Automatic detection of grapevine diseases in hyperspectral images

1st Diogo Silva

I. Introduction

Flavescence dorée is a grapevine disease that affects European vineyards [1], with vast economical consequences. Traditionally this disease is detected by performing chemical analysis in a laboratory, being very costly and time-demanding. Hyperspectral imaging (HSI) has a great number of applications, one of them being agriculture. This type of image collects information from across the electromagnetic spectrum, invisible to the human eye. However, it also comes with the curse of high dimensionality, containing a lot of redundant information. Thus, the objective of this paper is to study the use of AutoEncoders (AEs) to reduce the dimensionality of this type of images, more specifically the number of bands. Moreover, this work aims to evaluate the use of a trained encoder as a relevant feature extractor.

II. Methodology

This section describes the dataset and methods used for this work.

A. Data Collection

The dataset used consists of 35 hyperspectral leaf images from the Vinhão wine grape. Every leaf was submitted to chemical analysis, and the presence of Flavescence dorée was identified in 10 of them. All the images are fairly high resolution, 640x704, and each of them contains 272 different bands. In order to reduce the computational burden needed to process this huge ammount of information, all the images suffered a size compression of 80 % in width and height, and only 64 equally spaced bands were studied.

B. Methods

To evaluate the effectiveness of the use of an AE, as a meaningful band extractor, two approaches were taken:

1) Full-Image Approach: The full images were used to train an AE and a classifier.
2) Patched Approach: Every image was divided into patches of size 64x64, with 20% overlap, as if applying a convolution.

Figure 1 provides a global and brief overview of the developed work pipeline. For each approach, a baseline model was established, to compare and evaluate how these approaches match up to more standard CNNs.

1) Full-image Approach: In this first approach, the images are used as-is. First, the images are segmented with previously handmade binary masks. This segmentation is done using a bitwise "AND" operator across all channels of an image. Before attempting any novel approach, a baseline CNN was trained. It's 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.

To train this baseline model and later the fully-connected encoder a LOOCV (Leave-One-Out Cross Validation) strategy was used. 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.

Then to train an AE, the images were then split in a stratified fashion, keeping 80% to train and evaluate the AE and 20% to test the reconstruction capability. The encoder architecture is described in Figure 2. This 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 [2] 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 [3]. The "learning rate finder" method, introduced by Smith [3], 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, we save the loss, so when this method ends we can plot the loss against each learning rate. This plot is illustrated

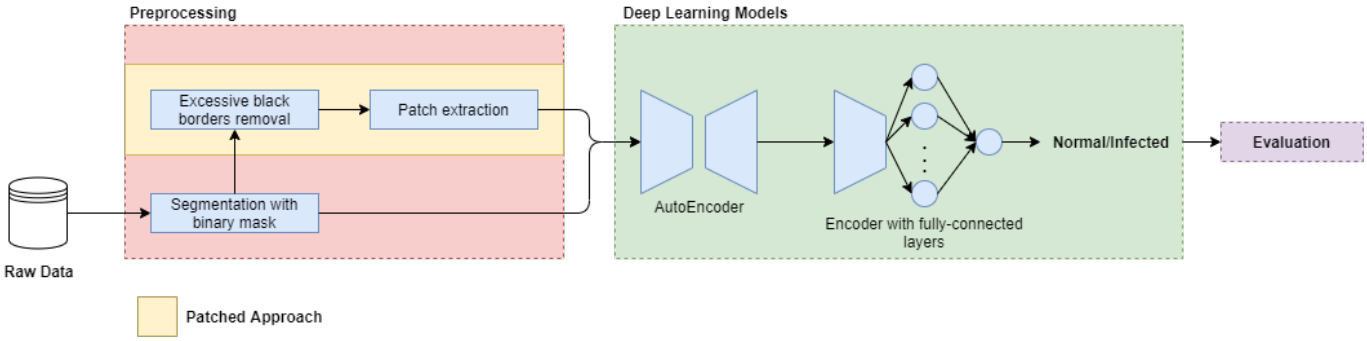

Fig. 1. Global pipeline overview.

