Computers and Electronics in Agriculture 197 (2022) 106945

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

SpectraNet–53: A deep residual learning architecture for predicting soluble solids content with VIS–NIR spectroscopy

J.A. Martins[a,*], R. Guerra[a,b], R. Pires[a], M.D. Antunes[a], T. Panagopoulos[a], A. Brázio[a], A.M. Afonso[a], L. Silva[a], M.R. Lucas[a], A.M. Cavaco[a]

[a] Center for Electronic, Optoelectronic and Telecommunications (CEOT), Campus de Gambelas, University of Algarve, Faro, Portugal
[b] Physics Department, FCT, Campus de Gambelas, University of Algarve, Faro, Portugal

ARTICLE INFO

Keywords:
Deep learning
Residual network
Near-infrared
Spectroscopy
Citrus

ABSTRACT

This work presents a new deep learning architecture, SpectraNet–53, for quantitative analysis of fruit spectra, optimized for predicting Soluble Solids Content (SSC, in °Brix). The novelty of this approach resides in being an architecture trainable on a very small dataset, while keeping a performance level on-par or above Partial Least Squares (PLS), a time-proven machine learning method in the field of spectroscopy. SpectraNet–53 performance is assessed by determining the SSC of 616 *Citrus sinensi* L. Osbeck 'Newhall' oranges, from two Algarve (Portugal) orchards, spanning two consecutive years, and under different edaphoclimatic conditions. This dataset consists of short-wave near-infrared spectroscopic (SW-NIRS) data, and was acquired with a portable spectrometer, in the visible to near infrared region, on-tree and without temperature equalization. SpectraNet–53 results are compared to a similar state-of-the-art architecture, DeepSpectra, as well as PLS, and thoroughly assessed on 15 internal validation sets (where the training and test data were sampled from the same orchard or year) and on 28 external validation sets (training/test data sampled from different orchards/years). SpectraNet–53 was able to achieve better performance than DeepSpectra and PLS in several metrics, and is especially robust to training overfit. For external validation results, on average, SpectraNet–53 was 3.1% better than PLS on RMSEP (1.16 *vs.* 1.20 °Brix), 11.6% better in SDR (1.22 *vs.* 1.10), and 28.0% better in R^2 (0.40 *vs.* 0.31).

1. Introduction

From a top-down perspective, it can be observed that contemporary agricultural practices have been progressing towards what is termed *Agriculture 4.0*, a term coined by the World Government Summit and Oliver Wyman in their 2018 report, *"Agriculture 4.0 – The Future of Farming Technology"* (Clercq et al., 2018). This report highlights four major development issues in agriculture: (i) a higher demand for food from increased demographics, (ii) the ever increasing scarcity of natural resources, (iii) the reduced productivity derived from climate change, and especially (iv) the environmental burden of food waste, as the food that never gets eaten globally represents a cultivating field with a landmass bigger than China, and will most likely end up in a landfill, decomposing into methane. Solving these issues requires thoughtful consideration on how to improve agricultural efficiency while mitigating any consequential and detrimental repercussions, from a meta- to a micro-scale, within the context of the limited resources available in the production and logistics chain. A key factor permeating most of these issues is achieving a very high degree of process efficiency and quality control, so that production and logistics-chain resources are adequately dimensioned and utilized, while preventing food waste as much as possible.

A major breakthrough in this new agricultural paradigm shift was on moving towards non-destructive measurements for estimating fruit (or vegetable) internal quality attributes (IQAs), especially using *visible–near diffuse reflectance infrared spectroscopy* (Vis–NIRS, from 400 to 2500 nm) (Nicolaï et al., 2007; Cavaco et al., 2021). It is a well known fruit screening technique, that has shown very good results in the rapid acquisition of important fruit information; a detailed review on this subject can be read in Li et al. (2018). This type of spectroscopy irradiates light-permeable substances with a broad-spectrum light (*e.g.*, tungsten-halogen, which has a spectrum ranging from 350 to 2500 nm),

* Corresponding author.
E-mail addresses: jamartins@ualg.pt (J.A. Martins), rguerra@ualg.pt (R. Guerra), rmpires@ualg.pt (R. Pires), mantunes@ualg.pt (M.D. Antunes), tpanago@ualg.pt (T. Panagopoulos), ambrazio@ualg.pt (A. Brázio), amafonso@ualg.pt (A.M. Afonso), lsilva.m@outlook.com (L. Silva), marciarosendo@cacial.com (M.R. Lucas), acavaco@ualg.pt (A.M. Cavaco).

https://doi.org/10.1016/j.compag.2022.106945
Received 8 July 2021; Received in revised form 2 April 2022; Accepted 5 April 2022
Available online 20 April 2022

which then scatters in the interior of the substance and is absorbed by its chemical compounds, before being partially reflected outside and sampled by a spectrometer (Tuchin, 2015). Any light changes, when compared with the base spectra of the incident light, signal the presence of a particular chemical compound, as its molecules absorbed specific wavelengths. The interaction of light and matter is crucial for quantifying several important organic compounds, like sugars in fruits, which are paramount for establishing fruit maturity and are highly correlated with consumer satisfaction (Magwaza and Opara, 2015) – sugars account for the majority of total *soluble solids concentration* (SSC), which is measured by a refractometer, alongside other soluble compounds. The interpretation of the spectra in terms of IQAs is, however, poised with several difficulties, since absorption peaks are usually very broad and may overlap each other. Furthermore, they are also largely unspecific, being caused by the vibrations of the O–H and C–H bonds, which are common to all organic molecules (Gauglitz and Vo-Dinh, 2003; Golic et al., 2003).

Despite being a topic with proliferous published literature, the most common spectra inference methods usually rely on classical machine learning approaches, most notably (and the field time-proven standard) based around the Partial Least Squares (PLS) regression. Unfortunately, PLS performance is hugely dependent on the choice of data preprocessing techniques for each dataset, so it is common to see many different approaches in the literature, making this a trial-and-error process – what works on one dataset may very well be detrimental even on another of the same substance, or have unexpectedly worse results. With these drawbacks in mind, the aim of this research was to explore the possibility of using a Deep Learning approach to create a more robust architecture for quantitative analysis, that could minimize the use of specific and custom-tailored pre-processing techniques for each dataset, while simultaneously being competitive with PLS even on small datasets.

Deep Learning is a field of Machine Learning that uses multiple computation layers, grounded on Artificial Neural Networks, to iteratively learn advanced data representations, usually with the aim of predicting or classifying dependent variables (Yann LeCun and Bengio, 2015). While there exists a sizable body of research on the use of deep learning for fruit detection and counting in trees, external defects assessment, or type categorization, the same is not true for generalized regression architectures built for spectroscopy-based IQA prediction, especially capable of being trained on small sample counts. A notable model in this field is DeepSpectra by Zhang et al. (2019) which is based on the Inception architecture developed for GoogLeNet (Szegedy et al., 2015). The authors validated this model on one-dimensional spectral information on four different Vis–NIRS small datasets: for corn (80 samples), pharmaceutical tablets (228 samples), wheat (882 samples) and soil patches (3792 samples), outperforming three simpler convolutional neural network (CNN) models and obtaining similar or better than PLS results, without requiring data preprocessing (Zhang et al., 2019). Two variations of DeepSpectra were used as comparison to the proposed architecture.

Mishra and Passos (2021) have explored deep multiblock CNN analysis for determining dry matter in mango fruit, by separating the visible (Vis) and near-infrared (NIR) regions into different convolutional blocks. It is an interesting approach, as it allows both blocks to use different hyperparamenters (*e.g.*, filter sizes), and thus be able to learn specific features for their inputs. This is especially helpful in architectures with a single or few convolutional layers (the authors used only one for each block), as deeper networks are expected to compensate this effect by starting on small convolutional windows and later increase their “field of view” by doing convolutions of convolutions, which combine simpler features into more complex ones. Still, both approaches could be combined to reduce the number of necessary convolutional layers required to process spectra which are reasonably different between NIR and Vis. In a different work, Passos and Mishra (2021) explored Neural Architecture (NA) and hyperparameter optimization on wheat classification (147,096 samples of 30 varieties), using as a rough baseline the CNN architecture of Zhou et al. (2020), although with only a single convolutional layer, and then optimized the number of fully connected and dropout layers necessary for obtaining the best possible results on the hyperparameter ranges searched. They were able to increase accuracy from 93% to 94.9% with an overall simpler network structure.

Other recent works are focusing their efforts on hyperspectral imaging Yu et al. (2018), which usually requires a more complex acquisition process than using a portable spectrometer, nonetheless there is already some research on field-acquired hyperspectral images of fruit peel data (Benelli et al., 2020). Related spectroscopy research fields have also shown promising results when using deep learning methods, most notably on inferring soil data properties using 2D fast Fourier transform (FFT) spectrograms (Padarian et al., 2019a; Xu et al., 2019), or on the detection of chlorophyll content in potato leaves, using a continuous wavelet transform (CWT)-based 2D spectrogram feeding a CNN (Zhao et al., 2022).

Spectrometer calibration transfer is also a topic of interest for deep learning applied to chemometrics. Yang et al. (2022) developed an interpretable deep learning model for this task, denoted Deep-TranSpectra (DTS). This model showed very good calibration transfer results, when using five different spectrometers, for predicting moisture and crude protein contents on a soybean meal and a wheat dataset.

Kiranyaz et al. (2019) have done a recent review on the theory and usage of 1D CNNs. They highlight that most published research focuses on architectures developed for classification tasks, constrained to limited labeled data but allowing for input signals with high variance. These are architectures generally with only one or two hidden CNN layers and less than 10 K total learnable parameters, designed for solving very specific problems. Overall, the literature on 1D Deep CNNs with $>$ 1 M parameters is still scarce, especially on those using more advanced methods, like Residual Networks (He et al., 2016a), and on regression tasks.

This article proposes a new Deep Learning approach based on 1D Convolutional Neural Networks (CNNs) that is able to process input fruit spectra and infer desired IQAs, currently focusing on SSC. Its main contributions are:

1. Developing a new and robust Deep Residual Learning Architecture for spectroscopic data that relies on minimal preprocessing, is robust to outlier input data and works on small datasets;
2. Evaluating the performance of this deep-learning architecture on a previously published dataset;
3. Achieving state-of-the-art performance results for SSC prediction on an external validation dataset;
4. Demonstrating that Deep 1D CNN architectures can be trained without significantly overfitting (to the point of compromising generalization capability), with careful choices in design philosophy.

For performance assessments, the method was applied on an orange (*Citrus sinensi* L. Osbeck ‘Newhall’) spectra dataset, previously explored with PLS (Cavaco et al., 2018), which allows for direct performance comparisons. This orange variety is a Protected Geographical Indication (PGI) fruit in the region of Algarve, Portugal, and is strategically important for both the portuguese national and export markets. To best preserve its organoleptic qualities and as a non-climateric fruit, its harvesting has to be done at optimal edible ripening stage. Specific PGI legislation controls its optimal harvest date (OHD), and is based on the SSC value, juice volume and Maturity Index (which is the ratio of SSC to Titratable Acidity) (Cavaco et al., 2018).

The remainder of this article is organized as follows: Section 2 describes the materials and methods used, namely for spectra collection and processing, as well as for fruit internal quality attributes assessment; section 3 describes the DeepSpectra network training parameters used for performance comparisons, and explains the different parts of the

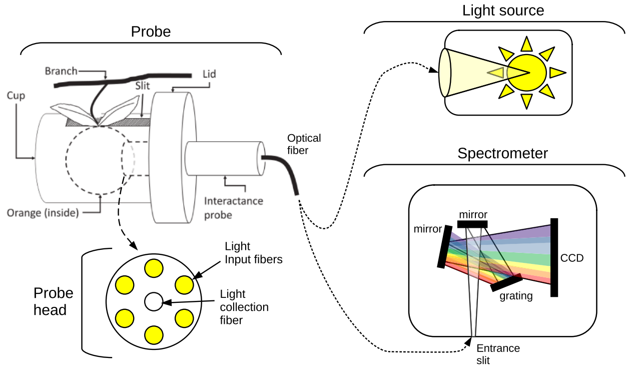
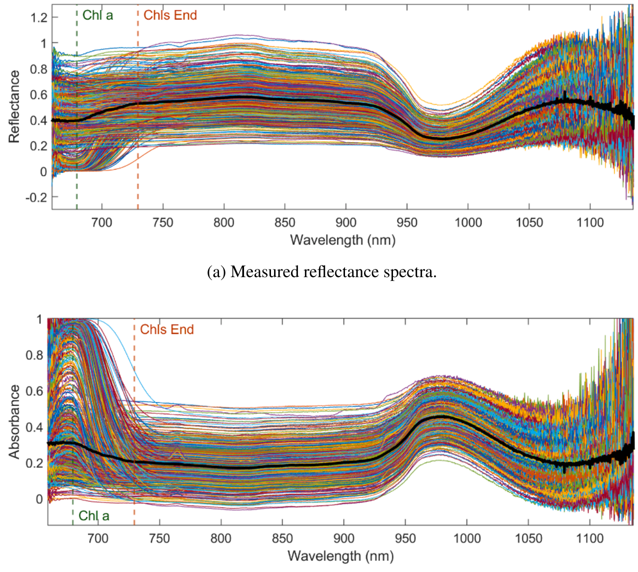


Fig. 1. Spectrometer setup.



(a) Measured reflectance spectra.

(b) Converted absorbance spectra.

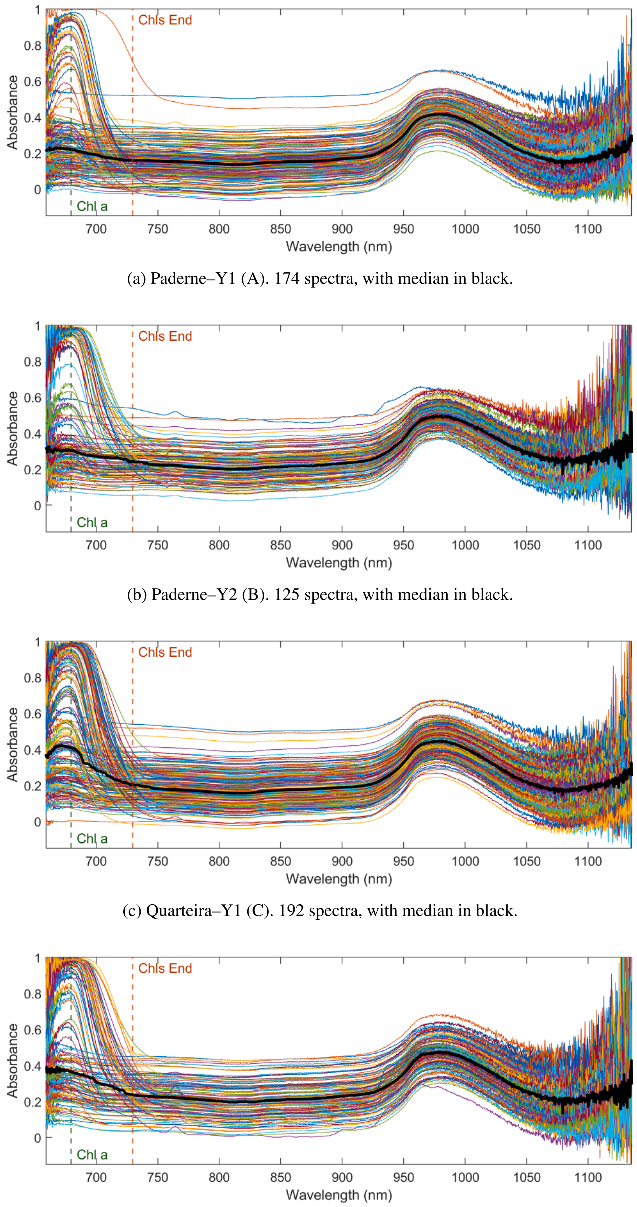
Fig. 2. The reflectance and absorbance-like spectra of the 616 oranges. Median wavelengths in black. The vertical dashed lines mark the Chlorophyll *a* absorption peak at 680 nm (in green) and the end of the Chlorophyll absorption influence, estimated from the data to be around 730 nm (in orange).

SpectraNet–53 architecture, detailing their design considerations; section 4 presents all performance assessments and comparisons with other methods. Finally, section 5 highlights the main achievements and contributions of this method.

