Improving Deep Learning on Hyperspectral Images of Grain by Incorporating Domain Knowledge from Chemometrics

Ole-Christian Galbo Engstrøm[1,2,3], Erik Schou Dreier[2], Birthe Møller Jespersen[3],
and Kim Steenstrup Pedersen[1,4]
[1]: Department of Computer Science (DIKU), University of Copenhagen, Denmark
[2]: FOSS Analytical A/S, Denmark
[3]: Department of Food Science (UCPH FOOD), University of Copenhagen, Denmark
[4]: Natural History Museum of Denmark (NHMD), University of Copenhagen, Denmark
ocge@foss.dk, esd@foss.dk, bm@food.ku.dk, kimstp@di.ku.dk

Abstract

We demonstrate how to design and apply domain-specific modifications to convolutional neural networks (CNNs) to improve model performance on hyperspectral images of grain kernels. We use hyperspectral images of grain kernels captured in the near-infrared wavelength range of 900 to 1700 nm as a case for supporting our argumentation. This part of the electromagnetic spectrum contains convoluted signals with chemical and physical information relevant to grain quality. For standard chemometric models, domain knowledge is used to select from a plethora of combinations of preprocessing techniques helpful in extracting relevant chemical and physical features for a given task. By incorporating domain-specific design modifications in the preexisting architectures of ResNet-18 and a simple CNN, we show that model performance can be increased significantly and that applying domain knowledge to CNNs is much more important than complexity is to their performance.

1. Introduction

General deep learning models for image analysis achieve good results across diverse domains [34, 37] including agricultural domains [8, 21, 26, 29, 38]. However, we argue that incorporating domain knowledge into the model design leads to significantly better results than those achievable by general deep learning models. We use grain quality analysis as an example to support our argument.

Grain quality analysis is a multi-parameter problem class consisting of physical and chemical properties [28, 31]. Historically, chemometricians have approached this problem class by combining physical and chemical knowledge with machine learning algorithms to predict quality parameters from near-infrared (NIR) spectra of grain [27, 28]. Application of chemometrics and NIR spectroscopy to grain quality analysis is an active field of research and continues as a primary analytical tool for grain quality analysis [9, 13, 28]. While research for grain quality analysis has been conducted within various subfields of computer vision [35, 45], NIR hyperspectral imaging (NIR-HSI) [20] is perhaps the most promising, as it combines spatial features with spectra containing chemical information regarding the biological quality of the grain [32] enabling analysis of both physical and chemical properties [9, 15].

Deep learning algorithms, specifically convolutional neural networks (CNNs), have seen widespread application within HSI for agricultural domains [21, 22, 46]. We know from chemometrics that the choice of spectral preprocessing is essential for downstream model performance [13], yet hard to choose before model validation [39, 42]. Likewise, this paper shows that spectral preprocessing is crucial for CNNs applied to hyperspectral images. However, while exhaustively searching the space of preprocessing techniques is possible for chemometric methods, the computation required for CNNs makes such a search infeasible. Therefore, guided by domain-specific knowledge from chemometrics and the physics of the imaging process, we design an extension to any pre-existing CNN that allows it to learn the optimal spectral preprocessing and, consequently, achieve significantly better performance than its plain counterpart.

To assess the importance of our extensions relative to employing sophisticated CNNs, we design a simple, shallow CNN and compare it to the well-known ResNet-18 from the high-performant ResNet-family [18]. We compare these models' performances to that of Partial Least Squares (PLS) [49], a standard method in chemometrics that we use as a baseline. We explain the models in detail in Sec. 3.

We use two datasets with hyperspectral images of grain kernels from Engstrøm *et al*. [14] and Dreier *et al*. [12], respectively. The first dataset contains reference values for the protein content in the grain kernels, while the second dataset contains class labels for the grain type of the kernels. We explain the datasets in detail in Sec. 2. For both datasets, it holds that the problems are in a low sample size scenario since obtaining this type of labeled data for model training is particularly costly. Problems like these can particularly benefit from adding constraints and prior knowledge into the model design [16, 30].

To summarise, our contributions are: (1) The design of simple extensions to any pre-existing CNN, facilitating end-to-end training on hyperspectral images of grain with no requirements for the application of advanced preprocessing techniques. (2) Unifying chemometrics and deep learning with an analysis of the CNN extensions' effects on both chemical and physical prediction tasks using domain knowledge from chemometrics. (3) Exemplifying that incorporating domain knowledge in the CNN design for agricultural image analysis is more critical than increasing model complexity.

In this article, we begin with a presentation of the grain quality analysis datasets in Sec. 2 followed by details of model designs in Sec. 3 and experimental designs in Sec. 4. We analyze and discuss the results in Sec. 5, which we conclude upon in Sec. 6.

2. Datasets

In this paper, we use two NIR-HSI datasets from Engstrøm *et al*. [14] and Dreier *et al*. [12], respectively. Both datasets consist of hyperspectral images with 224 uniformly distributed wavelength channels in the 900 nm - 1700 nm NIR range of the electromagnetic spectrum taken with a *Specim FX17* camera [43]. This spectral range contains chemical information relevant for grain quality assessment [32]. Dataset #1 consists of images of bulk wheat grain kernels with reference values for mean protein contents on a physical sample scale with values between 8.66% and 17.78% [14]. The grain in Dataset #1 is from the FOSS [4] European and World Grain Networks. Dataset #2 consists of images of different types of bulk rye and wheat kernels and contains class labels for the grain variety on an image level where each image has precisely a single grain variety [12]. The dataset includes one rye variety and seven wheat varieties. Thus, the tasks on the datasets are mean protein content regression and grain variety classification, respectively. For Dataset #1, we reuse the data split as provided by [14], who divided it into six splits; five for 5-fold cross-validation (CV) and one for testing. Similarly, for Dataset #2, we reuse the dataset split as provided by [12], who provide a training, validation, and testing split.

