Augmenting NIR Spectra in Deep Regression to Improve Calibration

Mark Wohlers[a,b], Andrew McGlone[a], Eibe Frank[b], Geoffrey Holmes[b]

[a] *The New Zealand Institute for Plant and Food Research Limited, Auckland, New Zealand*
[b] *Department of Computer Science, University of Waikato, Hamilton, New Zealand*

Abstract

Deep learning, particularly with convolutional neural networks, shows promise in modelling near-infrared spectroscopy (NIRS), but the lack of robust generalization across instruments often affects performance in practice. Here, we investigate a method to increase the robustness of this approach. The proposed method involves using a simple data augmentation technique during the training process. The performance of convolutional neural network regression is compared to partial least squares regression (PLSR) using kiwifruit data collected from multiple handheld devices over three seasons and mango data collected from a single device over four seasons. The results suggest that data augmentation for NIR spectra can prevent overfitting. In particular, augmenting the training data to mimic spectra collected over multiple devices results in a neural network model with improved performance over PLSR.

Keywords: near-infrared spectroscopy, convolutional neural networks, partial least squares regression, data augmentation

1. Introduction

Near-infrared spectroscopy (NIRS) has been a useful tool in non-destructive measurements of fruit quality [1]. The levels at which the NIR wavelengths are absorbed or scattered vary based on the sample's chemical makeup and structure. Using machine-learning modelling methods, recorded spectra can be used to make predictions of various fruit quality attributes such as dry matter content (DMC) or soluble solids content (SSC). The kiwifruit industry uses

a minimum DMC as minimum taste standard (MTS) with NIR sorting being successfully employed to recover high DMC fruit from populations failing this standard [2]. Other fruits have similar attribute specifications, for example, it is recommended that mango have an at-harvest DMC of at least 14 [2]. The use of deep learning models to make predictions based on NIR spectra has become increasingly popular [3]. However, fitting the models can require extensive optimisation to arrive at a final architecture and tuning parameters [4], which risks overfitting and poor generalization, especially if the data set is small.

Most traditional techniques, such as partial least squares regression (PLSR) [5], rely on pre-processing methods [6] to remove nuisance effects such as the confounding influence of light scattering due to variation in internal tissue structure. However, there are other effects such as temperature [7], operator, and random noise that can also influence the absorbances critical to generating accurate models. The devices themselves can be variable, between devices of the same model, or even across time for the same device [8]. The exact sampling position of fruit measurements can greatly affect the spectra as the internal composition and structure are much less uniform than what is typically observed in industrial NIRS analysis, such as that of liquid or powder samples. All of these effects can influence the quality of model predictions and should be taken into account.

There is evidence that deep learning models learn the appropriate pre-processing techniques automatically [3], but they may require larger data sets for training. In practice, access to such data may be limited due to situations where data collection is time-consuming, for example, when measuring fruit with what are typically slow handheld NIR devices [9]. One possible solution that is commonly employed in other domains, such as image classification, is to augment the observed data using synthetic generation to produce a larger training set [10], where training images are routinely altered, for example, by transformations such as rotation, clipping, and zooming, among others. While data augmentation has become increasingly popular and mature in other deep learning applications, there are fewer examples of applying it in the area of

NIRS.

Convolutional neural networks can be naturally applied to NIRS data by treating this data as a 1-dimensional signal [11], preserving the wavelength structure in the data. The idea of augmenting NIR spectra is not new and has previously been used with PLSR by adding Gaussian noise [12]. More recently, Bjerrum et al. [13] used convolutional neural networks to classify pharmaceutical samples based on NIR spectra. They found improved performance by augmenting the data using random offset, multiplicative, and slope effects. This technique was later expanded into a more general form [14] and found to perform well on a number of classification data sets. Mishra et al. [15] used an alternative method whereby raw spectra were altered by various permutations of pre-processing techniques. However, this can perhaps be viewed more accurately as a form of pre-processing rather than data augmentation, as the process does not increase the size of the training set.

Not all applications of augmentation to NIR data provide a positive result: Acquarelli et al. [16] reported that data augmentation did not improve performance. The authors speculate that this may be due to the difficulty of modelling noise as indicated by the range of pre-processing techniques needed to model the various data sets in their experiments. Details of the augmentation used were limited to adding perturbed samples.

An advantage of being able to realistically augment spectra is that it can produce robust models. Currently, variations amongst devices can lead to PLSR and other models being trained for each device individually. A robust model that generalises across devices would allow for new devices to be put to use earlier, reducing the need to collect a large amount of data to train an individual model specifically for each device.

This paper presents an alternative augmentation method for model training that improves generalisation to other devices. This is achieved by simulating changes in absorbances observed when measuring fruit from multiple devices. Augmented data are generated from a multivariate normal distribution (MVN) and incorporated into a data input pipeline API for easy implementation while

training deep learning models. The impact on training stability and the performance of trained models is assessed, including a comparison to PLSR.

2. Data Augmentation using Correlated Gaussian Noise

According to Blazhko [14], data augmentation "should strive to produce observations as close as possible to what could be obtained in reality". Gaussian noise has been used to augment NIR spectra in prior work [12]. However, the generation of the noise in that method is equivalent to sampling from a multivariate normal distribution with a diagonal covariance matrix. While the amount of noise added to each wavelength is based on the respective observed standard deviation, the noise between wavelengths is considered statistically independent. This is clearly not ideal when the aim is to produce realistic data: when repeated measurements are taken from the same fruit, it can be seen that spectral deviations from the mean are not independent. This is demonstrated in Figure 1 where spectra from multiple devices measured on the same fruit are roughly parallel over bands of wavelengths.

For example, spectra with higher than average absorbance at 550nm will also likely have higher than average absorbance at 650nm. The left panel of Figure 2 presents spectra from 205 kiwifruits with the fruits' average spectra subtracted. For the augmentation to generate spectra close to what "could be obtained in reality", a non-diagonal covariance matrix should be used, yielding samples such as the ones shown in the right panel of Figure 2.

The specification of an appropriate covariance structure for the investigation presented in this paper is based on observed spectra from a previously collected multi-instrument data set. A sample of 205 kiwifruits were each measured once by ten separate devices. The point of measurement across devices was taken from the same side at the equator of the fruit but may vary slightly in exact position. The spectra for each fruit were mean-centred (as in Figure 2: left panel). The Gaussian noise added for data augmentation was then generated by sampling from an MVN with mean 0 and covariance structure Σ (as in Figure 2 right

