CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Automatic detection of Flavescense Dorée grapevine disease in hyperspectral images using machine learning

Diogo M. Silva[a], Théo Bernardin[c], Kévin Fanton[c], Roshan Nepaul[c], Luís Pádua[a], Joaquim J. Sousa[a,b,*], António Cunha[a,b]

[a]*Universidade de Trás-os-Montes e Alto Douro, Quinta de Prado, 5000-811 Vila Real, Portugal*
[b]*Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência,Porto 4200-465, Portugal*
[c]*Polytech Annecy-Chambéry, 2 Av. du Lac d'Annecy, 73370 Le Bourget-du-Lac, France*

Abstract

The technological revolution that we have been witnessing recently has allowed components miniaturization and made electronic components accessible. Hyperspectral sensors benefited from these advances and could be mounted on unmanned aerial vehicles, which was unthinkable until recently. This fact significantly increased the applications of hyperspectral data, namely in agriculture, especially in the detection of diseases at an early stage. The vineyard is one of the agricultural sectors that has the most to gain from the use of this type of data, both by the economic value and by the number of diseases the plants are exposed to. The Flavescense dorée is a disease that attacks vineyards and may conduct to a significant loss. Nowadays, the detection of this disease is based on the visual identification of symptoms performed by experts who cover the entire area. However, this work remains tedious and relies only on the human eye, which is a problem since sometimes healthy plants are torn out, while diseased ones are left. If the experts think they have found symptoms, they take samples to send to the laboratory for further analysis. If the test is positive, then the whole vine is uprooted, to limit the spread of the disease. In this context, the use of hyperspectral data will allow the development of new disease detection methods. However, it will be necessary to reduce the volume of data used to make them usable by conventional resources. Fortunately, the advent of machine learning techniques empowered the development of systems that allow better decisions to be made, and consequently save time and money. In this article, a machine learning approach, which is based on an Autoencoder to automatically detect wine disease, is proposed.

* Corresponding author.
E-mail address: jjsousa@utad.pt.

1. Introduction

Traditional techniques to detect diseases in plants consist of in-field visual observation by an agronomist or laboratory tests [1]. Even though this technique can provide good results, it is only suitable for small areas because it is very time consuming and demanding. Moreover, subjectivity of visually identifying phytopathological problems can lead to misclassification and wrong conclusions even by experienced experts [2]. Therefore, having a crop consistently monitored can be very important to detect the first signs of a disease. However, to make it possible to detect the anomalous parts in images in early stages, the scanning sensor should allow the acquisition of image/data with appropriated characteristics, being spectral and spatial resolution among the most relevant.

Hyperspectral data stores reflectance values over a wide range of the electromagnetic spectrum, encompassing hundreds of narrow bands (some sensors can capture even thousands). This level of detail provides data that remains unseen for the human eyes and thus potentially produce more relevant results when regarding the detection of anomalous parts representing potential risk to plants development.

The Flavescense dorée (FD) grapevine disease is a bacterial disease and is caused by the phytoplasma. FD is one of the most important and best studied disease in Portugal and other European vine producing countries, causing a negative economic impact considered even as a major challenge for viticulture [3]. One of the difficulties in detecting the disease is that symptoms do not always appear every year and may only be present on one shoot or on a small number of shoots and different vine varieties are not equally susceptible to FD and may not present symptoms with equal intensity. Traditionally, field surveys are performed by experts or by people with experience in detecting plant disorders. However, the experts rely on the symptoms that exhibit in visible part, which frequently indicate middle to late stage of the disease development [4].

Therefore, hyperspectral imaging can be an excellent option to detect this disease (and others) at an early stage, enabling the mitigation of its economic impacts. However, the high number of spectral bands makes their handling very demanding from the point of view of computational resources. Considering that adjacent bands represent similar information, it makes sense to think about reducing the number of bands, thus reducing the computational effort required. The reduction in the number of bands will have to be carried out, ensuring the permanence of the bands whose information is relevant to determine contaminated plants. This task seems possible using artificial intelligence algorithms, namely machine learning.

This paper explores the use of Autoencoders (AEs) to reduce the dimensionality of hyperspectral images and evaluate the use of the obtained trained encoder as a relevant feature extractor for vine diseases detection.

2. Methodology

The methodology used is summarised in the pipeline represented in Fig. 1.