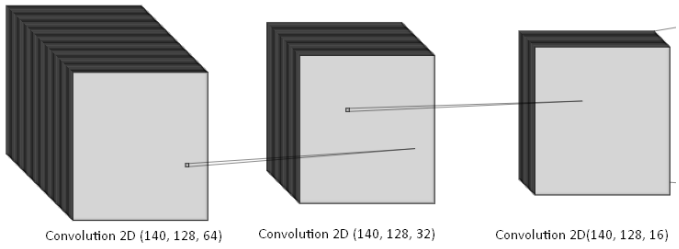

Fig. 2. Full image Approach: Encoder architecture.

in Figure 3. From analyzing this graph we can decide a good starting learning rate would be 1e-2 (higher bound), and a lower bound would be 1e-3. Here are the reasons we should choose the learning rate that produces the minimum loss: At the minimum loss the learning rate is already too big, it's at the edge of exploding. With this information, we can train the

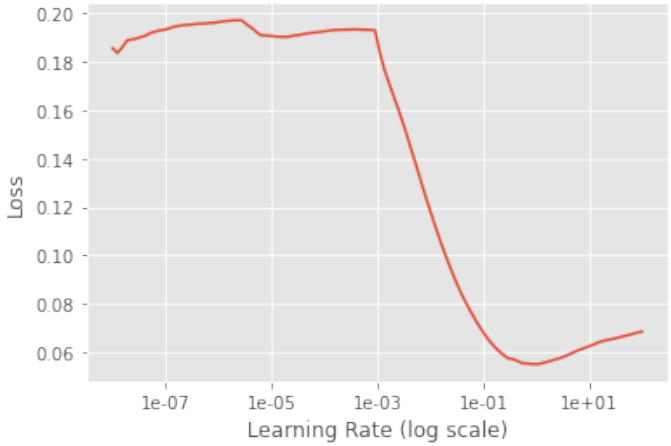

Fig. 3. Full-Image Approach: Loss plot of cyclical learning rate finder method.

AE starting with a learning rate of 1e-2 and reducing it by a factor of 0.2 when the loss is not improving after 15 epochs (stuck in local minima), until the minimum of 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 part two layers, each one with 256 neurons and ReLU activation, followed by a Dropout [4] layer (with dropout rate of 0.5), preventing the network from over-fitting. Moreover, early stopping is also applied. This means that if 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.

2) Patched Approach: The baseline model architecture for this approach is the 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.

In this approach, given the low amount of available data, every image is split into patches. But before applying any image patching, the data is split into six stratified folds: 5 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 preprocessed using some Digital Image Processing (DIP) techniques:

1) A image copy is made;
2) The copy converted to grayscale;
3) Then the copy is binarized, so that the background is black and the leaf area is white;
4) The image contours are found, keeping the biggest contour found (leaf);
5) A bounding box is computed around the contour;
6) The coordinates of the bounding box are used to perform auto-crop of excessive black borders on the original image.

Using this preprocessing technique avoids unnecessary costly convolutions. This new encoder architecture is described in Figure 4. The method used in the first approach to find an optimal learning rate for the AE is used again (Figure 5). To generate patches every image is convolved with a sliding window, of size 64x64, with 20% overlap between each patch.

