Postharvest Biology and Technology 183 (2022) 111741

Contents lists available at ScienceDirect

Postharvest Biology and Technology

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

Multi-output 1-dimensional convolutional neural networks for simultaneous prediction of different traits of fruit based on near-infrared spectroscopy

Puneet Mishra [a,*], Dário Passos [b]

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

ARTICLE INFO

Keywords:
Spectroscopy
Chemometrics
Calibration
Chemistry

ABSTRACT

In spectral data predictive modelling of fresh fruit, often the models are calibrated to predict multiple responses. A common method to deal with such a multi-response predictive modelling is the partial least-squares (PLS2) regression. Recently, deep learning (DL) has shown to outperform partial least-squares (PLS) approaches for single fruit traits prediction. The DL can also be adapted to perform multi-response modelling. This study presents an implementation of DL modelling for multi-response prediction for spectral data of fresh fruit. To show this, a real NIR data set related to SSC and MC measurements in pear fruit was used. Since DL models perform better with larger data sets, a data augmentation procedure was performed prior to data modelling. Furthermore, a comparative study was also performed between two of the most used DL architectures for spectral analysis, their multi-output and single-output variants and a classic baseline model using PLS2. A key point to note that all the DL modelling presented in this study is performed using novel automated optimisation tools such as Bayesian optimisation and Hyperband. The results showed that DL models can be easily adapted by changing the output of the fully connected layers to perform multi-response modelling. In comparison to the PLS2, the multi-response DL model showed ~13 % lower root mean squared error (RMSE), showing the ease and superiority of handling multi-response by DL models for spectral calibration.

1. Introduction

Multivariate data acquired with advanced analytical instruments such as spectrometers are rich in chemical and physical information related to the fruit analysed (Mishra et al., 2020; Saeys et al., 2019). For example, the visible and near-infrared (Vis-NIR) spectroscopy data consists of colour information, overtones of fundamental chemical bonds such as OH, CH, NH and SH, and light scattering information related to the physical structure of the fruit (Nicolai et al., 2007). Due to such a rich information captured by advanced analytical instruments, often, the user is interested to reap the full benefit of the information and try to predict multiple chemicals as well as physical properties in the samples (Mishra et al., 2021b). For example, in the case of fresh fruit analysis with diffuse reflectance near-infrared (NIR) spectroscopy, NIR calibrations are widely used to predict several chemical properties such as moisture content (MC) and soluble solids content (SSC) (Walsh et al., 2020). Another example of where multi-output NIR model was found to be helpful was related to the prediction of several types of fats and proteins in meat produce (Zomeño et al., 2012) and several soil properties (Ng et al., 2019). Hence, the need to predict multiple response variables with a single model explains the need to develop multi-response predictor models.

In the domain of chemometrics, the need for multi-response modelling is well understood and a popular method already widely used is the multi-response partial least-squares (PLS2) regression (Wold, 1987; Wold et al., 1984). PLS2 linearly transforms the space of the predictors and the responses jointly to obtain a new set of orthogonal variables called scores and corresponding loading vectors. The transformation is performed to maximize the covariance between the predictor and the response variables. The scores obtained from such a transformation can be used to regress either in a least-squares or support vector machines approach. Later, to regain back the multi-response predictions, the predicted scores for the response variable can be internally multiplied with the loading vectors. An enormous amount of

* Corresponding author.
E-mail address: puneet.mishra@wur.nl (P. Mishra).

https://doi.org/10.1016/j.postharvbio.2021.111741
Received 26 May 2021; Received in revised form 15 September 2021; Accepted 16 September 2021
Available online 22 September 2021

application of PLS2 exists in the chemometrics literature as well as in the metabolomics community (Mishra et al., 2021b; Stocchero et al., 2019). Given the recent (last decade) advances in artificial intelligence (AI) related to deep learning (DL) neural networks (NN), DL models are emerging as a potential tool to model the multivariate data related to analytical instruments such as spectrometers (Mishra and Passos, 2021a, b; Puneet and Passos, 2021). The DL approaches have already outperformed the PLS regression modelling, and many examples exist where DL has been used to predict single response variables (Mishra and Passos, 2021a; Yu et al., 2018b). From the many types of DL algorithms currently available (Alzubaidi et al., 2021), there are two types that have become popular for spectral modelling. The first is the 1-dimensional convolutional neural networks (1D CNNs) which allows joint features extraction and subsequent mapping to target variable (Bjerrum et al., 2017; Cui and Fearn, 2018; Ng et al., 2019), and the second is the autoencoders that allow data compression/encoding by comparing input features to output representations of the data. These encoded features (of lower dimension) can afterwards be used to develop regression models (Abdalla et al., 2019; Yu et al., 2018b). So far, 1D CNNs have the advantage over autoencoders as it directly combines a feature extraction block and a mapping of these features to the property of interest in the same model. Although CNNs methods are increasingly expanding into the space of applications for multivariate predictive modelling, most of spectra-related studies until now have focused on the use of 1D-CNNs for prediction of a single response (Bjerrum et al., 2017; Cui and Fearn, 2018; Xin et al., 2020; Yu et al., 2018a, b). Nonetheless, like PLS2 regression, 1D-CNNs can also be used to model multiple responses (Ng et al., 2019). Multi-target regression (also called multi-output or multi-task) modelling with 1D-CNNs can be performed by branching the NN structure or by changing the dimensions of the final output layer to the desired number of responses.

This study aimed to compare the multi-response deep spectral predictive modelling with multi-response PLS based analysis for fruit traits prediction with NIR spectroscopy. A comparison between a one convolutional layer DL model with a more complex three convolutional layers model was also explored to find if complex DL models can outperform simple DL models. Furthermore, the study also aimed to explore/validate a novel automated DL model optimisation approach based on Bayesian optimization methods to train and optimise the DL models efficiently.