Collected raw data is pre-processed and organized in train, validation and test set. Two approaches were implemented to evaluate the effectiveness of the use of an AE: patched approach and full image approach. For both approaches, the AEs are trained, and their encoders are used as features extractors in the classifiers used for vine diseases detection. Finally, the models are evaluated, and both approaches compared.
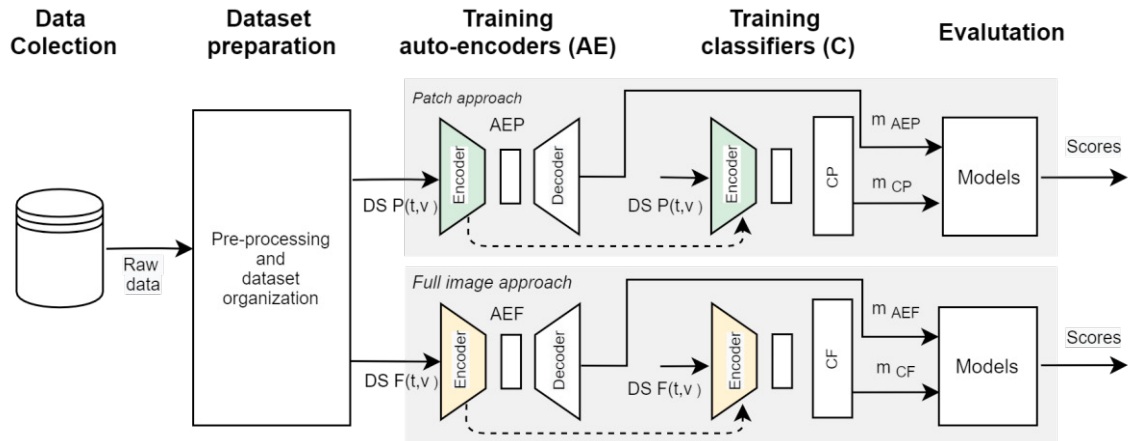
Fig. 1. Methodology pipeline. Legend: DS P(t,v) – Images patch training and validation dataset; DS F(t,v) – Full image training and validation datasets; AEP – AE patched images; AEF – AE full images; CP – Patch classifier; CF – Full image classifier; m_{AEP} – Patch trained AE model; m_{AEF} – Full image trained AE model; m_{CP} – Patch image trained C model; m_{CF} – Full image trained C model.

2.1. Data collection

The dataset used consists of 35 hyperspectral leaves images from the Vinhão wine grape. Every leaf was submitted to chemical analysis, and Flavescence dorée was identified in 10 of them. All the images are high resolution, 640x704, and each of them contains 272 different bands.

2.2. Dataset preparation

First, handmade binary masks of leaves were done to segment all leaf images. It was done using a bitwise "AND" operator across all channels of an image.

Then, two datasets were prepared for models training and evaluation, the DS_F for the full image approach and the DS_P for the patched approach. In the DS_F, full images were used, and in the DS_P, images were divided into 64x64 patches, with 20% overlap, as if applying a convolution.

For the full image approach AE, the images were then split in 80% to train and evaluate the AE and 20% to test the reconstruction capability, using the Leave-One-Out Cross-Validation (LOOCV) strategy. Due to the lack of computational power, we decided to reduce the resolution of all images by 80% (from 704x640 to 140x128), as well as reducing the number of bands from 272 to 64 equally spaced bands. It is assumed that sequential bands contain similar information, and through our approach, it will still be possible to choose only the most representative of these 64. With better GPUs, it would be possible to scale this approach to the original 272 bands.

For the patched approach, however, the data was split into six stratified folds: five folds for performing cross-validation; and one final holdout fold for testing. This ensures that patches from the same image don't end up in different folds, contaminating the samples. Since every image contains excessive black borders, which would produce patches with no information (all black), every image is pre-processed according to the following procedure:
- An image copy is made;
- The copy is converted to grayscale;
- The copy is binarized: the background is black, and the leaf area is white;
- The image contours are found, keeping the biggest contour found;
- A bounding box is computed around the contour;
- The coordinates of the bounding box are used to perform auto-crop of excessive black borders on the original image.

The use of this pre-processing technique avoids unnecessary costly convolutions to extract patches with no leaf part (all black).