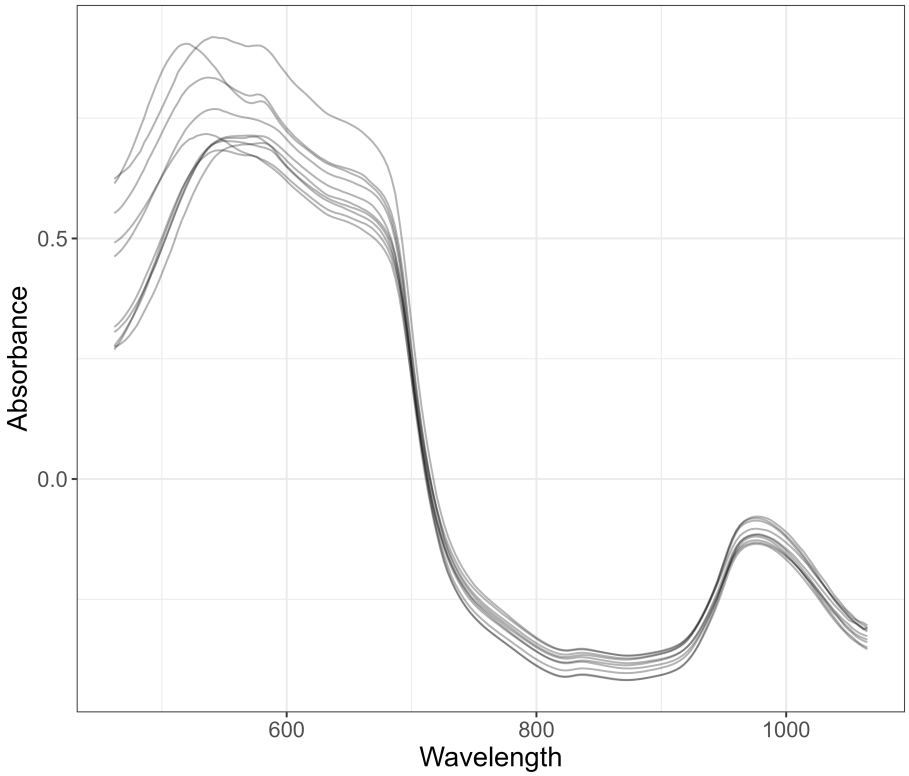

Figure 1: Example of NIR measurements from multiple devices on the same fruit.

panel), where Σ is estimated as the sample covariance matrix of this centred data set. More formally, $\mathbf{\Sigma} = [\sigma_{jk}]$ with $\sigma_{jk} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$ where x_{ij} is the j_{th} wavelength of the i_{th} sample, and $\bar{x}_j$ is the sample mean of wavelength j.

Note that this means the augmentation is based on within-device measurement error as well as spectral variation between devices.

3. Materials and Methods

3.1. Datasets

Two data sets are used for the empirical results presented in this paper. Both contain NIR measurements that were recorded using handheld F-750 produce

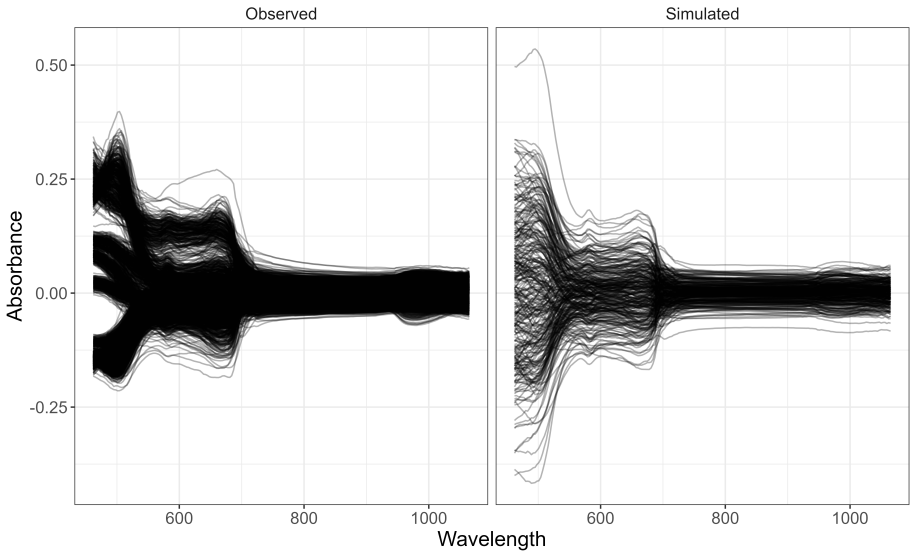

Figure 2: Example of observed (left) and generated noise from an MVN distribution with the respective empirical covariance matrix (right).

quality meters by Felix instruments. The recorded spectra contained wavelengths between 402nm and 1137nm. This was further trimmed to 202 wavelengths from 459nm to 1059nm in steps of 3nm. Outliers were removed from the training set based on Hotelling T2 scores from a PLSR with 20 latent variables and confirmed via visual inspection.

3.1.1. Kiwifruit Dataset

The first data set contains spectra from 4,956 kiwifruits collected over three seasons from two sites approximately 500km apart. Depending on the site, each fruit was measured once by two or three devices. After the non-destructive NIRS was performed, two destructive industry-standard fruit quality measures of DMC and SSC for each fruit were recorded. These measurements have been shown to be related to consumer liking responses [17]. The DMC was recorded as a percentage of the fresh weight of an equatorial slice (3mm thick) taken from the fruit and measured prior to drying in a convection oven at 60 °C for 24 hours. The SSC was recorded as the average °Brix, measured by refractometry, of the

juice squeezed separately from the stem and stylar ends of the fruit. The data are generally made up of groups of five fruit from the same vine at each time point. A separate analysis indicated that there is a high amount of variation amongst vines, which is possibly due to phenotypic differences such as tricome (hair) density on the skin, shape, core size, etc. The large geographical distance between the two data collection sites can also produce variation in DMC and SSC, among other measures. In total, there were five handheld devices used to measure the fruit across the two sites: two at a Kerikeri site and three at a Te Puke site. Kerikeri is located in the Bay of Islands region of New Zealand, and Te Puke is located in the Bay of Plenty region. The training set was taken as observations collected between 21st March 2017 and 6th March 2019, the validation set was the data collected immediately after the training set until 27th March 2019, and the test set was collected after the 3rd of April 2019. These sets comprise of 3762, 594, and 600 fruit with 8392, 1382, and 1500 total spectra, respectively.

The distributions of DMC measured at each site were slightly different, and as the devices were nested within site, there may be some confounding of device and site effects. This could result in a model erroneously using information about a site-specific device to predict SSC and/or DMC and may negatively affect the cross-site predictive performance. This was investigated by training models based on the training data above from each of the two sites and testing on the respective other site.

The estimation of the covariance structure used for augmentation of the training data, discussed in the previous section, is based on a historical data set collected prior to 21st March 2017. More specifically, we had access to a data set where all five devices measured 205 fruit in 2017. These data were not included in the training, validation, or testing sets to retain the independence of the covariance matrix.

3.1.2. Mango Dataset

The second data set contains DMC measurements from 4675 mangoes with multiple scans per fruit, resulting in 11,691 NIRS spectra, taken from supplementary material in [18] [19]. The data were collected from ten cultivars over four seasons from two Australian growing regions using one F-750 device. The data set was analysed using deep learning in [15], which provides a useful baseline to compare our augmentation method to. However, Mishra and Passos [15] used additional outlier removal that we did not replicate here, so our results using their methods, which we re-implemented, likely differ from the source text. The dataset used in [15] is freely available from the author's github site.[1] In our investigation, the training, tuning, and validation sets were used as specified in the source data set [18].

The mango dataset contained spectra from a single device so the covariance could not be estimated (as was done for the kiwifruit dataset). With the assumption that the variation amongst devices would be similar for different fruit relative to the absorbance levels, we used a scaled version of the data set 1 augmentation method for the mango data set. More specifically, we scaled the covariance matrix used in the augmentation of the mango data: each wavelength was scaled by its respective standard deviation in the kiwifruit training set. The mango data set was then normalised, as further outlined below, so that augmentation was done on the same relative scale as for the kiwifruit data set.