2. Materials and methods

2.1. Spectra collection and processing

As described in Cavaco et al. (2018), fruits were sampled from two orchards (*Paderne* and *Quarteira* in Algarve, Portugal) during two consecutive harvest seasons. The oranges (*Citrus sinensi* L. Osbeck 'Newhall') used in these tests were picked randomly at the eye level (circa 1.60 m height) of the canopy of each of the 25 geo-referenced trees, chosen across two commercial orchards of local cooperative CACIAL, under different edaphoclimatic conditions. Sampling was performed through time, starting from early ripening stage up to late harvest, in both orchards, in the following periods: October 2015–February 2016 (harvest season 1) and November 2016–February 2017 (harvest season 2).



(a) Paderne–Y1 (A). 174 spectra, with median in black.

(b) Paderne–Y2 (B). 125 spectra, with median in black.

(c) Quarteira–Y1 (C). 192 spectra, with median in black.

(d) Quarteira–Y2 (D). 125 spectra, with median in black.

Fig. 3. The absorbance spectra of the 616 oranges, divided into orchard–year pairs. Median wavelengths in black. The vertical dashed lines mark the Chlorophyll *a* absorption peak at 680 nm (in green) and the end of the Chlorophylls absorption influence at 730 nm (in orange).

Spectra were obtained with a JAZ spectrometer (Ocean Optics, USA) in the Vis–NIR range of 680–1100 nm, using a tungsten light source and a customized fiber optic probe, made from a bifurcated fiber with an interactance probe in one extreme, with the other two legs connected to the spectrometer and light source, respectively, as shown in Fig. 1. The interactance probe is made of a receiving fiber in the middle connected to the spectrometer, and six emission fibers connected to the light source, arranged in a 5 mm circle around the middle fiber. For the absolute reference material, a disk of white Spectralon was used (WS-1, Ocean Optics, USA), held at an adequate constant height below the interactance probe. A custom closed-lid cup was designed and produced for measuring fruits in the tree while blocking as much sunlight as possible. After a fruit was introduced in the cup, the lid was closed and the interactance probe was slid through a small hole until it makes contact with the fruit. This process minimizes any harm to the fruit stem. For additional details, please refer to Cavaco et al. (2018).

In total, 616 orange spectra were acquired, with 1421 wavelengths each, between 659 and 1135.6 nm. Fig. 2(a) shows the reflectance (*R*)

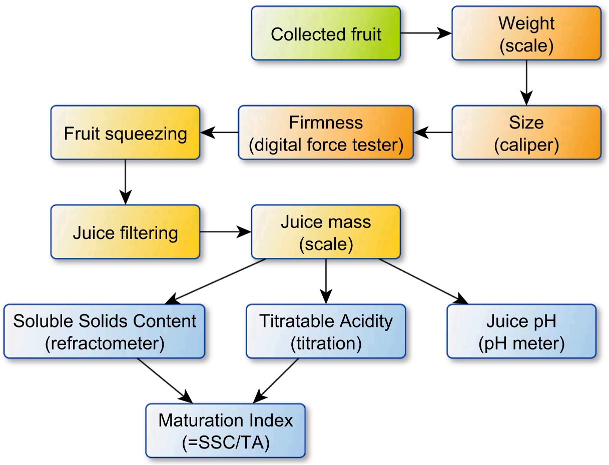

Collected fruit
Weight (scale)
Size (caliper)
Firmness (digital force tester)
Fruit squeezing
Juice filtering
Juice mass (scale)
Soluble Solids Content (refractometer)
Titratable Acidity (titration)
Juice pH (pH meter)
Maturation Index (=SSC/TA)

Fig. 4. Internal Quality Attributes assessment procedure.

data captured by the spectrometer. As the collection geometries for fruit and reference were not the same, it is possible to have $R > 1$ for some fruits and wavelengths. Fig. 2(b) shows the same spectra converted by an absorbance-like transform (A), which was done by truncating negative values to zero, *i.e.*, $R \in [0,\infty)$, and computing $A = -\log_{10}(R + 0.1)$, so that $A \in (-\infty, 1]$. This differs from the usual absorbance formula by the addition of a 0.1 constant, resulting in $\max(A) = -\log_{10}(0.1) = 1$. Also, at $R = 1, A = -\log_{10}(1.1) = -0.0414$. The motivation for this transformation is to limit the exponential scaling of low reflectance values into infinite absorbance, promoting a more linear relation between neighbor and low reflectance values (especially below 0.2). This will help to stabilize neural network weights at the first convolutional layer, during training.

This dataset spans two orchards for two consecutive years (seasons), with the first orchard *Paderne* having 174 spectra for *Y1* and 125 for *Y2*, while the second orchard *Quarteira* has 192 spectra for *Y1* and 125 for *Y2*. These will further be referenced either by the orchard–year pair, or the letters A to D (A: *Paderne–Y1*, B: *Paderne–Y2*, C: *Quarteira–Y1*, D: *Quarteira–Y2*). The spectra of these four major dataset partitions are respectively shown in Fig. 3. From visual inspection alone, the figures already show some differences between orchards, and even for consecutive years of the same orchard. This is highlighted when looking at the differences between the median spectra of each figure.

As for spectra preprocessing, no wavelenghts were discarded (Cavaco et al. (2018) eliminated all wavelengths below 750 nm and above 1100 nm), and used neither signal derivatives nor smoothing. Also, no malformed spectra were rejected, so all were included, even those contaminated by sunlight (signalled by a dip in the 760–765 nm range: 4% in orchard–year A, 18% in B and C, and 22% in D, for a total of 86 samples) or with unusual high or low absorbance values (with a norm outside the 3σ band of the mean norm; around 2–3% of the total remaining samples). In addition, no outliers were rejected during training or testing, so all samples of the dataset were used. These choices reflect one of the major goals of this work in creating and assessing an architecture resilient to several spectra-related errors.

2.2. Internal quality attributes assessment

Destructive analysis was done at a lab facility (Cavaco et al., 2018), with fruits at room temperature (*i.e.*, around 20°C), as depicted in Fig. 4. Each fruit was weighted and measured at the maximum equatorial diameter, with a caliper tool. Next, each fruit was individually squeezed in an orange automatic squeezer. After filtration, the total juice percentage (w/w), SSC, total titratable acidity (TA) and juice pH were assessed using standard procedures: juice was sampled into a digital

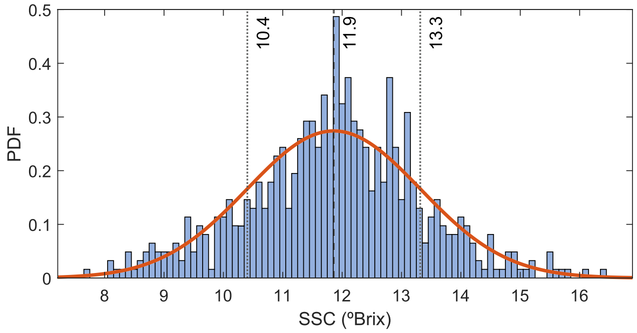


Fig. 6. Overall SSC (°Brix) histogram for the 616 orange samples in the dataset, with corresponding Probability Density Function. The dashed line represents the mean value and the dotted lines the standard variation around the mean. The bins are set at 0.1°Brix.

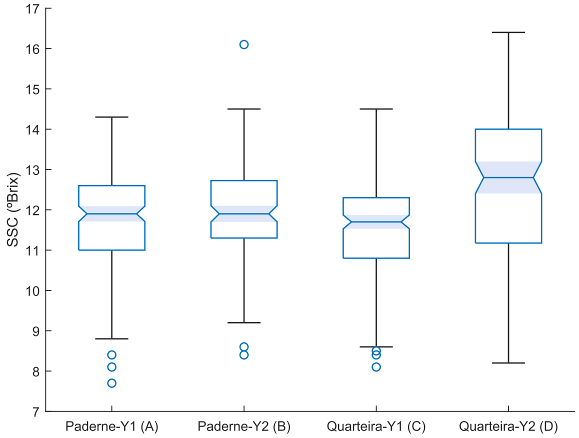


Fig. 7. SSC (°Brix) for orchard–year pairs, with shaded 95% confidence intervals. Statistical significance of the differences was assessed with a one-way ANOVA $[F(3,612) = 12.45, p < 10^{-7}]$. A Tukey post hoc test reveals significant pairwise differences between D and the rest ($p < 0.01$), but not between A, B and C.

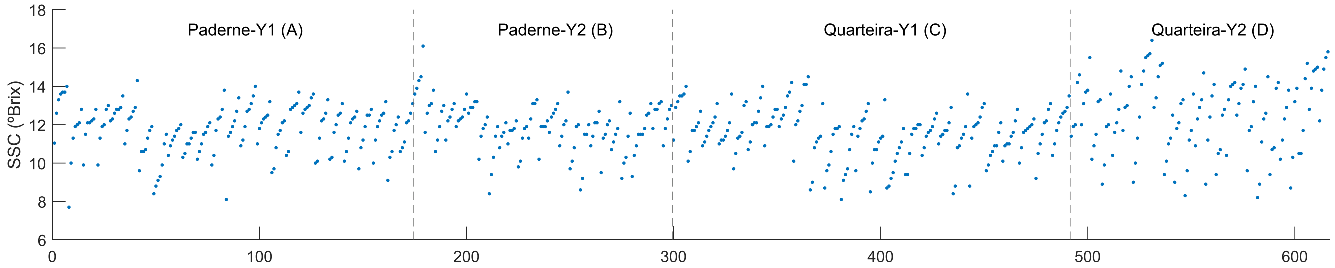


Fig. 5. SSC (°Brix) for the 616 orange samples in the dataset. Each orchard–year pair is separated by vertical dotted lines. Samples are shown ordered by increasing SSC value, for each orange tree. The variation is expected, as they were picked from early ripening stage up to late harvest.

Table 1
Combinations of train/test pairs for internal and external validation.

Number of orchard-year pairs in the training set	Internal validation (IV)			External validation (EV)		
	Example of a training/ test pair	Possible combinations		Example of a training/ test pair	Possible combinations	
1	80% of A/ 20% of A	$C_1^4 = 4$	A, B, C, D	A/B	$C_1^4 \times 3 = 12$	A/B, A/C, A/D, B/A, B/C, B/D,° C/A, C/B, C/D, D/A, D/B, D/C
2	80% of AB/ 20% of AB	$C_2^4 = 6$	AB, AC, AD, BC, BD, CD	AB/C	$C_2^4 \times 2 = 12$	AB/C, AB/D, AC/B, AC/D, AD/B, AD/C, BC/A, BC/D, BD/A, BD/C, CD/A, CD/B
3	80% of ABC/ 20% of ABC	$C_3^4 = 4$	ABC, ABD, ACD, BCD	ABC/D	$C_3^4 \times 1 = 4$	ABC/D, ABD/C, ACD/B, BCD/A
4	80% of ABCD/ 20% of ABCD	$C_4^4 = 1$	ABCD			

Notes. A is Paderne-Y1, B is Paderne-Y2, C is Quarteira-Y1, and D is Quarteira-Y2.

refractometer (Atago Model PAL-1, Atago Co. Ltd., Tokyo, Japan) for SSC determination; pH was measured with a digital pH meter (TitroLine 6000, SI Analytics GmbH, Germany); TA, which represents the mass percentage of citric acid per 100 mL of juice (%), was assessed by diluting 5 mL of orange juice with 5 mL of distilled water and subjecting the solution to volumetric alkaline titration with sodium hydroxide (0.1 N NaOH), until reaching a pH of 8.2, using an automatic potentiometric titrator (TitroLine 6000, SI Analytics GmbH Germany). The Maturation Index (MI) is simply the ratio $\frac{SSC}{TA}$.

Of all the attributes measured, the goal of this research is to achieve the best possible predictions for SSC, represented in Fig. 5, as it is a crucial attribute for both harvest date prediction and sweetness perception. The overall SSC distribution is represented in Fig. 6. However, the other IQAs are important for increasing SSC performance prediction, which will be discussed below, in Sections 3.2.4 and 3.2.5.

Fig. 7 shows the statistical differences between orchard–year pairs, which fall within the expected variations of orchards as a result of edaphoclimatic variability, due to many factors: *e.g.*, the soil types are different between locations, as well as the local and general climatic conditions (*e.g.*, colder *vs.* warmer years). A major takeaway from this is that prediction models should always span multiple locations and seasons, to account for the influence of edaphoclimatic variables.

2.3. Dataset partition

To allow for results comparison with Cavaco et al. (2018), an identical dataset division was performed, as shown in Table 1. This arrangement consists of 15 Internal Validation (IV) and 28 External Validation (EV) sets:

a) for the IV condition, the samples of each set were further divided into a training set (80%) and a test set (20%), using interleaved indices (*i.e.*, for each sequential 5 samples, the first 4 are for training and the 5th is for testing);
b) for EV, the test set was always a different orchard–year pair (*i.e.*, a holdout dataset) than the one(s) used for training.

3. Neural networks

3.1. Revisiting DeepSpectra

DeepSpectra (Zhang et al., 2019) was implemented as a baseline for comparison, which is a state-of-the-art deep learning architecture created for quantitative spectral analysis, based on the Inception model (Szegedy et al., 2015). For the training hyperparameters, the same ones that the authors proposed for their “Wheat” dataset assessment were used, which estimated protein content. The authors trained DeepSpectra on 775 wheat samples, from 7 crop years, and tested on 107 samples, from a single crop year. Spectra ranged from 400 to 2498 nm, on a 2 nm resolution, with a total of 1050 wavelengths. This was the dataset that had the closest number of features and samples to the one presented above (it has 1421 wavelengths of 616 oranges). In detail, the network hyperparamenters were: **kernel size 1:** 7-pts, **kernel size 2:** 3-pts, **kernel size 3:** 5-pts, **stride 1:** 3-pts, **stride 2:** 2-pts, **hidden number:** 32 neurons, **mini-batch size:** 128, **dropout rate:** 10%, **regularization coefficient:** $\lambda = 0.01$, **learning rate:** 10^{-2}, with a decay of 10^{-3} (-0.1%) at each epoch, and **epochs:** 10 (the training epochs were not reported on Zhang et al. (2019), so a similar value to the network architecture presented in this paper was used, which will be addressed in Section 3.2.5).

It is important, however, to keep in mind the following cautionary note: the spectral range used in the original DeepSpectra paper (400–2498 nm) is much broader than the range of the spectra studied here (659–1135.6 nm), which means that the hyperparameters used for DeepSpectra may not deliver optimal performance when applied to narrower and possibly less informative spectra, as is the present case.

3.2. SpectraNet–53: A deep learning architecture for fruit spectra

One of the first issues to contend in creating a deep learning architecture suitable for processing small datasets is the low amount of fruit samples available for training, as a sizable portion of samples need to be holdout for testing. It is well known that one of the common constraints of deep learning is that it requires a sizeable amount of training data to achieve performant models, otherwise the quality of inference will suffer. Working around this limitation is not trivial, and requires carefully designing an architecture to be as much as possible impervious to this issue.

After empirically evaluating many different architectures, the best one was based on a Deep Residual Network with 53 layers (detailed at the end of the article, in Table 4 and Fig. 15), stacking six Residual Unit blocks, and a careful choice of several key components and methods to allow it to generalize as best as possible. These will be described in the following subsections.

The software used for implementing this architecture (as well as DeepSpectra) was MathWorks MATLAB R2021a, with partial use of its Deep Learning Toolbox.

3.2.1. Residual units

A high layer count in a deep neural network is usually advantageous for achieving a higher capacity of representing increasingly complex information, however, classical CNN architectures suffer from several problems that restrict training on architectures with many layers (He et al., 2016a):

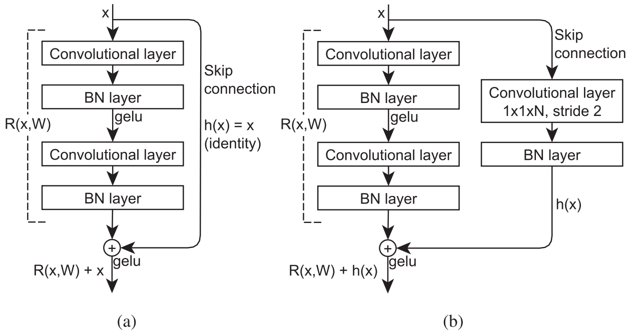


Fig. 8. Residual Units (He et al., 2016a). Diagram (a) shows a RU with an identity skip connection, while (b) shows a RU with an 1x1 convolutional shortcut with a stride of 2, used to downscale the identity connection before addition. BN are Batch Normalization (Ioffe and Szegedy, 2015) layers.

a) Gradients tend to vanish or explode, reducing the capability of the network to achieve a lower training error and consequently increase the prediction errors on test data;
b) Current solvers are typically inefficient in optimizing very deep CNNs, as they try to optimize all layers simultaneously, even when doing so is detrimental.