2.3. Autoencoders training

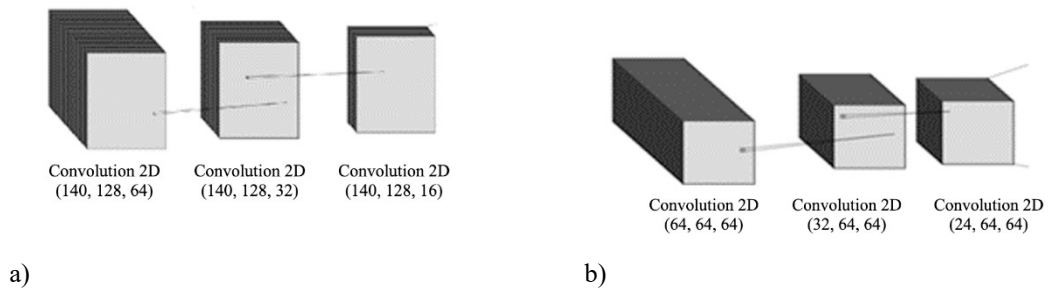The encoders architecture are described in Fig. 2 a) and b).

Fig. 2. Encoders architecture: a) Full image approach, b) Patched approach.

For the full image approach, the encoder sequentially compresses the number of bands until the image has only 16 channels (25% of the original channels). This is called a latent space representation of the original image. At each convolution the encoder is forced to compress only the meaningful channels, discarding the irrelevant ones. The decoder performs transposed convolutions [5] to gradually transform the image back to the original shape.

To optimize and converge the AE, an optimal learning rate must be used, as it is widely known that it's the most important hyper-parameter to tune in a neural network [6]. The "learning rate finder" method, introduced by Smith [6], is implemented to find the optimal interval that the learning rate can take. This method suggests starting with a very low learning rate (10e-8), and at each mini-batch, this learning rate is multiplied by a certain factor, until it reaches a very high value (10 for instance), or the loss explodes. At each iteration, the loss is saved, so when this method ends, the loss against each learning rate can be plotted. This plot is illustrated in Fig. 3 a).
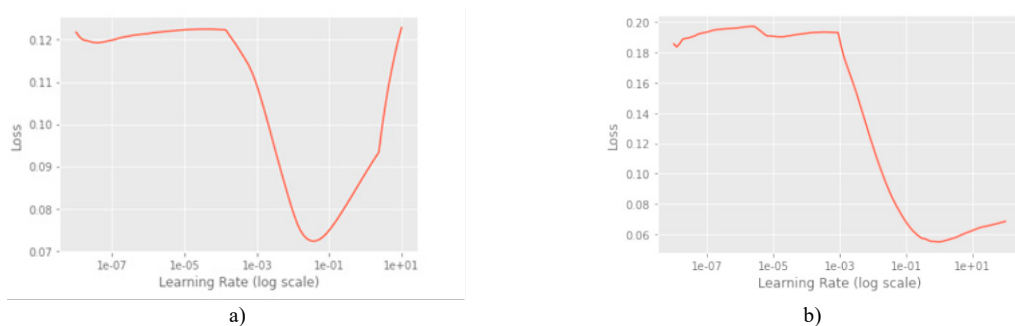

Fig. 3. Loss plot of cyclical learning rate finder method a) Full image approach, b) Patched approach.

The learning rate that produces the minimum loss should be chosen, with this information, we can efficiently train the AE. From analyzing the plot Fig. 3 a), a good starting learning rate can be decided: would be 1e-2 for the higher bound, and for the lower bound 1e-3. After the AE is trained and successfully converging, the encoder part is extracted so that the latent space representation is feed to a fully-connected neural network. This fully-connected consist of two layers, each one with 256 neurons and ReLU activation, followed by a Dropout [7] layer (with a dropout rate of 0.5), preventing the network from overfitting. Both encoders are trained using the Adam optimizer, for a maximum number of 1000 epochs. This huge number of epochs is because we apply early stopping. This means that if the loss on validation data (20% of training data) is not improving after 20 epochs, we stop the training and restore the weights from the best epoch. Moreover, to achieve a better convergence, if the loss on validation data is stagnated after 15 epochs, we reduce the learning rate by a factor of 0.2.

For the patched approach, the encoder architecture is described in Fig. 2 b). We use the same method to find an optimal learning rate for the AE, and the plot is illustrated in Fig. 3 b).

2.4. Classifier training

For each approach, a baseline model and an encoder + classifier were established.

Full image approach

The baseline model architecture can be summarized as follows:
- 2D Convolutional Layer (32 filters) with LeakyReLU activation, followed by Max Pooling 2D;
- 2D Convolutional Layer (64 filters) with LeakyReLU activation, followed by Max Pooling 2D;
- 2D Convolutional Layer (64 filters) with LeakyReLU activation, followed by Max Pooling 2D;
- Flatten Layer;
- Dense Layer (64 neurons) with LeakyReLU activation, followed by Dropout Layer (0.5);
- Dense Layer (64 neurons) with LeakyReLU activation, followed by Dropout Layer (0.5);
- Dense Layer (1 neuron) with Sigmoid activation.