3.2. Methods

Deep learning models were fit using TensorFlow 2.4.0 on a desktop computer containing an Intel Xeon E3-1270 CPU, 64 gigabytes of main memory, and an NVIDIA Quadro M4000 GPU with 8 gigabytes of device memory. Partial least squares regression (PLSR) was conducted using the Scikit-learn package [20]. The Hyperopt [21] package performed the hyperparameter optimisation over

[1] `https://github.com/dario-passos/DeepLearning_for_VIS-NIR_Spectra/tree/master/notebooks/Tutorial_on_DL_optimization/datasets`

100 iterations for each outcome measure. Each deep learning model was run for 100 epochs using the Adam [22] optimiser. During hyperparameter tuning, the He Normal method was used to initialise model weights. However, this was changed to He Uniform for the final models as it was later found to give more consistent convergence. All analyses were conducted in Python 3.6.

3.2.1. Pre-processing

Prior work presented in [3] suggests that the convolutional layers applied to NIR data learn the appropriate pre-processing operations and, as such, convolutional neural networks do not require the methods usually employed with such data. However, we did apply column-wise normalisation to both data sets, such that every feature had zero mean and a standard deviation of one.This normalisation was found to improve model convergence. When using the method of Bjerrum et al. spectra were insteadaltered by subracting the grand mean and dividing by twice the overall standard deviation as described in [13]. For the PLSR models, a Savitsky-Golay second-derivative filter, itself a convolutional filter, with a window size of 17, and polynomial order 2, was used for the kiwifruit data set. Other methods tried were varying window size, taking the first derivative, and using unaltered spectra. These methods were not found useful.

3.2.2. Data Augmentation

The data augmentation was implemented using the TensorFlow Dataset API. This created a pipeline where the training data are read from disk, shuffled, and divided into random (mini-)batches to fit on the GPU for training. A random sample from the specified MVN is added by applying the TensorFlow probability library together with the tf.data.Dataset data pipeline. This allows the random addition to be repeated a number of times for each instance in the batch, with only augmented instances being used in model training. For hyperparameter tuning, 50 augmented spectra were generated per observed instance due to time constraints of model fitting. This was increased to 100 for final model training with optimised parameters. For comparison, the method described in

[13], where spectra are altered through random offset, slope, and multiplication transformation, was also used.

3.2.3. Deep Learning

The architecture used in the deep-learning models consisted of a combination of convolutional layers followed by dense layers, with a final dense layer with a single neuron exhibiting a linear activation function, as is typical when neural networks are applied to regression problems. All other layers used a rectified linear unit (ReLU) activation function.

The models used the approach presented by [13], where the number of layers was chosen through optimisation. This is discussed in more detail in the next subsection. Kiwifruit final models were trained for 1000 epochs at a set learning rate. Performance on the validation set was inspected to ensure the appropriateness of this training regime. For the mango data set, a learning rate scheduler was used to be consistent with the analysis used in [15]. This meant the learning rate was halved if there was no improvement in the validation loss after 25 epochs, and training halted if no improvement was found after 50.

3.2.4. Hyperparameter Tuning

The performance of convolutional networks can be influenced by the particular architecture used and other hyperparameters such as the learning rate, batch size, and the amount of augmentation performed. In particular, finding an appropriate network architecture involves tuning a number of hyperparameters, which can be difficult and time-consuming. Bayesian optimisation was used for this task as it is more time-efficient when model training is slow compared to methods such as grid or random searches [21]. A full list of the search space for the hyperparameters is given in Table 1. It is similar to the one presented in Bjerrum et al. [13] with the exception that it also includes the number of convolutional and dense layers to be used in the model as parameters to be tuned. The minimum validation set mean square error (MSE) over the full number of epochs was used as the criterion to be optimised. To reduce the variability of

	Parameter	Search Space
1	Convolutional layers	{1, 2}
2	Dense layers	{0, 1, 2}
3	Batch size	{100, 500, 1000}
4	Learning rate	0.0001 - 0.1
5	Generated data size	{1, 10, 30, 50}
6	Conv. layers: no. kernels	2-40
7	Conv. layers: filter size	5-150
8	Dense layers: no. neurons	4-1000

Table 1: Hyperparameter search space

this measure, a moving average smoother with window size 10 was applied to the MSE prior to calculating the minimum. The stability of the optimisation process was further improved by running each hyperparameter configuration three times and taking the mean performance. The Bayesian optimisation was conducted using the Hyperopt package [21] in Python. Only the training and validation data were used during hyperparameter optimisation, not the test set, to avoid optimistic performance estimates on the test set. Note that due to the large training time, a relatively small number of hyperparameter iterations is performed considering the size of the search space, which may result in non-optimal solutions presented here. The optimization time was highly dependent on the architecture being evaluated. It took approximately a week to evaluate the 100 iterations, with the bulk of the time spent on those models with a large number of convolutional filters across multiple layers.

3.2.5. Partial Least Squares Regression

PLSR with second derivative Savitsky-Golay pre-processing was used as a baseline to compare against deep learning. The number of latent variables used in the PLSR models was selected based on the minimum MSE of the respective validation dataset. We also considered unaltered spectra with no pre-

processing or augmentation applied. Unless specified otherwise, PLSR results include second derivative Savitsky-Golay preprocssing.

4. Results and Discussion

4.1. Kiwifruit Results

First, we briefly discuss the outcome of hyperparameter tuning. Then, we study the effect of data augmentation on training the neural networks and PLSR, respectively, before comparing them. Cross-validation of the kiwifruit data by site is also considered.

4.1.1. Selection of Network Architectures

For dry matter, the optimal architecture for the convolutional neural network found during the optimisation required only a single convolutional layer of 122 filters with a kernel size of 13, followed by a single linear output neuron. Interestingly, after training this architecture for 200 epochs, only 10 of the convolutional filters ever fired over all training, validation, and test data sets. Based on this result, a reduced model was trained with only ten filters and a lower learning rate (0.001). It provided similar performance but was more robust to overfitting.

The hyperparameter tuning for predicting SSC gave a solution with two convolutional layers with 113 filters of size 32 and 93 filters of size 28, respectively. Again, no dense layer was needed. Investigating the output of the filters, all in the first convolutional layer provided a non-zero output for at least one observation of the training set. However, the second convolutional layer consisted of only two filters that gave non-zero output and so the model could be pruned. However, it was decided to leave this architecture unchanged.

As discussed above, hyperparameter optimisation was based on the mean of three runs of a given configuration. During this process, it was noted that there were occasions where two of the runs had a low MSE, but the third did not converge with both high training and validation errors resulting in a poor

overall loss. On these occasions, it was obvious that the model optimisation process was not progressing as the training error was stuck at a value far greater than if all predictions were set to the grand mean. This does cast some doubt on the optimised parameters as perhaps a better configuration that was missed could be superior if different initialisation or optimisation routines were used. Because the hyperparameter space did not permit very deep architectures, a deeper model was also trained to evaluate the suitability of the MVN augmentation for different architectures. This consisted of five convolutional layers using exponential linear unit (ELU) activation functions with 4, 4, 8, 16, and 24 filters of sizes 9, 6, 7, 5, and 3 respectively. Each convolutional layer was followed by batch normalisation. This was connected to a single dense layer with a single neuron and linear activation function.