Deep Residual Networks (ResNets), specifically, were first introduced by He et al. (2016a) and are arguably one of the most important deep learning architectural innovations of the last decade. They work around these issues by using Residual Unit (RU) blocks, which are stacked into modularized architectures. An RU can be expressed as:

$$y_l = \mathrm{R}(x_l, W_l) + h(x_l), \tag{1}$$

$$x_{l+1} = \mathrm{A}(y_l), \tag{2}$$

where y_l and x_{l+1} correspond to the output and input of the l-th unit, $W_l = \{W_{l,k} | [0]_{1\leqslant k\leqslant K}\}$ is the set of weights and biases of the l-th RU with K layers, $\mathrm{R}(x_l, W_l)$ is the residual function to be learned by the network during training, $h(x_l)$ is the chosen identity mapping type for the skip connection, and A is a non-linear activation function (He et al., 2016b) (Gaussian Error Linear Units were used for SpectraNet, as described below in Section 3.2.1). The optimization goal of the network during training is for each RU to learn its residual function $\mathrm{R}(x_l, W_l)$. Block diagrams of the two SpectraNet RU configurations are shown in Figs. 8 (a) and (b), and their inner layers discussed in the next subsections.

Shortcut connections are a key advantage of residual architectures, as they allow for RU stacking without explicitly hindering training performance, which can be crucial for allowing a more rich feature space, capable of achieving meaningful performance metrics when using spectroscopic data. Also, the fact that a small dataset was used, requires the network to be able to differentiate spectra from each other as best as possible with only a few training samples, so the higher feature space is most certainly an advantage.

Table 4 and Fig. 15 show the complete 1D ResNet–53 architecture used in this work, composed of six stacked Residual Units. Each table row is a consecutive layer that is connected to the one above and below, unless stated otherwise. Additon layers 10, 17, 26, 33, 42 and 49 are responsible for adding skip connections with the main path (*i.e.*, they output y_l, the sum of both terms of Eq. (1)), and are followed by a non-linear activation layer which will output x_{l+1}, as defined in Eq. (2).

Convolutional layers. These layers replicate the concept of a neuron's receptive field (RF), implemented as a filter window that convolves and moves spatially along the input data axis. It is able to detect important features in the correlated data, similarly to how the human vision system relies on different types of on-off cell RFs. Filter windows can have arbitrary sizes, and each of its points has a learnable weight, with a global bias for the whole window. For example, looking at Table 4, the convolution layer at L2 has 32 filters of size 17, along with the 32 respective bias. When these 32 filters are applied to each of the 1421 points of the input tensor, the result is 1 × 1421 × 32 activations. These filters are all stacked together in the channel dimension, with their windows overlapping each other and applied point-by-point into a convolutional layer, allowing the detection and extraction of different types of important features in the underlying data, with various degrees of complexity/abstraction depending on the layer position in the overall network architecture (*i.e.*, deeper layers operate on more abstract feature representations). Due to the sliding window effect of these convolutional windows, these layers are also very efficient in retrieving characteristic representations from the spacial structure of data, which has historically increased their popularity in image recognition and categorization systems for computer vision (Vinet and Zhedanov, 2011).

Gaussian Error Linear Units (GELUs). SpectraNet heavily relies on Gaussian Error Linear Units (GELUs) (Hendrycks and Gimpel, 2016), which is a state-of-the-art non-linear activation function, with several desirable features:

a) not only it prevents the vanishing gradients problem, it weights inputs by their magnitude for both positive and negative values, instead of gating them by their sign. This accelerates training and protects neurons from "dying" to a badly conditioned learning rate, which can happen when using the common Rectified Linear Unit (ReLU) activation function; consequently, GELU also allows for much faster training by using a higher initial learning rate; and
b) most importantly, the transfer function itself is the expected transformation of a stochastic regularizer, which reduces the need for other regularization measures (*e.g.*, adding noise to intermediate network layers or requiring Dropout (Srivastava et al., 2014)).

Formally, as described in Hendrycks and Gimpel (2016), the neuron input x is multipled by $m \sim \mathrm{Bernoulli}[p = \Phi(x)]$, with $\Phi(x) = \mathrm{P}(X \leqslant x)$ and $X \sim \mathscr{N}(0, 1)$. As $E[\mathrm{Bernoulli}(p) = p]$, the expected value of the transformation is:

$$E[mx] = xE[m] = Ix \times \Phi(x) + 0x \times [1 - \Phi(x)] = x\Phi(x). \tag{3}$$

$\Phi(x)$ is the CDF of a gaussian distribution, which is adequate for neuron inputs following a Batch Normalization (BN) layer (Ioffe and Szegedy, 2015). As inputs have a higher probability of being dropped as x decreases, the transformation is stochastic while still depending on the input. The standardized GELU function, with $\mu = 0$ and $\sigma = 1$, can be approximated (Hendrycks and Gimpel, 2016) as:

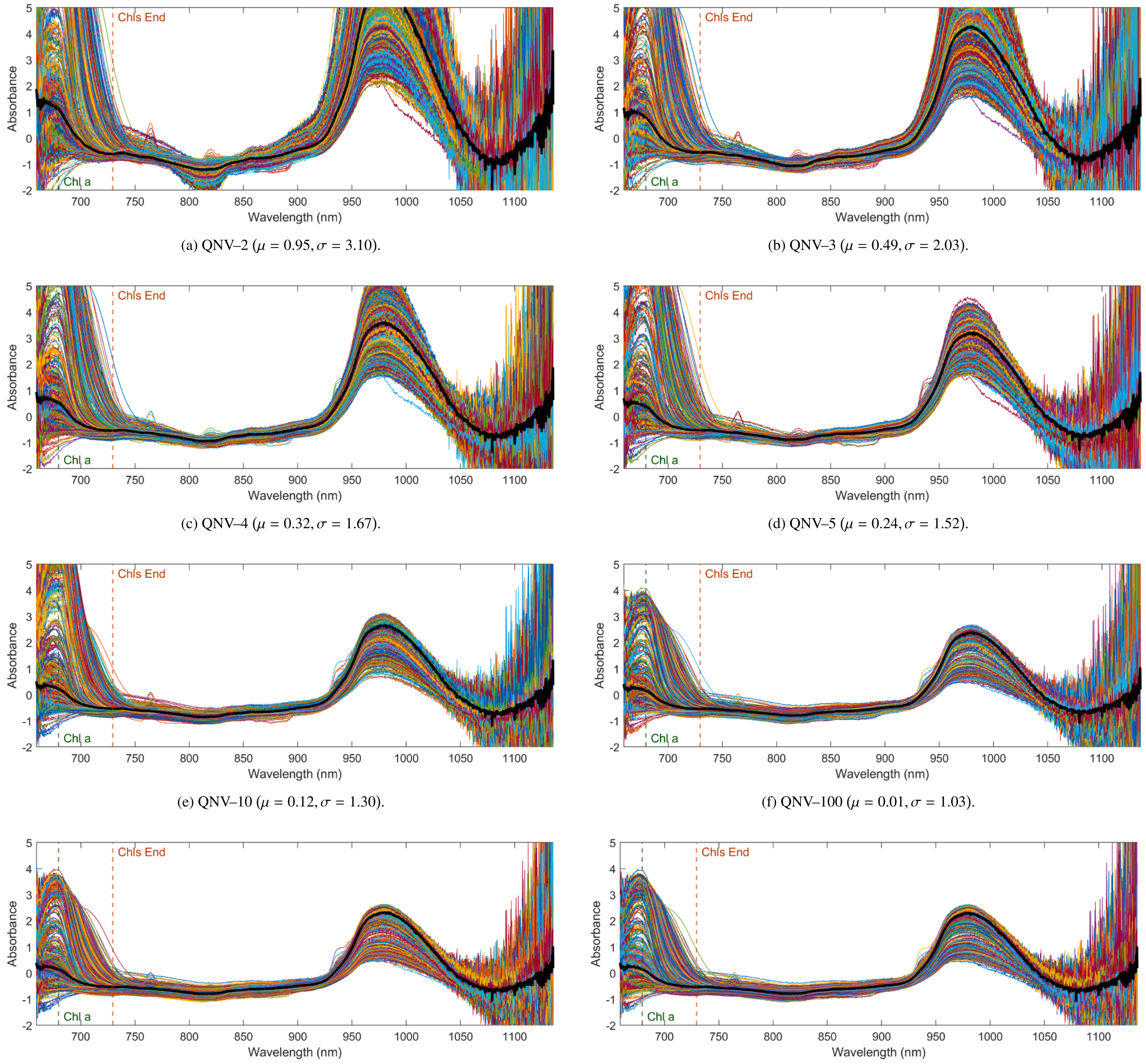
$$\mathrm{GELU}(x) = x\Phi(x) = x\mathrm{P}(X \leqslant x) \approx x\sigma(1.702x), \forall x \in \mathbb{R}. \tag{4}$$

However, having μ and σ as learnable hyperparameters was chosen instead, aiming for the best possible performance; GELU can thus be rewritten as:

$$\mathrm{GELU}(x|\mu,\sigma) = x\mathrm{CDF}(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} \exp^{\frac{-(x-\mu)^2}{2\sigma^2}} dx, \forall x \in \mathbb{R}. \tag{5}$$

Identity Mappings. He et al. (2016b) have shown that different types of RU identity mappings can be used for the shortcut connections, with varying degrees of performance. This work used both the original configuration (He et al., 2016a), as shown in Fig. 8(a), and another with 1 × 1 convolutions (Fig. 8(b)). However, both used exclusively GELU activations, as this solves an important problem the original configuration had with ReLU: only positive inputs for subsequent RUs after each addition layer are possible, which is undesirable, as this shapes the forward propagated signal to increase monotonically, and most likely hurts the representational ability of each RU. On the other hand, a GELU activation is able to output a $(-\infty, \infty)$ signal by varying its learned mean and standard deviation.

SpectraNet also used 1 × 1 convolutional shortcuts (He et al.,



(a) QNV–2 ($\mu = 0.95, \sigma = 3.10$).

(b) QNV–3 ($\mu = 0.49, \sigma = 2.03$).

(c) QNV–4 ($\mu = 0.32, \sigma = 1.67$).

(d) QNV–5 ($\mu = 0.24, \sigma = 1.52$).

(e) QNV–10 ($\mu = 0.12, \sigma = 1.30$).

(f) QNV–100 ($\mu = 0.01, \sigma = 1.03$).

(g) QNV–1000 ($\mu = 0.00, \sigma = 1.01$).

(h) SNV ($\mu = 0, \sigma = 1$).

Fig. 9. Signal standardization comparison between absorbance spectra, from QNV–2 to QNV–1000 and with SNV. In parenthesis are the mean and standard deviation when considering all signals shown. QNV–5 was a good compromise, which was used throughout this work.

2016b) (Fig. 8(b)) on RUs three and five, at L24 and L40 of Table 4, which allow for the layer activations in the convolutional units to be halved in the feature space and upsampled in the channel dimension. These convolutional shortcuts should allow for a better representational ability than just identity shortcuts, as their solution space is a superset of the latter. Historically, 1 × 1 convolutions usually display a larger training error than of identity shortcuts, which is commonly due to optimization issues (He et al., 2016b); however this was not the case for the current architecture. Batch Normalization (BN) (Ioffe and Szegedy, 2015) was also applied after every 1 × 1 convolution, before addition and subsequent GELU activation.

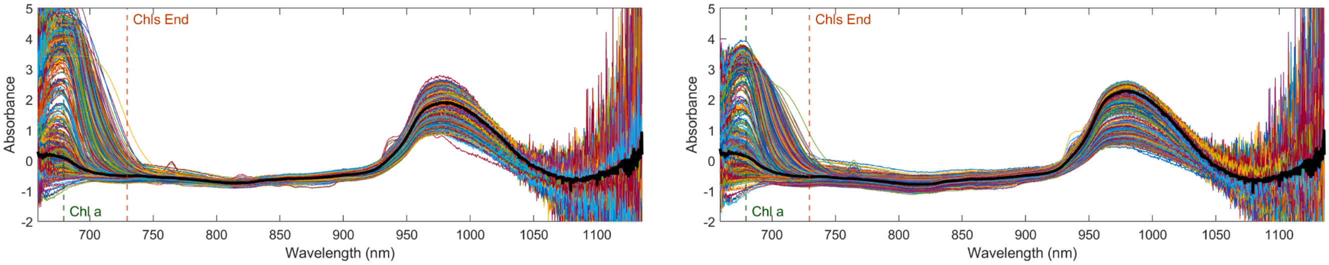
3.2.2. *Quantile normal variate transform*

Neural network inputs should be standardized or constrained (*e.g.*, $[-1,1]$) to accelerate training, stabilize the weights of the first layers and have increased resiliency to signal outliers. Both methods have their strengths and drawbacks, with the most common method for non-image data being signal standardization (*i.e.*, subtracting the mean and dividing by the standard deviation), also referred as Standard Normal Variate (SNV). SNV is somewhat robust to outliers, if they are not too frequent as to have a significant effect on the signal's mean and standard deviation – otherwise the mean range becomes too compressed after standardization, which is not ideal.

This work proposes a new method (as far as the authors can ascertain), entitled Quantile Normal Variate (QNV) with the aim of delivering a more robust signal standardization to the input layer of the network. QNV is thus defined as:

$$\mathrm{QNV}(x,n) = \frac{x - \mu_{\mathrm{Q}(x,n)}}{\sigma_{\mathrm{Q}(x,n)}}, \qquad (6)$$

where $\mathrm{Q}(x,n)$ are the quantiles of n evenly spaced cumulative probabilities (*i.e.*, $\frac{1}{(n+1)}, \frac{2}{(n+1)}, \ldots, \frac{n}{(n+1)}$) for an integer $n > 1$ (for non-existent



(a) QNV–5 $\left(\mu_{L2} = 0.247,\ \sigma_{L2} = 2.374,\ \gamma_{L2} = 0.996,\ \beta_{L2} = 7.61 \times 10^{-4}\right)$.

(b) SNV $\left(\mu_{L2} = 2.00 \times 10^{-8},\ \sigma_{L2} = 0.999,\ \gamma_{L2} = 0.996,\ \beta_{L2} = -8.4475 \times 10^{-5}\right)$.

Fig. 10. Absorbance spectra differences after BN at Layer 2, for QNV–5 and SNV input transforms, for one network of the EV–3 ACD/B set; μ is the mini-batch mean, σ the standard deviation, γ the scaling factor and β the bias (Ioffe and Szegedy, 2015).