2. Materials and method

2.1. Data sets

In this study, a near-infrared (NIR) spectroscopy data set related to European 'Conference' pear (*Pyrus communis* L.) fruit was used for the prediction to two internal quality properties: moister content (MC) and solid soluble content (SSC). All fruit were sourced from a local fruit distributor in The Netherlands. A total of 551 pears were measured with NIRS and laboratory reference analysis. The spectral measurements were performed with a portable spectrometer (Felix F-750, Camas, WA, USA) in the range of 310–1135 nm, with a spectral resolution of 8–13 nm. In this study, the analysis was restricted to the spectral range (720–997 nm) because it has most of the information about the chemical overtones for moisture (OH) and sugars (CH) (Osborne, 2006). The spectrometer used a Tungsten lamp for illumination and a built-in white painted reference standard for estimating the reflectance. The data acquisition was performed at the mid belly part of the fruit. For all batches, after spectral measurements, a 1 cm thick slice was cut from the equator of the fruit and divided into four equal parts. Two of these parts were used to determine MC and SSC (in relative percentage units %). MC was found by recording the weight of the parts before and after drying in a hot-air oven (FP 720, Binder GmbH, Tuttlingen, Germany) at 80 °C for 96 h, as performed in a recent study (Mishra et al., 2021a). SSC of extracted pear fruit juice was determined using a handheld refractometer (HI 96801, Hanna Instruments Inc, Woonsocket, RI, USA) as performed in a recent study (Mishra et al., 2021a). The measured values for the MC distribution range from 80.12 % to a max of 99.94 % (mean = 84.83, std = 1.71), while the SSC distribution ranges from 8.4 % to 16.4 % (mean = 12.59 %, std = 1.33 %).

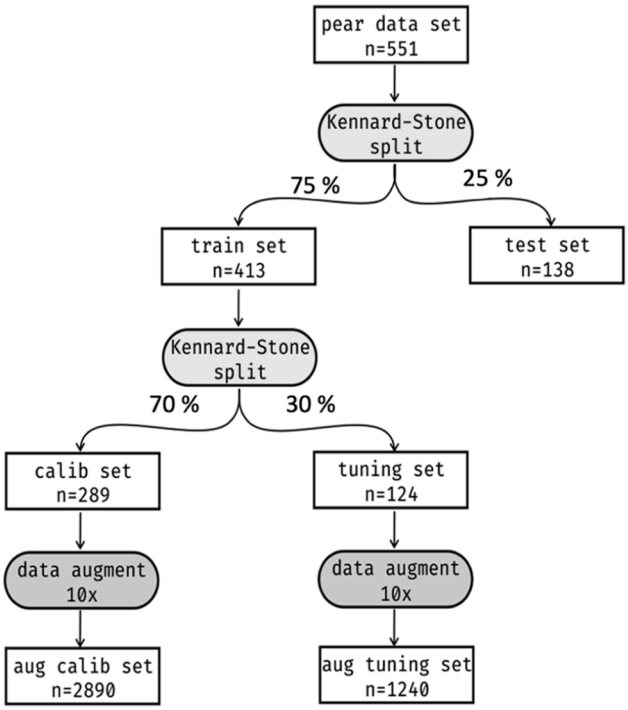

Fig. 1. Data partition and augmentation scheme.

2.2. Data partition and augmentation

The local measurement performed in the laboratory resulted in a total of 551 spectra and corresponding reference measurements. However, 551 samples from a DL modelling perspective can be considered as a small data set size and data augmentation was needed. The initial data set was partitioned into train (75 %) and test (25 %) subsets using the Kennard-Stone (Kennard and Stone, 1969) algorithm. The train set was further divided into calibration (70 %) and tuning (30 %) subsets using the same K–S algorithm and subsequently augmented using the method described in (Bjerrum et al., 2017). Model train and hyperparameters optimization was performed with the augmented calibration and tuning sets, while the test set was used to measure the final performance of the model. A complete scheme of the data split can be found in Fig. 1.

The augmentation was performed by adding random variations in offset, multiplication, and slope to existing samples. This procedure simulates slightly different spectra acquisition scenarios (e.g., background lighting, instrumental offsets, etc.) so that for the same target value, they were created multiple (slightly different) copies of the original spectra. Offset was varied ±0.10 times the standard deviation of the training set as described in (Bjerrum et al., 2017). Multiplication was carried out with 1 ± 0.10 times the standard deviation of the training set, and the slope was adjusted randomly between 0.95 and 1.05. The process was repeated 10 times for the training subsets (calibration and tuning), thus resulting in a total of 2890 calibration and 1240 tuning spectra. Both reference properties were augmented accordingly. The data augmentation serves two purposes; first was that it increases the data size to become suitable for DL modelling (help avoid overfitting) and second, it adds extra variability to the data set such that the models can become robust to unseen variations. Finally, all spectra were standardized (mean = 0 and standard deviation = 1). The models were