4.1.2. Augmentation for Deep Learning

Training the shallow and deep convolutional models was greatly improved using the MVN augmented data for both DMC and SSC. Figure 3 shows the training history for a deep architecture predicting DMC and SSC on the training and validation sets with three runs for non-augmented and MVN augmented data. Training with observed data alone quickly leads to overfitting and poor performance on the validation data.

Additionally, the data were augmented using the method described in Bjerrum [13] for comparison, see Figure 4. Here we compare three architectures: the one used in Bjerrum et al.[13], our (optimised) shallow CNN architecture, and a deep architecture. For both the optimised and reduced architectures, the data set augmented with our method (MVN) consistently reached a lower validation MSE and reached it in fewer epochs. This was not true of the training set, where the Bjerrum augmentation gave the lowest MSE, implying that MVN augmentation provided some protection against overfitting. Similar observations can be made for the Bjerrum architecture.

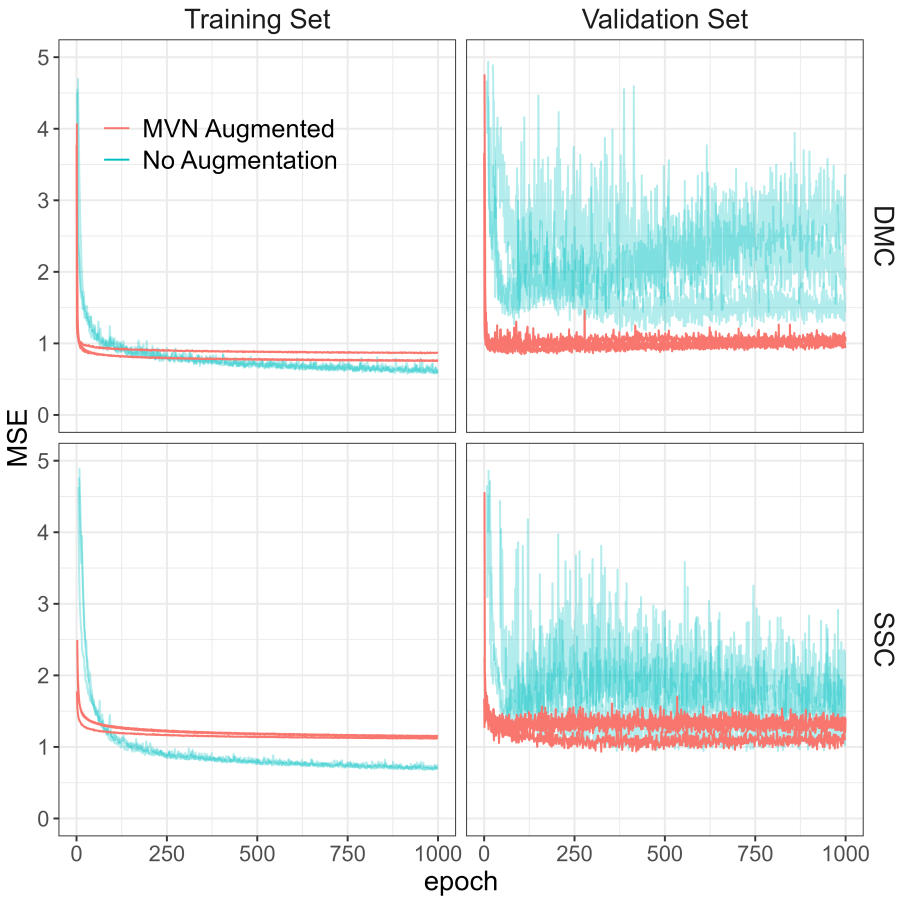

Figure 3: Comparison of results with and without data augmentation. The left column shows MSE on the training data, the right column MSE on the validation data.

4.1.3. Kiwifruit cross-validation by site

Fitting PLSR models to the MVN augmented data set was not consistently better than fitting on the Savitsky-Golay second derivative pre-processed data, see Table 2, in contrast to the consistently positive effect of augmentation in the case of CNNs (Table 3). Note that the results are based on PLSR models trained on devices from the training site only and tested on all data from the excluded site. The MVN augmented data considered thus far uses a covariance matrix for all devices, including those from the test site. Here, augmented results based

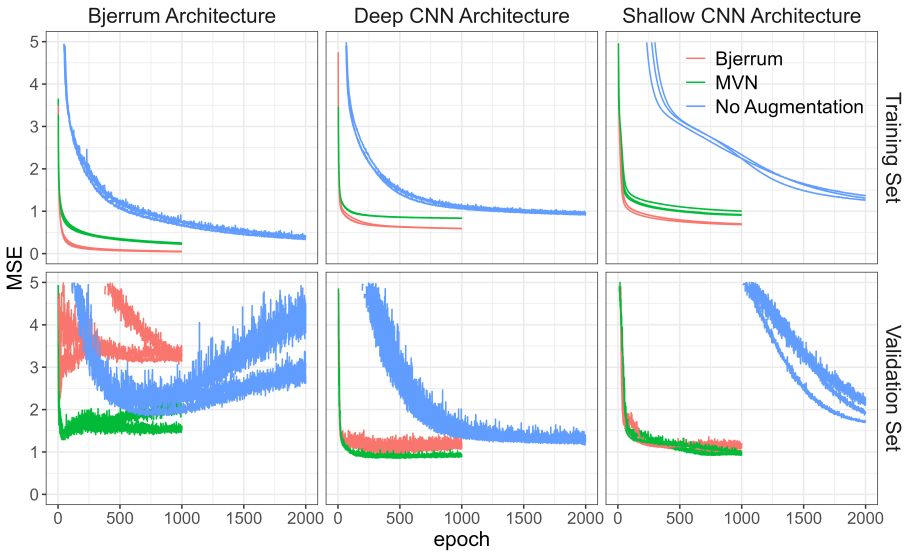

Figure 4: Comparison of the effect of different data augmentation methods on model fitting to predict DMC using different CNN architectures: deep, shallow, and that outlined in Bjerrum et al.. The top row shows MSE on the training data for three runs of each data augmentation method; the bottom row shows MSE on the validation data.

on a covariance matrix excluding the test set devices are also presented. The results are inconsistent for DMC prediction, as some devices have lower RMSE for models trained on the MVN augmented data, while others are higher when compared to models trained on the non-augmented data. TP3 in particular had poorer DMC predictions when MVN augmentation was applied. In general, the MVN augmentation performs superior predicting SSC. In almost all tests, the full covariance augmentation performed better than the covariance with the test devices excluded. This was consistent with the results from the deep CNN models shown in Table 3. It indicates that the augmentation method is sensitive to the covariance matrix, and it is beneficial to include the devices of interest in its estimation and/or estimate on a wide range of similar devices.