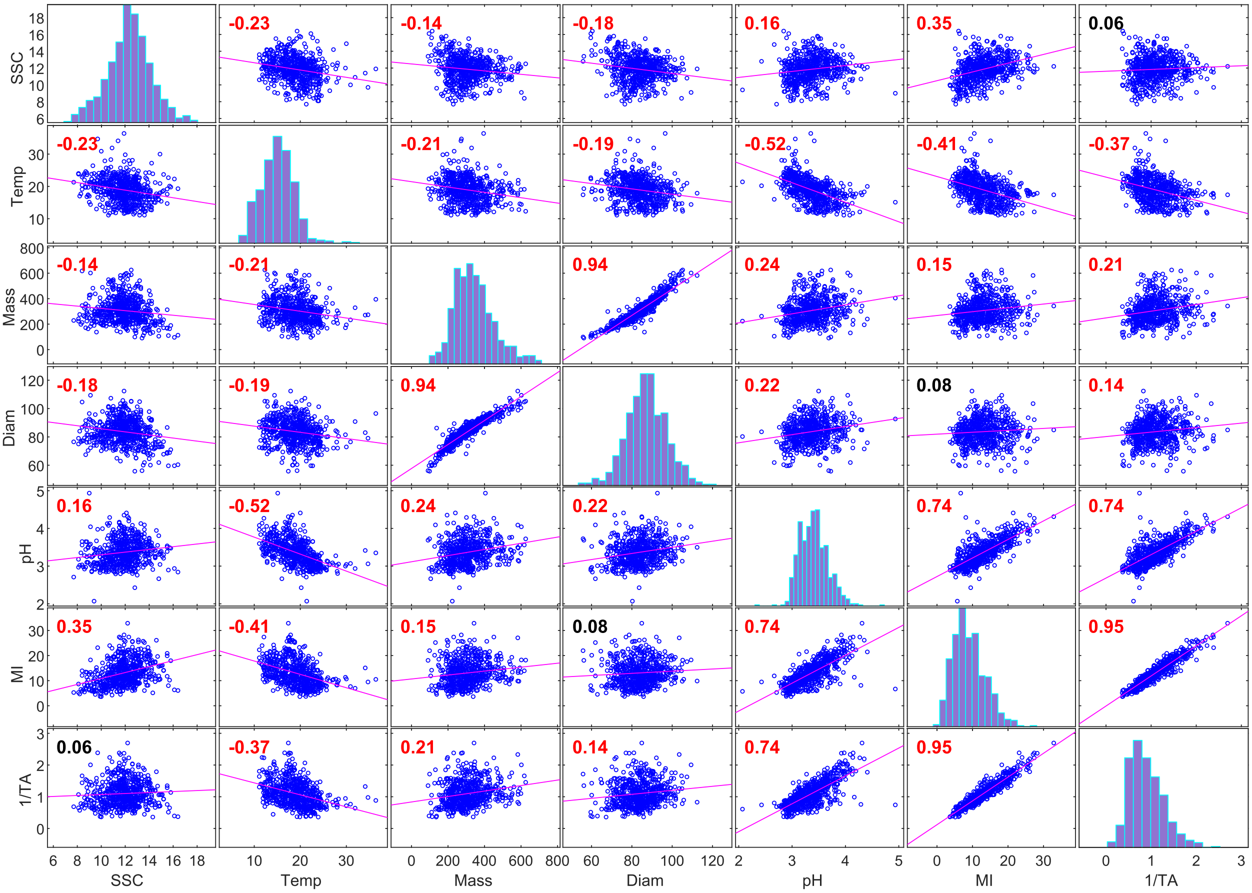


Fig. 11. Histograms and correlation plots for the different IQAs used in training. MI is not used, but is shown for completeness: as $\mathrm{MI} = \frac{\mathrm{SSC}}{\mathrm{TA}}$, the uncorrelated TA^{-1} is used instead. For each pair, the Pearson's correlation coefficients are shown in black or in red when significant ($\alpha = 0.05$) in a two-tailed correlation test. The units are, from left to right: %, °C, g, mm, pH, %·100 mL/g of citric acid, and 100 mL/g of citric acid.

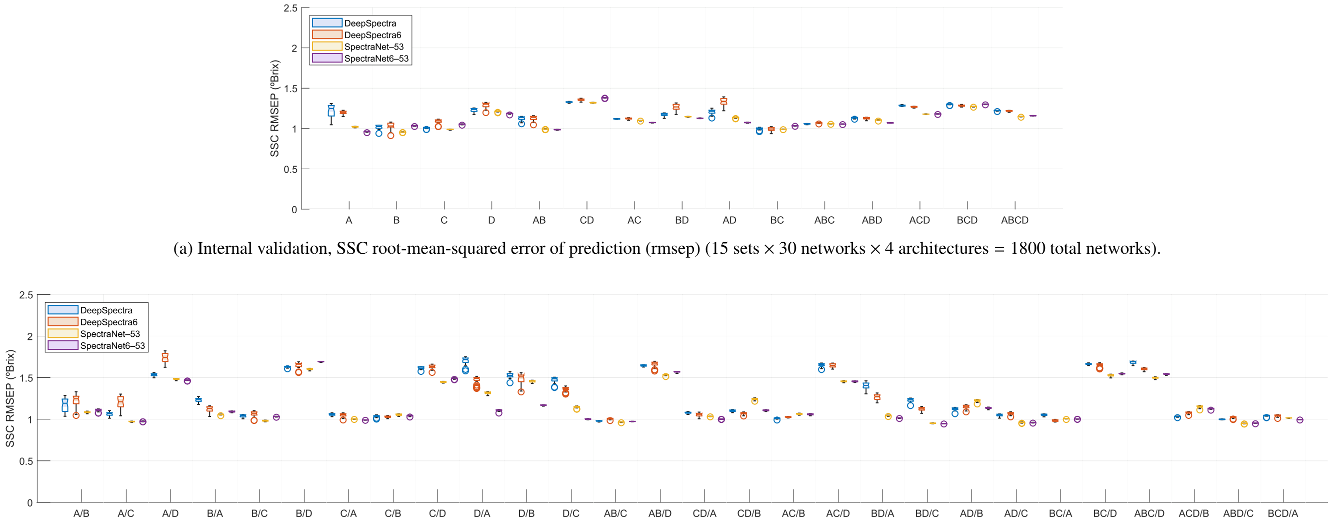
quantiles in the data, linear interpolation is used), μ is the mean of the resulting quantiles and σ is the standard deviation of the quantile values.

QNV tries to simultaneously reduce the skew effect of low-percentile and high-percentile outliers, by using quantiles as representative samples of the underlying signal. By increasing n, QNV approaches SNV (as $n \rightarrow \infty$), due to the increasingly number of quantiles, until they effectively sample all data points. Fig. 9 shows QNV examples at various n–spaced quantiles for the 616 orange spectra, and the corresponding SNV transform at Fig. 9(h). For this dataset, QNV at 1000 quantiles is already very similar to SNV, as shown in Fig. 9(g).

3.2.3. *Batch normalization after the input layer*

The seminal paper of Ioffe and Szegedy (2015) on Batch Normalization (BN) has been one of the most innovative contributions to deep learning of the last decade, allowing a new degree of training stabilization, acceleration and performance for many deep neural architectures, resulting in the use of BN layers as standard in most state-of-the-art methods. For a recent discussion on the merits of BN (*i.e.*, internal covariate shift reduction, objective function smoothing and especially length-direction decoupling), please refer to Kohler et al. (2020).

Similarly to Simon et al. (2016), a Batch Normalization (BN) layer was used directly after the input layer, which is still quite uncommon in the literature. This had two main benefits:



(a) Internal validation, SSC root-mean-squared error of prediction (rmsep) (15 sets × 30 networks × 4 architectures = 1800 total networks).

(b) External validation, SSC root-mean-squared error of prediction (rmsep) (28 sets × 30 networks × 4 architectures = 3360 total networks).

Fig. 12. Root-mean-squared error of prediction (rmsep) of the 30 neural networks trained in each of the 15 internal and 28 external validation sets, times the 4 architectures, with notched and shaded 95% confidence intervals. From left to right, at each set: DeepSpectra (blue), DeepSpectra6 (red), SpectraNet–53 (yellow) and SpectraNet6–53 (purple).

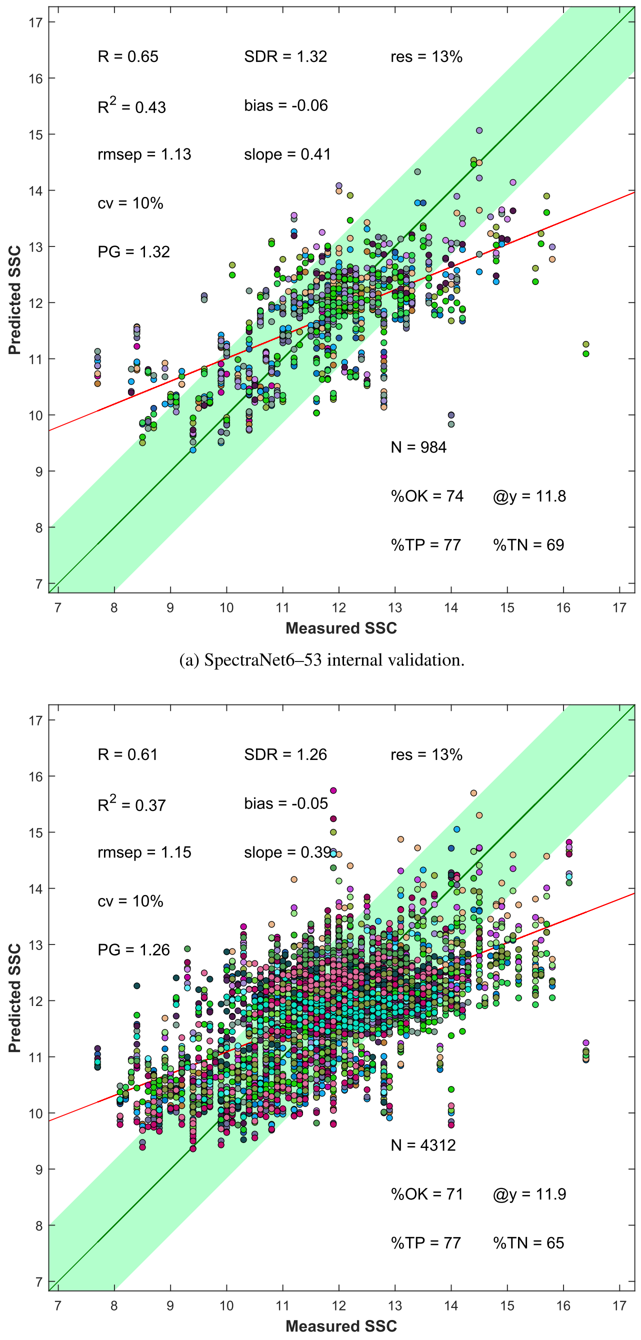
Table 2
Summary of the SSC performance results obtained in IV and EV.

		N	rmsec	rmsep	SDR	PG	R^2	CV(%)	Bias	Slope
SSC (%), SEL = 0.1% **DeepSpectra** (baseline)	IV	1	1.08 ± 0.15	1.12 ± 0.12	**1.29 ± 0.22**	**1.26 ± 0.24**	**0.41 ± 0.16**	9.40 ± 1.06	0.15 ± 0.34	0.44 ± 0.15
		2	1.14 ± 0.15	1.16 ± 0.11	**1.26 ± 0.09**	**1.24 ± 0.13**	**0.39 ± 0.06**	9.77 ± 0.82	0.13 ± 0.21	0.40 ± 0.05
		3	1.11 ± 0.11	1.19 ± 0.12	**1.25 ± 0.04**	**1.21 ± 0.08**	**0.36 ± 0.04**	10.13 ± 0.86	0.04 ± 0.15	0.33 ± 0.04
		4	1.10	1.22	**1.23**	**1.18**	**0.34**	10.47	0.09	0.35
	EV	1	1.08 ± 0.14	1.34 ± 0.26	**1.07 ± 0.18**	**1.08 ± 0.20**	**0.33 ± 0.11**	11.23 ± 2.04	0.22 ± 0.65	0.38 ± 0.14
		2	1.12 ± 0.15	1.25 ± 0.27	**1.14 ± 0.11**	**1.21 ± 0.29**	**0.38 ± 0.11**	10.44 ± 1.93	0.05 ± 0.58	0.36 ± 0.10
		3	1.12 ± 0.10	1.19 ± 0.33	**1.20 ± 0.07**	**1.30 ± 0.37**	**0.41 ± 0.09**	9.92 ± 2.37	−0.10 ± 0.56	0.39 ± 0.09
SSC (%), SEL = 0.1% **DeepSpectra6** (6 outputs)	IV	1	1.10 ± 0.15	1.15 ± 0.11	**1.25 ± 0.18**	**1.22 ± 0.20**	**0.36 ± 0.16**	9.67 ± 0.92	0.14 ± 0.28	0.35 ± 0.18
		2	1.16 ± 0.19	1.20 ± 0.14	**1.22 ± 0.07**	**1.20 ± 0.09**	**0.36 ± 0.05**	10.11 ± 1.08	0.17 ± 0.22	0.32 ± 0.04
		3	1.10 ± 0.10	1.19 ± 0.11	**1.25 ± 0.04**	**1.22 ± 0.08**	**0.37 ± 0.04**	10.06 ± 0.76	0.04 ± 0.16	0.38 ± 0.04
		4	1.10	1.22	**1.23**	**1.19**	**0.35**	10.42	0.12	0.37
	EV	1	1.11 ± 0.14	1.34 ± 0.26	**1.07 ± 0.14**	**1.09 ± 0.23**	**0.31 ± 0.10**	11.18 ± 1.87	0.22 ± 0.57	0.29 ± 0.11
		2	1.12 ± 0.14	1.23 ± 0.27	**1.16 ± 0.08**	**1.23 ± 0.30**	**0.39 ± 0.10**	10.24 ± 1.91	−0.03 ± 0.56	0.37 ± 0.07
		3	1.12 ± 0.09	1.19 ± 0.28	**1.20 ± 0.05**	**1.28 ± 0.33**	**0.41 ± 0.09**	9.91 ± 1.94	0.02 ± 0.53	0.39 ± 0.06
SSC (%), SEL = 0.1% **SpectraNet-53**	IV	1	0.91 ± 0.12	1.04 ± 0.11	**1.37 ± 0.16**	**1.34 ± 0.18**	**0.47 ± 0.10**	8.76 ± 0.81	−0.08 ± 0.21	0.44 ± 0.13
		2	0.92 ± 0.10	1.11 ± 0.12	**1.32 ± 0.07**	**1.29 ± 0.12**	**0.43 ± 0.06**	9.39 ± 0.88	0.02 ± 0.20	0.43 ± 0.05
		3	0.90 ± 0.06	1.15 ± 0.09	**1.29 ± 0.06**	**1.25 ± 0.08**	**0.40 ± 0.05**	9.78 ± 0.60	0.05 ± 0.17	0.41 ± 0.05
		4	0.89	1.15	**1.31**	**1.25**	**0.42**	9.85	0.13	0.42
	EV	1	0.92 ± 0.12	1.22 ± 0.23	**1.17 ± 0.15**	**1.19 ± 0.26**	**0.38 ± 0.09**	10.18 ± 1.65	0.09 ± 0.46	0.39 ± 0.12
		2	0.92 ± 0.08	1.17 ± 0.22	**1.22 ± 0.12**	**1.29 ± 0.32**	**0.40 ± 0.08**	9.75 ± 1.55	0.04 ± 0.39	0.42 ± 0.11
		3	0.91 ± 0.04	1.15 ± 0.24	**1.23 ± 0.12**	**1.31 ± 0.33**	**0.41 ± 0.09**	9.63 ± 1.67	0.00 ± 0.42	0.42 ± 0.12
SSC (%), SEL = 0.1% **SpectraNet6-53**	IV	1	1.02 ± 0.18	1.06 ± 0.09	**1.36 ± 0.21**	**1.32 ± 0.20**	**0.48 ± 0.12**	8.87 ± 0.55	−0.20 ± 0.24	0.41 ± 0.10
		2	1.02 ± 0.12	1.11 ± 0.14	**1.32 ± 0.10**	**1.29 ± 0.15**	**0.43 ± 0.08**	9.37 ± 1.02	−0.09 ± 0.20	0.41 ± 0.06
		3	1.00 ± 0.08	1.15 ± 0.11	**1.29 ± 0.08**	**1.26 ± 0.11**	**0.40 ± 0.07**	9.76 ± 0.79	−0.02 ± 0.19	0.39 ± 0.05
		4	1.01	1.16	**1.30**	**1.24**	**0.40**	9.92	0.05	0.37
	EV	1	1.02 ± 0.16	1.18 ± 0.24	**1.20 ± 0.09**	**1.25 ± 0.37**	**0.38 ± 0.08**	9.87 ± 1.64	−0.05 ± 0.37	0.37 ± 0.10
		2	1.02 ± 0.10	1.15 ± 0.24	**1.24 ± 0.09**	**1.31 ± 0.34**	**0.40 ± 0.08**	9.58 ± 1.63	−0.06 ± 0.35	0.39 ± 0.08
		3	1.02 ± 0.06	1.15 ± 0.27	**1.23 ± 0.11**	**1.32 ± 0.35**	**0.41 ± 0.08**	9.64 ± 1.87	−0.05 ± 0.42	0.40 ± 0.09

Notes. N is the number of orchard–year pairs used for training, SEL the standard error of laboratory. The other abbreviations are explained in Section 4. All values correspond to a mean ± standard deviation.
In bold: values attaining a minimum standard of model performance, consisting of SDR > 1, PG > 1 or $R^2 > 0.16$ (meaning that $|R| > 0.4$).

a) the inputs are scaled and shifted automatically during training, as learnable hyperparameters, which can result in a dataset-specific standardization that is more fine-tuned and beneficial to the overall architecture; and
b it is a form of data augmentation, as the same input will slightly change across mini-batches, due to the per-batch normalization statistics.

