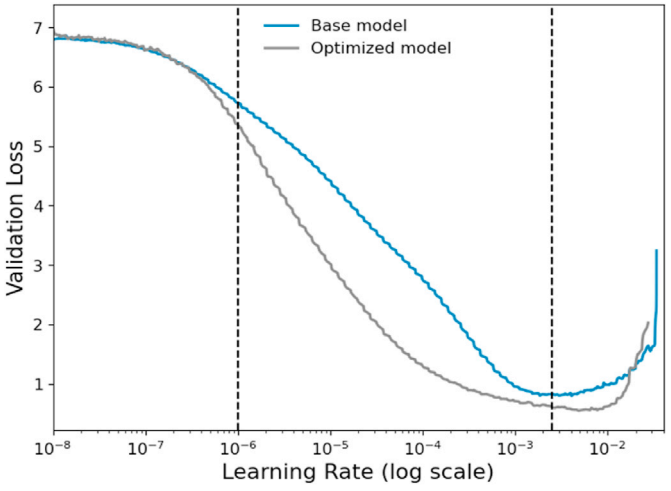

Fig. 4. Validation Loss as a function of learning rate. The dashed vertical lines mark the optima boundaries for LR to be used during training. The blue line was obtained with the base model and the grey line was obtained 'a posteriori' with the final CNN model, after the hyperparameters optimisation process. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

3.2. Deep learning model optimisation for automated neural architecture identification

The first hyperparameter to be optimised is the learning rate, LR. Since the automated optimisation pipeline involves probing a few thousand models, each of which is relatively expensive in terms of computation time, using a large LR during training accelerates the process. However, if the LR is too large the stochastic gradient descent algorithm might overshoot potential minima and stride around it never converging, or even diverging completely. For this reason, it is useful to use a LR scheduler that dynamically adapts the LR during training, hence the adoption of the 'ReduceLROnPlateau' strategy (as explained in M&M) is justified. To find an aggressive LR that allows for rapid initial convergence and still supplies stable solutions, the LR range test proposed by [53] was used. It was assumed, based on earlier experience, that the LR behaviour from the base CNN architecture behaves similarly to the final optimised model. This initial assumption was later confirmed after the optimal CNN was obtained. In Fig. 4, the blue curve shows the impact of the LR on the solutions obtained with the base CNN (without optimisation). The validation loss decreases steadily from around LR $= 1 \times 10^{-7}$ to around LR $= 2.5 \times 10^{-3}$, after which it starts increasing again. This interval where the validation loss decreases steadily with LR is the region of interest for this kind of test. The right most LR value was then chosen as the initial LR for the Adam optimisation algorithm, and LR $= 1 \times 10^{-6}$ is chosen as the minimum LR that ReduceLROnPlateau can schedule.

With the LR already set, the rest of the hyperparameters was obtained by the TPE + Hyperband optimisation pipeline automatically. The TPE algorithm was initialized with 50 trials of random search to ensure a uniform exploration of the hyperparameter space, and therefore, preventing that TPE focused just on the first local minima it founds. The optimisation study considered a total of 1000 trials, each of which had a maximum training budget of 450 epochs per model. The best model, that achieved an accuracy of ~95% on test set, was obtained for the NA depicted in Fig. 5 and the hyperparameters summarized in Table 2.

More details about the classification statistics for each class, the precision, recall and F1-score for the independent test set are provided as supplementary material.

3.3. Deep learning model optimisation for automated hyperparameters identification

In terms of computation time, the TPE + Hyperband method used here was compared with a standard random search over the same range of values. Over the predefined budget of 1000 models, TPE + Hyperband took 62 h to run, achieving an accuracy of 94.9% on the test set. In contrast, random search took 179 h and achieved an accuracy of 93.1% on the same test set. That corresponds to a factor of 3 in computation time while achieving an inferior performance. Of the main causes of this difference is the pruning capability of Hyperband that prunes/ends trials that perform poorly. Fig. 6 show the evolution of the accuracy achieved as a function of trial number. The most promising trial are coloured blue, the "best so far" trials are coloured red and the pruned trials are coloured grey. A key point to note is that by achieving an accuracy of 94.9%, this study outperforms the best-known classification accuracy of 93% on the wheat data set [22].

Fig. 7 illustrates how the optimisation using Bayesian Optimisation is processed. In the first trials (white coloured dots), the optimisation randomly samples all hyperparameter intervals and, progressively (bluer dots), the validation accuracy improves as the algorithm zooms in into a region of optimal values.