Interestingly, when applying PLSR to augmented data, the optimum number of latent variables (LV) found to minimise the RMSE of the non-augmented validation data set was often large (Table 2). Including excessive LVs in the model

has the risk of overfitting; this appears true when augmenting with the covariance matrix excluding the test site devices. The test RMSE improved when using fewer LVs, such as the optimal number found with the non-augmented data. For comparison, results are also presented in brackets using 30 LVs. This number is based on observed performance in previous analyses with the same devices. Using the full covariance matrix in the augmentation seemed more robust and did not see a dramatic improvement when reducing the number of LVs. A possible explanation for the overfitting is that the validation data includes data from the same devices as the rest of the training data while the test set contains data from a different device. Nevertheless, caution is recommended when selecting the number of latent variables while applying PLSR to augmented data.

Large differences occurred between sites and devices in terms of the spectra produced. In particular, the Te Puke site had one device, TP3, that had substantially higher absorbance levels than the other devices in the lower wavelengths (Figure 5). This is reflected in the performance estimate obtained when performing cross-validation on a per-site basis. When TP3 was not included in the training set, the subsequent validation predictions on that device were far poorer than others. When it was included in the training, it did not have a detrimental effect on predicting the Te Puke sites. However, using the MVN augmentation method, PLSR improved significantly, see Figure 6, by reducing the offset in the predictions.

Dataset			PLSR RMSE			
Training	Test	Measure	No-Aug	MVN Aug (excl test)	MVN Aug	LVs
Kerikeri	TP1	DMC	1.26(1.29)	1.49(1.36)	1.13(1.14)	29/84/89
Kerikeri	TP2	DMC	1.15(1.15)	1.72(1.06)	1.14(1.19)	29/84/89
Kerikeri	TP3	DMC	1.48(1.48)	2.48(1.75)	2.00(2.34)	29/84/89
Te Puke	KK1	DMC	0.90(0.87)	0.96(0.99)	0.96(0.95)	18/16/16
Te Puke	KK2	DMC	1.06(1.04)	1.36(1.76)	1.05(1.04)	18/16/16
Kerikeri	TP1	SSC	4.15(4.23)	3.22(1.26)	1.19(1.25)	38/86/81
Kerikeri	TP2	SSC	6.31(5.00)	3.29(1.28)	1.2(1.29)	38/86/81
Kerikeri	TP3	SSC	5.98(4.49)	3.8(2.71)	1.46(1.49)	38/86/81
Te Puke	KK1	SSC	1.27(3.37)	2.33(1.45)	1.18(1.22)	12/43/48
Te Puke	KK2	SSC	2.42(4.99)	2.28(2.10)	1.48(1.58)	12/43/48

Table 2: Results of PLSR trained on non-augmented data and PLSR trained on MVN-augmented data, from either Te Puke (TP1, TP2, TP3 training sets) or Kerikeri (KK1, KK2 training sets), based on both, the covariance excluding all devices from the test site (excl test) and including all devices. The number of latent variables (LVs) for each of the three models is presented separated by "/". RMSE presented in brackets is from the respective PLSR model with 30 LVs. All PLSR models used Savitsky-Golay 2nd derivative pre-processing.

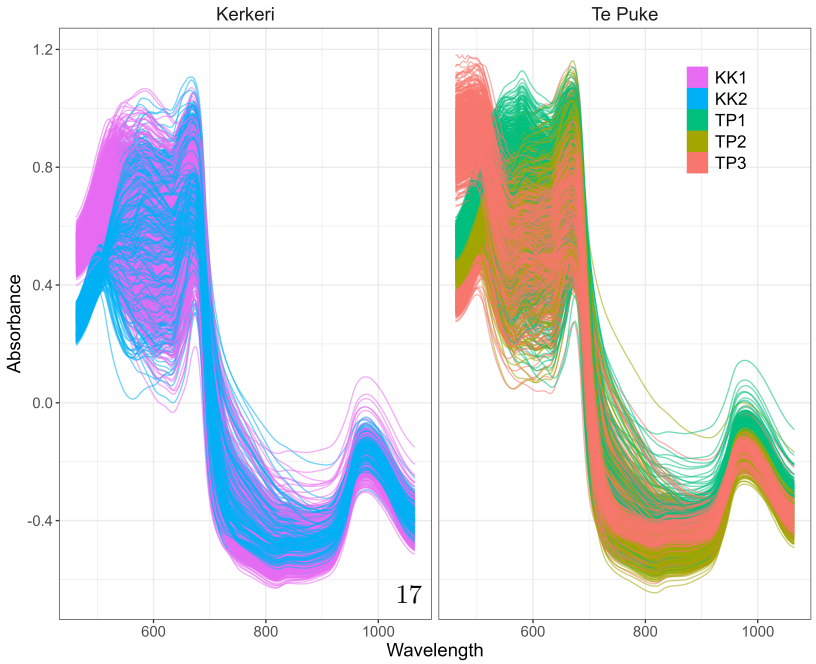

Figure 5: NIR spectra by device. Scans within site are on the same fruit.

Dataset			Deep CNN RMSE			
Training	Test	Measure	No-Aug	Bjerrum Aug	MVN Aug (excl test)	MVN Aug
Kerikeri	TP1	DMC	2.63	1.14	1.43	**1.09**
Kerikeri	TP2	DMC	2.15	2.7	**1.08**	1.17
Kerikeri	TP3	DMC	2.75	2.85	**1.93**	2.88
Te Puke	KK1	DMC	1.43	1.33	**0.92**	0.98
Te Puke	KK2	DMC	1.46	1.78	1.8	**1.07**
Kerikeri	TP1	SSC	1.6	3.81	2.17	**1.2**
Kerikeri	TP2	SSC	1.71	5.65	1.48	**1.29**
Kerikeri	TP3	SSC	1.64	6.41	2.12	**1.35**
Te Puke	KK1	SSC	2.35	1.84	1.73	**1.13**
Te Puke	KK2	SSC	2.51	1.66	2.14	**1.44**

Table 3: Results of Deep CNN trained on non-augmented data, data augmented using the method of Bjerrum, and MVN-augmented training data, from either Te Puke (TP1, TP2, TP3 training sets) or Kerikeri (KK1, KK2 training sets), based on both, the covariance excluding all devices from the test site (excl test) and including all devices.

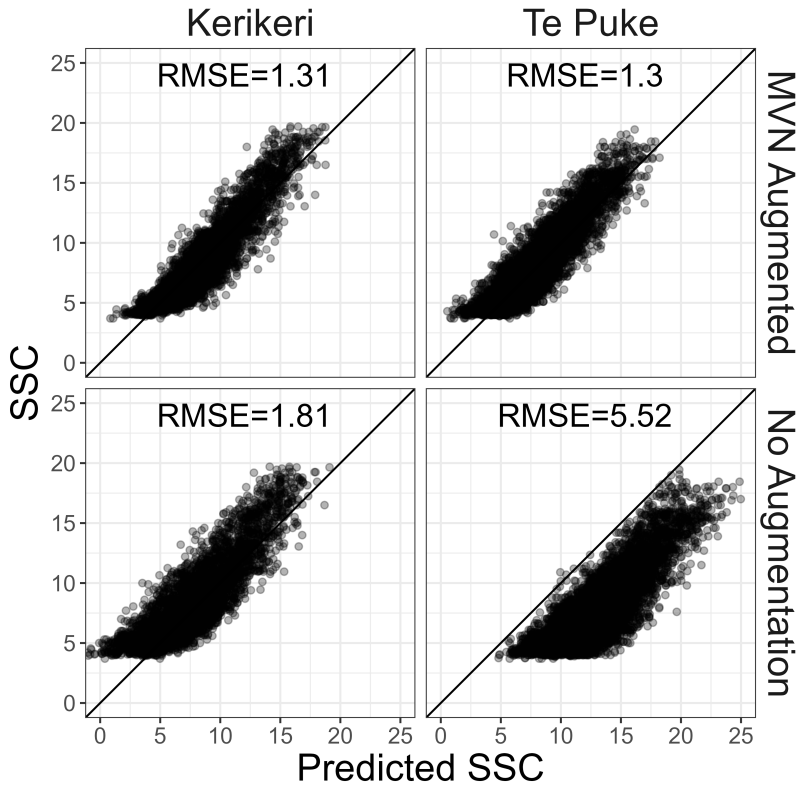

Figure 6: PLSR test set SSC predictions based on non-augmented (bottom) or augmented (top) data calibration. In the left column are models trained on Te Puke data and tested on Kerikeri data. In the right column are models trained on Kerikeri data and tested on Te Puke data.