Fig. 10(a) shows the effect of this step after the QNV–5 transform used in all trained networks, and Fig. 10(b) exemplifies what would happen if SNV was used instead – a trained network for this case only slightly adjusted the zero mean and unity standard deviation of its inputs, while the QNV–5 transform after BN shown in Fig. 10(a) is visibly different than Fig. 9(d).



(a) SpectraNet6–53 internal validation.

(b) SpectraNet6–53 external validation.

Fig. 13. Regression scatter plots, using *representative* networks – for each of the 43 sets (15 IV and 28 EV), a representative neural network was selected out of the 30 in each set, which was the one with an rmsep value closest to the median of the set. Statistics are computed across all representative networks. Each marker color corresponds to a different set.

3.2.4. *Multi-objective learning*

CNN-based deep neural networks have previously shown improved results on simultaneously predicting multiple outputs instead of only a single one. They are able to create shared representations of information, using internal correlations between targets as cues during prediction (Padarian et al., 2019b; Ramsundar et al., 2015), as well as increasing prediction accuracy by reducing overfitting (Ruder, 2017). The performance of SpectraNet–53 when trained with only one output (SSC) will be compared to the improvement effect on SSC when trained with five additional outputs, as described below.

Table 3
SSC results comparison with Cavaco et al. (2018).

			PLS		SpectraNet6-53		% Improvement	
		N	Mean	Std	Mean	Std	Mean	Std
rmsec	IV	1	0.76	0.12	1.02	0.18	–	–
		2	0.80	0.09	1.02	0.12	–	–
		3	0.84	0.05	1.00	0.08	–	–
		4	0.86	–	1.01	–	–	–
	EV	1	0.82	0.17	1.02	0.16	–	–
		2	0.82	0.09	1.02	0.10	–	–
		3	0.85	0.08	1.02	0.06	–	–
rmsep	IV	1	1.06	0.21	1.06	0.09	0%	133%
		2	1.02	0.12	1.11	0.14	−8%	−14%
		3	1.01	0.07	1.15	0.11	−12%	−36%
		4	1.00	–	1.16	–	−14%	–
	EV	1	**1.27**	0.20	**1.18**	0.24	**8%**	**−17%**
		2	**1.17**	0.15	**1.15**	0.24	**2%**	**−38%**
		3	**1.15**	0.17	**1.15**	0.27	**0%**	**−37%**
SDR	IV	1	1.27	0.20	1.36	0.21	7%	−5%
		2	1.36	0.13	1.32	0.10	−3%	30%
		3	1.39	0.09	1.29	0.08	−7%	13%
		4	1.40	–	1.30	–	−7%	–
	EV	1	**1.05**	0.17	**1.20**	0.09	**14%***	**89%**
		2	**1.11**	0.14	**1.24**	0.09	**12%***	**56%**
		3	**1.13**	0.14	**1.23**	0.11	**9%**	**27%**
R^2	IV	1	0.39	0.20	0.48	0.12	23%	67%
		2	0.47	0.09	0.43	0.08	−9%	13%
		3	0.48	0.06	0.40	0.07	−17%	−14%
		4	0.49	–	0.40	–	−18%	–
	EV	1	**0.26**	0.14	**0.38**	0.08	**46%***	**75%**
		2	**0.33**	0.14	**0.40**	0.08	**21%**	**75%**
		3	**0.34**	0.13	**0.41**	0.08	**21%**	**63%**
Bias	IV	1	0.02	0.03	−0.20	0.24	–	–
		2	0.00	0.02	−0.09	0.20	–	–
		3	−0.01	0.01	−0.02	0.19	–	–
		4	0.00	–	0.05	–	–	–
	EV	1	0.11	0.50	−0.05	0.37	–	–
		2	−0.05	0.48	−0.06	0.35	–	–
		3	−0.03	0.50	−0.05	0.42	–	–

Notes. The improvement percentage is calculated by dividing the respective columns and subtracting 1. The numerator and denominator are chosen based on the direction of improvement: for rmsec and rmsep means, and std columns, lower values are better; for sdr and R^2 means, higher values are better.
For EV: green and red values represent positive or negative double-digit (and yellow values represent single-digit) percentage differences. The values marked by * have significantly different means at $p < 0.05$, when compared by a Welch's t-test Delacre et al., 2017.

3.2.5. *Network hyperparameters*

The next paragraphs briefly describe the network hyperparameters chosen for training.

Output variables: Two architectures were trained, SpectraNet–53 with only SSC as output, and SpectraNet6–53 with five additional outputs: Fruit Temperature, Mass, Equatorial Diameter, pH and the inverse of the Titratable Acidity (TA^{-1}) – as MI $= \frac{SSC}{TA}$, the uncorrelated TA^{-1} is used instead. These are depicted in Fig. 11, with SSC having significant correlations with the first five IQAs, respectively, and indirectly with TA^{-1}.

Output scaling: All the outputs were standard-scaled (SNV).

Absorvance: As previously stated (Section 2.1), all reflectance spectra were converted into absorbance-like values, such that $x_A = log_{10}(\frac{1}{x_R} + 0.1)$.

Input scaling: The input data was normalized using a Quantile Normal Variate transform (Section 3.2.2) with 5 quantiles (QNV–5). As seen in Fig. 9(d), this yielded an average signal 24% higher than SNV, with a 52% higher standard deviation. QNV–5

Table 4
SpectraNet–53 Architecture.

Layer	Type	Activations	Learnable Parameters		Total Param.
	Input	1 × 1421 × 1	—		0
1	Batch Normalization	1 × 1421 × 1	Offset: 1 × 1	Scale: 1 × 1	2
2	Convolution	1 × 1421 × 32	Weights: 1 × 17 × 1 × 32	Bias: 1 × 1 × 32	576
3	Batch Normalization	1 × 1421 × 32	Offset: 1 × 1 × 32	Scale: 1 × 1 × 32	64
4	GELU	1 × 1421 × 32	Mean: 1 × 1 × 32	Std: 1 × 1 × 32	64
5	S1U1 Convolution	1 × 1421 × 32	Weights: 1 × 17 × 32 × 32	Bias: 1 × 1 × 32	17.440
6	S1U1 Batch Normalization	1 × 1421 × 32	Offset: 1 × 1 × 32	Scale: 1 × 1 × 32	64
7	S1U1 GELU	1 × 1421 × 32	Mean: 1 × 1 × 32	Std: 1 × 1 × 32	64
8	S1U1 Convolution	1 × 1421 × 32	Weights: 1 × 17 × 32 × 32	Bias: 1 × 1 × 32	17.440
9	S1U1 Batch Normalization	1 × 1421 × 32	Offset: 1 × 1 × 32	Scale: 1 × 1 × 32	64
10	Addition (L9 + L4)	1 × 1421 × 32	—		0
11	GELU	1 × 1421 × 32	Mean: 1 × 1 × 32	Std: 1 × 1 × 32	64
12	S1U2 Convolution	1 × 1421 × 32	Weights: 1 × 17 × 32 × 32	Bias: 1 × 1 × 32	17.440
13	S1U2 Batch Normalization	1 × 1421 × 32	Offset: 1 × 1 × 32	Scale: 1 × 1 × 32	64
14	S1U2 GELU	1 × 1421 × 32	Mean: 1 × 1 × 32	Std: 1 × 1 × 32	64
15	S1U2 Convolution	1 × 1421 × 32	Weights: 1 × 17 × 32 × 32	Bias: 1 × 1 × 32	17.440
16	S1U2 Batch Normalization	1 × 1421 × 32	Offset: 1 × 1 × 32	Scale: 1 × 1 × 32	64
17	Addition (L16 + L11)	1 × 1421 × 32	—		0
18	GELU	1 × 1421 × 32	Mean: 1 × 1 × 32	Std: 1 × 1 × 32	64
19	S2U1 Convolution (stride 2)	1 × 711 × 64	Weights: 1 × 17 × 32 × 64	Bias: 1 × 1 × 64	34.880
20	S2U1 Batch Normalization	1 × 711 × 64	Offset: 1 × 1 × 64	Scale: 1 × 1 × 64	128
21	S2U1 GELU	1 × 711 × 64	Mean: 1 × 1 × 64	Std: 1 × 1 × 64	128
22	S2U1 Convolution	1 × 711 × 64	Weights: 1 × 17 × 64 × 64	Bias: 1 × 1 × 64	69.696
23	S2U1 Batch Normalization	1 × 711 × 64	Offset: 1 × 1 × 64	Scale: 1 × 1 × 64	128
24	*Skip Conn.* — Convolution (input L18, stride 2)	1 × 711 × 64	Weights: 1 × 1 × 32 × 64	Bias: 1 × 1 × 64	2.112
25	*Skip Conn.* — Batch Normalization	1 × 711 × 64	Offset: 1 × 1 × 64	Scale: 1 × 1 × 64	128
26	Addition (L23 + *L25*)	1 × 711 × 64	—		0
27	GELU	1 × 711 × 64	Mean: 1 × 1 × 64	Std: 1 × 1 × 64	128
28	S2U2 Convolution	1 × 711 × 64	Weights: 1 × 17 × 64 × 64	Bias: 1 × 1 × 64	69.696
29	S2U2 Batch Normalization	1 × 711 × 64	Offset: 1 × 1 × 64	Scale: 1 × 1 × 64	128
30	S2U2 GELU	1 × 711 × 64	Mean: 1 × 1 × 64	Std: 1 × 1 × 64	128
31	S2U2 Convolution	1 × 711 × 64	Weights: 1 × 17 × 64 × 64	Bias: 1 × 1 × 64	69.696
32	S2U2 Batch Normalization	1 × 711 × 64	Offset: 1 × 1 × 64	Scale: 1 × 1 × 64	128
33	Addition (L32 + L27)	1 × 711 × 64	—		0
34	GELU	1 × 711 × 64	Mean: 1 × 1 × 64	Std: 1 × 1 × 64	128
35	S3U1 Convolution (stride 2)	1 × 356 × 128	Weights: 1 × 17 × 64 × 128	Bias: 1 × 1 × 128	139.392
36	S3U1 Batch Normalization	1 × 356 × 128	Offset: 1 × 1 × 128	Scale: 1 × 1 × 128	256
37	S3U1 GELU	1 × 356 × 128	Mean: 1 × 1 × 128	Std: 1 × 1 × 128	256
38	S3U1 Convolution	1 × 356 × 128	Weights: 1 × 17 × 128 × 128	Bias: 1 × 1 × 128	278.656
39	S3U1 Batch Normalization	1 × 356 × 128	Offset: 1 × 1 × 128	Scale: 1 × 1 × 128	256
40	*Skip Conn.* — Convolution (input L34, stride 2)	1 × 356 × 128	Weights: 1 × 1 × 64 × 128	Bias: 1 × 1 × 128	8.320
41	*Skip Conn.* — Batch Normalization	1 × 356 × 128	Offset: 1 × 1 × 128	Scale: 1 × 1 × 128	256
42	Addition (L39 + *L41*)	1 × 356 × 128	—		0
43	GELU	1 × 356 × 128	Mean: 1 × 1 × 128	Std: 1 × 1 × 128	256
44	S3U2 Convolution	1 × 356 × 128	Weights: 1 × 17 × 128 × 128	Bias: 1 × 1 × 128	278.656
45	S3U2 Batch Normalization	1 × 356 × 128	Offset: 1 × 1 × 128	Scale: 1 × 1 × 128	256
46	S3U2 GELU	1 × 356 × 128	Mean: 1 × 1 × 128	Std: 1 × 1 × 128	256
47	S3U2 Convolution	1 × 356 × 128	Weights: 1 × 17 × 128 × 128	Bias: 1 × 1 × 128	278.656
48	S3U2 Batch Normalization	1 × 356 × 128	Offset: 1 × 1 × 128	Scale: 1 × 1 × 128	256
49	Addition (L48 + L43)	1 × 356 × 128	—		0
50	GELU	1 × 356 × 128	Mean: 1 × 1 × 128	Std: 1 × 1 × 128	256
51	Global Average Pooling	1 × 1 × 128	—		0
52	Dropout (20%)	1 × 1 × 128	—		0
53	Fully Connected	1 × 1 × 6	Weights: 6 × 128	Bias: 6 × 1	774
	MSE evaluation	—	—		0

Table 5
Summary of the SSC performance results obtained in IV and EV.

		N	rmsec	rmsep	SDR	PG	R^2	CV(%)	Bias	Slope
Temperature	IV	1	2.87 ± 0.55	3.15 ± 1.14	**1.11 ± 0.02**	**1.15 ± 0.23**	**0.21 ± 0.05**	16.60 ± 5.92	−0.33 ± 0.77	0.20 ± 0.06
(°C)		2	3.06 ± 0.34	3.35 ± 0.46	**1.13 ± 0.05**	**1.12 ± 0.17**	**0.21 ± 0.07**	17.71 ± 2.37	−0.20 ± 0.48	0.20 ± 0.07
		3	3.11 ± 0.27	3.30 ± 0.21	**1.15 ± 0.05**	**1.15 ± 0.14**	**0.23 ± 0.06**	17.49 ± 1.02	0.00 ± 0.33	0.22 ± 0.06
		4	3.16	3.23	**1.16**	**1.18**	**0.25**	17.19	0.17	0.22
	EV	1	2.91 ± 0.48	3.82 ± 0.45	0.91	0.94	**0.19 ± 0.11**	20.19 ± 2.48	−0.29 ± 2.02	0.19 ± 0.11
		2	3.09 ± 0.22	3.61 ± 0.30	0.96	**1.04 ± 0.16**	**0.21 ± 0.11**	19.08 ± 1.75	−0.29 ± 1.72	0.21 ± 0.08
		3	3.09 ± 0.14	3.52 ± 0.31	0.98	**1.08 ± 0.12**	**0.24 ± 0.14**	18.63 ± 1.86	−0.26 ± 1.82	0.23 ± 0.07
Mass	IV	1	79.95 ± 16.32	83.34 ± 10.58	**1.06 ± 0.07**	**1.10 ± 0.14**	0.12	27.42 ± 4.24	−9.92 ± 8.44	0.10 ± 0.06
(g)		2	81.51 ± 7.40	87.24 ± 8.17	**1.06 ± 0.04**	**1.09 ± 0.10**	0.13	28.24 ± 3.35	−7.68 ± 15.67	0.13 ± 0.07
		3	81.98 ± 4.80	88.73 ± 1.80	**1.07 ± 0.02**	**1.08 ± 0.06**	0.14	28.57 ± 1.96	−8.16 ± 15.76	0.14 ± 0.05
		4	83.23	86.89	**1.08**	**1.11**	0.13	28.08	−3.91	0.13
	EV	1	79.92 ± 13.91	95.15 ± 20.14	0.97	**1.01 ± 0.34**	0.14	31.06 ± 5.62	−21.86 ± 34.36	0.14 ± 0.10
		2	81.31 ± 6.54	93.82 ± 18.37	0.98	**1.06 ± 0.30**	0.14	30.66 ± 5.19	−11.62 ± 38.46	0.15 ± 0.09
		3	82.23 ± 4.14	91.98 ± 20.31	1.00 ± 0.08	**1.10 ± 0.35**	0.14	30.01 ± 5.49	−7.75 ± 40.01	0.13 ± 0.07
Equatorial Diameter	IV	1	7.43 ± 1.30	7.40 ± 1.47	**1.07 ± 0.08**	**1.17 ± 0.14**	0.12	8.88 ± 1.97	0.43 ± 0.80	0.12 ± 0.05
(mm)		2	7.59 ± 0.54	7.83 ± 1.18	**1.05 ± 0.03**	**1.12 ± 0.14**	0.11	9.35 ± 1.65	0.22 ± 1.66	0.12 ± 0.05
		3	7.61 ± 0.32	8.03 ± 0.84	**1.05 ± 0.02**	**1.09 ± 0.13**	0.11	9.56 ± 1.23	−0.12 ± 1.60	0.11 ± 0.05
		4	7.69	7.96	**1.06**	**1.10**	0.10	9.47	−0.16	0.09
	EV	1	7.37 ± 1.18	9.07 ± 1.52	0.94	0.96	0.12	10.85 ± 1.90	−0.66 ± 4.45	0.15 ± 0.08
		2	7.52 ± 0.55	8.80 ± 1.55	0.97	**1.01 ± 0.25**	0.11	10.53 ± 1.94	−0.28 ± 3.98	0.12 ± 0.08
		3	7.58 ± 0.34	8.55 ± 1.66	0.99	**1.05 ± 0.28**	0.10	10.23 ± 2.08	−0.21 ± 3.69	0.10 ± 0.06
Juice pH	IV	1	0.26 ± 0.05	0.23 ± 0.04	**1.22 ± 0.13**	**1.39 ± 0.32**	**0.35 ± 0.14**	7.00 ± 1.18	0.02 ± 0.04	0.29 ± 0.08
(pH)		2	0.26 ± 0.02	0.27 ± 0.04	**1.16 ± 0.08**	**1.20 ± 0.22**	**0.27 ± 0.11**	8.16 ± 1.09	0.02 ± 0.05	0.25 ± 0.11
SEL = 0.07		3	0.26 ± 0.01	0.29 ± 0.01	**1.15 ± 0.03**	**1.13 ± 0.09**	**0.26 ± 0.06**	8.51 ± 0.40	0.00 ± 0.04	0.23 ± 0.06
		4	0.27	0.28	**1.17**	**1.17**	**0.28**	8.27	−0.02	0.22
	EV	1	0.25 ± 0.03	0.34 ± 0.05	0.92	0.92	**0.18 ± 0.08**	10.22 ± 1.29	0.02 ± 0.21	0.16 ± 0.05
		2	0.26 ± 0.02	0.33 ± 0.04	0.95	0.98	**0.20 ± 0.10**	9.87 ± 1.21	0.02 ± 0.19	0.18 ± 0.06
		3	0.26 ± 0.01	0.33 ± 0.04	0.95	0.99	**0.21 ± 0.11**	9.83 ± 1.24	0.01 ± 0.21	0.19 ± 0.07
TA^{-1}	IV	1	0.28 ± 0.03	0.30 ± 0.05	**1.12 ± 0.13**	**1.14 ± 0.21**	**0.22 ± 0.13**	28.58 ± 3.58	0.04 ± 0.09	0.24 ± 0.13
(100 mL/g citric acid)		2	0.29 ± 0.03	0.32 ± 0.04	**1.15 ± 0.04**	**1.14 ± 0.13**	**0.26 ± 0.09**	29.69 ± 1.24	0.04 ± 0.07	0.24 ± 0.09
SEL = 100 mL/0.03 g		3	0.30 ± 0.03	0.33 ± 0.02	**1.17 ± 0.05**	**1.17 ± 0.07**	**0.29 ± 0.08**	29.79 ± 0.65	0.03 ± 0.05	0.24 ± 0.09
		4	0.32	0.33	**1.16**	**1.19**	**0.26**	29.58	0.01	0.22
	EV	1	0.28 ± 0.03	0.43 ± 0.11	0.83	0.83	40.78 ± 11.64	40.78 ± 11.64	0.17 ± 0.06	0.17 ± 0.06
		2	0.29 ± 0.03	0.42 ± 0.08	0.84	0.91	39.45 ± 6.82	39.45 ± 6.82	0.17 ± 0.08	0.17 ± 0.08
		3	0.30 ± 0.03	0.42 ± 0.09	0.84	0.94	39.52 ± 6.45	39.52 ± 6.45	0.16 ± 0.03	0.16 ± 0.03