Moreover, the TPE implementation available in the Optuna package allowed to instantiate multivariate kernels over the hyperparameter

Table 2
Optimised neural architecture and hyperparameters obtained from the automated optimisation.

NA/Hyperparameters	Values for Optimised CNN
Number of intermediate FC layers	4
Number of units p/FC layer	[310, 456, 250, 282]
Conv. filter size	3
Number of Dropout layers	4
Dropout Rate p/Dropout layer	[0.035, 0.135, 0.405, 0.28]
L2 regularization β	3×10^{-5}
Batch size	832
Learning rate in Adam ()	$LR_{max} = 2.5 \times 10^{-3}$, $LR_{min} = 1 \times 10^{-6}$

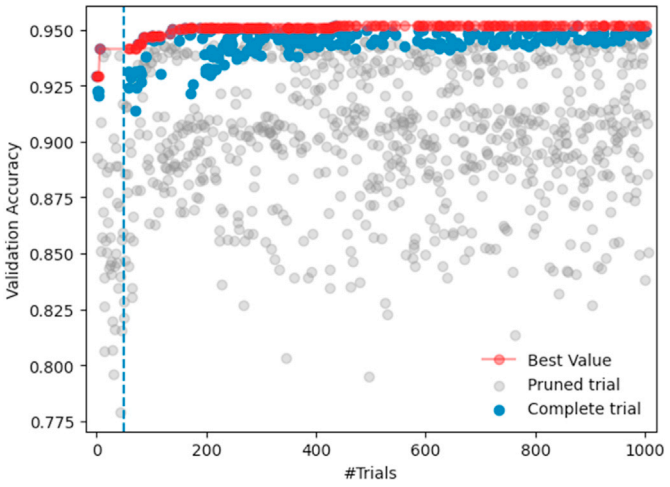

Fig. 6. Optimisation history of validation accuracy (×100 %) over 1000 trials. Red represent the evolution of the best values, blue the complete trials and in grey the unpromising trials that were terminated earlier (pruned) by the Hyperband algorithm. The dashed line marks the first 50 trials that were obtained using random search and that serve as starting point for TPE. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

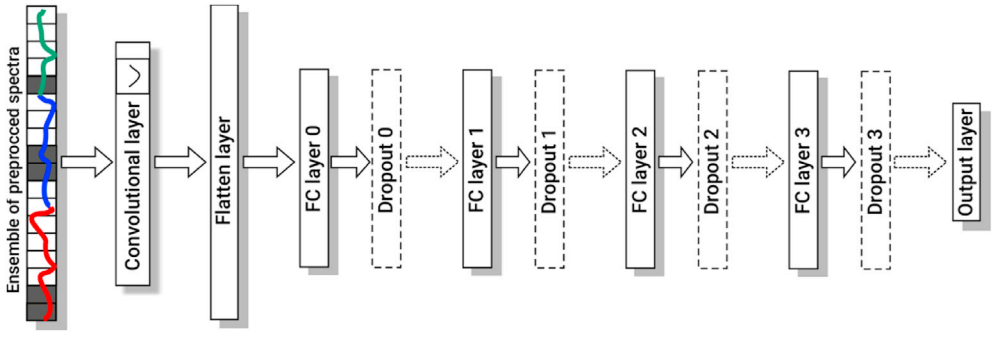

Fig. 5. The optimal neural architecture obtained with the automated neural architecture optimisation.