4.1.4. Comparison of CNN and PLSR

The results of the PLSR and deep learning models, when trained on the entire training set of all devices and tested on the respective test data, applying augmentation based on the covariance matrix established using all devices, are summarised in Table 4. In general, the deep convolutional neural network resulted in lower RMSE across the SSC test set, and to a lesser extent, DMC test set, compared to the PLSR model.

Results of models with and without augmentation and pre-processing trained on the full training set, with data from both sites included, are summarised in

		DMC RMSE			SSC RMSE		
Device	N	Shallow CNN	Deep CNN	PLSR	Shallow CNN	Deep CNN	PLSR
TP1	300	**1.23**	1.32	1.29	1.47	1.22	**1.21**
TP2	300	**1.28**	1.36	1.38	1.38	**1.33**	1.54
TP3	300	1.29	1.35	**1.00**	1.60	**1.28**	1.29
KK1	300	**1.00**	1.07	1.28	1.31	**1.30**	1.33
KK2	298	0.91	0.96	**0.82**	1.35	**1.31**	1.40

Table 4: Kiwifruit DMC and SSC test set results on 600 individual fruit (300 per site). PLSR includes Savitsky-Golay 2nd derivative pre-processing. CNN's both use MVN augmentation

Table 5. Both the deep models as well as the shallow models found using hyperparameter optimisation benefit from MVN augmentation. To a lesser extent the augmentation method of Bjerrum et al. improved performance in the DMC predictions for the CNN models. PLSR however showed little change in performance across the different pre-processing techniques. This is somewhat surprising as Savitsky-Golay pre-processing proved effective when separately fitting individual PLSR models to each device (data not shown). We observed that CNNs in general worked better than PLSR which is consistent with other studies [23] [13] [3].

Model	Measure	No Aug	Savitsky-Golay 2nd	Bjerrum Aug	MVN Aug
PLSR	DMC	1.20	1.18	1.22	**1.16**
Shallow CNN	DMC	19.38	18.61	1.45	1.20
Deep CNN	DMC	1.28	1.47	1.23	1.22
PLSR	SSC	1.33	1.36	1.34	1.37
Shallow CNN	SSC	1.77	2.25	1.71	1.40
Deep CNN	SSC	1.72	1.44	1.78	**1.29**

Table 5: DMC and SSC RMSE of the test set for PLSR, shallow CNN models, and deep CNN models, trained on data that was not augmented or pre-processed, Savitsky-Golay 2nd derivative data, data augmented using Bjerrum et al.'s method, or data augmented using MVN.

4.2. Mango Results

Similar to the kiwifruit results, training on the augmented data led to faster convergence on the Mango data. The validation training history of the shallow architecture, inherited from the kiwifruit experiments for DMC, is shown in Figure 7. The difference in the length of the lines is due to the learning rate scheduler, which will reduce the learning rate and later terminate training if no further improvement in the validation MSE is found. The augmented data improves the training of the network. This improvement was also observed while training the deep network albeit with a higher RMSE than the shallow network. This is consistent with the kiwifruit data, where a simpler model was sufficient in predicting DMC. The simpler architecture of a single convolutional layer performed well compared to the PLSR models. Of the PLSR models, the 2nd derivative Savitsky-Golay method produced slightly better overall results and is reported here.

Figure 8 shows the validation and test performance of repeated training of shallow and deep networks with and without augmentation. The PLSR baseline RMSEs are also included for reference. It can be seen that the deep-learning models without data augmentation are variable in terms of prediction RMSE. The shallow deep-learning model with only one convolutional layer had consistently better predictions when augmentation was used. However, the more complex deep model took far longer to train, providing worse results, if only slightly, than the PLSR model. A comparison of the test set predictions for the PLSR and shallow network is shown in Figure 9, with the shallow CNN model providing a better fit when comparing RMSE.

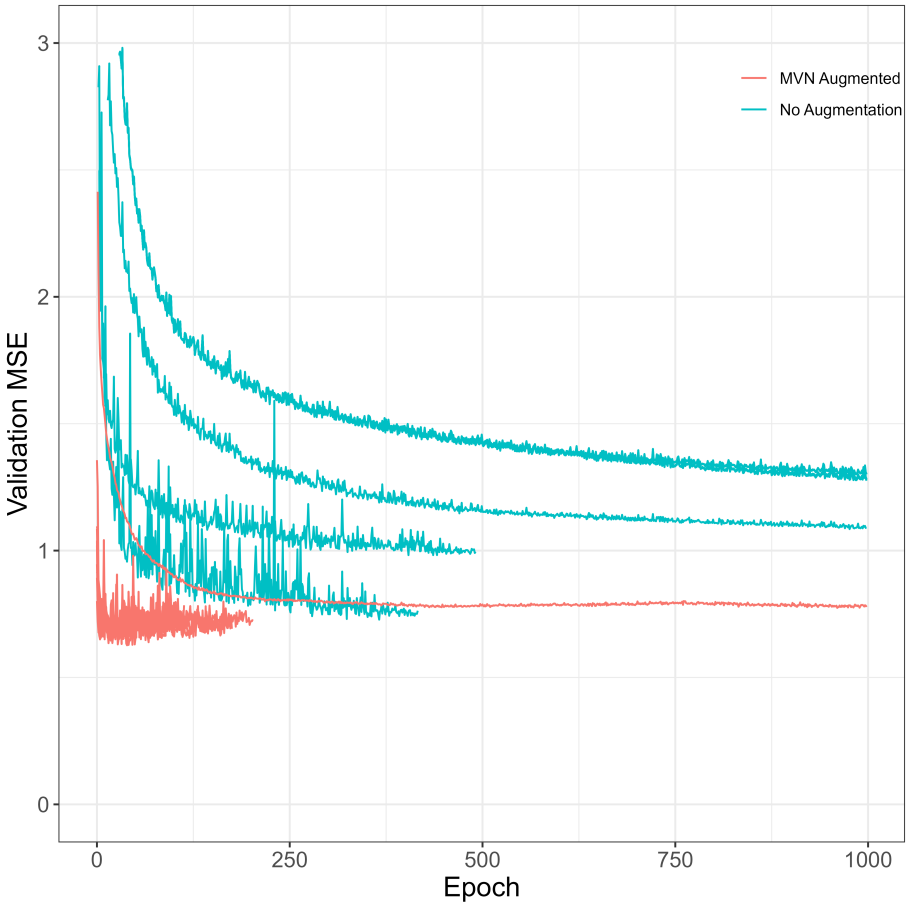

Figure 7: Comparison of validation MSE for shallow model trained on non-augmented and MVN augmented mango data. Training includes early stopping in some cases. The figure includes the histories of ten runs for each model trained on the observed and augmented data. Note that some lines overlap.