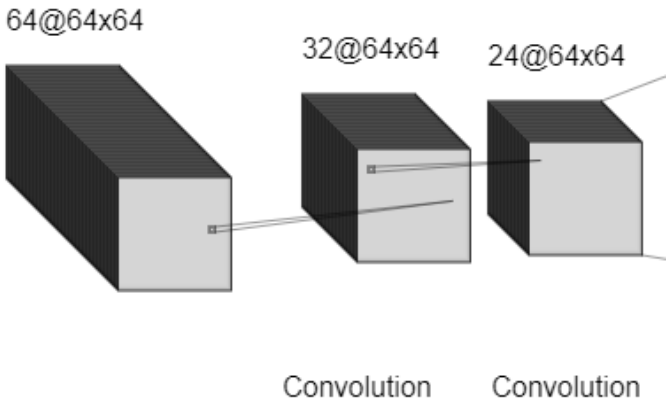

Fig. 4. Patched Approach: Encoder architecture.

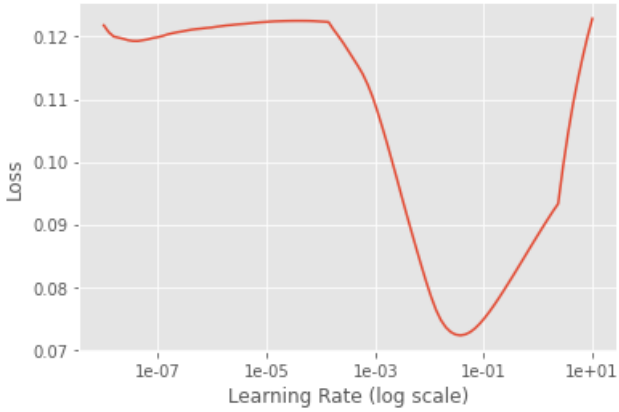

Fig. 5. Patched Approach: Loss plot of cyclical learning rate finder method.

III. Results

A. Full-Image Approach

First, 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 7, and compute the metrics that are displayed in Table I. As noticeable, the performance of this classifier is poor. The recall of the positive class (infected), is not much better than a non-skill classifier. This means that this model is struggling to identify the positive samples.

TABLE I
Full-Image Approach: Metrics of the baseline model.

	AUC	Accuracy	Recall	F1-Score	Precision
Normal	0.75	0.74	0.80	0.82	0.83
Infected			0.6	0.57	0.55

After training the AE, seven images were used to test the reconstruction capability of this model. The presented loss for this test was 2.12e-04, which is very low, meaning that the AE is able to compress and decompress the number of image bands very well. These reconstructions are presented in Figure 6 (3 random bands were used to display the reconstructions).

The encoder part is then extracted and used to feed fully-connected layers. This new network is then evaluated using the same cross-validation strategy used in the baseline model. The new confusion matrix is displayed in Figure 8, and the metrics results in Table II. When comparing the AE approach to the baseline model, we can observe that the recall of the positive class 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.

TABLE II
Full-Image Approach: Metrics of the fully-connected encoder.

	AUC	Accuracy	Recall	F1-Score	Precision
Normal	0.83	0.83	0.84	0.87	0.91
Infected			0.8	0.73	0.67

B. Patched Approach

First, the baseline model was trained. At each folds the train and validation loss were saved. These histories were then used to plot the loss curves (Figure 10).

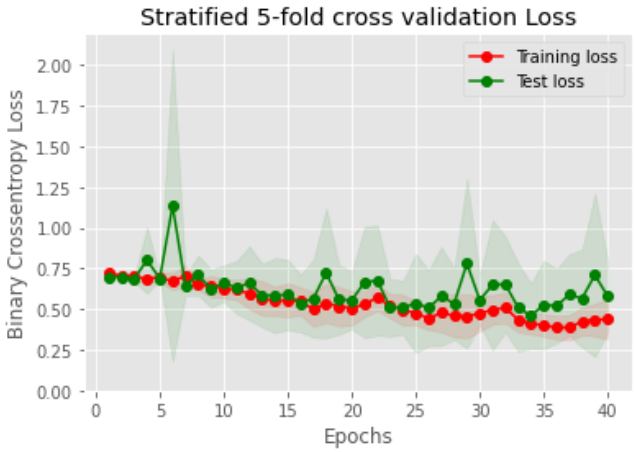

Fig. 10. Patched Approach: Cross-validation loss histories.