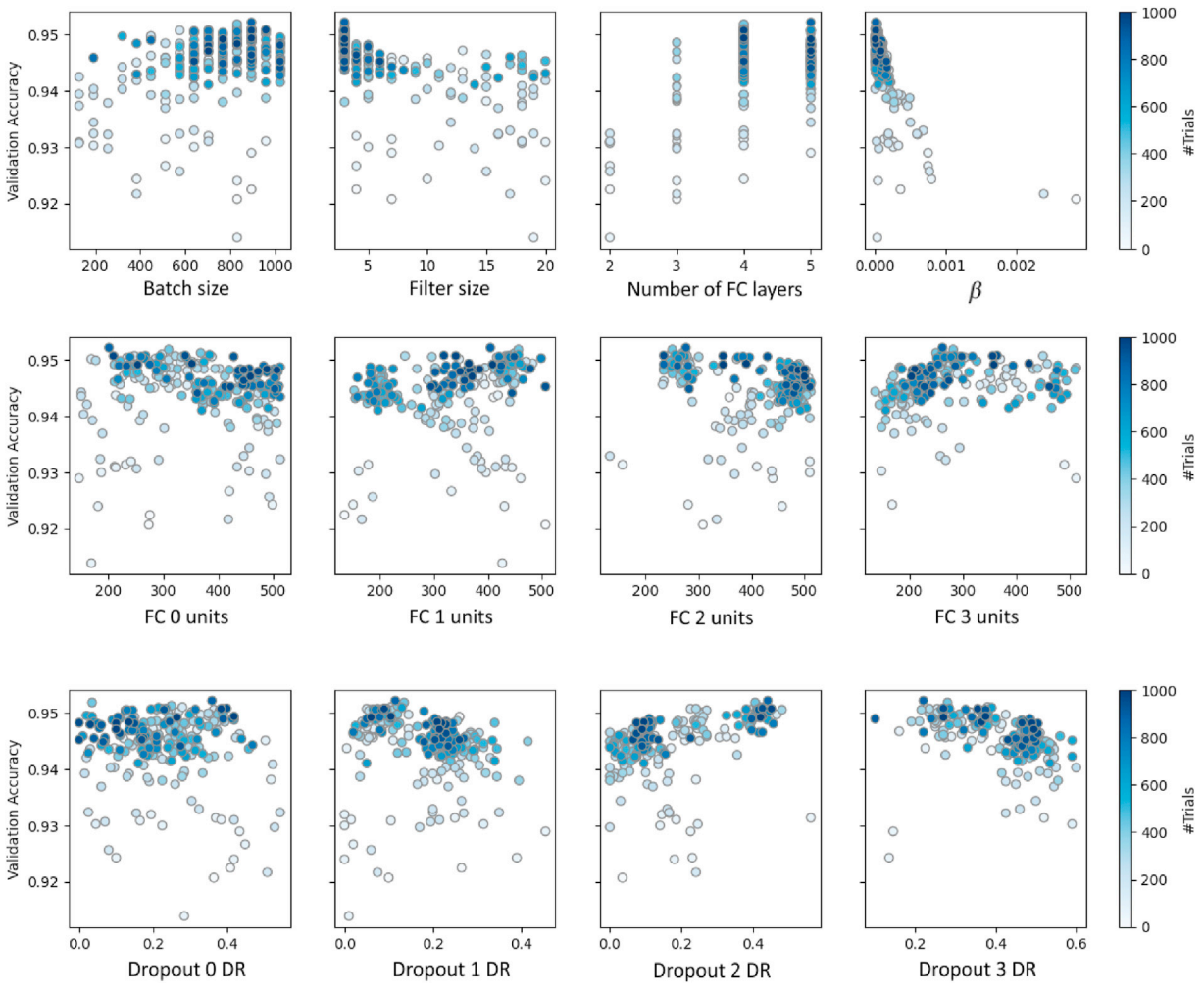

Fig. 7. Validation accuracy ($\times 100$ %) as a function of hyperparameters search space. The tone of blue shading shows the order of the trial. In the begging of the optimisation process (light blues), the tested hyperparameters returned low accuracies, but as the optimisation proceeded (darker blues), hyperparameters drifted towards higher accuracies. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

space. This means that the algorithm was able to find relations between different hyperparameters (e.g. dependencies or adversarial effects) and take advantage of those. In terms of the importance, for each hyperparameter during the optimisation process, one can estimate the contribution of each by using the fANOVA hyperparameter importance evaluation algorithm [51]. This algorithm fits a random forest regression model that predicts the objective value given a parameter configuration. Fig. 8 shows the relative importance of each hyperparameter for the optimisation. This can be helpful in the sense that it can serve as guide to engineer future models. However, one must be cautious with this kind of analysis because the range of values used for each hyperparameter should also be taken into consideration.

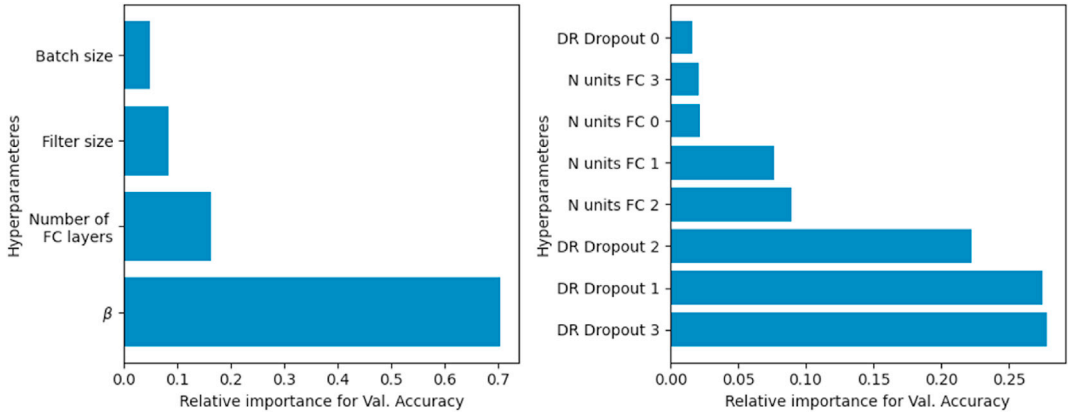

Fig. 8. Hyperparameters importance for the optimisation process was computed using the fANOVA algorithm [51] that uses random forest regression model to predict the objective value given by a hyperparameter configuration.