The hyperspectral images in both datasets have a spatial size that varies slightly due to the line scan nature of the camera. A typical size, however, is approximately 800×500 pixels. Both datasets contain images of sparsely and densely packed grain with varying grain density across the images. The authors of [14] and [12] use a similar but slightly different strategy for cropping the hyperspectral images into a collection of crops of 128×128 pixels. Both use the grain density ratio to decide whether a hyperspectral image crop should be retained or discarded in the final dataset. The grain density ratio of a hyperspectral image crop is defined as the number of pixels containing grain to the total amount of pixels. The computation of this ratio is based on a binary semantic segmentation mask produced by thresholding based on Otsu [33] threshold selection provided with both datasets. Where Engstrøm *et al*. [14] use a grain density ratio of 0.1, Dreier *et al*. [12] use one of 0.5. In our work, we opt to use 0.1 for both datasets. Both Engstrøm *et al*. [14] and Dreier *et al*. [12] crop with 50% overlap in both spatial dimensions. However, whereas Engstrøm *et al*. [14] start cropping at the first row and column containing grain, as determined by the segmentation mask, Dreier *et al*. [12] start cropping at the center of the image. In our work, we opt for a third approach, where we start cropping at the top left corner. For Dataset #1, our approach yields $69,630$ hyperspectral image crops to be used in the 5-fold CV and $17,783$ hyperspectral image crops for testing. For Dataset #2, our approach yields $15,376$ hyperspectral image crops for training, $7,967$ hyperspectral image crops for validation, and $3,274$ hyperspectral image crops for testing. We summarize the datasets in Table 1.

Engstrøm *et al*. [14] and Dreier *et al*. [12] use a different approach for reducing the raw 224 channels. In this work, we opt for the method used by Engstrøm *et al*. [14] consisting of removing the first and last 10 channels, which are noisy due to low camera sensitivity, and binning the remaining 204 channels by averaging neighboring pairs, reducing the number of channels to 102. Engstrøm *et al*. [14] and Dreier *et al*. [12] transform the reflectance images to absorbance images and mask the resulting absorbance, zeroing, any non-grain pixels. We apply the same transformation.

Additionally, for each hyperspectral image crop, a mean grain spectrum is computed for use by chemometric models that take a spectrum as input. We derive the mean grain spectrum by averaging the spectrum of each grain pixel in the hyperspectral image crop, determined by the segmentation mask. Figure 1 shows a masked hyperspectral image crop, a spectrum at a single grain pixel, and a mean grain spectrum over the hyperspectral image crop.

3. Models

As a chemometric baseline model to determine if the comparatively large CNNs utilizing the entire spatio-spectral image yield increased performance over chemo-

Dataset	# image crops	Task
#1	5-fold CV: $69,630$. Test: $17,783$.	Protein content regression. Range: $8.66\% - 17.78\%$.
#2	Train: $15,376$. Val: $7,967$. Test: $3,274$.	Grain variety classification. 1 rye variety and 7 wheat varieties.

Table 1. The number of image crops and the associated task for the two datasets.

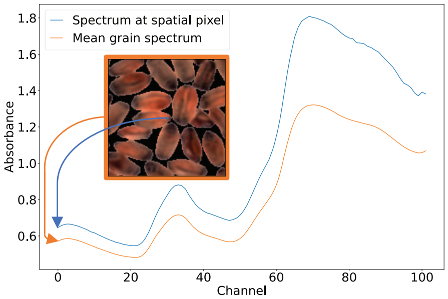

Figure 1. A masked hyperspectral image crop of spatial size 128 × 128 pixels along with the spectrum at one of the pixels and the mean spectrum of the crop computed over all the grain pixels, both containing 102 wavelength channels.

metric methods using only a mean spectrum, we implement PLS [49]. On Dataset #1, we apply PLS regression (PLS-R) [50, 51]. On Dataset #2, we apply PLS discriminant analysis (PLS-DA) [6]. PLS-DA is a method that uses a binary encoding of the categorical variables and performs PLS-R from the input to the binary-encoded categorical variables. For multiple target variables, as in multiclass classification, we can choose between two PLS-DA methods [25]. We can train an ensemble of PLS-DA algorithms in a one-versus-all manner for each class or train a single multiclass classifier for all classes using one-hot encoding of the categorical variables, commonly referred to as a PLS2-DA model. Our experiments showed that PLS2-DA was consistently better than an equivalent PLS-DA ensemble. Thus, from now on, we will focus on PLS2-DA regarding chemometric methods for classification on Dataset #2.

We implement two plain CNN-based models to which we apply and assess our extensions. The first plain CNN is a ResNet-18 [18] as used by Engstrøm *et al.* [14] where they swap the ordering of rectified linear unit (ReLU) and batch normalization (BN) [19]. The second is a much simpler 5-layer CNN, which we name SimpleNet, where each convolution layer has the number of filters with the same kernel size as those within the corresponding ResNet block. Following the structure of ResNet-18, each convolution layer uses a ReLU activation followed by a BN layer. Figure 2 shows the architecture of SimpleNet. Both networks have a fully connected linear output layer with one neuron for the protein regression task on Dataset #1 and a fully connected layer with eight neurons and a softmax activation for the grain variety classification task of Dataset #2.

Inspired by Engstrøm *et al.* [14], we experiment with two types of extensions and combinations of these extensions. The first extension consists of adding a 3-dimensional convolution (Conv3D) layer as the first layer in the network. A Conv3D layer facilitates learning spectral smoothing and derivative filters, commonly used in NIR spectroscopy [13, 39]. We denote such a layer by Conv3D$_{n,(h\times w\times d)}$ where n is the number of filters and h, w, and d are the kernel's height, width, and depth, respectively. The second extension, which we name a Downsampler (Ds$_m$), consists of adding an initial 2-dimensional convolution (Conv2D) layer that applies dimensionality reduction to the spectral dimension with m number of filters, each applying a 1×1 kernel for every spectral channel. The number of filters, m, in Ds $_m$ resembles the number of components chosen by PLS. We denote a PLS model using m components by A$_m$. While Engstrøm *et al.* [14] experiment with two combinations of these extensions, a Conv3D layer followed by a Ds$_3$ layer and a Ds$_3$ without a prior Conv3D layer, we extend this also to include a Conv3D layer without a subsequent Ds layer. Figure 2 shows the combinations of extensions we employ.