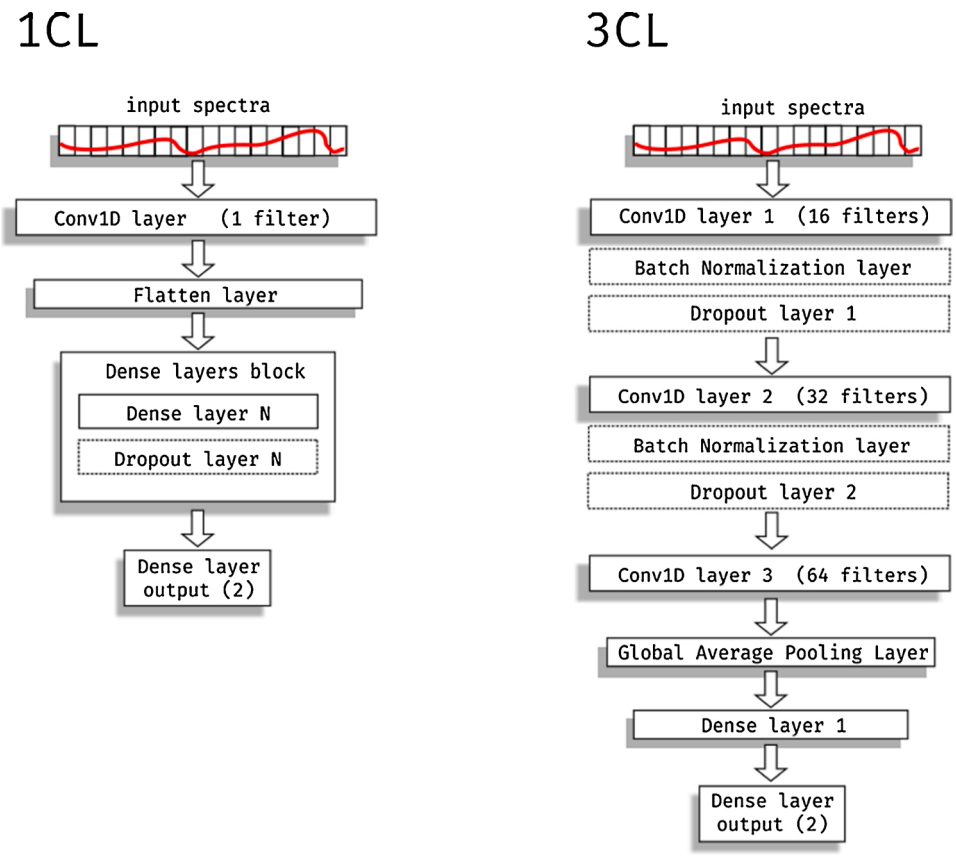

Fig. 2. The two deep learning model architectures studied in this work with, 1 convolutional layer (1CL) and 3 convolutional layers (3CL).

fitted/trained on the augmented calibration set and continuously evaluated and improved based on the metrics obtained for the tuning set. Once the model was calibrated, as a last step, it was tested on the test set. The evaluation metrics chosen are the Root Mean Squared Error (RMSE) and the Coefficient of Determination (R^2) defined as

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}, \quad (1)$$

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}, \quad (2)$$

$$SDR = \frac{RMSE \text{ if } \hat{y} = \bar{y}}{RMSE} \quad (3)$$

Where, N explains the number of samples, y_n corresponds to a measured target value, $\hat{y}_n$ explain a predicted target value and $\bar{y}$ was the mean value of the target variable. The Standard Deviation Ratio (SDR) was also defined for latter use in results comparisons.

2.3. Multi-response 1D convolutional neural networks

In this study, two different 1-dimensional CNNs (or 1D-CNN) architectures were used for comparison purposes. The first architecture, a one convolutional layer DL model derived from (Cui and Fearn, 2018), was constituted by 1 convolutional layer, with 1 filter (also called kernel) with a fixed width and stride = 1, followed by a block of fully connected (FC or dense) layers and dropout layers pairs, with different number of units each and a final output layer with 2 units (corresponding to target variable MC and SSC). The second architecture, a three convolutional layers DL model derived from (Malek et al., 2018; Zhang et al., 2019, 2020) uses 3 convolutional layers with 16, 32 and 64 filters correspondently. The first two convolutional layers were followed by batch normalization and dropout layers to decrease problems related to overfitting and increase the stability during training. This three-layers convolutional block was followed by a global average pooling layer (GAP), a dense layer with a fixed number of units and a final output dense layer with two units. An illustration of the two architectures can be found in Fig. 2. The main idea behind the CNN architecture was that the convolution layers can extract relevant spectral features (certain peaks, valleys, etc.) and the dense layer(s) further combine these features in a non-linear way to predict the target variables at the output. In both architectures, a linear activation function was used in the output layers (due to the regression task) and exponential linear unit (ELU) activations were used in all the other convolutional and dense layers. The units' weights in all layers were initialized using the 'He_normal' initialization procedure (He et al., 2015) and optimized (during model training) using the adaptive moment optimizer algorithm, Adam (Kingma and Ba, 2014). Adam's learning rate (LR) was automatically adapted along the training using an iterative algorithm that dynamically decreases the LR to ensure that the best convergence behaviour was achieved. The loss function used during the gradient descent optimization was the mean squared error (MSE) complemented by an L2 penalty (β) on the model weights (L2 layer regularization) to decrease the problem of overfitting. A fixed batch size of 128 samples and a maximum of 500 epochs for training were used in conjunction with an 'Early Stopping' approach to further help prevent overfitting issues.

The CNNs hyperparameters in both architectures were optimized using the Bayesian optimization pipeline recently presented in earlier study (Passos and Mishra, 2021). This pipeline uses Tree-structured Parzen Estimators (TPE) (Bergstra et al., 2011) and the Hyperband (Li et al., 2017) algorithm to effectively probe the hyperparameter space in search for optimal solutions and a LR range test to optimize the initial

Table 1
Ranges used for Neural Architecture Search (NAS) and hyperparameters (HP) search in the optimization pipeline. *The initial LR is determined based on the LR range finder test.