Notes. N is the number of orchard–year pairs used for training, SEL the standard error of laboratory. The other abbreviations are explained in Section 4. All values correspond to a mean ± standard deviation.
In bold: values attaining a minimum standard of model performance, consisting of SDR > 1, PG > 1 or $R^2 > 0.16$ (meaning that $|R| > 0.4$).

is a conservative parameter choice that still achieves a good separation of the spectra at wavelengths 960–984 nm, which chemically correspond to the 2nd overtone of OH stretching.

Solver: Standard ADAM was used, with default parameters.

Learning Rate: 10^{-4}, with a drop multiplier of 0.7 (−30%) at each epoch.

Epochs: All the results presented in the following section are from networks trained with 10 epochs, which was generally the minimum value where the training error starts to stabilize for most networks. This is a good early stop point, as network hyperparameters are higher than the size of the dataset, so keeping a low number of epochs helps to prevent overfit. Still, in Appendix A this topic is developed further, with networks trained for many more epochs, until the training loss stabilizes (*i.e.*, stops decreasing for 10 epochs). This is helpful in determining if the architecture has an innate tendency to overfit to the training data, or the architectural choices were enough to prevent that effect.

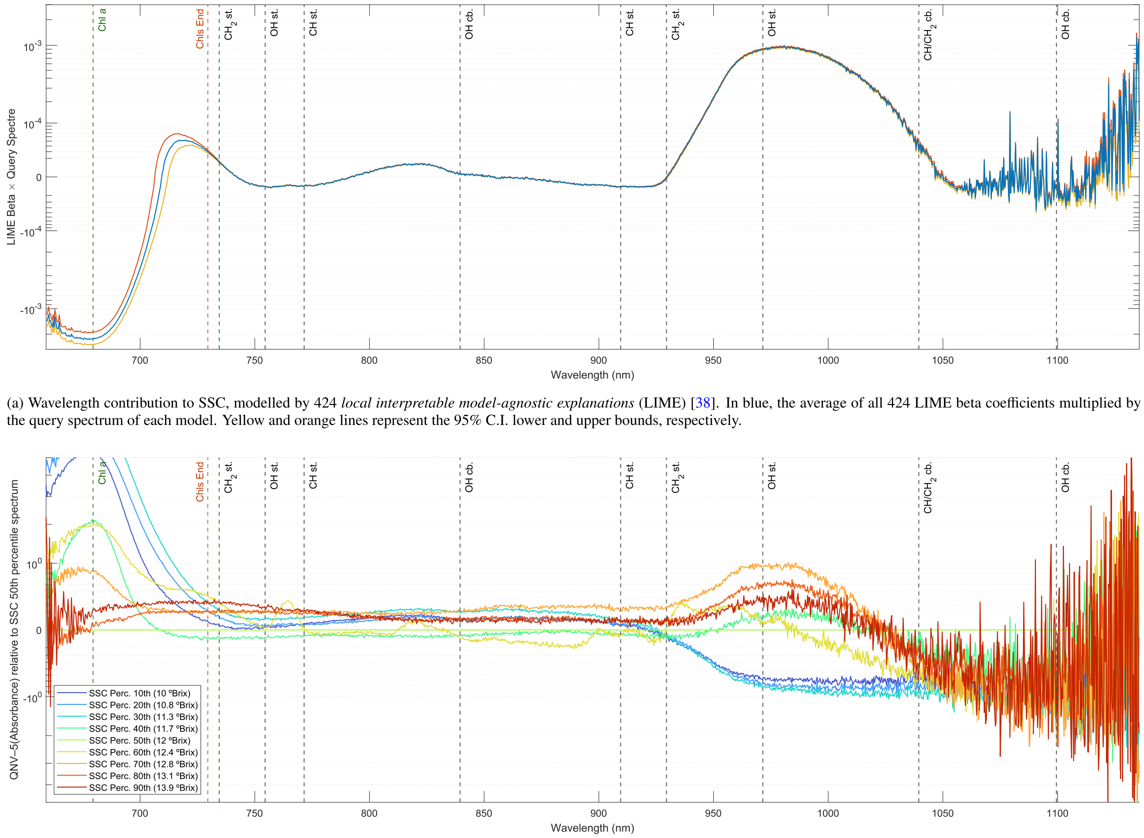
Mini-batch size: The training mini-batch size was set at 32. Higher values would most likely reduce the benefit of using BN after input for data augmentation, due to the low amount of training samples for each set, as inter-batch statistics would tend to be very similar. Thus, a low mini-batch size increases the number of different input mini-batches for each epoch, taking advantage of the benefits described in Section 3.2.3.

Convolutional layers: a 17-pt filter window was used in all convolutional layers. For the first layer, each filter window captures 5.7 nm of spectral data. The optimal value for this parameter is expected to depend on the wavelength-separation of each dataset, the number of stacked RUs, or even on the IQA being assessed (i.e., it is feature specific).

Dropout layer: L52 had a drop probability value of 20%.

L2 Regularization: Yes, with a coefficient $\lambda = 0.05$. L2 Regularization (also commonly referred as *weight decay*) adds a regularizing term to the weights of the loss function, reducing the overfit possibility from a large gradient descent update. If $L(x)$ is the expected loss function, then $L_R(x) = L(x) + \lambda\Omega(W)$, with $\Omega(W) = \frac{1}{2}W^TW$.

Gradient Clipping: Yes, as another overfitting prevention measure. For each learnable parameter, any L2–norm of gradients higher than 0.5 were truncated, so that any L2–norm $\leqslant 0.5$. This avoids gradient explosion by stabilizing and allowing training



(a) Wavelength contribution to SSC, modelled by 424 *local interpretable model-agnostic explanations* (LIME) [38]. In blue, the average of all 424 LIME beta coefficients multiplied by the query spectrum of each model. Yellow and orange lines represent the 95% C.I. lower and upper bounds, respectively.

(b) Spectra distribution by SSC percentiles, for the 424 training spectra of the EV–3 ABD/C. Blue to red lines represent the closest spectra to each SSC percentile, from the 10th to 90th. Data is normalized by subtracting the 50th percentile spectra. Vertical dashed lines represent Chlorophyll and sugar bands.

Fig. 14. Wavelength contribution for SSC prediction, explained using LIME (Ribeiro et al., 2016).

at initial higher learning rates, while reducing the vulnerability to outlier gradients.

Layer initialization: Convolutional layers are initialized with *He* weights (He et al., 2015), and the final fully connected layer (L53) is initialized with *Glorot* weights (Glorot and Bengio, 2010).

4. Results and discussion

Two variations of the DeepSpectra network were trained: one with a single output variable (SSC), further referred as *DeepSpectra*, and another with the same 6 outputs of SpectraNet6–53, named as *DeepSpectra6*. This allows for a direct comparison between all four architecture types, as the later case has effectively the same inputs and outputs as SpectraNet6–53.

A total of 5160 networks were trained, with 30 networks for each of the 15 IV and 28 EV sets, times the 4 architectures. Fig. 12 shows the distribution of SSC results for the root-mean-squared error of prediction (rmsep), for both IV (a) and EV (b). DeepSpectra and DeepSpectra6 had generally worse results than both SpectraNets, especially for sets trained on A or D spectra. The benefit of 6 outputs in DeepSpectra6 was mostly evident on difficult EV cases, especially in sets trained on D and tested on A spectra. However, it had mixed results for IV sets, where DeepSpectra performance was generally better. Overall, SpectraNet–53 and SpectraNet6–53 SSC rmsep values were the lowest and most consistent across sets, and the networks of both architectures only very slightly diverge from one another in this performance parameter. However, SpectraNet6–53 is notably better on EV sets trained on D spectra.

Table 2 shows the key performance metrics on the following parameters: **N** – number of orchard–year pairs in the training set; **rmsec** – root mean squared (rms) error in the training (calibration) set; **rmsep (iv/eV)** – rms error of prediction in the test (validation) set; **sdr** – standard deviation ratio $= \frac{\mathrm{std}(y)}{\mathrm{rmsep}}$, where y represents the reference test data; **pg** – prediction gain $= \frac{\mathrm{std}(y')}{\mathrm{rmsep}}$, where y' represents the reference training data; $\mathbf{R^2}$ – squared correlation coefficient, *i.e.*, coefficient of determination; **cv** – coefficient of variation (%), defined as $\frac{\mathrm{rmsep}}{\mathrm{mean}(y)} \times 100$; **bias** – the $\mathrm{mean}(\hat{y}) - \mathrm{mean}(y)$, where $\hat{y}$ represents the predicted test data; **slope** – the slope of the linear regression in y *vs.* $\hat{y}$. Each value represents a mean of means ± the standard deviation of the means, *i.e.*, for each set, the results of all 30 networks are averaged and then those averages are used to determine the mean and standard deviation inside each of the orchard–year $\{1,2,3,4\}$–pairs used for IV and $\{1,2,3\}$–pairs used for EV, as detailed in Section 2.3 and Table 1. A minimum standard of model performance was set as SDR > 1 (*i.e.*, predictions are better than a random guess around the test population mean), PG > 1 (*i.e.*, predictions are better than a random guess around the training population mean) and $R^2 > 0.16$ (*i.e.*, $|R| > 0.4$ meaning there is some moderate correlation between predicted and true values). Only SSC results are shown for DeepSpectra6 and SpectraNet6–53, as the predictions for the remaining IQAs are mostly of average quality (the remaining SpectraNet6–53 results are included in Table 5, at the end of the article). This is not unexpected, as the network architecture and training parameters were focused on finding good SSC-related descriptors in spectra, and it was

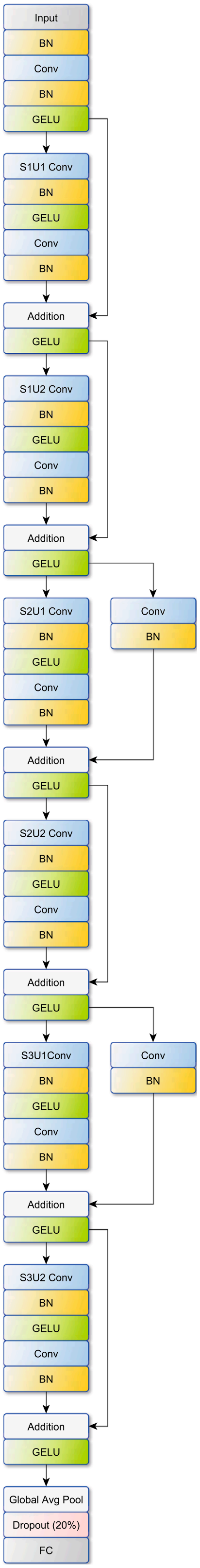


Fig. 15. Architecture.

not guaranteed that those descriptors would be appropriate predictors for other IQAs.

Analyzing Table 2, all architectures are able to surpass the minimum standards of model performance, shown in bold. SpectraNet6–53 showed the best overall performance, with the lowest rmsep values for EV. SDR and PG values are well above 1 for both IV and EV, with SDR and R^2 variation being especially stable at IV 2–4 and EV. R^2 values were around 40% in IV and EV. Bias was mostly negative, suggesting that the network has a tendency to underestimate SSC predictions.

Fig. 13 displays SpectraNet6–53 scatter plots for all 15 IV and 28 EV predictions, made with the *representative* neural network of each set: for each of the 43 sets, out of the 30 possible networks, the network with an rmsep value closest to the median the set was selected. Aggregated statistics across all networks are also shown. Additional performance metrics are shown for the following parameters: **res** – resolution, defined as $\frac{\text{rmsep}}{\max(y)-\min(y)} \times 100$, which is a measure of how the prediction error resolves the range of y–variation; **N** – the total number of IV or EV points in each figure; **%OK** – the percentage of test samples which are correctly assigned below or above the population mean (shown as **y**); this value is further detailed in **% TN** and **%TP**, representing the percentages of correct assignments below (True Negative) or above (True Positive) the population mean.

Both IV and EV conditions display comparable metrics across all sets, with slightly better performance on the IV case, as expected.

4.1. Comparison with previous works

Table 3 shows a direct comparison with Cavaco et al. (2018). The training rmsec values are not very meaningful (*i.e.*, they might signify overfit) and are only shown for completeness. The IV results are also not very interesting to compare to, as PLS is very good for prediction when built upon few, but representative samples of a population (especially after filtering outliers) – a neural network would need more samples to beat its performance. However, EV results require each method to create some kind of an internal representation of knowledge, applicable to samples that may come from different populations other than the ones previously trained upon, which can be quite hard on classical models. Thus, analysing SpectraNet6–53 EV performance:

a) For rmsep, both methods had similar means, with a slight SpectraNet6–53 advantage; however it had worse standard deviations. This is expected, due to the outlier spectra present in the SpectraNet–53 trials.
b) SDR had an increase between 9 and 14% on the mean values (significantly higher on EV–1 and 2), and between 27 to 89% on the standard deviation, which is a solid improvement over PLS – especially considering that no outlier spectra (and corresponding SSC values) were discarded. This results in a higher $\text{std}(y)$, as the rmsep values are very close for both methods, particularly for the EV–2 and 3 sets.
c) R^2 was higher than PLS, with mean values between 21 to 46% higher (a very significant increase on EV–1), and a reduction of 63 to 75% on the standard deviation.