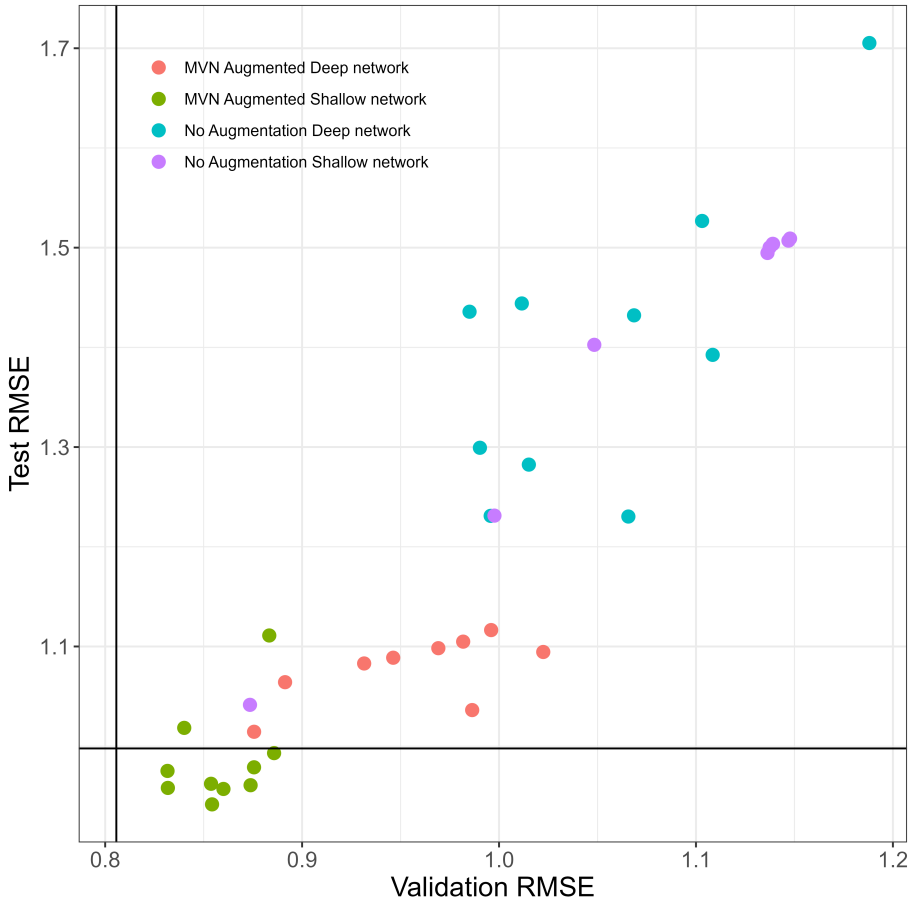

Figure 8: Validation vs. test RMSE, lines represent PLSR RMSE. Each point represents a randomly initialised model for the given architecture, with and without augmentation. The horizontal and vertical black lines indicate the PLSR test and validation RMSEs respectively.

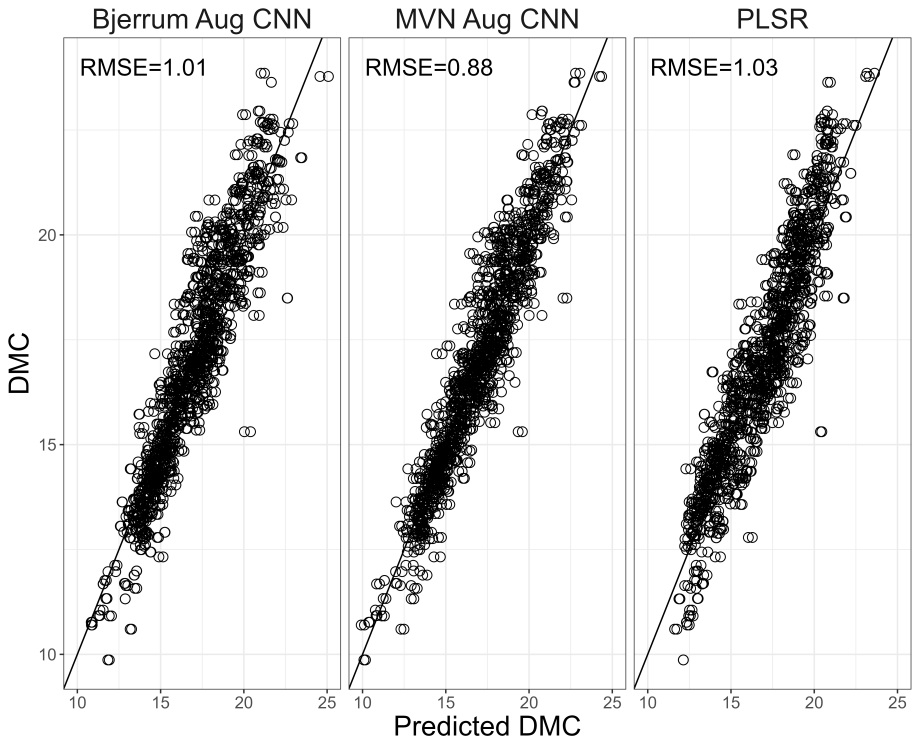

Figure 9: Comparison of model fits on the Mango test set for a shallow CNN trained on Bjerrum augmented data (left), MVN augmented data (middle) and PLSR with Savitsky-Golay second derivative pre-processing (right).

5. Conclusion

We have presented a data augmentation method based on sampling spectra from a multivariate normal distribution with empirically estimated covariance matrices. The primary benefit was observed when training convolutional neural networks: faster convergence and better validation performance were observed. The PLSR analysis also benefited from the same augmentation when the covariance matrix included information relating to between-device information. This proved useful when predicting on devices not included in the training set, albeit with most of the performance increase observed when the test devices were included in the covariance matrix estimation.With a reasonable estimate of the covariance matrix, there is potential to apply the method to other devices of the same type, irrespective of the measurement unit. This potential was demonstrated by successfully applying the technique to the independent Mango dataset.

The augmentation was particularly useful in training convolutional neural networks, both shallow and deeper architectures which is consistent with the results found by [13].

The inclusion of other sources of variation in the covariance matrix should be investigated. As the devices are nested within the site, the effects of operators and sites, and the cultivars measured there, are confounded with device. Therefore, while the augmentation simulates variation due to devices, it would be more robust to include other sources. This would require a new set of repeated measurements on fruit under various scenarios.

The suggested augmentation technique may help improve the quality of non-destructive measures of fruit quality, which will aid recovery of fruit that would otherwise be deemed to not meet an industry export standard.

6. Data

The absorbance spectra, DMC and anonymised sample metadata used in the kiwifruit analysis will be made available pending approval. Code used in

the analyses will be available on acceptance.