Name	Type	(Base value) [Search Interval] / step	1 / 3 convolutional layers (CL)
Convolutional filter size	HP	(25) [5 – 79] / 2	1CL
Number of Dense layers	NA	(3) [1 – 5] / 1	1CL
Number of units p/ Dense layer	HP	(36, 18, 12) [8 – 128] / 8	1CL
L2 regularization β	HP	(0.003) [0.00025 – 0.05] / 0.00025	1CL
Number of Dropout layers	NA	(0) [1 – 5] / 1	1CL
Dropout rate p/ Dropout layer	HP	[0 – 0.6] / 0.005	1CL
Initial Learning rate	HP	(*) [1×10^{-7} – 0.4]	1CL
Convolutional 1 filter size	HP	(9) [3 – 25] / 2	3CL
Convolutional 2 filter size	HP	(7) [3 – 25] / 2	3CL
Convolutional 2 filter size	HP	(5) [3 – 25] / 2	3CL
Convolutional 1 filter stride	HP	(5) [1 – 11] / 1	3CL
Convolutional 2 filter stride	HP	(3) [1 – 11] / 1	3CL
Convolutional 2 filter stride	HP	(3) [1 – 11] / 1	3CL
Dropout Rate 1	HP	(0.1) [0 – 0.6] / 0.005	3CL
Dropout Rate 2	HP	(0.) [0 – 0.6] / 0.005	3CL
Number of units in Dense layer	HP	(64) [8 – 256] / 8	1CL
L2 regularization β	HP	(0.01) [0.00025 – 0.05] / 0.00025	3CL
Initial Learning rate	HP	(*) [1×10^{-8} – 0.1]	3CL

LR. The optimization process starts by using a base hyperparameter configuration and improves it iteratively. Table 1 shows the base initial hyperparameter values, their search ranges and step for both architectures. For the one convolutional layer architecture, this optimization pipeline also acts as Neural Architecture Search method (NAS), since it was configured to find the appropriate number of dense and dropout layers for the final model. The number of dropout layers effectively implemented was always the same as the number of dense layers, but the optimization process can find a Dropout Rate (DR) for a certain dropout layer to be zero, which in practice was equivalent to disabling it.

The optimization pipeline fits a CNN model (with a given set of hyperparameters) to the calibration set, computes the RMSE on the tuning set, and uses that metric as the optimization objective function. The computed RMSE in this optimization phase was the averaged RMSE for both MC and SSC outputs. For each of the architectures, 1000 models were probed during the optimization. As a baseline benchmark for comparing the multi-output CNNs results, a PLS2 (Wold, 1987) analysis was carried out using the PLS function in the SciKit-Learn library (0.23.2) (https://scikit-learn.org/stable/).

All models and their optimization routines were implemented using the Python (3.6) language, TensorFlow/Keras (2.5.0) (Abadi et al., 2016) and the Optuna 2.9.1 library (Akiba et al., 2019), running on a workstation equipped with a NVidia GPU (GeForce RTX 1070 Ti), an Intel® Core™ i7-4770k @3.5 GHz and 16 GB RAM, running Microsoft Windows 10 OS.

3. Results

Prior to showing the DL modelling, a benchmark analysis with PLS2 was performed. A summary of the results from PLS2 analysis for MC and SSC prediction based on this pear fruit data set are shown in Fig. 3. PLS2 optimization was done by fitting multiple PLS models with varying number of Latent Variables (LVs) in the range (1–50) to the calibration set and computing their RMSE on the calibration and tuning sets. The criteria used to choose the optimal LVs = 7 (Fig. 3 A) was the point where the RMSE of the calibration and tuning sets diverge. This criterion usually ensures a better generalization capability to the PLS model. The RMSE on the test set for MC and SSC were 0.67 % (Fig. 3 B) and 0.65 % (Fig. 3 C), respectively.

The LR optimizations for both architectures were performed using the initial hyperparameter values. This procedure allowed to find the LR interval where the CNN effectively learns (i.e., the loss improves during

Table 2
Optimized hyperparameters for one and three convolutional layers deep model. *The number of dropout layers corresponds to the number of dropout layers with DR≠ 0.

1 Convolutional layer model hyperparameters	Value	3 Convolutional layer model hyperparameters	Value
Convolutional filter size	7	Convolutional 1 filter size	25
Number of Dense layers	3	Convolutional 2 filter size	15
Number of units p/ Dense layer	[64, 120, 96]	Convolutional 3 filter size	11
L2 regularization β	0.002	Convolutional 1 filter stride	7
Number of Dropout layers*	2	Convolutional 2 filter stride	10
Dropout rate p/ Dropout layer	[0.4, 0.05, 0.]	Convolutional 3 filter stride	7
Initial Learning rate	0.004	Dropout Rate 1	0.015
		Dropout Rate 2	0.185
		L2 regularization β	0.015
		Number of units in Dense layer	96
		Initial Learning rate	0.003

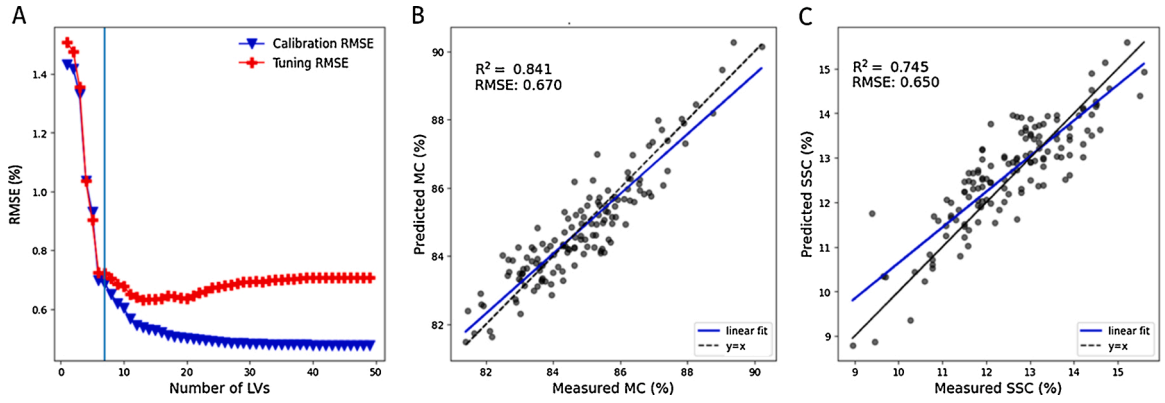

Fig. 3. A summary of PLS2 calibration, (A) Latent variable optimization (vertical line marks LVs = 7), (B) performance for predicting MC (%) in the test set, and (C) performance for predicting SSC (%) in the internal test set.