Both Engstrøm *et al.* [14] and Dreier *et al.* [12] employ grayscale variants of the plain ResNet-18 with the latter also using a 3-dimensional variant of the plain ResNet-18 where each 2-dimensional convolution and pooling layer is replaced with an equivalent 3-dimensional layer. We follow this practice and employ grayscale and 3-dimensional variants of plain ResNet-18 and plain SimpleNet. The grayscale variant determines if the problems can be solved using a purely spatial approach by averaging the spectral dimension and feeding the resulting grayscale image to the CNN. In contrast, the 3-dimensional variant lends itself naturally to the hyperspectral image, accounting for spatio-spectral features.

3.1. Implementation Details

The CNN and PLS models are implemented using TensorFlow [1], and Keras [10]. While 32 bits of floating point precision is sufficient for the CNNs, the PLS models require 64 bits of floating point precision to converge with increasing values of A$_m$. Unlike CNNs, PLS does not compute a bias coefficient, which is necessary as it assumes a proportionality between the input spectrum and the target variable. To account for this, we can either augment the input spectrum with a constant extra channel or center the target variable. We opt for the latter approach and center the protein content around the mean of each training split in Dataset #1 before training PLS-R. For PLS-DA, the issue is nullified by applying sigmoid, ensuring that a PLS output of zero yields an equiprobable class prediction when combined with the binary encoding of target variables. For PLS2-DA, using one-hot encoding nullifies the issue altogether.

For the CNNs, we initialize all weights with the Kaiming He Normal Distribution [17] and initialize all biases to zero. Unless explicitly stated otherwise, convolution and pooling layers apply zero-padding. For CNN regressors on Dataset #1, we use the root mean squared error (RMSE) as the loss function. For CNN classifiers on Dataset #2, the loss function is the weighted categorical cross entropy (CCE) with balanced class weights from scikit-learn [36]. In practice, the class weights are close to 1 as Dataset #2 is very balanced. In both cases, we apply L2-regularization (L2) with a regularization parameter of 10^{-3}. For SimpleNet, L2 is applied to all weights. For ResNet-18, we follow the guidelines by Kim *et al*. [23] regarding when to apply L2 to BN weights, termed γ, by Ioffe and Szegedy [19]. Additionally, for all other layers in ResNet-18, we apply L2 to their weights. We use stochastic gradient descent (SGD) optimization with a batch size of 32, a momentum of 0.9, and an initial learning rate of 0.1. We multiply the learning rate by 0.1 if the validation RMSE or CCE plateaus for 10 epochs. When reducing the learning rate, we restore the current best weights and continue the training from that point on. If the validation RMSE or validation CCE plateaus for 50 epochs or reaches a total of 1000 epochs, training is halted and the best weights restored. In practice, all CNNs were halted by the 50 epochs plateau criterion. While training CNNs, we apply data augmentation to the hyperspectral image crops by uniformly randomly flipping them vertically and horizontally.

PLS can be implemented with several different algorithms. We choose to implement PLS using Improved Kernel PLS Algorithm #1 [11] as it is both fast [2] and numerically stable [3]. For PLS-R on Dataset #1, we choose A_m for each of the CV splits as the number of PLS components that yields the lowest validation RMSE. Although previously attempted [44], computing CCE on the output of PLS2-DA is not a good metric of its performance. PLS2-DA outputs a vector that does not necessarily encode a discrete probability distribution. While applying the softmax function to the output of a PLS2-DA model does not change its categorical prediction, when interpreted as an arg max of the prediction vector, it does not encode a probability distribution that allows for meaningful application of CCE. A perfect PLS2-DA model will predict the ground truth before any softmax application. Thus, any subsequent application of softmax would yield a non-zero CCE. Therefore, for PLS2-DA on Dataset #2, we choose A_m based on the highest weighted validation categorical accuracy, which is unaffected by the softmax application.

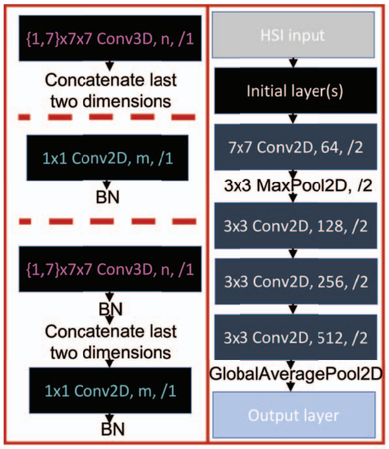

Figure 2. Top left: $\mathrm{Conv3D}_{n,(\{1,7\}\times7\times7)}$. Middle left: Ds_m. Bottom left: $\mathrm{Conv3D}_{n,(\{1,7\}\times7\times7)}$ followed by Ds_m. Right: The architecture of SimpleNet. The notation is kernel $h \times w$, number of filters, /stride.

4. Experiments

We wish to analyze whether we can use domain-specific knowledge regarding preprocessing of NIR spectra to design end-to-end trained CNNs. Preprocessing of NIR spectra can significantly increase the performance of subsequent linear models such as PLS [39]. However, training CNNs is several orders of magnitude slower than training PLS models, rendering it impossible to test a wide range of spectral preprocessing techniques for CNNs.