We can observe that the baseline model is not overfitting, however the metric results are not very great. Table IV shows the cross-validation metrics (with standard deviation) and the holdout (unseen) data results. After training the new AE,

TABLE III
Patched Approach: Cross-validation and Holdout fold metrics results of the baseline model.

	Accuracy	AUC
Cross-validation	0.70 (+/- 0.16)	0.66 (+/- 0.24)
Holdout fold	0.63	0.58

110 images are used to test the AE reconstruction capability

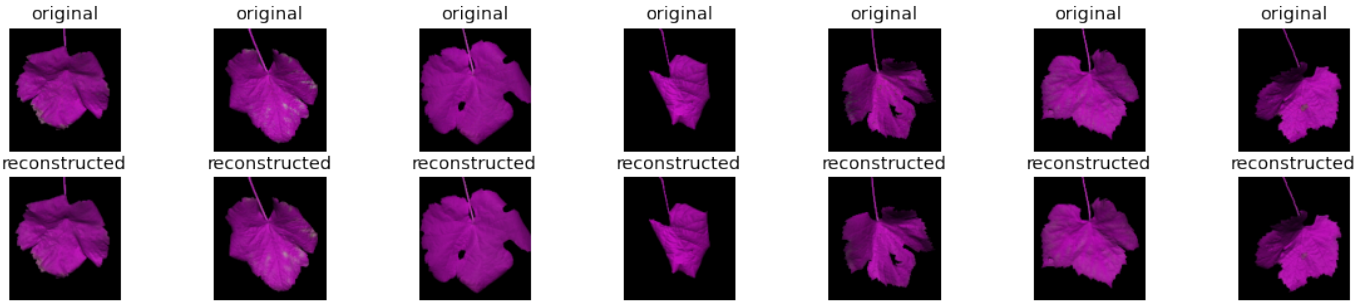

Fig. 6. Full-Image Approach: AE reconstructions.

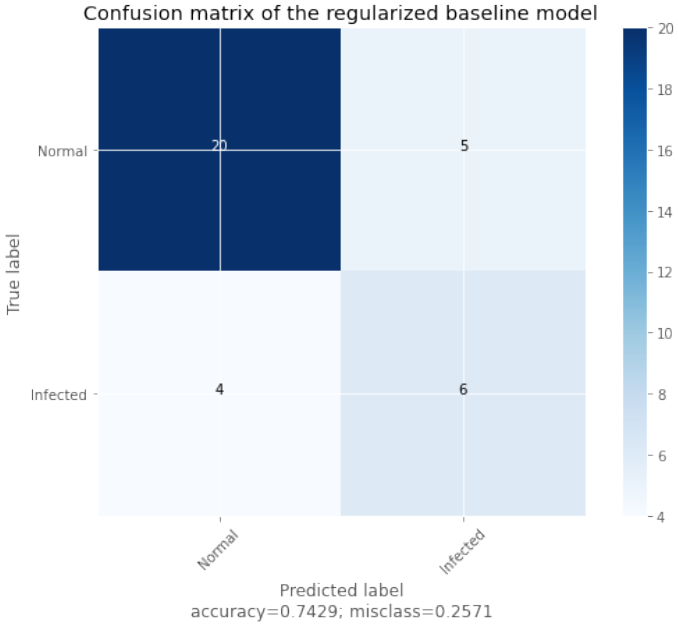

Fig. 7. Full-Image Approach: Confusion matrix of baseline CNN model.