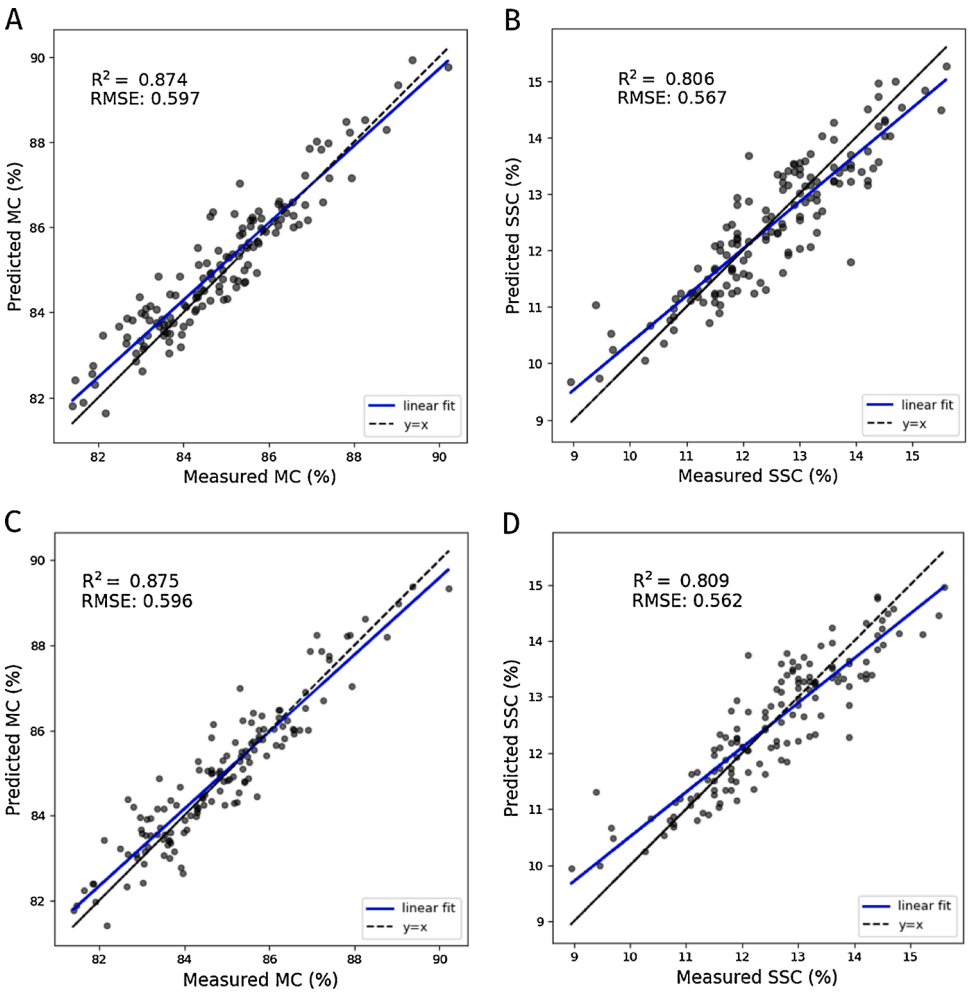

Fig. 4. Performance of multi-response deep learning model for simultaneously predicting moister (MC %) (A, C), and (B, D) soluble solids content (SSC %). Top row (A, B) are results obtained with one convolutional layer CNN model and the bottom row (C, D) correspond to results obtained with three convolutional layer CNN model, on the best run of the optimization benchmark.

training). The higher LR boundary of the LR range finder test was used as initial LR for the Adam optimizer. This procedure suggested an initial LR = 0.004 for one convolutional layer model and LR = 0.003 for three convolutional layers model. The other optimal hyperparameters found for both architectures are shown in Table 2.

The optimal models found for both DL model architectures where then tested on the test set yielding a lower RMSE and better linearity than the ones obtained using PLS2. The best performing models found during the optimization are shown in Fig. 4.

To have a more robust measure of the model's performance, the average RMSE and standard deviation for ten repeated runs over a maximum of 800 epochs was computed. For one convolutional layer DL model, $RMSE_{MC} = 0.551 \pm 8.8\times10^{-7}$ % and $RMSE_{SSC} = 0.548 \pm 9.6\times10^{-8}$ %. For the three convolutional layer DL model architecture $RMSE_{MC} = 0.596 \pm 8.1\times10^{-7}$ % and $RMSE_{SSC} = 0.562 \pm 6.4\times10^{-7}$ %. The results obtained with both architectures were similar with an advantage for the one convolutional layer DL model, despite the three convolutional layers DL model being more complex (37250 trainable parameters for three convolutional layers DL model vs. 25122 for one convolutional layer DL model). Over the ten repeated runs, the one convolutional layer DL model only stopped improving around epoch 680, while the three convolutional layers DL model stopped training earlier, at epoch 450. This shows that the maximum of 500 epochs allowed during the optimization task capped the performance of one convolutional layer DL model and explains the slightly higher values found for the best model of the pipeline (Fig. 4 A, B). Moreover, three convolutional layers DL model has a higher number of hyperparameters that must be optimized, and that process was much more computationally costly. In the reference workstation used, the optimization batch of 1000 models for three convolutional layers DL model took 23 h 30 min compared to the 18 h 43 min that one convolutional layer DL model optimization took. These optimization times completely dwarfed the PLS2 optimization that took less than a 1 min to compute. PLS2 simplicity is still one of its most attractive features and its speed cannot be outperformed by any neural models.