3.4. Model robustness and wavelength importance for classification

As it was mentioned earlier, dropout layers work by randomly shutting down connections between FC layers. This process is highly stochastic and produces small fluctuations in the results every time a model is trained. However, a robust model should be invariant to this type of fluctuation. A robust model should also perform similarly if trained with different configurations of the data and in the case where the initial weights of all units are different [63,64]. To assess the robustness of the optimised model, the mean accuracy over the following experiments was considered: 1) train 10 models from scratch (to probe the effect of dropout stochasticity); 2) train 10 models in 10 fold cross-validation (shuffling that data at each epoch) on an extended data set composed by pooling the original train and validation sets together (to also probe the effect of different data partitions); 3) train 10 models using different values of weight initialization each time (provided by changing the seed of the 'He Normal' initialization). Table 3 supplies a summary of the mean accuracy per experiment. All experiments provided the same overall level of performance with a slight decrease in experiment 3.

Due to their complexity and highly non-linear character, deep neural networks inner works are not easy to understand. Interpretability of these kind of algorithms is an active research area and some tools are starting to appear that can help experts interpret how these models work. One of these techniques, that can be adapted to 1D data like the spectra used in this study, is called Gradient-weighted Class Activation Mapping (grad-CAM) [54]. This technique uses the back propagation gradient information from the classifier layer into the last convolutional layer of the CNN to decide the importance of each feature for a particular class. For the CNN used here, this corresponds to tracking activity directly to the conv. layer in the model easing interpretation of the extracted features, i.e., there is no down sampling usually caused using polling layers in the conv. block. This is also an advantage of the used architecture. To show how this technique can be implemented in the spectral classification pipeline, grad-CAM scores were computed for one spectrum of 5 different classes of wheat (Fig. 9). The higher the grad-CAM score, the more important that specific wavelength band was for the classification process. Although this study included 30 classes, it was difficult to discuss the key features for all the 30 classes in the text, hence for demonstration purpose only 5 classes are shown in Fig. 9. It can be noted in Fig. 9 that higher weights were attained for several spectral regions correspond to chemical overtones related to CH, NH and OH bonds which are highly abundant in wheat kernels. This post-prediction feature analysis based on Grad-CAM also showed the advantages of using an ensemble of pre-processed data as for each type of pre-processing the CNN was able to extract relevant information from different bands which eventually lead to an improved performance of the DL model compared to earlier works.

3.5. Some optimisation tips and future improvements

In this study, a simple CNN architecture based on a single convolutional layer was used, however, NA optimisation presented in this study can be extended to account for multiple filters in the conv. layer and, if the complexity of the data is higher, include diverse types of conv. blocks with multiple conv. layers and channels, pooling layers, etc. The NA search component of this pipeline could be further applied to the automatic optimisation of the number of frozen layers in transfer learning scenarios [16]. In the case of using ensembles of classical chemometric pre-processed data, like it was presented in this study, one can also differently treat each pre-processing type as a different data block and use a multiblock input architecture that allows individual filter optimisation [55]. The ensemble input data approach is useful because it saves time (by not having to optimise the CNN for each pre-processing in separate) and allows for the model to pick complementary information from each pre-processed spectrum to improve classification.

For a swifter optimisation using TPE + Hyperband, one option would be to start the hyperparameter search from a subset of values that supply an already satisfactory solution. If the user has some intuition of a subset of hyperparameters that return a good objective function (metric), this subset should be enqueued into the search trials. By doing this, Hyperband will only complete training on the sets of hyperparameters that show better performance than the enqueued set, therefore reducing computation time. A possible downside of this strategy is that this may cripple the exploration of new hyperparameter regions and might focus the attention of the TPE on a local minimum and not a global one.

In terms of robustness of the final model, the optimisation pipeline could allow for random initialization of the model's weights at every trial (by using a random seed in the 'HeNormal' initialization). In principle, the optimisation algorithm would drift towards hyperparameters that are more resilient to initial fluctuations, hence making the model more robust.