Our experiments with preprocessing techniques for CNNs are two-fold. First, we apply only simple preprocessing, such as channel-wise centering and scaling, and allow the CNNs to learn their own more advanced spectral preprocessing by applying different types of Conv3D layers. We compare this approach with optimizing the spectral preprocessing technique on PLS. Optimizing the spectral preprocessing technique on PLS allows us to test a wide range of combinations and apply the best one to the CNNs to assess if the optimal preprocessing technique for PLS is also beneficial to CNNs.

4.1. Spectral Preprocessing Optimization

Preprocessing techniques within NIR spectroscopy generally fall within the two categories of scatter correction and spectral derivatives, aiming to reduce the effects of physical phenomena on the spectrum, such as scattering, thus revealing the chemical information in the spectra [39, 42]. In our experiments, the scatter correction techniques are represented by Standard Normal Variate (SNV) [7], and detrending (Detrend) [7], while a wide range of Savitzky-Golay filters (SG) represents the spectral derivative techniques [40].

SNV standardizes each spectrum by subtracting its mean and dividing by its standard deviation. Detrend subtracts from each spectrum a polynomial that has been fitted to that spectrum. We use a second-order polynomial as is standard for Detrend [39]. When used in unison, it is recommended to apply SNV before Detrend [7, 39].

SG applies convolution in the spectral dimension with a pre-defined filter that is determined by a window length w, polynomial order p, and derivative order d. The specific filter is denoted $\text{SG}_{w,p,d}$. Simply computing the derivative using finite differences would be infeasible due to noise inflation. SG tackles this issue by combining smoothing and spectral derivatives in a single filter by fitting a polynomial within a given window, thus getting a more acceptable signal-to-noise ratio [39]. We do not use padding when convolving with SG as this will introduce spectral artifacts [39]. We denote the lack of padding in SG and Conv3D by prepending a † to the name. When combining scatter correction (SNV) with differentiation (SG), scatter correction should always be applied first as the scatter correction techniques were designed for correction on raw spectra [39]. We test $\text{SG}_{w,p,d}$ with all unique combinations of $w \in [\![3, 23]\!]$, $p \in [\![0, 3]\!]$, $d \in [\![0, 2]\!]$, using only odd w. For each value of d, two subsequent values of p will yield the same filter coefficients. We remove any redundant filter coefficients and end with a total of 54 uniquely different $\text{SG}_{w,p,d}$ filters.

We also experiment with spectral centering (Center) and scaling (Scale) by computing for each training split the mean spectrum and standard deviation spectrum for channel-wise subtraction and subsequent channel-wise division, respectively. This preprocessing technique is commonly applied to PLS-R [51] and is beneficial to the convergence of neural networks [24, 41, 48]. Indeed, confirming that the literature also applies to our hyperspectral setting, we initially trained CNNs with and without Center → Scale and witnessed increased performance when applying Center → Scale. Thus, from now on, we focus on CNNs with Center → Scale. The mean spectrum and standard deviation spectrum are computed over all grain pixels using a two-pass version of the robust Welford's algorithm [47]. When we apply the spectral preprocessing techniques in sequence, the ordering is first-to-last, SNV → Detrend → †SG → Center → Scale. This ordering entails that the mean and standard deviation spectra used in the Center and Scale operations are computed after any previous preprocessing application as they depend on any prior preprocessing applications. When used on hyperspectral images, spectral preprocessing is applied only to the grain pixels, and the background is masked out.

4.2. Comparative Studies

For PLS on both datasets, we experiment with applying SNV, SNV → Detrend, SNV → †SG and neither scatter correction nor spectral derivatives. For each of these, we experiment with applying Center, Center → Scale, and neither Center nor Scale. For the CNNs, we initially experiment with the commonly used Center → Scale. Afterward, we extend this to include the preprocessing methods found optimal for PLS on Dataset #1 and Dataset #2, respectively. The best preprocessing technique for CV of PLS-R on Dataset #1 was SNV → $\dagger\text{SG}_{7,2,2}$ and for PLS2-DA validation on Dataset #2 Center was best. Using the optimal preprocessing techniques, the average optimal number of components for the CV of the best PLS-R model is A_{15}, and for validation of the best PLS2-DA, the optimum is A_{17}. In Figure 3, we show how the different combinations of SNV, Detrend, and †SG with their optimal combination of Center and Scale affect the performance of PLS in both datasets.

Based on the results of the preprocessing experiments with PLS, we design modifications of ResNet-18 and SimpleNet to understand if and how these results transfer to CNNs. These modifications include adding Ds_{15} for CNNs on Dataset #1 and Ds_{17} for Dataset #2. Additionally, as $\dagger\text{SG}_{7,2,2}$ is part of the optimal preprocessing for PLS-R on Dataset #1, we include $\dagger\text{Conv3D}_{1,(1\times1\times7)}$ as it shares the same window length and, as such, can learn the same filter if necessary. Engstrøm *et al.* [14] showed that $\dagger\text{Conv3D}_{1,(1\times1\times7)}$ can cause instability during training. Therefore, we experiment with adding additional spectral †Conv3D filters and a spatio-spectral version $\dagger\text{Conv3D}_{3,(1\times1\times7)}$ and $\dagger\text{Conv3D}_{1,(7\times7\times7)}$, both having the same spectral window length, to see if this can alleviate the instability during training, which seemingly is the case.

All CNN results, and the best PLS results, laying the foundation for the dataset-specific CNN modifications, are shown in Table 2. Additionally, for each dataset, we compare our best CNN and best PLS model with those of Engstrøm *et al.* [14] and Dreier *et al.* [12], respectively, in Table 3. We show confusion matrices and training and validation loss curves in the supplementary material. Engstrøm *et al.* [14] report better performance for PLS-R than for CNN models. While closing this performance gap, our best CNN and PLS-R models achieve lower RMSE than the best from Engstrøm *et al.* [14]. Furthermore, by constructing an ensemble CNN, we outperform PLS-R and the equivalent ensemble PLS-R. We create the ensemble by taking a uniform average of the predictions from each of the five cross-validated models. On Dataset #2, our best PLS2-DA model significantly outperforms the equivalent from Dreier *et al.* [12]. Our best CNN does not beat the best CNN from Dreier *et al.* [12]. However, they use a grain density of $\geq 50\%$, whereas ours is $\geq 10\%$. When we evaluate our model on test images with at least 50% grain density, we close much of the performance gap, indicating that lower grain density makes the classification task more difficult. The study by Dreier *et al.* [12] support this indication by showing that their classification accuracy diminishes greatly when grain density is $< 50\%$.