So far, it has been shown that the multi-output DL model outperformed the PLS2 analysis by a substantial fraction for both MC and SSC prediction. However, to understand if the multi-output DL model provides any advantage over developing two individual single output DL models, a performance comparison was performed. Using the one convolutional layer DL model architecture, two single output DL models were implemented and optimized individually using the same method and hyperparameters intervals previously described. The structure of the DL model the same as the one presented for one convolutional layer DL model (Fig. 2) but having only 1 unit in the final output layer. The two models are separately trained and optimized (Table 3) to predict MC and SSC individually.

The final metrics (for the average of 10 model runs) on the test set obtained on MC were $RMSE_{MC} = 0.538 \pm 7\times10^{-7}$ %, and for SSC, $RMSE_{SSC} = 0.539 \pm 8\times10^{-6}$ %. In both cases, the performance of the DL models improved when compared to their multi-output counterpart. This behaviour was expected because the neural network can focus on

Table 3
Optimized hyperparameters for single output 1 convolutional layer deep learning models for individual MC and SSC prediction. *The number of dropout layers corresponds to the number of dropout layers with DR≠ 0.

One convolutional layer model hyperparameters	Value (for MC)	Value (for SSC)
Convolutional filter size	7	19
Number of Dense layers	3	3
Number of units p/ Dense layer	[64, 88, 104]	[8, 32, 24]
L2 regularization β	0.00125	0.00325
Number of Dropout layers*	2	3
Dropout rate p/ Dropout layer	[0.14, 0.0, 0.05]	[0.245, 0.27, 0.535]
Initial Learning rate	0.004	0.04

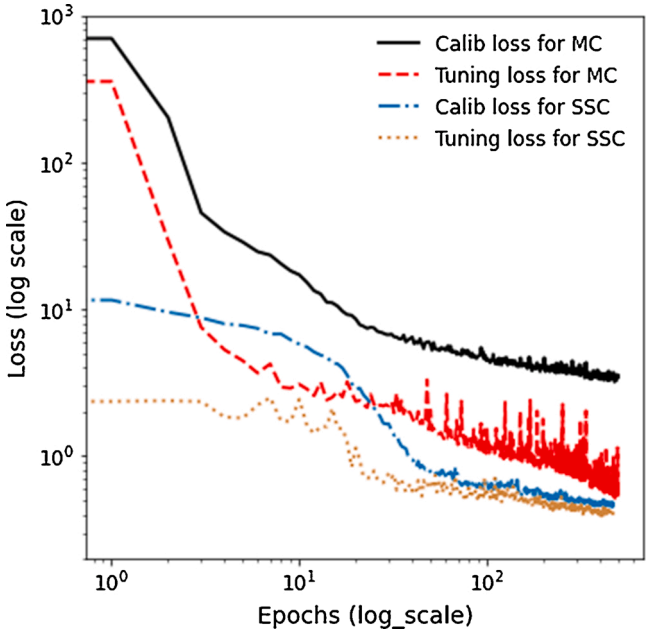

Fig. 5. Training history for the calibration and tuning losses for the optimized single-output models. Blue (dot-dashed) and orange (dotted) lines correspond to MC training. Black (solid) and red (dashed) lines correspond to SSC training (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

learning just one target variable. One of the downsides of having to optimize two models was the computational time requirements, as in this case, it was around 14 h (1000 models for both MC and SSC).

4. Discussion

In this study, the RMSE obtained with PLS based multi-response modelling were in the range of ~0.65 % for both the MC and SSC. In recent studies using pears, the prediction errors were in the similar range, for example, RMSEP ~0.6 % (Sun et al., 2009), RMSEP = 0.68 % (He et al., 2016), RMSEP = ~0.6 % (Yuan et al., 2020), and RMSEP = 0.7 % (Cruz et al., 2021). However, the performance of the multi-response DL models in this study was better than the results reported in the earlier studies using a single response model suggesting the superiority of DL models to have better performance even while modelling multi-responses.

One of the important things to consider when modelling multiple target variables is their value distribution. In this specific type of data (pear NIRS, MC and SSC data) the bibliographic survey and a preliminary data analysis showed that the expected errors for prediction for MC and SSC are of the same magnitude. This must be considered because, during the training of the multi-output DL models, the stochastic gradient descent algorithm will try to minimize the combined loss (MSE in this case) for both target variables. In the case of regression problems for multiple target variables with quite different amplitudes/ranges, the user could use normalization/scaling methods to ensure similar ranges of target variables and thus inducing a more balanced response from the DL models. One also must account for the fact that some target variables are easier to map/predict than others. This means that the DL models might struggle to adjust the layers weights to produce a non-linear function approximation to one of the target variables in detriment of the other. This seems to be (partially) the case in this study, where the single-output CNNs performed slightly better than their multi-output analogue. Based on the results obtained from all models (CNNs and PLS2) MC was arguably more difficult to predict. This was substantiated by the SDR computation, e.g., for the single-output one convolutional layer DL model, where $SDR_{MC} = 3.124$ and $SDR_{SSC} = 2.385$. In fact, by looking at the training history of the single-output one convolutional layer DL model (Fig. 5) one can see that the loss values for the MC model starts at much higher values than that of the SSC model. In fact, in the first 50 epochs the MC model reduces the loss error 2 orders of magnitude. It was expected that in the multi-output DL model this behaviour was also present, with MC improvement driving the neural network learning in the first epochs and then catching on to the SSC learning.