A key point to note is that, based on the primary hypothesis of this study, which was to demonstrate that a simple DL model with proper NA adjustment and hyperparameter optimisation can lead to equal or even better performance than more complex DL models, the result obtained with a simple 1D CNN layer DL model showed that the hypothesis was found to be true as this study achieved almost ~2% better classification accuracy compared to that obtained in previous studies with complex DL architectures. One other point to note is that, although due to computational cost reasons, this study used a limited search space for the NAS and hyperparameter optimisation, the results obtained suggest that the explored search space was sufficient to prove the hypothesis that a simple DL model with proper optimisation can achieve performance better than a complex DL architecture model. In the future, based on the application, computation power and available time, the user can explore wider search spaces if required. However, the authors would like to emphasise that, the simpler the model, the better it is to interpret and use in practical applications. For example, to deploy this model in an actual physical sorting machine, it is much more desirable to have a simpler, lighter model that can run on modest microprocessors, instead of a very complex structure that requires much more computational resources.

Table 3
Accuracies (%) of the final model for calibration, validation and test set obtained under different settings.

	Calibration	Validation	Test	Accuracy on same test set from earlier study [22]
Best model Accuracy	98.7	95.2	94.9	93
Mean of 10 models (init seed fixed)	98.5 ± 0.13	95.1 ± 0.04	94.9 ± 0.04	
Mean of 10 models in CV	97.8 ± 0.43	97 ± 0.40	94.9 ± 0.29	
Mean of 10 models (init random seed)	99.2 ± 0.53	94.5 ± 0.18	94.4 ± 0.25	

4. Conclusions

This study presented a novel approach based on the combination of ensemble chemometric pre-processing paired with advanced optimisation techniques to automatically achieve the optimal neural architecture and the hyperparameters for the DL model. The method was showed on a real near-infrared spectral dataset for wheat variety classification. The results reached showed that the automated approach resulted in improved performance of deep classification models compared to randomly chosen neural architecture and hyperparameters. The results were not only improved, but a best-known classification accuracy of 94.9% was reached on the wheat variety classification dataset. Furthermore, this study highlighted a series of steps that are conducive to computationally affordable optimisation and to achieve robust models in terms of neural architecture engineering and hyperparameters. This

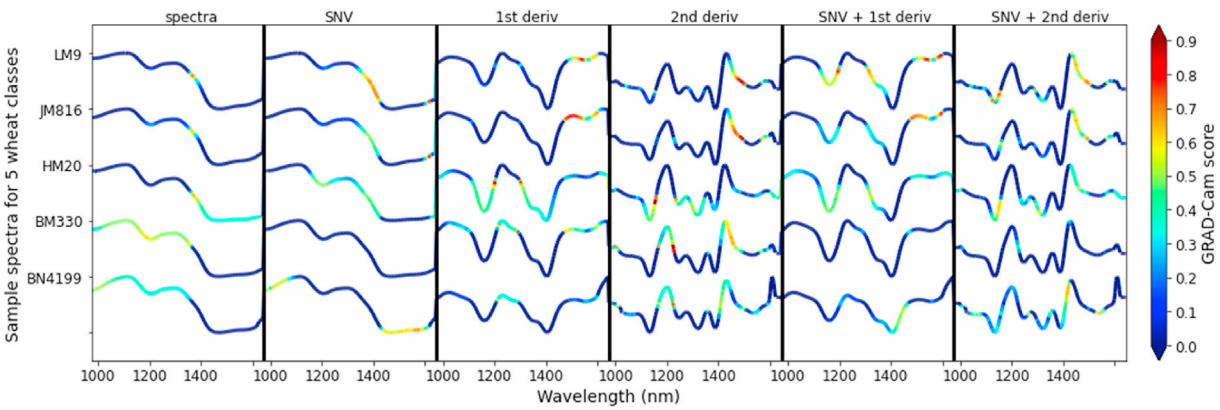

Fig. 9. Representation of the grad-CAM scores for wheat samples from 5 different classes. For an easier interpretation, the ensemble input spectra are divided into their original pre-processing methods, stacked vertically and coloured according to their grad-CAM score. Red segments correspond to the spectral features that the CNN relied the most to perform the classification. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

study not only achieved higher accuracy compared to earlier works but achieved it with a simpler model having only 1 convolutional layer, thus, showing the benefit of proposed model optimisation strategy. Furthermore, the use of Grad-CAM feature visualisation approach allows a clear visualisation of the important spectral band's contribution to the classification modelling closing the loop on the analysis pipeline. In terms of time, the proposed optimisation strategy achieved the optimal model in 3 times less time compared to random search for optimal neural architecture and hyperparameters. The method presented here can be generalised to wise range spectral data modelling problems.