In summary, when considering average external validation results, SpectraNet6–53 was 3.1% better than PLS on RMSEp (1.16 *vs.* 1.20), 11.6% better in SDR (1.22 *vs.* 1.10), and 28.0% better in R^2 (0.40 *vs.* 0.31).

4.2. Wavelength contribution for SSC prediction

From a physical and chemical standpoint, good prediction models are also important in allowing some understanding of the specific wavelengths that contribute for predicting SSC. To this end, a *local interpretable model-agnostic explanations* (LIME) (Ribeiro et al., 2016) method was used. LIME can generate a synthetic dataset from a query vector, with a chosen number of important predictors. To this end, a SpectraNet–53 representative network of the EV–3 ABD/C set (network #27) was used, which had 424 training spectra. This was the network with the lowest error in EV–3, as shown in Fig. 12(b). For obtaining a comprehensive representation of the wavelength contribution, 424 LIME models were fitted, one for each ABD spectra (*i.e.*, each is a query vector). The models used 1421 predictors (wavelengths), and LIME generated 5000 synthetic spectra per query vector. Next, the beta coefficients of the resulting in 424 LIME models were multiplied by their respective query vector, represented by the blue line in Fig. 14(a). Per-wavelength 95% C.I. were also determined, shown as yellow and orange, for the lower and upper values, respectively. The most important wavelengths for SSC will have higher absolute values and tighter C.I. ranges. Marked on the image are the Chlorophyll bands, and the most important bands for sugars, which account for the majority of SSC. These are related to OH and CH vibrations (Golic et al., 2003), namely: **band 1** – composed of the 3rd overtone of OH stretching, from 740 nm (less H–bonded) to 770 nm (more H–bonded), represented at 755 nm, and the 4th overtones of stretching vibrations of CH (772 nm) and CH_2 (730–740 nm), represented at 735 nm; **band 2** – 2nd overtone of the OH combination band (stretching and bending) at 840 nm; **band 3** – 3rd overtone of CH stretching at 910 nm; **band 4** – 3rd overtone of CH_2 stretching at 930 nm; **band 5** – 2nd overtone of OH stretching represented at 972 nm, ranging from 960 nm (less H–bonded) to 984 nm (more H–bonded); **band 6** – CH and CH_2 combination band at 1040 nm; **band 7** – 1st overtone of the OH combination band (stretching and bending) at 1100 nm.

Of notice are the wavelenghts below 709 nm (zero crossing), especially the Chlorophyll-a peak at 680 nm, which tends to correlate inversely with SSC, while the most positive contributions are those between 930 nm (CH_2 st.) and 1050 nm, which include OH st. circa 972 nm, and the CH/CH_2 cb. at 1040 nm.

5. Conclusion and final remarks

This article presents a novel 1D residual deep learning neural network architecture for fruit spectra IQA prediction, namely SSC. Its performance was evaluated on a small 'Newhall' orange dataset, and assessed under various validation conditions, as well as compared to state-of-the-art methods in the field. The SpectraNet architecture was able to achieve two very uncommon results: (a) achieving state-of-the-art results on a very small dataset, with as low as 125 training samples on a network with 1.3 M parameters; and (b) extensive training of the architecture (shown in Appendix A) demonstrated its ability to consistently achieve good performance results, without overfitting to training data.

This research is also an interesting contribution to the state-of-the-art in 1D Convolutional Neural Network design, which usually relies on very restrict architectures to prevent overfitting, typically with only one or two hidden layers and less than 10 K parameters. The presented architecture had 53 layers with 1.304.968 parameters, and was still able to generalize properly.

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.

Acknowledgements

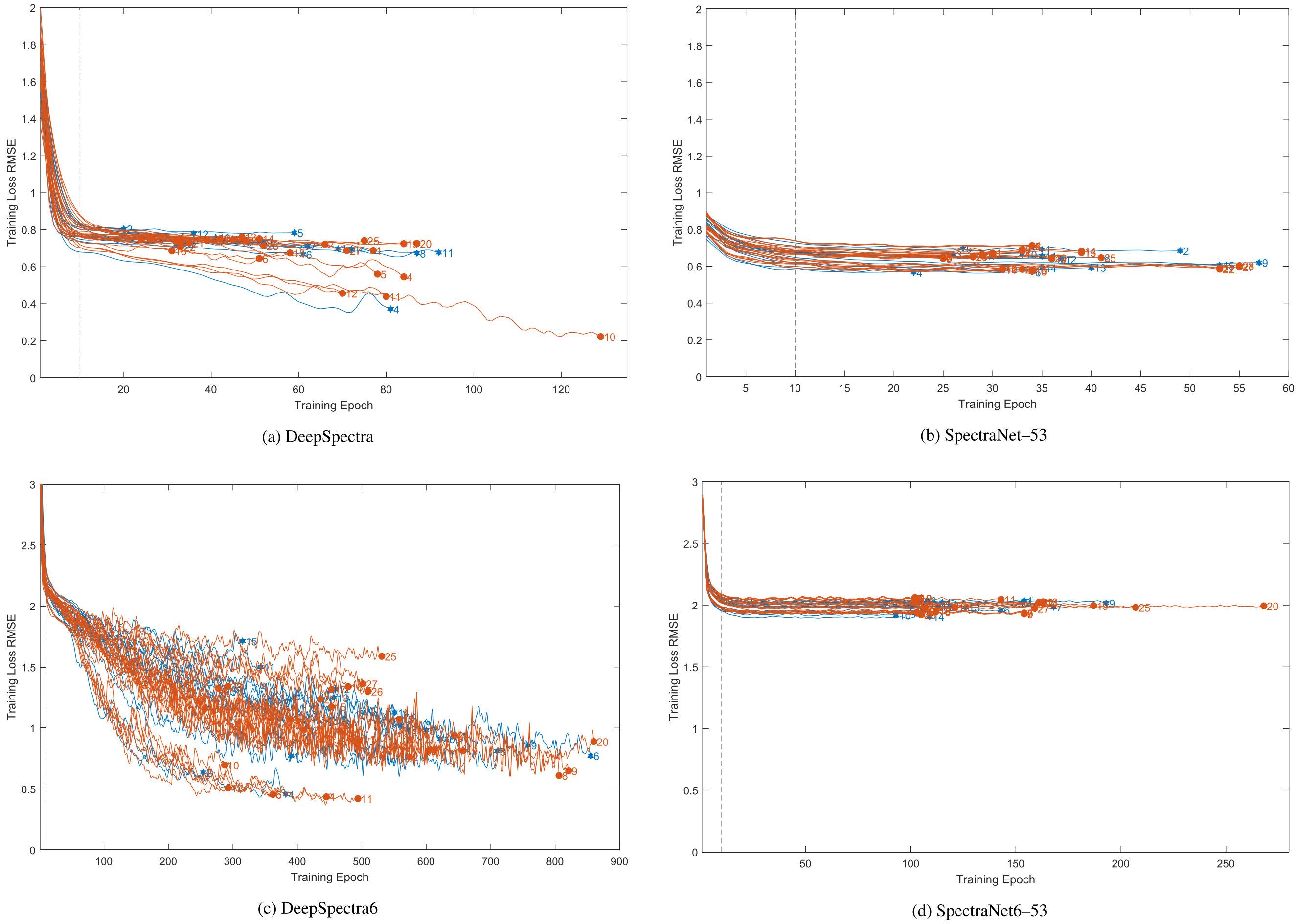
This work was developed within the Centre for Electronics, Optoelectronics and Telecommunications (CEOT), and supported by the Portuguese Science and Technology Foundation (FCT), within the post-doc fellowship SFRH/BPD/101634/2014 to A. M. Cavaco, national funds of DL57/2016/CP1361/CT0040 to J. A. Martins and projects UID/MULTI/00631/2019, UIDP/00631/2020 CEOT PROGRAMÁTICO and UIDB/00631/2020 CEOT BASE. Part of the work was also supported by projects NIBAP (ALG-01-0247-FEDER-037303) and OtiCalFrut (ALG-01-0247-FEDER-033652).

Appendix A. Training on many epochs

For evaluating the overfit tendency of the networks discussed in this article, a new batch of 5160 networks was trained, again with 30 for each of the 15 IV and 28 EV sets. This time, the networks were trained until either: (I) a maximum of 1000 epochs was reached, or (II) the training loss stopped decreasing, which was determined by a moving window (of 10 consecutive epochs) where the minimum training error of the window did not decrease. The last network trained was used for prediction.

Fig. A.16 shows the training loss rmse for the four architectures, with a dashed vertical line on the 10 epochs, which was the stopping criteria used previously. There is a clear difference between DeepSpectra and SpectraNet training loss: (a) DeepSpectra has a tendency to require many more epochs to stabilize, and (b) the training loss is sometimes prone to oscillations, most especially for DeepSpectra6. On the other hand, SpectraNet quickly stabilizes its training error, suggesting it is not prone to overfit on the training data, even after many epochs.

Fig. A.17 compares the networks trained on 10 epochs *vs.* on "max epochs". DeepSpectra networks display mixed results: some sets have a lower SSC prediction error, while others increase. For SpectraNets, the results are extremely similar between conditions, once again suggesting that the architecture appears to be resilient to overfit. Table A.6 displays the "max epochs" numerical results for these architectures, with three color highlights to help compare these results to Table 2: green values represent metrics with greater than 0.01 better performance, yellow signals metrics within a 0.01 absolute difference, and red represent cases with a greater than 0.01 worse performance. Some things of note: (a) DeepSpectra rmsec results are lower, especially for DeepSpectra6, which has more than halved its training error; however, while some of the EV results are better for DeepSpectra, they were similar or worse than before for DeepSpectra6, due to overfit; (b) SpectraNet results are very similar to previous ones, with most within a 0.01 difference; this is an impressive result, considering that SpectraNet–53 trained around 2–6 times longer, and SpectraNet6–53 trained around 8–27 times longer, without any of them showing tendency to overfit.



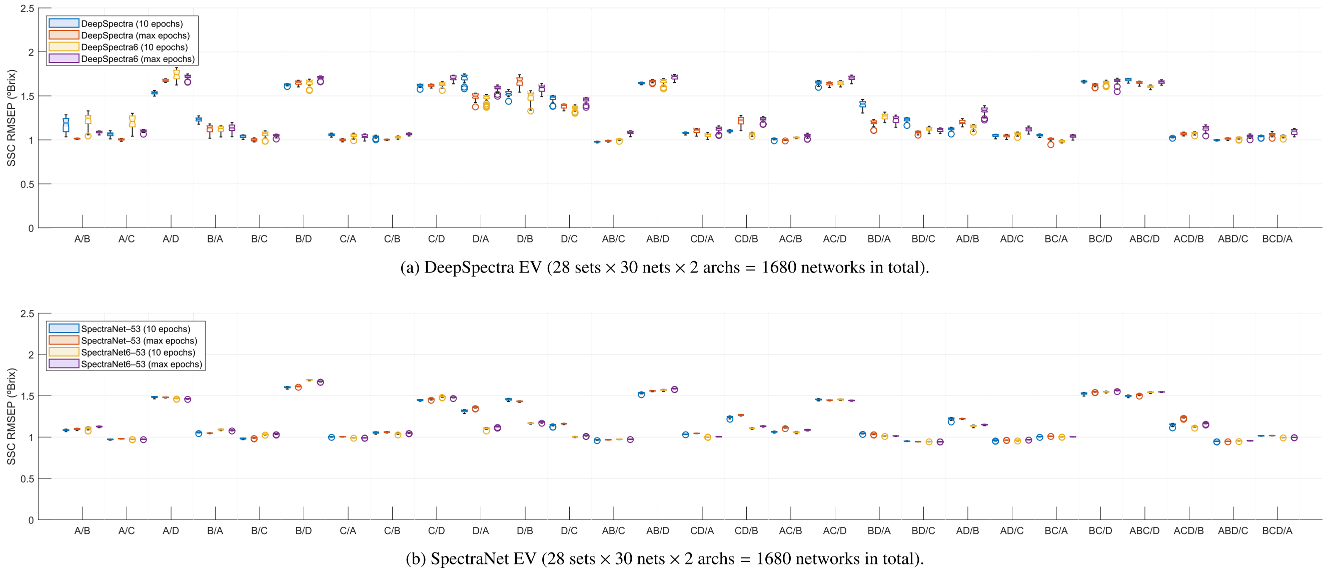
(a) DeepSpectra

(b) SpectraNet–53

(c) DeepSpectra6

(d) SpectraNet6–53

Fig. A.16. Training Loss RMSE results for the representative networks of each IV and EV set, selected as in Fig. 13 (the network closest to the mode of each set's rmsep). Blue lines represent IV sets 1–15 and orange lines represent EV sets 1–28. The dashed vertical line represents the 10 epochs threshold.

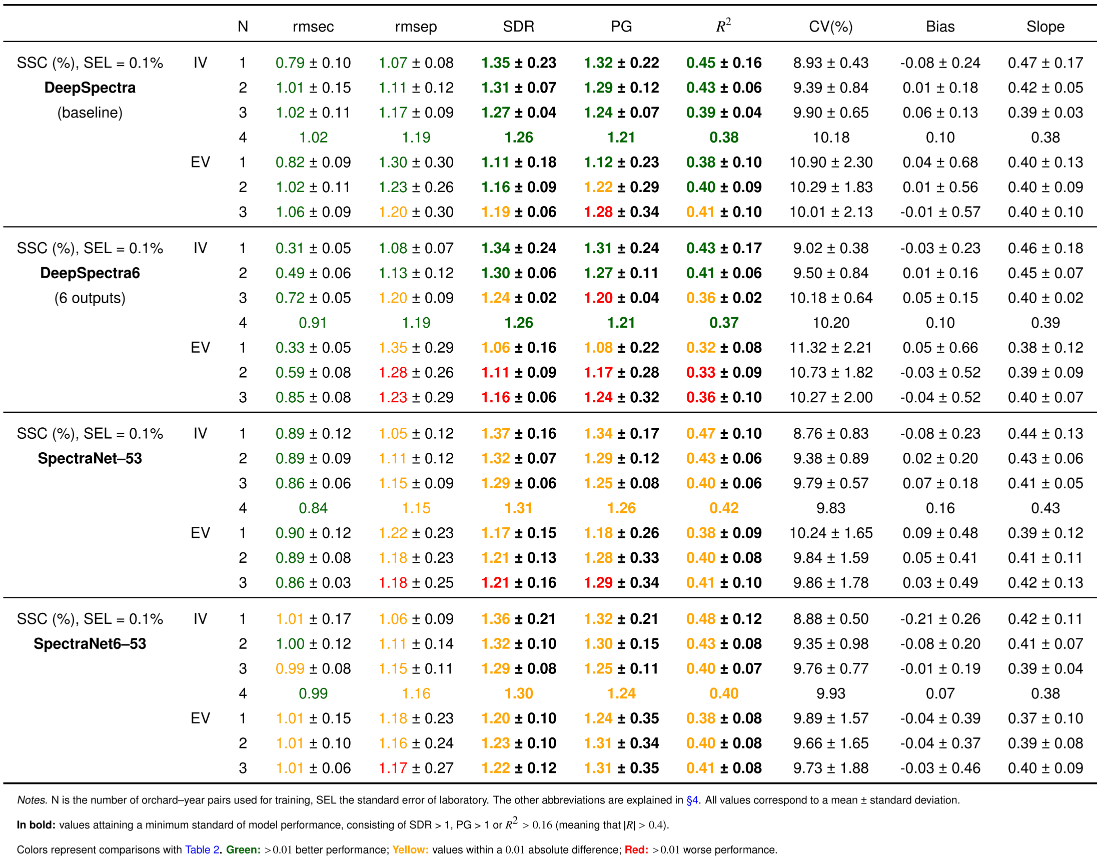


(a) DeepSpectra EV (28 sets × 30 nets × 2 archs = 1680 networks in total).