A final note regarding the modelling presented in this study is that the tuning loss (computed for validation) in the optimized DL models was lower than the calibration loss (effective training). This was since the Bayesian optimization pipeline used the minimization of the error in the tuning set as objective function. At this point it might be useful to remember that that is different than what is usually found in the standard training process of NNs. During training, the stochastic gradient descent algorithm finds the best model parameters (weights and biases) to fit the data, but here, prior to that, the Bayesian optimization searches for the model hyperparameters that minimizes the pipeline objective function thus leading to models that might perform better in the tuning set than in the calibration set. This opens the possibility to other types of optimization objective functions like e.g., multi-objectives, minimization of a metric and maximization of another, etc. It can also be used in scenarios where a neural network has several output branches that can be used, for example, for simultaneous regression type predictions (SSC, MC, etc) and classification tasks (cultivar, presence of internal browning, etc.). The goal would be to be able to find a neural network architecture that could effectively separate the spectral responses due to the influence of different target variables for the problem at hand.

5. Conclusions

This study demonstrated the potential of the DL models to multi-response modelling for fruit traits prediction by changing the final output layer of the DL to the desired number of responses. The results showed that the DL model for simultaneously predicting MC and SSC in pear fruit achieved up to ~13 % lower RMSE compared to the traditional latent space based multi-response PLS modelling. Furthermore, a simple DL model using only one convolutional layer was sufficient to achieve a better performance compared to an extraordinarily complex multiple convolutional layers-based DL model. From results, it can be understood that the spectral data may not require large DL models with multiple convolutional layers and as a single convolutional layer model with proper optimisation may be sufficient, hence, it is suggested to the scientific community to first explore simple DL models before moving to more complex neural architectures. Although it is incredibly important to perform the proper optimisation of DL model using the automated approaches as demonstrated in this study. Since a multi-output model is a special case of multiple single output model, hence, as expected, the performance of the multi-output DL models was topped by the performance of the individually developed single-output DL models for

different responses. However, in practical industrial settings, where the best model performance is desired, multi-output DL models are practical and handy as they allow a single model to integrate multiple responses. Such a single model can be easy to handle, update and transfer using advances DL approaches such as transfer learning.

CRediT authorship contribution statement

Puneet Mishra: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft. **Dário Passos:** Conceptualization, Methodology, Software, Formal analysis, Writing - original draft.

Declaration of Competing Interest

The authors report no declarations of interest.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. Savannah, GA, USA: USENIX Association. https://doi.org/10.5555/3026877.3026899.

Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., He, Y., 2019. Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure. Comput. Electron. Agric. 167, 105091 https://doi.org/10.1016/j.compag.2019.105091.

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage, AK, USA: Association for Computing Machinery. https://doi.org/10.1145/3292500.3330701.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data 8, 53. https://doi.org/10.1186/s40537-021-00444-8.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization. In: Proceedings of the 24th International Conference on Neural Information Processing Systems. Granada, Spain: Curran Associates Inc.. https://doi.org/10.5555/2986459.2986743.

Bjerrum, E.J., Glahder, M., Skov, T., 2017. Data augmentation of spectral data for convolutional neural network (CNN) based deep chemometrics. arXiv preprint arXiv:1710.01927 doi: arXiv:1710.01927.

Cruz, S., Guerra, R., Brazio, A., Cavaco, A.M., Antunes, D., Passos, D., 2021. Nondestructive simultaneous prediction of internal browning disorder and quality attributes in 'Rocha' pear (Pyrus communis L.) using VIS-NIR spectroscopy. Postharvest Biol. Technol. 179, 111562 https://doi.org/10.1016/j.postharvbio.2021.111562.

Cui, C., Fearn, T., 2018. Modern practical convolutional neural networks for multivariate regression: applications to NIR calibration. Chemom. Intell. Lab. Syst. 182, 9–20. https://doi.org/10.1016/j.chemolab.2018.07.008.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. IEEE International Conference on Computer Vision (ICCV) 1026–1034 doi: arXiv:1512.03385.

He, X.M., Fu, X.P., Rao, X.Q., Fang, Z.H., 2016. Assessing firmness and SSC of pears based on absorption and scattering properties using an automatic integrating sphere system from 400 to 1150 nm. Postharvest Biol. Technol. 121, 62–70. https://doi.org/10.1016/j.postharvbio.2016.07.013.

Kennard, R.W., Stone, L.A., 1969. Computer aided design of experiments. Technometrics 11, 137–148. https://doi.org/10.2307/1266770.

Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 doi: arXiv:1412.6980.

Li, L., Jamieson, K.G., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., 2017. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. International Conference on Learning Representations doi: N/A.

Malek, S., Melgani, F., Bazi, Y., 2018. One-dimensional convolutional neural networks for spectroscopic signal regression. J. Chemom. 32, e2977. https://doi.org/10.1002/cem.2977.

Mishra, P., Passos, D., 2021a. A synergistic use of chemometrics and deep learning improved the predictive performance of near-infrared spectroscopy models for dry matter prediction in mango fruit. Chemom. Intell. Lab. Syst. 104287 https://doi.org/10.1016/j.chemolab.2021.104287.

Mishra, P., Passos, D., 2021b. Realizing transfer learning for updating deep learning models of spectral data to be used in a new scenario. Chemom. Intell. Lab. Syst. 104283 https://doi.org/10.1016/j.chemolab.2021.104283.

Mishra, P., Biancolillo, A., Roger, J.M., Marini, F., Rutledge, D.N., 2020. New data preprocessing trends based on ensemble of multiple preprocessing techniques. Trac Trends Anal. Chem. 116045 https://doi.org/10.1016/j.trac.2020.116045.