Mod. #	$\text{Conv3D}_{n,(h\times w\times d)}$	Ds_m (CNN) A_m (PLS)	Preprocessing	# param. $(\times 10^6)$ ResNet-18 / SimpleNet	RMSE (%) ± SEM (%) on Dataset #1 ResNet-18 / SimpleNet	Accuracy on Dataset #2 ResNet-18 / SimpleNet
Dataset #1 and Dataset #2						
1	None	None	Center → Scale	11.5 / 1.87	0.93 ± 0.06 / 0.92 ± 0.06	0.96 / 0.95
2	None	None	Center → Scale → Grayscale	11.2 / 1.56	1.43 ± 0.07 / 1.55 ± 0.05	0.91 / 0.82
3	Fully Conv3D	None	Center → Scale	33.2 / 4.67	0.92 ± 0.04 / 0.86 ± 0.03	0.96 / 0.91
4	$\text{Conv3D}_{1,(1\times1\times7)}$	None	Center → Scale	11.5 / 1.87	0.74 ± 0.05 / 0.76 ± 0.05	0.90 / 0.95
5	$\text{Conv3D}_{3,(1\times1\times7)}$	None	Center → Scale	12.1 / 2.51	0.72 ± 0.03 / 0.73 ± 0.04	0.95 / 0.92
6	$\text{Conv3D}_{1,(7\times7\times7)}$	None	Center → Scale	11.5 / 1.87	0.74 ± 0.04 / 0.81 ± 0.04	0.86 / 0.91
7	$\text{Conv3D}_{3,(7\times7\times7)}$	None	Center → Scale	12.1 / 2.51	0.68 ± 0.03 / 0.71 ± 0.03	0.86 / 0.86
8	None	Ds_3	Center → Scale	11.2 / 1.56	1.18 ± 0.08 / 1.25 ± 0.13	0.97 / 0.93
9	$\text{Conv3D}_{1,(1\times1\times7)}$	Ds_3	Center → Scale	11.2 / 1.56	0.82 ± 0.13 / 0.71 ± 0.02	0.91 / 0.94
10	$\text{Conv3D}_{3,(1\times1\times7)}$	Ds_3	Center → Scale	11.2 / 1.56	0.66 ± 0.01 / 0.71 ± 0.01	0.94 / 0.92
11	$\text{Conv3D}_{1,(7\times7\times7)}$	Ds_3	Center → Scale	11.2 / 1.56	0.69 ± 0.01 / 0.71 ± 0.03	0.88 / 0.93
12	$\text{Conv3D}_{3,(7\times7\times7)}$	Ds_3	Center → Scale	11.2 / 1.56	0.67 ± 0.02 / 0.71 ± 0.02	0.92 / 0.93
Dataset #1						
PLS-R	-	A_{15}	SNV → †$\text{SG}_{7,2,2}$	-	0.67 ± 0.01	-
13	None	Ds_{15}	SNV → †$\text{SG}_{7,2,2}$	11.2 / 1.60	0.67 ± 0.02 / 0.72 ± 0.02	-
14	†$\text{Conv3D}_{1,(1\times1\times7)}$	Ds_{15}	SNV	11.2 / 1.60	1.61 ± 0.20 / 1.36 ± 0.28	-
15	†$\text{Conv3D}_{3,(1\times1\times7)}$	Ds_{15}	SNV	11.2 / 1.61	0.90 ± 0.23 / 0.71 ± 0.02	-
16	†$\text{Conv3D}_{1,(7\times7\times7)}$	Ds_{15}	SNV	11.2 / 1.60	0.89 ± 0.23 / 0.92 ± 0.22	-
Dataset #2						
PLS2-DA	-	A_{17}	Center	-	-	0.93
17	None	Ds_{17}	Center	11.24 / 1.61	-	0.94 / 0.93

Table 2. Performance, number of parameters, and preprocessing of all CNN modifications and the best PLS-R and PLS2-DA models. The first 12 modifications are inspired or developed by Engstrøm *et al.* [14] and Dreier *et al.* [12] and are applied to Dataset #1 and Dataset #2. Modifications $13 - 16$ are designed based on the best preprocessing methods for PLS-R on Dataset #1 and are used only on Dataset #1. Modification 17 is designed based on the best preprocessing methods for PLS2-DA on Dataset #2 and is used only on Dataset #2. All results shown are computed on the test splits of the respective datasets.

5. Discussion

The spectral dimension is critical to protein content regression, as evident by the high RMSE of the gray-scale

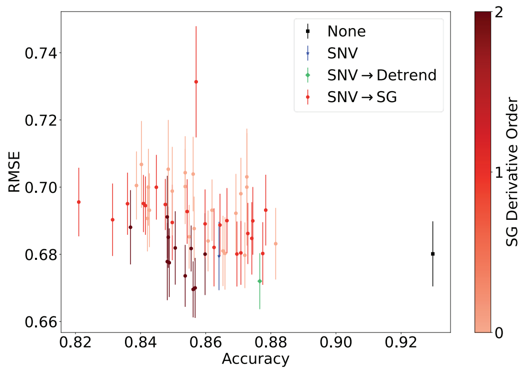

Figure 3. PLS-R and PLS2-DA performance (± SEM) on the test sets of Dataset #1 (vertical axis) and Dataset #2 (horizontal axis). The models shown here share Center and Scale applications with the model that achieves the lowest validation RMSE and the highest validation accuracy, respectively. For PLS2-DA, Center has been applied, whereas this has not been applied for PLS-R. The red markers correspond to models that have applied SNV → †SG preprocessing. Here, we choose the red shade based on the derivative order of the †SG filter. Similar plots for the window length and polynomial order are provided in the supplementary material. The outlier with a high RMSE is likely due to the combination of a very large †SG window length, a low derivative order, and a low polynomial order.