(b) SpectraNet EV (28 sets × 30 nets × 2 archs = 1680 networks in total).

Fig. A.17. SSC root-mean-squared error of prediction (rmsep) of the 30 neural networks, for each of the 15 internal and 28 external validation sets, and the 4 architectures, with shaded 95% confidence intervals. From left to right, at each set: one output with 10 epochs (blue, previously shown in Fig. 12(b)), one output with max epochs (red), six outputs with 10 epochs (yellow, also previously shown) and six outputs with max epochs (purple).

Table A.6
Summary of the SSC performance results obtained in IV and EV for the max epochs case.



		N	rmsec	rmsep	SDR	PG	R^2	CV(%)	Bias	Slope
SSC (%), SEL = 0.1%	IV	1	0.79 ± 0.10	1.07 ± 0.08	**1.35 ± 0.23**	**1.32 ± 0.22**	**0.45 ± 0.16**	8.93 ± 0.43	-0.08 ± 0.24	0.47 ± 0.17
DeepSpectra		2	1.01 ± 0.15	1.11 ± 0.12	**1.31 ± 0.07**	**1.29 ± 0.12**	**0.43 ± 0.06**	9.39 ± 0.84	0.01 ± 0.18	0.42 ± 0.05
(baseline)		3	1.02 ± 0.11	1.17 ± 0.09	**1.27 ± 0.04**	**1.24 ± 0.07**	**0.39 ± 0.04**	9.90 ± 0.65	0.06 ± 0.13	0.39 ± 0.03
		4	1.02	1.19	**1.26**	**1.21**	**0.38**	10.18	0.10	0.38
	EV	1	0.82 ± 0.09	1.30 ± 0.30	**1.11 ± 0.18**	**1.12 ± 0.23**	**0.38 ± 0.10**	10.90 ± 2.30	0.04 ± 0.68	0.40 ± 0.13
		2	1.02 ± 0.11	1.23 ± 0.26	**1.16 ± 0.09**	**1.22 ± 0.29**	**0.40 ± 0.09**	10.29 ± 1.83	0.01 ± 0.56	0.40 ± 0.09
		3	1.06 ± 0.09	1.20 ± 0.30	**1.19 ± 0.06**	**1.28 ± 0.34**	**0.41 ± 0.10**	10.01 ± 2.13	-0.01 ± 0.57	0.40 ± 0.10
SSC (%), SEL = 0.1%	IV	1	0.31 ± 0.05	1.08 ± 0.07	**1.34 ± 0.24**	**1.31 ± 0.24**	**0.43 ± 0.17**	9.02 ± 0.38	-0.03 ± 0.23	0.46 ± 0.18
DeepSpectra6		2	0.49 ± 0.06	1.13 ± 0.12	**1.30 ± 0.06**	**1.27 ± 0.11**	**0.41 ± 0.06**	9.50 ± 0.84	0.01 ± 0.16	0.45 ± 0.07
(6 outputs)		3	0.72 ± 0.05	1.20 ± 0.09	**1.24 ± 0.02**	**1.20 ± 0.04**	**0.36 ± 0.02**	10.18 ± 0.64	0.05 ± 0.15	0.40 ± 0.02
		4	0.91	1.19	**1.26**	**1.21**	**0.37**	10.20	0.10	0.39
	EV	1	0.33 ± 0.05	1.35 ± 0.29	**1.06 ± 0.16**	**1.08 ± 0.22**	**0.32 ± 0.08**	11.32 ± 2.21	0.05 ± 0.66	0.38 ± 0.12
		2	0.59 ± 0.08	1.28 ± 0.26	**1.11 ± 0.09**	**1.17 ± 0.28**	**0.33 ± 0.09**	10.73 ± 1.82	-0.03 ± 0.52	0.39 ± 0.09
		3	0.85 ± 0.08	1.23 ± 0.29	**1.16 ± 0.06**	**1.24 ± 0.32**	**0.36 ± 0.10**	10.27 ± 2.00	-0.04 ± 0.52	0.40 ± 0.07
SSC (%), SEL = 0.1%	IV	1	0.89 ± 0.12	1.05 ± 0.12	**1.37 ± 0.16**	**1.34 ± 0.17**	**0.47 ± 0.10**	8.76 ± 0.83	-0.08 ± 0.23	0.44 ± 0.13
SpectraNet–53		2	0.89 ± 0.09	1.11 ± 0.12	**1.32 ± 0.07**	**1.29 ± 0.12**	**0.43 ± 0.06**	9.38 ± 0.89	0.02 ± 0.20	0.43 ± 0.06
		3	0.86 ± 0.06	1.15 ± 0.09	**1.29 ± 0.06**	**1.25 ± 0.08**	**0.40 ± 0.06**	9.79 ± 0.57	0.07 ± 0.18	0.41 ± 0.05
		4	0.84	1.15	**1.31**	**1.26**	**0.42**	9.83	0.16	0.43
	EV	1	0.90 ± 0.12	1.22 ± 0.23	**1.17 ± 0.15**	**1.18 ± 0.26**	**0.38 ± 0.09**	10.24 ± 1.65	0.09 ± 0.48	0.39 ± 0.12
		2	0.89 ± 0.08	1.18 ± 0.23	**1.21 ± 0.13**	**1.28 ± 0.33**	**0.40 ± 0.08**	9.84 ± 1.59	0.05 ± 0.41	0.41 ± 0.11
		3	0.86 ± 0.03	1.18 ± 0.25	**1.21 ± 0.16**	**1.29 ± 0.34**	**0.41 ± 0.10**	9.86 ± 1.78	0.03 ± 0.49	0.42 ± 0.13
SSC (%), SEL = 0.1%	IV	1	1.01 ± 0.17	1.06 ± 0.09	**1.36 ± 0.21**	**1.32 ± 0.21**	**0.48 ± 0.12**	8.88 ± 0.50	-0.21 ± 0.26	0.42 ± 0.11
SpectraNet6–53		2	1.00 ± 0.12	1.11 ± 0.14	**1.32 ± 0.10**	**1.30 ± 0.15**	**0.43 ± 0.08**	9.35 ± 0.98	-0.08 ± 0.20	0.41 ± 0.07
		3	0.99 ± 0.08	1.15 ± 0.11	**1.29 ± 0.08**	**1.25 ± 0.11**	**0.40 ± 0.07**	9.76 ± 0.77	-0.01 ± 0.19	0.39 ± 0.04
		4	0.99	1.16	**1.30**	**1.24**	**0.40**	9.93	0.07	0.38
	EV	1	1.01 ± 0.15	1.18 ± 0.23	**1.20 ± 0.10**	**1.24 ± 0.35**	**0.38 ± 0.08**	9.89 ± 1.57	-0.04 ± 0.39	0.37 ± 0.10
		2	1.01 ± 0.10	1.16 ± 0.24	**1.23 ± 0.10**	**1.31 ± 0.34**	**0.40 ± 0.08**	9.66 ± 1.65	-0.04 ± 0.37	0.39 ± 0.08
		3	1.01 ± 0.06	1.17 ± 0.27	**1.22 ± 0.12**	**1.31 ± 0.35**	**0.41 ± 0.08**	9.73 ± 1.88	-0.03 ± 0.46	0.40 ± 0.09

Notes. N is the number of orchard–year pairs used for training, SEL the standard error of laboratory. The other abbreviations are explained in §4. All values correspond to a mean ± standard deviation.

In bold: values attaining a minimum standard of model performance, consisting of SDR > 1, PG > 1 or $R^2 > 0.16$ (meaning that $|R| > 0.4$).

Colors represent comparisons with Table 2. **Green:** >0.01 better performance; **Yellow:** values within a 0.01 absolute difference; **Red:** >0.01 worse performance.

Notes. N is the number of orchard–year pairs used for training, SEL the standard error of laboratory. The other abbreviations are explained in Section 4. All values correspond to a mean ± standard deviation.
In bold: values attaining a minimum standard of model performance, consisting of SDR > 1, PG > 1 or $R^2 > 0.16$ (meaning that $|R| > 0.4$).
Colors represent comparisons with Section 2. **Green:** > 0.01 better performance; **Yellow:** values within a 0.01 absolute difference; **Red:** > 0.01 worse performance.

References

Benelli, A., Cevoli, C., Fabbri, A., 2020. In-field Vis/NIR hyperspectral imaging to measure soluble solids content of wine grape berries during ripening. In: 2020 IEEE Int. Work. Metrol. Agric. For. IEEE, pp. 99–103. https://doi.org/10.1109/MetroAgriFor50201.2020.9277621. URL: https://ieeexplore.ieee.org/document/9277621/.

Cavaco, A.M., Pires, R., Antunes, M.D., Panagopoulos, T., Brázio, A., Afonso, A.M., Silva, L., Lucas, M.R., Cadeiras, B., Cruz, S.P., Guerra, R., 2018. Validation of short wave near infrared calibration models for the quality and ripening of 'Newhall' orange on tree across years and orchards. Postharvest Biol. Technol. 141, 86–97. https://doi.org/10.1016/j.postharvbio.2018.03.013. URL: https://doi.org/10.1016/j.postharvbio.2018.03.013 https://linkinghub.elsevier.com/retrieve/pii/S092552141731195X.

Cavaco, A.M., Passos, D., Pires, R.M., Antunes, M.D., Guerra, R., 2021. Nondestructive Assessment of Citrus Fruit Quality and Ripening by Visible-Near Infrared Reflectance Spectroscopy. In: Citrus [Working Title]. IntechOpen. https://doi.org/10.5772/intechopen.95970.

Clercq, M.D., Vats, A., Biel, A., 2018. Agriculture 4.0: the Future of Farming Technology. In: World Gov. Summit Collab. with OliverWyman, pp. 30. URL: https://www.worldgovernmentsummit.org/api/publications/document?id=95df8ac4-e97c-6578-b2f8-ff0000a7ddb6.

Delacre, M., Lakens, D., Leys, C., 2017. Why psychologists should by default use welch's t-Test instead of student's t-Test. Int. Rev. Soc. Psychol. 30, 92–101. https://doi.org/10.5334/irsp.82.

Gauglitz, G., Vo-Dinh, T. (Eds.), 2003. Handbook of Spectroscopy. Wiley. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/3527602305. https://doi.org/10.1002/3527602305.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: J. Mach. Learn. Res., vol. 9, 2010a, pp. 249–256. https://doi.org/10.1109/ICCV.2015.123. arXiv:1502.01852.

Golic, M., Walsh, K., Lawson, P., 2003. Short-wavelength near-infrared spectra of sucrose, glucose, and fructose with respect to sugar concentration and temperature. Appl. Spectrosc. 57, 139–145. https://doi.org/10.1366/000370203321535033. URL: https://journals.sagepub.com/doi/10.1366/000370203321535033.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. IEEE International Conference on Computer Vision (ICCV) 9 (1), 1026–1034. https://doi.org/10.1109/ICCV.2015.123.

He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition. In: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., volume 2016-Decem. IEEE, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90. URL: http://image-net.org/challenges/LSVRC/2015/ http://ieeexplore.ieee.org/document/7780459/ http://arxiv.org/abs/1512.03385. arXiv:1512.03385.

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity Mappings in Deep Residual Networks. In: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). LNCS, vol. 9908. Springer, pp. 630–645. https://doi.org/10.1007/978-3-319-46493-0_38. arXiv:1603.05027.

Hendrycks, D., Gimpel, K., 2016. Gaussian Error Linear Units (GELUs). URL: http://arxiv.org/abs/1606.08415. arXiv:1606.08415.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: 32nd Int. Conf. Mach. Learn. ICML 2015, vol. 1, pp. 448–456. arXiv:1502.03167.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J., 2019. 1D Convolutional Neural Networks and Applications: A Survey. Mech. Syst. Signal Process. 151, 107398. https://doi.org/10.1016/j.ymssp.2020.107398. URL: https://doi.org/10.1016/j.ymssp.2020.107398 https://linkinghub.elsevier.com/retrieve/pii/S0888327020307846 http://arxiv.org/abs/1905.03554. arXiv:1905.03554.

Kohler, J., Daneshmand, H., Lucchi, A., Hofmann, T., Zhou, M., Neymeyr, K., 2019. Exponential convergence rates for Batch Normalization: The power of length-direction decoupling in non-convex optimization. In: AISTATS 2019–22nd Int. Conf. Artif. Intell. Stat., 2020. arXiv:1805.10694.

Li, B., Lecourt, J., Bishop, G., 2018. Advances in non-destructive early assessment of fruit ripeness towards defining optimal time of harvest and yield prediction–a review. Plants 7, 1–20. https://doi.org/10.3390/plants7010003.

Magwaza, L.S., Opara, U.L., 2015. Analytical methods for determination of sugars and sweetness of horticultural products–A review. https://doi.org/10.1016/j.scienta.2015.01.001.

Mishra, P., Passos, D., 2021. Deep multiblock predictive modelling using parallel input convolutional neural networks. Anal. Chim. Acta 1163, 338520. https://doi.org/10.1016/j.aca.2021.338520.

Nicolaï, 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. https://doi.org/10.1016/j.postharvbio.2007.06.024.

Padarian, J., Minasny, B., McBratney, A.B., 2019a. Using deep learning to predict soil properties from regional spectral data. Geoderma Reg. 16, e00198. https://doi.org/10.1016/j.geodrs.2018.e00198. URL: https://www.sciencedirect.com/science/article/pii/S2352009418302785?via%3Dihub.

Padarian, J., Minasny, B., McBratney, A.B., 2019b. Using deep learning for digital soil mapping. Soil 5, 79–89. https://doi.org/10.5194/soil-5-79-2019. URL: https://www.soil-journal.net/5/79/2019/.

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.

Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., Pande, V., 2015. Massively Multitask Networks for Drug Discovery. URL: http://arxiv.org/abs/1502.02072. arXiv:1502.02072.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. Why should i trust you? Explaining the predictions of any classifier. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13–17-Augu, 2016, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778.

Ruder, S., 2017. An Overview of Multi-Task Learning in Deep Neural Networks, arXiv. URL: http://arxiv.org/abs/1706.05098. arXiv:1706.05098.

Simon, M., Rodner, E., Denzler, J., 2016. ImageNet pre-trained models with batch normalization. URL: http://www.inf-cv.uni-jena.de/Research/CNN+Models.html and https://github.com/cvjena/cnn-models. http://arxiv.org/abs/1612.01452. arXiv:1612.01452.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07–12-June, 2015, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594. arXiv:1409.4842.

Tuchin, V.V., 2015. Tissue Optics and Photonics: Light-Tissue Interaction. J. Biomed. Photonics Eng. 98–134. https://doi.org/10.18287/jbpe-2015-1-2-98.

Vinet, L., Zhedanov, A., 2011. A 'missing' family of classical orthogonal polynomials. J. Phys. A Math. Theor. 44, 2571–2575. https://doi.org/10.1088/1751-8113/44/8/085201 arXiv:1011.1669.

Xu, Z., Zhao, X., Guo, X., Guo, J., 2019. Deep Learning Application for Predicting Soil Organic Matter Content by VIS-NIR Spectroscopy. Comput. Intell. Neurosci. 2019, 1–11. https://doi.org/10.1155/2019/3563761. URL: https://www.hindawi.com/journals/cin/2019/3563761/.

Yang, J., Li, J., Hu, J., Yang, W., Zhang, X., Xu, J., Zhang, Y., Luo, X., Ting, K.C., Lin, T., Ying, Y., 2022. An interpretable deep learning approach for calibration transfer among multiple near-infrared instruments. Comput. Electron. Agric. 192, 106584. https://doi.org/10.1016/j.compag.2021.106584.

Yann LeCun, G.H., Bengio, Yoshua, 2015. Deep learning. Nature.

Yu, X., Lu, H., Wu, D., 2018. 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.

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.

Zhao, R., An, L., Tang, W., Gao, D., Qiao, L., Li, M., Sun, H., Qiao, J., 2022. Deep learning assisted continuous wavelet transform-based spectrogram for the detection of chlorophyll content in potato leaves. Comput. Electron. Agric. 195, 106802. https://doi.org/10.1016/j.compag.2022.106802. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168169922001193.

Zhou, L., Zhang, C., Taha, M.F., Wei, X., He, Y., Qiu, Z., Liu, Y., 2020. 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 https://doi.org/10.3389/fpls.2020.575810. URL: https://pubmed.ncbi.nlm.nih.gov/33240294/.