The encoder + classifier was composed by the encoder of Fig. 2 a), feeding its latent space representation to the following fully-connected layers:
- Flatten Layer;
- Dense Layer (128 neurons) with LeakyReLU activation, followed by Dropout Layer (0.5);
- Dense Layer (128 neurons) with LeakyReLU activation, followed by Dropout Layer (0.5);
- Dense Layer (1 neuron) with Sigmoid activation.

They were trained using the LOOCV strategy. This allows us to have a better estimate of how the model performs, on unseen data, in cases where the available data is scarce. With this strategy, each sample is used once as test data (singleton), while the remaining samples are used for training.

Patched approach

The baseline model architecture for this approach is very similar to the baseline model of the first approach, but instead of 64 neurons in the fully-connected layers, this model presents 128 neurons. It was found that a greater number of neurons for this approach yielded better results.

The encoder + classifier of this approach is essentially the same as the previous one. The only difference is the encoder used, in this case, the one in Fig. 2 b). The extracted latent representation feeds the same fully-connected layers of the full image approach.

2.5. Evaluation

To evaluate the full image approach models, we use standard classification metrics: accuracy; precision: recall: area under the curve (AUC); F1-score; and confusion matrix.

In the case of the patched approach, it was used a cross-validation strategy, and for the metrics, we chose to use accuracy and AUC.

3. Results

3.1. Full image approach

The baseline model was trained. At each fold, the predicted label is saved in an array, as well as the expected label. These two arrays are then used to plot a confusion matrix showed in Fig. 4 a), and the computed metrics that are

displayed in Table 1. As noticeable, the performance of the baseline is poor. The recall is not much better than a no-skill classifier. This means that this model is struggling to identify the positive samples.

Table 1. Results of full image approach models.

	Accuracy	**Precision**	**Recall**	**AUC**	**F1-Score**
Baseline model	0.74	0.55	0.6	0.75	0.57
Encoder+classifier	0.83	0.67	0.8	0.83	0.7

Fig. 4. Full image approach: a) Confusion matrix of baseline CNN model, b) Confusion matrix of the Encoder + classifier.

After training the AE, seven images were used to test the reconstruction capability of this model, reaching 2.12e-04 as loss, which is very low, meaning that the AE can compress and decompress the number of image bands very well. These reconstructions images are presented in Fig. 5.

The Encoder + classifier was evaluated using the same cross-validation strategy used in the baseline model. The new confusion matrix is displayed in Fig. 4 b), and the metrics results are also displayed in Table 1.
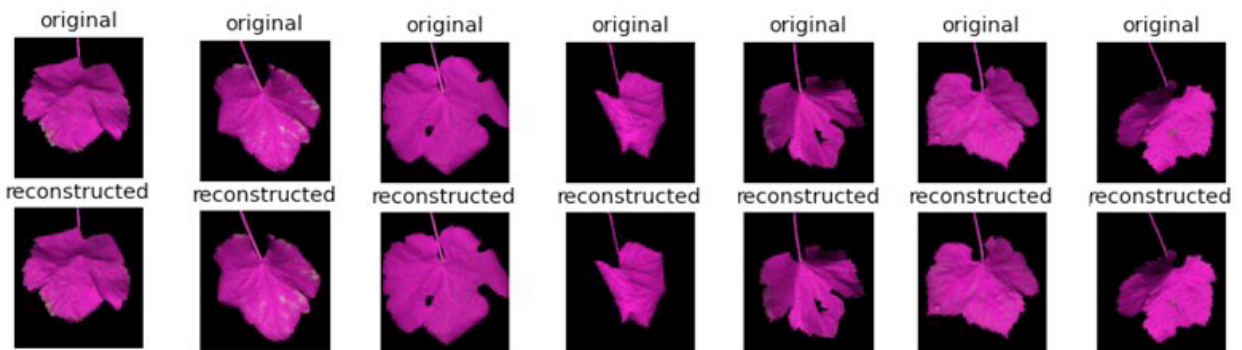

Fig. 5. Full image approach: Autoencoder reconstructions. Three random bands were used to display the reconstructions.

When comparing the AE approach to the baseline model, we can observe that the recall of the positive class has improved, as well as all the other metrics. However, since the amount of data is low, these results are not statistically significant. Hence, the patched approach.

3.2. Patched approach

The baseline model was trained. At each fold, the train and validation loss was saved. These histories were then used to plot the loss curves (Fig. 6).