Best model	CNN	PLS
RMSE (%) **on Dataset #1**		
Ours	0.66 ± 0.01	0.67 ± 0.01
Ensemble (ours)	**0.59**	0.67
Engstøm *et al.* [14]	0.90 ± 0.05	0.75 ± 0.01
Acc. (%) **on Dataset #2**		
Ours	**96.7** (98.9)	93.0 (96.2)
Dreier *et al.* [12]	(**99.8**)	(79.9)

Table 3. Comparison of the best CNNs and PLS models with prior work on both datasets using their respective test splits. The numbers in parentheses indicate performances on a grain density of ≥ 50%. We use **bold** to highlight the best model on each dataset.

variant, modification #2, shown in Table 2 and supported by the results of Engstrøm *et al.* [14]. To achieve optimal performance, however, the spectral dimension requires delicate treatment. Removing physical phenomena in the spectra by preprocessing them with a combination of SNV, Detrend, and †SG increases the downstream model's ability to perform the chemometric regression analysis of determining protein content. While PLS can obtain decently low RMSE without this preprocessing, as seen in Figure 3, the importance of removing physical phenomena in the spectra is profound for CNNs. Inspecting Table 2 we see a vast decrease in RMSE for modification #13 applying SNV → $\dagger\text{SG}_{7,2,2}$ when compared with the plain modification #1 applying the classical Center → Scale preprocessing.

Instead of trying different †SG filters for CNNs, it is possible to have them learn their own spectral preprocessing filters using †Conv3D. These learned filters converge towards †SG for protein content regression. An example of this is shown for the model with the lowest RMSE on the protein regression, ResNet-18 modification #10, in Figure 4. This phenomenon of Conv3D learning to approximate some variation of †SG, usually a higher-order polynomial and derivative, is a tendency for every CNN trained on Dataset #1 with an initial Conv3D, spectral and spatio-spectral alike. Plots showing this tendency for our other models are provided in the supplementary material.

As shown, applying SNV → $\dagger\text{SG}_{7,2,2}$, found optimal by PLS-R, to the CNNs yields sublime results. However, by applying only SNV and allowing the CNNs to subsequently learn the remaining spectral preprocessing by training $\dagger\text{Conv3D}_{1,(1\times1\times7)}$ as done in modification #14, results worsen significantly. We hypothesize that applying a local centering and scaling of the spectra, e.g., as done by SNV, leads to instability during training. However, this issue seems to be somewhat relieved by increasing the number of spectral filters either directly as done in modification #15 or indirectly by extending †Conv3D across the spatial dimensions as done in modification #16. This hypothesis is supported by the studies done by Engstrøm *et al.* [14] whom experience the same instability for their model using $\text{Conv3D}_{1,(1\times1\times7)}$ and significantly better stability for $\text{Conv3D}_{3,(1\times1\times7)}$ and $\text{Conv3D}_{1,(7\times7\times7)}$ alike. They apply Center → Scale with mean and standard deviation spectra being computed locally on an image crop basis. While this is not as local as SNV, it is significantly more local than our global strategy.

These effects can lead to a hypothesis that increasing the number of Conv3D filters will lead to even better results. However, inspecting the $\text{Conv3D}_{3,(1\times1\times7)}$ layer learned by ResNet-18 modifications #10 as shown in Figure 4, two of the filters learn zero-responses and contribute nothing to its downstream responses. Indeed, inspecting the same model's Ds_3, shown in Figure 5, where all three filters learn zero responses for the parts of the input signal's spectral dimension that correspond to the zero-response filters of the prior $\text{Conv3D}_{3,(1\times1\times7)}$ layer. This tendency is prevalent for all our models employing $\text{Conv3D}_{3,(1\times1\times7)}$. Similar plots for the remaining models are provided in the supplementary material. Additionally, the fact that modification #2, the Fully Conv3D CNNs, the first layers of which employ $\text{Conv3D}_{64,(7\times7\times7)}$, do not provide any benefit over the standard 2D CNNs, further supports that applying as many Conv3D filters as possible is not beneficial.

Grain variety classification can be solved reasonably well using only spatial or only spectral information, as evident in Table 2 by the grayscale modification #2 and PLS2-DA. These results are supported by those of Dreier *et al.* [12]. However, simultaneously utilizing spectral and spatial information enables better performance than utilizing either alone, as evident by the plain CNNs with modification #1. For this task, the treatment of the spectral dimension is also essential, as evident for the PLS2-DA experiments shown in Figure 3. Here removing physical phenomena of the spectral dimension is detrimental to the classification accuracy. Not removing the physical phenomena of the spectra allows the PLS2-DA model to achieve much better accuracy than applying any combination of SNV, Detrend, or †SG. The results correspond with previous studies of grain classification using spectral analysis. They indicate that grain classification relies on the grains' physical characteristics and that spectral preprocessing has a detrimental effect on accuracy [5]. While less critical to the performance than for PLS2-DA, adding a single initial Conv3D layer to the CNNs offers no increase in classification accuracy. Inspecting the responses learned by the filters of Conv3D for the classification CNNs reveals that they do learn responses approximating those of †SG but always of derivative order 0 or 1, performing smoothing and removal of only additive effects, while maintaining the multiplicative effects. These results indicate that even when offered the opportunity as a trainable Conv3D layer, the CNNs learn to maintain the physical phenomena of the spectra to a higher degree than filters learned by the protein content regression CNNs. Further, inspecting Figure 3 shows a tendency towards high-order derivatives yielding better protein content regression performance. At the same time, this is not the case for grain variety classification, where the best physical-phenomena-removing preprocessing technique uses a 0 order derivative. Interestingly, when learning multiple Conv3D, the classification CNNs share the tendency of having one filter learning a dominant response. Furthermore, inspecting their subsequent Ds, it is evident that only the part of the input corresponding to the dominant Conv3D filter learns a non-minuscule response. These results indicate that employing multiple spectral filters only stabilizes the model during training but is not needed to learn multiple responses for any single-task prediction. While choosing the optimal value of A for PLS is essential to reduce the risk of overfitting, the role of Ds in CNNs is less profound. Engstrøm *et al.* [14], witness increased regression performance by employing Ds_3 to the plain ResNet-18 while in our studies, we notice an adverse effect for the same modification. However, it is usually beneficial when combined with a prior Conv3D layer. For grain variety classification, the effect seems to be minuscule.