CRediT authorship contribution statement

Dário Passos: Conceptualization, Software, Methodology, Writing – review & editing. **Puneet Mishra:** Conceptualization, Methodology, Software, Writing – original draft, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2021.104354.

References

[1] P. Firmani, S. De Luca, R. Bucci, F. Marini, A. Biancolillo, Near infrared (NIR) spectroscopy-based classification for the authentication of Darjeeling black tea, Food Contr. 100 (2019) 292–299.
[2] P. Mishra, A. Nordon, J. Tschannerl, G. Lian, S. Redfern, S. Marshall, Near-infrared hyperspectral imaging for non-destructive classification of commercial tea products, J. Food Eng. 238 (2018) 70–77.
[3] K.-X. Mu, Y.-Z. Feng, W. Chen, W. Yu, Near infrared spectroscopy for classification of bacterial pathogen strains based on spectral transforms and machine learning, Chemometr. Intell. Lab. Syst. 179 (2018) 46–53.
[4] C. Zhu, Y. Kanaya, R. Nakajima, M. Tsuchiya, H. Nomaki, T. Kitahashi, K. Fujikura, Characterization of microplastics on filter substrates based on hyperspectral imaging: laboratory assessments, Environ. Pollut. 263 (2020), 114296.
[5] R.M. Balabin, R.Z. Safieva, E.I. Lomakina, Gasoline classification using near infrared (NIR) spectroscopy data: comparison of multivariate techniques, Anal. Chim. Acta 671 (2010) 27–35.
[6] N. Fuenffinger, S. Arzhantsev, C. Gryniewicz-Ruzicka, Classification of ciprofloxacin tablets using near-infrared spectroscopy and chemometric modeling, Appl. Spectrosc. 71 (2017) 1927–1937.
[7] N. Sinelli, E. Casiraghi, D. Tura, G. Downey, Characterisation and classification of Italian virgin olive oils by near- and mid-infrared spectroscopy, J. Near Infrared Spectrosc. 16 (2008) 335–342.
[8] L. Wang, D.-W. Sun, H. Pu, J.-H. Cheng, Quality analysis, classification, and authentication of liquid foods by near-infrared spectroscopy: a review of recent research developments, Crit. Rev. Food Sci. Nutr. 57 (2017) 1524–1538.
[9] C. Pasquini, Near infrared spectroscopy: a mature analytical technique with new perspectives – a review, Anal. Chim. Acta 1026 (2018) 8–36.
[10] Y. Roggo, P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond, N. Jent, A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies, J. Pharmaceut. Biomed. Anal. 44 (2007) 683–700.
[11] N. Prieto, O. Pawluczyk, M.E.R. Dugan, J.L. Aalhus, A review of the principles and applications of near-infrared spectroscopy to characterize meat, fat, and meat products, Appl. Spectrosc. 71 (2017) 1403–1426.
[12] D. Ruiz-Perez, H. Guan, P. Madhivanan, K. Mathee, G. Narasimhan, So you think you can PLS-DA? BMC Bioinf. 21 (2020) 2.
[13] S. Wold, M. Sjöström, SIMCA: a method for analyzing chemical data in terms of similarity and analogy, chemometrics: theory and application, Am. Chem. Soc. (1977) 243–282.
[14] I.O. Afara, J.K. Sarin, S. Ojanen, M.A.J. Finnilä, W. Herzog, S. Saarakkala, R.K. Korhonen, J. Töyräs, Machine learning classification of articular cartilage integrity using near infrared spectroscopy, Cell. Mol. Bioeng. 13 (2020) 219–228.
[15] M.D.G. Neves, R.J. Poppi, Authentication and identification of adulterants in virgin coconut oil using ATR/FTIR in tandem with DD-SIMCA one class modeling, Talanta 219 (2020), 121338.
[16] P. Mishra, D. Passos, Realizing Transfer Learning for Updating Deep Learning Models of Spectral Data to Be Used in a New Scenario, Chemometrics and Intelligent Laboratory Systems, 2021, p. 104283.
[17] P. Mishra, D. Passos, A Synergistic Use of Chemometrics and Deep Learning Improved the Predictive Performance of Near-Infrared Spectroscopy Models for Dry Matter Prediction in Mango Fruit, Chemometrics and Intelligent Laboratory Systems, 2021, p. 104287.
[18] Z. Xin, S. Jun, T. Yan, C. Quansheng, W. Xiaohong, H. Yingying, A deep learning based regression method on hyperspectral data for rapid prediction of cadmium residue in lettuce leaves, Chemometr. Intell. Lab. Syst. 200 (2020), 103996.
[19] X.J. Yu, H.D. Lu, D. Wu, Development of deep learning method for predicting firmness and soluble solid content of postharvest Korla fragrant pear using Vis/NIR hyperspectral reflectance imaging, Postharvest Biol. Technol. 141 (2018) 39–49.
[20] X. Yu, H. Lu, Q. Liu, Deep-learning-based regression model and hyperspectral imaging for rapid detection of nitrogen concentration in oilseed rape (Brassica napus L.) leaf, Chemometr. Intell. Lab. Syst. 172 (2018) 188–193.
[21] C. Cui, T. Fearn, Modern practical convolutional neural networks for multivariate regression: applications to NIR calibration, Chemometr. Intell. Lab. Syst. 182 (2018) 9–20.
[22] L. Zhou, C. Zhang, M.F. Taha, X. Wei, Y. He, Z. Qiu, Y. Liu, Wheat kernel variety identification based on a large near-infrared spectral dataset and a novel deep learning-based feature selection method, Front. Plant Sci. 11 (2020) 1682.
[23] G. Polder, P.M. Blok, H.A.C. de Villiers, J.M. van der Wolf, J. Kamp, Potato virus Y detection in seed potatoes using deep learning on hyperspectral images, Front. Plant Sci. 10 (2019).
[24] Å. Rinnan, F.v.d. Berg, S.B. Engelsen, Review of the most common pre-processing techniques for near-infrared spectra, Trac. Trends Anal. Chem. 28 (2009) 1201–1222.
[25] J. Gerretzen, E. Szymańska, J.J. Jansen, J. Bart, H.-J. van Manen, E.R. van den Heuvel, L.M.C. Buydens, Simple and effective way for data preprocessing selection based on design of experiments, Anal. Chem. 87 (2015) 12096–12103.
[26] J. Engel, J. Gerretzen, E. Szymańska, J.J. Jansen, G. Downey, L. Blanchet, L.M.C. Buydens, Breaking with trends in pre-processing? Trac. Trends Anal. Chem. 50 (2013) 96–106.