It can be observed that the baseline model is not overfitting, however, the metric results are not very great. Table 2 shows the cross-validation metrics (with standard deviation) and the holdout (unseen) data results.
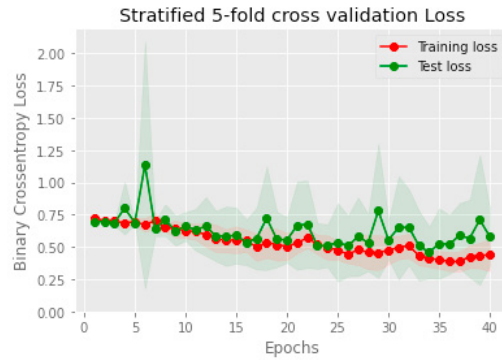
Fig. 6. Patched approach: Cross-validation loss histories.

Table 2. Result of patched approach models, cross-validation and holdout fold.

	Cross-validation		Holdout Fold	
	Accuracy	**AUC***	**Accuracy**	**AUC**
Baseline model	0.70 (+/- 0.16)	0.66 (+/- 0.24)	0.63	0.58
Encoder + classifier	0.70 (+/- 0.9)	0.86 (+/- 0.12)	0.83	0.92

* average (standard deviation)

The combination, encoder + classifier, is also displayed in Table 2. The metrics significantly improved. The model is now able to consistently distinguish an infected sample from a normal one, as noticeable by the significant increase in the AUC score.

After training the AE, 110 image patches are used to test the AE reconstruction capability of this model. The obtained loss was 4.71e-04 (Fig. 6). Fig. 7 shows the reconstruction of ten random patches.
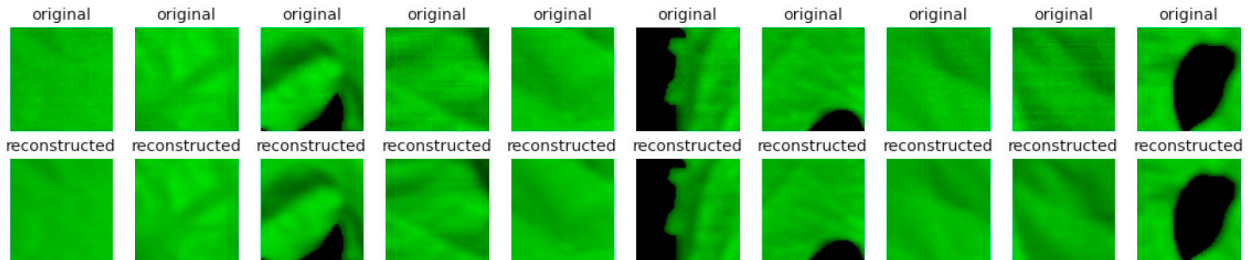

Fig. 7. Patched approach: AE reconstructions.

4. Conclusion

Both studied approaches, especially the patched approach, show that there is a great potential in applying AE for hyperspectral imaging feature extraction, fighting the problem of high dimensionality. Furthermore, this paper shows that it is possible to perform automatic detection of grapevine diseases. In short, for the patched approach, when predicting the presence of FD on test samples, the use of an encoder for feature extraction allowed a 0.2 accuracy increase, and a 0.34 increase in AUC score, together with a significant reduction of the computational burden, compared to a standard CNN model. This could potentially help agriculture, increasing the production, and therefore, the profits.

As future work the following approaches are considered :
- Full image approach: Use of square images. This would allow for strided convolutions;

- Patched approach: Use of strided convolutions;
- Patched approach: Use of majority voting, for deciding the label of a sample, on different patches of the same image;
- Experiment with full-scale data: using all the 272 bands, and high-resolution images.

Acknowledgements

References

[1] Riley, M. B., Williamson, M. R., & Maloy, O. (2002). American Phytopathological Society. American Phytopathological Society. https://www.apsnet.org/pages/default.aspx.

[2] Akila, M., & Deepan, P. (2018). Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm. International Journal of Engineering Research, 6(07), 5.

[3] Chuche, J., & Thiéry, D. (2014). Biology and ecology of the Flavescence dorée vector Scaphoideus titanus: A review. Agronomy for Sustainable Development, 34(2), 381–403. https://doi.org/10.1007/s13593-014-0208-7.

[4] Lowe, A., Harrison, N., & French, A. P. (2017b). Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress. Plant Methods, 13. https://doi.org/10.1186/s13007-017-0233-z.

[5] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. 2016.

[6] Leslie N. Smith. Cyclical learning rates for training neural networks. In Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, pages 464–472. Institute of Electrical and Electronics Engineers Inc., jun 2017.

[7] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.