The proper choice of spectral preprocessing method is challenging to assess before model validation for chemometric models [39, 42], yet it is required for achieving optimal performance [13]. Our CNN modifications can learn this spectral preprocessing autonomously. In this study, we have devised a blueprint for CNN design that aids in bridging the gap between generic deep learning for image analysis and domain-specific problems within chemometrics.

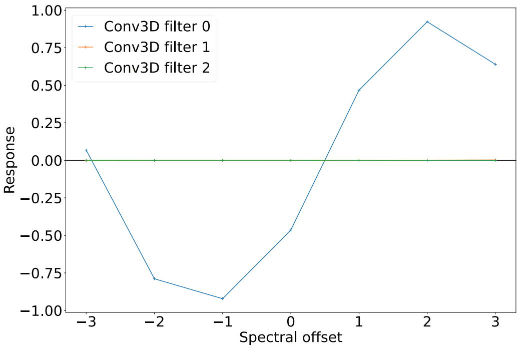

Figure 4. The three filters of $\text{Conv3D}_{3,(1\times1\times7)}$ from ResNet-18 modification #10 validated on validation split 1 of Dataset #1. Conv3D filter 1 and Conv3D filter 2 lie on top of each other at the line Response = 0.

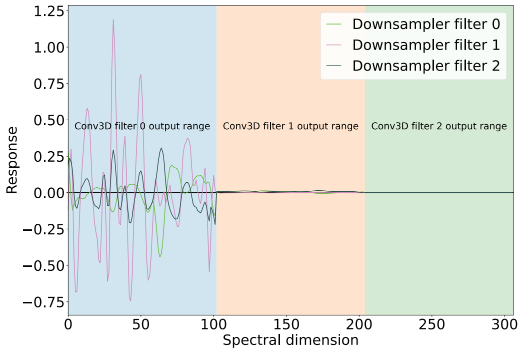

Figure 5. The three filters of Ds_3 from ResNet-18 modification #10 validated on validation split 1 of Dataset #1. The background area is colored with respect to the previous Conv3D filter responsible for producing this part of the input to the DS_3.

6. Conclusion

In this study, we have shown that by applying knowledge about the relationship between physical and chemical properties to the task at hand, we can design a modification to a standard CNN that significantly improves its predictive performance on hyperspectral images of grain. Adding an initial Conv3D layer allows the CNN to learn to remove irrelevant physical effects from the spectrum, thereby performing significantly better than its plain counterpart for regression analysis on the chemical parameter of protein content. However, when physical properties contain information valuable to the task, Conv3D filters offer no benefit over plain CNNs. Indeed, the study shows that using the proper modification is much more critical to the CNN's performance than the complexity of the network, as demonstrated by our SimpleNet modifications with a single initial Conv3D layer outperforming the plain ResNet-18 on protein content regression.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 3

[2] A. Alin. Comparison of pls algorithms when number of objects is much larger than number of variables. *Statistical papers*, 50(4):711, 2009. 4

[3] M. Andersson. A comparison of nine pls1 algorithms. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 23(10):518–529, 2009. 4

[4] Foss Analytical A/S. https://www.fossanalytics.com/. 2

[5] Y. Bao, C. Mi, N. Wu, F. Liu, and Y. He. Rapid classification of wheat grain varieties using hyperspectral imaging and chemometrics. *Applied Sciences*, 9(19):4119, 2019. 8

[6] M. Barker and W. Rayens. Partial least squares for discrimination. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(3):166–173, 2003. 3

[7] R. J. Barnes, M. S. Dhanoa, and S. J. Lister. Standard normal variate transformation and de-trending of near-infrared diffuse reflectance spectra. *Applied spectroscopy*, 43(5):772–777, 1989. 4, 5

[8] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis. Machine learning in agriculture: A comprehensive updated review. *Sensors*, 21(11):3758, 2021. 1

[9] N. Caporaso, M. B. Whitworth, and I. D. Fisk. Near-infrared spectroscopy and hyperspectral imaging for non-destructive quality assessment of cereal grains. *Applied spectroscopy reviews*, 53(8):667–687, 2018. 1

[10] F. Chollet et al. Keras. `https://keras.io`, 2015. 3

[11] B. S. Dayal and J. F. MacGregor. Improved pls algorithms. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 11(1):73–85, 1997. 4

[12] E. S. Dreier, K. M. Sørensen, T. Lund-Hansen, B. M. Jespersen, and K. S. Pedersen. Hyperspectral imaging for classification of bulk grain samples with deep convolutional neural networks. *Journal of Near Infrared Spectroscopy*, 30(3):107–121, 2022. 2, 3, 5, 6, 7

[13] Z. Du, W. Tian, M. Tilley, D. Wang, G. Zhang, and Y. Li. Quantitative assessment of wheat quality using near-infrared spectroscopy: A comprehensive review. *Comprehensive Reviews in Food Science and Food Safety*, 21(3):2956–3009, 2022. 1, 3, 8