[27] P. Mishra, A. Biancolillo, J.M. Roger, F. Marini, D.N. Rutledge, New data preprocessing trends based on ensemble of multiple preprocessing techniques, Trac. Trends Anal. Chem. (2020), 116045.
[28] J.-M. Roger, J.-C. Boulet, M. Zeaiter, D.N. Rutledge, Pre-processing Methods☆, Reference Module in Chemistry, Molecular Sciences and Chemical Engineering, Elsevier, 2020.
[29] P. Mishra, J.M. Roger, D.N. Rutledge, E. Woltering, SPORT pre-processing can improve near-infrared quality prediction models for fresh fruits and agro-materials, Postharvest Biol. Technol. 168 (2020), 111271.
[30] J.-M. Roger, A. Biancolillo, F. Marini, Sequential preprocessing through ORThogonalization (SPORT) and its application to near infrared spectroscopy, Chemometr. Intell. Lab. Syst. 199 (2020), 103975.
[31] P. Mishra, J.M. Roger, F. Marini, A. Biancolillo, D.N. Rutledge, Parallel Pre-processing through Orthogonalization (PORTO) and its Application to Near-Infrared Spectroscopy, Chemometrics and Intelligent Laboratory Systems, 2020, p. 104190.
[32] P. Mishra, T. Verkleij, R. Klont, Improved prediction of minced pork meat chemical properties with near-infrared spectroscopy by a fusion of scatter-correction techniques, Infrared Phys. Technol. 113 (2021), 103643.
[33] P. Mishra, S. Lohumi, Improved prediction of protein content in wheat kernels with a fusion of scatter correction methods in NIR data modelling, Biosyst. Eng. 203 (2021) 93–97.
[34] P. Mishra, F. Marini, A. Biancolillo, J.-M. Roger, Improved Prediction of Fuel Properties with Near-Infrared Spectroscopy Using a Complementary Sequential Fusion of Scatter Correction Techniques, Talanta, 2020, p. 121693.
[35] P. Mishra, A. Nordon, J.-M. Roger, Improved prediction of tablet properties with near-infrared spectroscopy by a fusion of scatter correction techniques, J. Pharmaceut. Biomed. Anal. (2020) 113684.
[36] P. Mishra, D.N. Rutledge, J.-M. Roger, K. Wali, H.A. Khan, Chemometric pre-processing can negatively affect the performance of near-infrared spectroscopy models for fruit quality prediction, Talanta (2021) 122303.
[37] R.J. Barnes, M.S. Dhanoa, S.J. Lister, Standard normal variate transformation and de-trending of near-infrared diffuse reflectance spectra, Appl. Spectrosc. 43 (1989) 772–777.
[38] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (1964) 1627–1639.
[39] T. Isaksson, T. Næs, The effect of multiplicative scatter correction (MSC) and linearity improvement in NIR spectroscopy, Appl. Spectrosc. 42 (1988) 1273–1284.
[40] G. Rabatel, F. Marini, B. Walczak, J.-M. Roger, VSN: variable sorting for normalization, J. Chemometr. 34 (2020), e3164.
[41] B.G. Osborne, Near-Infrared Spectroscopy in Food Analysis, Encyclopedia of Analytical Chemistry, 2006.
[42] A. Zela, A. Klein, S. Falkner, F. Hutter, Towards Automated Deep Learning: Efficient Joint Neural Architecture and Hyperparameter Search, 2018, 06906. ArXiv, abs/1807.
[43] S. Falkner, A. Klein, F. Hutter, BOHB: robust and efficient hyperparameter optimization at scale, Proc. 35th Int. Conf. Mach. Learn. Proc. Mach. Learn. Res. 80 (2018) 1437–1446.