Mishra, P., Marini, F., Brouwer, B., Roger, J.M., Biancolillo, A., Woltering, E., Echtelt, E. H.-v., 2021a. Sequential fusion of information from two portable spectrometers for improved prediction of moisture and soluble solids content in pear fruit. Talanta 223, 121733. https://doi.org/10.1016/j.talanta.2020.121733.

Mishra, P., Woltering, E., Brouwer, B., Hogeveen-van Echtelt, E., 2021b. Improving moisture and soluble solids content prediction in pear fruit using near-infrared spectroscopy with variable selection and model updating approach. Postharvest Biol. Technol. 171, 111348 https://doi.org/10.1016/j.postharvbio.2020.111348.

Ng, W., Minasny, B., Montazerolghaem, M., Padarian, J., Ferguson, R., Bailey, S., McBratney, A.B., 2019. Convolutional neural network for simultaneous prediction of several soil properties using visible/near-infrared, mid-infrared, and their combined spectra. Geoderma 352, 251–267. https://doi.org/10.1016/j.geoderma.2019.06.016.

Nicolai, B.M., Beullens, K., Bobelyn, E., Peirs, A., Saeys, W., Theron, K.I., Lammertyn, J., 2007. Nondestructive measurement of fruit and vegetable quality by means of NIR spectroscopy: a review. Postharvest Biol. Technol. 46, 99–118. https://doi.org/10.1016/j.postharvbio.2007.06.024.

Osborne, B.G., 2006. Near-infrared spectroscopy in food analysis. Encyclopedia of Analytical Chemistry. https://doi.org/10.1002/9780470027318.a1018.

Passos, D., Mishra, P., 2021. An automated deep learning pipeline based on advanced optimisations for leveraging spectral classification modelling. Chemom. Intell. Lab. Syst. 215, 104354 https://doi.org/10.1016/j.chemolab.2021.104354.

Puneet, M., Passos, D., 2021. Deep multiblock predictive modelling using parallel input convolutional neural networks. Anal. Chim. Acta 338520. https://doi.org/10.1016/j.aca.2021.338520.

Saeys, W., Do Trong, N.N., Van Beers, R., Nicolai, B.M., 2019. Multivariate calibration of spectroscopic sensors for postharvest quality evaluation: a review. Postharvest Biol. Technol. 158 https://doi.org/10.1016/j.postharvbio.2019.110981.

Stocchero, M., Locci, E., d'Aloja, E., Nioi, M., Baraldi, E., Giordano, G., 2019. PLS2 in metabolomics. Metabolites 9, 51. https://doi.org/10.3390/metabo9030051.

Sun, T., Lin, H.J., Xu, H.R., Ying, Y.B., 2009. Effect of fruit moving speed on predicting soluble solids content of' Cuiguan' pears (Pomaceae pyrifolia Nakai cv. Cuiguan) using PLS and LS-SVM regression. Postharvest Biol. Technol. 51, 86–90. https://doi.org/10.1016/j.postharvbio.2008.06.003.

Walsh, K.B., Blasco, J., Zude-Sasse, M., Sun, X., 2020. Visible-NIR 'point' spectroscopy in postharvest fruit and vegetable assessment: the science behind three decades of commercial use. Postharvest Biol. Technol. 168, 111246 https://doi.org/10.1016/j.postharvbio.2020.111246.

Wold, S., 1987. PLS Modeling With Latent Variables in Two or More Dimensions. doi: N/A.

Wold, S., Albano, C., Dunn, W.J., Edlund, U., Esbensen, K., Geladi, P., Hellberg, S., Johansson, E., Lindberg, W., Sjöström, M., 1984. Multivariate data analysis in chemistry. In: Kowalski, B.R. (Ed.), Chemometrics: Mathematics and Statistics in Chemistry. Dordrecht: Springer Netherlands, pp. 17–95. https://doi.org/10.1007/978-94-017-1026-8_2.

Xin, Z., Jun, S., Yan, T., Quansheng, C., Xiaohong, W., Yingying, H., 2020. A deep learning based regression method on hyperspectral data for rapid prediction of cadmium residue in lettuce leaves. Chemom. Intell. Lab. Syst. 200, 103996 https://doi.org/10.1016/j.chemolab.2020.103996.

Yu, X., Lu, H., Liu, Q., 2018a. Deep-learning-based regression model and hyperspectral imaging for rapid detection of nitrogen concentration in oilseed rape (Brassica napus L.) leaf. Chemom. Intell. Lab. Syst. 172, 188–193. https://doi.org/10.1016/j.chemolab.2017.12.010.

Yu, X.J., Lu, H.D., Wu, D., 2018b. 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, 39–49. https://doi.org/10.1016/j.postharvbio.2018.02.013.

Yuan, L.M., Mao, F., Chen, X.J., Li, L.M., Huang, G.Z., 2020. Non-invasive measurements of' Yunhe' pears by vis-NIRS technology coupled with deviation fusion modeling approach. Postharvest Biol. Technol. 160 https://doi.org/10.1016/j.postharvbio.2019.111067.

Zhang, X., Lin, T., Xu, J., Luo, X., Ying, Y., 2019. DeepSpectra: an end-to-end deep learning approach for quantitative spectral analysis. Anal. Chim. Acta 1058, 48–57. https://doi.org/10.1016/j.aca.2019.01.002.

Zhang, X., Xu, J., Yang, J., Chen, L., Zhou, H., Liu, X., Li, H., Lin, T., Ying, Y., 2020. Understanding the learning mechanism of convolutional neural networks in spectral analysis. Anal. Chim. Acta 1119, 41–51. https://doi.org/10.1016/j.aca.2020.03.055.

Zomeño, C., Juste, V., Hernández, P., 2012. Application of NIRS for predicting fatty acids in intramuscular fat of rabbit. Meat Sci. 91, 155–159. https://doi.org/10.1016/j.meatsci.2012.01.009.