[14] O.-C. G. Engstrøm, E. S. Dreier, and K. S. Pedersen. Predicting protein content in grain using hyperspectral deep learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1372–1380, 2021. 2, 3, 5, 6, 7, 8

[15] L. Feng, S. Zhu, F. Liu, Y. He, Y. Bao, and C. Zhang. Hyperspectral imaging for seed quality and safety inspection: A review. *Plant methods*, 15(1):1–25, 2019. 1

[16] E. Gallup, T. Gallup, and K. Powell. Physics-guided neural networks with engineering domain knowledge for hybrid process modeling. *Computers & Chemical Engineering*, page 108111, 2023. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 4

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 3, 4

[20] D. Ishikawa, M. Ishigaki, and A. A. Gowen. Nir imaging. *Near-Infrared Spectroscopy: Theory, Spectral Analysis, Instrumentation, and Applications*, pages 517–551, 2021. 1

[21] A. Kamilaris and F. X. Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018. 1

[22] A. Khan, A. D. Vibhute, S. Mali, and C. H. Patil. A systematic review on hyperspectral imaging technology with a machine and deep learning methodology for agricultural applications. *Ecological Informatics*, page 101678, 2022. 1

[23] B. J. Kim, H. Choi, H. Jang, D. G. Lee, W. Jeong, and S. W. Kim. Guidelines for the regularization of gammas in batch normalization for deep residual networks. *arXiv preprint arXiv:2205.07260*, 2022. 4

[24] Y. LeCun, I. Kanter, and S. Solla. Second order properties of error surfaces: Learning time and generalization. *Advances in neural information processing systems*, 3, 1990. 5

[25] L. C. Lee, C.-Y. Liong, and A. A. Jemain. Partial least squares-discriminant analysis (pls-da) for classification of high-dimensional (hd) data: a review of contemporary practice strategies and knowledge gaps. *Analyst*, 143:3526–3539, 2018. 3

[26] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8):2674, 2018. 1

[27] M. Manley. Near-infrared spectroscopy and hyperspectral imaging: non-destructive analysis of biological materials. *Chemical Society Reviews*, 43(24):8200–8214, 2014. 1

[28] M. Manley and P. J. Williams. Applications: food science. *Near-Infrared Spectroscopy: Theory, Spectral Analysis, Instrumentation, and Applications*, pages 347–359, 2021. 1

[29] V. Meshram, K. Patil, V. Meshram, D. Hanchate, and S. D. Ramkteke. Machine learning in agriculture domain: A state-of-art survey. *Artificial Intelligence in the Life Sciences*, 1:100010, 2021. 1

[30] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan. Incorporating prior domain knowledge into deep neural networks. In *2018 IEEE international conference on big data (big data)*, pages 36–45. IEEE, 2018. 2

[31] J. G. Nuttall, G. J. O'leary, J. F. Panozzo, C. K. Walker, K. M. Barlow, and G. J. Fitzgerald. Models of grain qual-

ity in wheat—a review. *Field crops research*, 202:136–145, 2017. 1

[32] B. G. Osborne. Near-infrared spectroscopy in food analysis. *Encyclopedia of analytical chemistry: applications, theory and instrumentation*, 2006. 1, 2

[33] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transaction on Systems, Man and Cybernetics*, SMC-9(1):62–66, 1979. 2

[34] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh. Deep learning vs. traditional computer vision. In *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1*, pages 128–144. Springer, 2020. 1

[35] D. I. Patrício and R. Rieder. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and electronics in agriculture*, 153:69–81, 2018. 1

[36] F. Pedregosa, 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, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4

[37] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017. 1

[38] C. Ren, D.-K. Kim, and D. Jeong. A survey of deep learning in agriculture: techniques and their applications. *Journal of Information Processing Systems*, 16(5):1015–1033, 2020. 1

[39] Å. Rinnan, F. van den Berg, and S. B. Engelsen. Review of the most common pre-processing techniques for near-infrared spectra. *TrAC Trends in Analytical Chemistry*, 28(10):1201–1222, 2009. 1, 3, 4, 5, 8

[40] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964. 4

[41] Nicol N Schraudolph. Centering neural network gradient factors. In *Neural Networks: Tricks of the Trade*, pages 207–226. Springer, 2002. 5

[42] K. M. Sørensen, F. van den Berg, and S. B. Engelsen. Nir data exploration and regression by chemometrics—a primer. *Near-Infrared Spectroscopy: Theory, Spectral Analysis, Instrumentation, and Applications*, pages 127–189, 2021. 1, 4, 8

[43] Specim. Specim fx17, https://www.specim.fi/products/specim-fx17/. 2

[44] P. J. Trainor, A. P. DeFilippis, and S. N. Rai. Evaluation of classifier performance for multiclass phenotype discrimination in untargeted metabolomics. *Metabolites*, 7(2):30, 2017. 4

[45] P. Vithu and J. A. Moses. Machine vision system for food grain quality evaluation: A review. *Trends in Food Science & Technology*, 56:13–20, 2016. 1

[46] C. Wang, B. Liu, L. Liu, Y. Zhu, J. Hou, P. Liu, and X. Li. A review of deep learning used in the hyperspectral image analysis for agriculture. *Artificial Intelligence Review*, 54(7):5205–5253, 2021. 1

[47] B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962. 5

[48] S. Wiesler and H. Ney. A convergence analysis of log-linear training. *Advances in Neural Information Processing Systems*, 24, 2011. 5

[49] H. Wold. Estimation of principal components and related models by iterative least squares. *Multivariate analysis*, pages 391–420, 1966. 1, 3

[50] S. Wold, C. Albano, W. J. Dunn, K. Esbensen, S. Hellberg, E. Johansson, M. Sjöström, H. Martens, and J. Russwurm. Food research and data analysis. *London: H. Martens and H. Russwurn Jr*, 1983. 3

[51] S. Wold, M. Sjöström, and L. Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130, 2001. 3, 5