[44] J. Wang, J. Xu, X. Wang, Combination of Hyperband and Bayesian Optimization for Hyperparameter Optimization in Deep Learning, 2018. ArXiv, abs/1801.01596.
[45] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: bandit-based configuration evaluation for hyperparameter optimization, in: International Conference on Learning Representations, 2017.
[46] K.G. xJamieson, A.S. Talwalkar, Non-stochastic Best Arm Identification and Hyperparameter Optimization, 2016. ArXiv, abs/1502.07943.
[47] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: 25th Annual Conference on Neural Information Processing Systems, NIPS 2011), 2011.
[48] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: Proceedings of the 25rd ACM (SIGKDD) International Conference on Knowledge Discovery and Data Mining, 2019.
[49] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026–1034.
[50] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, ArXiv, 2014 arXiv:1412.6980.
[51] F. Hutter, H. Hoos, K. Leyton-Brown, An efficient approach for assessing hyperparameter importance, Proc. 31st Int. Conf. Mach. Learn., PMLR 32 (1) (2014) 754–762.
[52] Z. Shen, R.A. Viscarra Rossel, Automated spectroscopic modelling with optimised convolutional neural networks, Sci. Rep. 11 (2021) 208.
[53] L.N. Smith, Cyclical learning rates for training neural networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 464–472.
[54] R.R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, Int. J. Comput. Vis. 128 (2019) 336–359.
[55] P. Mishra, D. Passos, Deep multiblock predictive modelling using parallel input convolutional neural networks, Anal. Chim. Acta (2021) 338520.
[56] L.N. Smith, A Disciplined Approach to Neural Network Hyper-Parameters: Part 1 - Learning Rate, Batch Size, Momentum, and Weight Decay, 2018, 09820. ArXiv, abs/1803.
[57] L. Li, K.G. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, A.S. Talwalkar, Massively Parallel Hyperparameter Tuning, 2018, 05934. ArXiv, abs/1810.
[58] R. Roscher, B. Bohn, M.F. Duarte, J. Garcke, Explainable machine learning for scientific insights and discoveries, IEEE Access 8 (2020) 42200–42216.
[59] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digit. Signal Process. 73 (2018) 1–15.
[60] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, in: Proceedings of Workshop at International Conference on Learning Representations, 2014, p. 6034. ArXiv, abs/1312.
[61] A. Nguyen, J. Yosinski, Understanding neural networks via feature visualization: a survey, in: W. Samek, G. Montavon, A. Vedaldi, L. Hansen, K.R. Müller (Eds.), Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science vol. 11700, Springer, Cham, 2019.
[62] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[63] P. Madhyastha, D. Batra, On model stability as a function of random seed, Proc. 23rd Conf. Comput. Nat. Language Learn. (CoNLL) (2019) 929–939.
[64] E. Cyr, M. Gulian, R. Patel, M. Perego, N. Trask, Robust Training and Initialization of Deep Neural Networks: an Adaptive Basis Viewpoint, 2020, 04862. ArXiv, abs/1912.
[65] M. Abadi, et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, Software available from, 2015. tensorflow.org.