7. Acknowledgements

We thank Harpreet Kaur (The New Zealand Institute for Plant and Food Research Limited) and Liz Popowski at the kiwifruit breeding centre (KBC) for their help collating the kiwifruit measurements and overseeing the collection of such a rich data set. We also thank the KBC staff of Susan Murphy at Te Puke, Ann Krebs, Gustavo Hernandez-Grijota, Lisa Anderson at Kerikeri for collecting the NIR spectra over a number of seasons, as well as the KBC fast lab staff who collected the SCC and DMC data. The research received financial support from the New Zealand Ministry of Business Innovation & Employment (MBIE) as part of the Endeavour funded project *Perfecting storage life prediction for delivery of high quality fruit.*

References

[1] H. Wang, J. Peng, C. Xie, Y. Bao, Y. He, Fruit Quality Evaluation Using Spectroscopy Technology: A Review, Sensors (Basel, Switzerland) 15 (5) (2015) 11889. doi:10.3390/S150511889.
URL /pmc/articles/PMC4481958//pmc/articles/PMC4481958/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC4481958/

[2] K. B. Walsh, V. A. McGlone, D. H. Han, The uses of near infra-red spectroscopy in postharvest decision support: A review, Postharvest Biology and Technology 163 (2020) 111139. doi:10.1016/J.POSTHARVBIO.2020.111139.

[3] C. Cui, T. Fearn, Modern practical convolutional neural networks for multivariate regression: Applications to NIR calibration, Chemometrics and Intelligent Laboratory Systems 182 (2018) 9–20. doi:10.1016/j.chemolab.2018.07.008.

[4] X. Zhang, J. Yang, T. Lin, Y. Ying, Food and agro-product quality evaluation based on spectroscopy and deep learning: A review, Trends in Food Science & Technology 112 (2021) 431–441. doi:10.1016/J.TIFS.2021.04.008.

[5] P. Geladi, B. R. Kowalski, Partial least-squares regression: a tutorial, Analytica Chimica Acta 185 (C) (1986) 1–17. doi:10.1016/0003-2670(86)80028-9.

[6] A. Rinnan, F. v. d. Berg, S. B. Engelsen, Review of the most common preprocessing techniques for near-infrared spectra, TrAC Trends in Analytical Chemistry 28 (10) (2009) 1201–1222. doi:10.1016/J.TRAC.2009.07.007.

[7] K. B. Walsh, J. Blasco, M. Zude-Sasse, X. Sun, Visible-NIR 'point' spectroscopy in postharvest fruit and vegetable assessment: The science behind three decades of commercial use, Postharvest Biology and Technology 168 (2020) 111246. doi:10.1016/J.POSTHARVBIO.2020.111246.

[8] E. Bouveresse, D. L. Massart, Standardisation of near-infrared spectrometric instruments: A review, Vibrational Spectroscopy 11 (1) (1996) 3–15. doi:10.1016/0924-2031(95)00055-0.

[9] H. Kaur, R. Künnemeyer, A. McGlone, Comparison of hand-held near infrared spectrophotometers for fruit dry matter assessment:, http://dx.doi.org/10.1177/0967033517725530 25 (4) (2017) 267–277. doi:10.1177/0967033517725530.
URL https://journals.sagepub.com/doi/10.1177/0967033517725530

[10] C. Shorten, T. M. Khoshgoftaar, A survey on Image Data Augmentation for Deep Learning, Journal of Big Data 6 (1) (2019) 1–48. doi:10.1186/S40537-019-0197-0/FIGURES/33.
URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[11] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D. J. Inman, 1D convolutional neural networks and applications: A survey, Mechanical Systems and Signal Processing 151 (2021) 107398. doi:10.1016/J.YMSSP.2020.107398.

[12] A. K. Conlin, E. B. Martin, A. J. Morris, Data augmentation: an alternative approach to the analysis of spectroscopic data, Chemometrics and Intelligent Laboratory Systems 44 (1-2) (1998) 161–173. doi:10.1016/S0169-7439(98)00071-9.

[13] E. J. Bjerrum, M. Glahder, T. Skov, Data Augmentation of Spectral Data for Convolutional Neural Network (CNN) Based Deep Chemometrics (10 2017).
URL http://arxiv.org/abs/1710.01927

[14] U. Blazhko, V. Shapaval, V. Kovalev, A. Kohler, Comparison of augmentation and pre-processing for deep learning and chemometric classification of infrared spectra, Chemometrics and Intelligent Laboratory Systems 215 (2021) 104367. doi:10.1016/J.CHEMOLAB.2021.104367.

[15] P. Mishra, D. Passos, A synergistic use of chemometrics and deep learning improved the predictive performance of near-infrared spectroscopy models for dry matter prediction in mango fruit, Chemometrics and Intelligent Laboratory Systems 212 (February) (2021). doi:10.1016/j.chemolab.2021.104287.

[16] J. Acquarelli, T. van Laarhoven, J. Gerretzen, T. N. Tran, L. M. Buydens, E. Marchiori, Convolutional neural networks for vibrational spectroscopic data analysis, Analytica Chimica Acta 954 (2017) 22–31. doi:10.1016/J.ACA.2016.12.010.

[17] F. R. Harker, B. T. Carr, M. Lenjo, E. A. MacRae, W. V. Wismer, K. B. Marsh, M. Williams, A. White, C. M. Lund, S. B. Walker, others, Consumer liking for kiwifruit flavour: A meta-analysis of five studies on fruit quality, Food Quality and Preference 20 (1) (2009) 30–41.

[18] N. T. Anderson, K. B. Walsh, P. P. Subedi, C. H. Hayes, Achieving robustness across season, location and cultivar for a NIRS model for intact mango fruit dry matter content, Postharvest Biology and Technology 168 (2020) 111202. doi:10.1016/J.POSTHARVBIO.2020.111202.

[19] N. T. Anderson, K. B. Walsh, J. R. Flynn, J. P. Walsh, Achieving robustness across season, location and cultivar for a NIRS model for intact mango fruit dry matter content. II. Local PLS and nonlinear models, Postharvest Biology and Technology 171 (2021) 111358. doi:10.1016/J.POSTHARVBIO.2020.111358.

[20] F. Pedregosa, R. Weiss, M. Brucher, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
URL http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html%5Cnhttp://arxiv.org/abs/1201.0490

[21] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D. D. Cox, Hyperopt: A Python library for model selection and hyperparameter optimization, Computational Science and Discovery 8 (1) (2015) 014008. doi:10.1088/1749-4699/8/1/014008.
URL https://iopscience.iop.org/article/10.1088/1749-4699/8/1/014008https://iopscience.iop.org/article/10.1088/1749-4699/8/1/014008/meta

[22] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, International Conference on Learning Representations, ICLR, 2015.
URL https://arxiv.org/abs/1412.6980v9

[23] P. Mishra, D. Passos, F. Marini, J. Xu, J. M. Amigo, A. A. Gowen, J. J.

Jansen, A. Biancolillo, J. M. Roger, D. N. Rutledge, A. Nordon, Deep learning for near-infrared spectral data modelling: Hypes and benefits, TrAC Trends in Analytical Chemistry 157 (2022) 116804. doi:10.1016/J.TRAC.2022.116804.