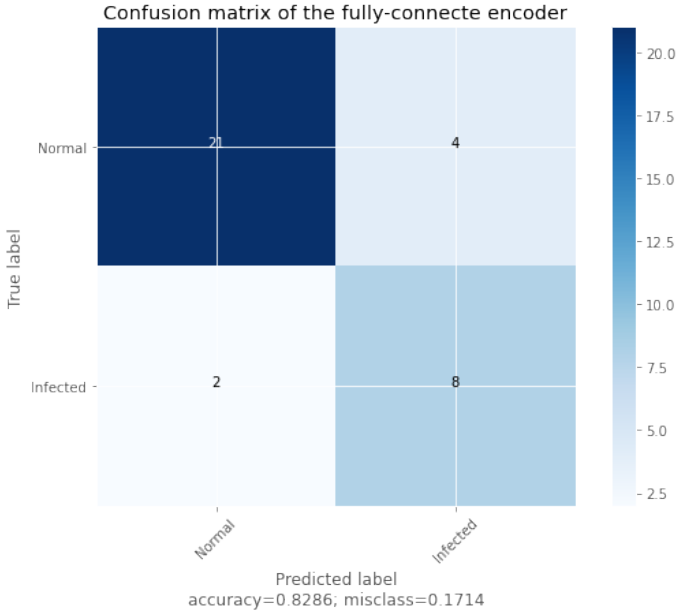

Fig. 8. Full-Image Approach: Confusion matrix of the fully-connected encoder.

of this model. The loss was 4.71e-04. Figure 9 shows the reconstruction of ten random patches.

The encoder of this AE was then used to feed the latent space representation into fully-connected layers. This new network is then evaluated using the same cross-validation strategy used in the baseline model. The new results are displayed in Table IV. Our metrics improved a lot. The model is now able to consistently distinguish a infected sample from a normal one.

TABLE IV
PATCHED APPROACH: CROSS-VALIDATION AND HOLDOUT FOLD METRICS RESULTS OF THE FULLY-CONNECTED ENCODER.

	Accuracy	AUC
Cross-validation	0.70 (+/- 0.9)	0.86 (+/- 0.12)
Holdout fold	0.83	0.92

IV. CONCLUSION

Both studied approaches, specially the patched approach, show that there is a great potential in applying AE for HSI feature extraction, fighting the problem of high dimensionality. Furthermore, this paper shows that it is possible to perform automatic detection of grapevine diseases. This could potentially help agriculture, increasing the production, and therefore, the profits.

As future work it would be of great interest to try the following:

- Full-Image Approach: Use of squared 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.

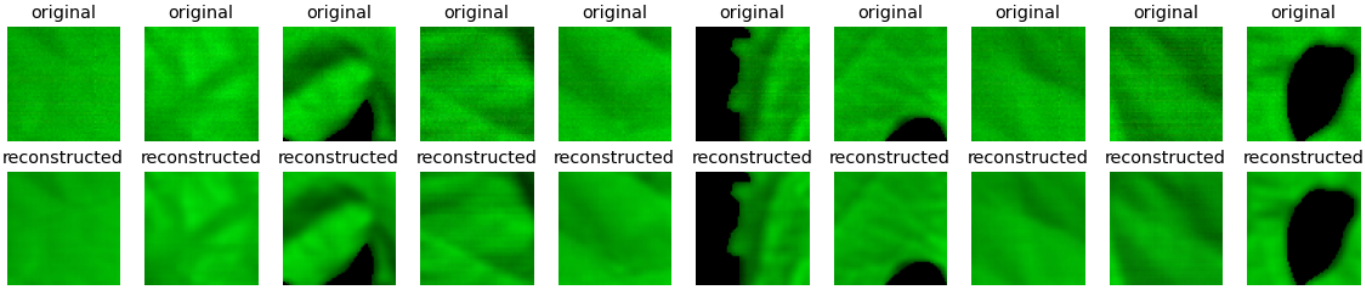

Fig. 9. Patched Approach: AE reconstructions.

REFERENCES

[1] Julien Chuche and Denis Thiéry. Biology and ecology of the Flavescence dorée vector Scaphoideus titanus: A review, feb 2014.

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

[3] 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.

[4] 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.