Journal of Mathematics in Industry
a SpringerOpen Journal

RESEARCH Open Access

# Unsupervised deep learning techniques for automatic detection of plant diseases: reducing the need of manual labelling of plant images

Alessandro Benfenati[1], Paola Causin[2*], Roberto Oberti[3] and Giovanni Stefanello[2]

*Correspondence:
paola.causin@unimi.it
[2]Dept. of Mathematics, University of Milan, Milan, Italy
Full list of author information is available at the end of the article

**Abstract**

Crop protection from diseases through applications of plant protection products is crucial to secure worldwide food production. Nevertheless, sustainable management of plant diseases is an open challenge with a major role in the economic and environmental impact of agricultural activities. A primary contribution is expected to come from precision crop protection approaches, with treatments tailored to spatial and time-specific needs of the crop, in contrast to the current practice of applying treatments uniformly to fields. In view of this, image-based automatic detection of early disease symptoms is considered a key enabling technology for high throughput scouting of the crop, in order to timely target the treatments on emerging infection spots. Thanks to the unprecedented performance in image-recognition problems, Deep Learning (DL) methods based on Convolutional Neural Networks (CNNs) have recently entered the domain of plant disease detection. This work develops two DL approaches for automatic recognition of powdery mildew disease on cucumber leaves, with a specific focus on exploring unsupervised techniques to overcome the need of large training set of manually labelled images. To this aim, autoencoder networks were implemented for unsupervised detection of disease symptoms through: i) clusterization of features in a compressed space; ii) anomaly detection. The two proposed approaches were applied to multispectral images acquired during in-vivo experiments, and the obtained results were assessed by quantitative indices. The clusterization approach showed only partially capability to provide accurate disease detection, even if it gathered some relevant information. Anomaly detection showed instead to possess a significant potential of discrimination which could be further exploited as a prior step to train more powerful supervised architectures with a very limited number of labelled samples.

**Keywords:** Plant disease detection; Anomaly detection; Unsupervised deep learning; Powdery mildew; Multispectral imaging; Precision agriculture

## 1 Introduction

Plant disease management plays a pivotal role in securing high quality and abundant yield of agricultural products. Disease control is mainly obtained by plant protection products, commonly referred to as pesticides, that are repeatedly applied uniformly across entire fields. However, it is well established that most crop diseases exhibit an uneven spatio-temporal distribution, with randomly sparse patchy structures evolving around discrete foci (*i.e.* localized initial infection spots), especially at the early stages of development [1]. This pattern offers great opportunities to develop precision crop protection solutions, *i.e.* the application of precision agriculture (PA) concept to protection operations, with spatial and temporal variation of the treatments as needed by the crop [2], leading to great reduction of pesticide use with associated benefits in term of costs and environmental impact [3].

A fundamental requirement for the full implementation of precision crop protection systems is the capability of automatically detect symptoms of disease at early stages in order to timely target the treatments on emerging infection spots, and preventing their establishment and following epidemic expansion. To this aim, different disease sensing methods and technologies have been applied, including molecular analysis, spectroscopy, fluorimetry, analysis of volatile organic compounds [4, 5], but imaging-based approaches emerges as the most widely studied techniques for disease sensing applications [6, 7].

Indeed, computer vision has an inherent great potential since symptoms of crop disease very often cause a signature on plant organs which can be automatically detected by adequate image-analysis techniques. Disease symptoms have been detected and identified by analyzing color or reflectance, texture, shape features specifically extracted from leaf or plant images [8, 9], or by computing spectral vegetation indices (VIs), *i.e.* algebraic combinations of pixel values in two or more spectral channels, which can enhance the features differences between healthy and diseased tissue [10–12]. While these approaches rely on human experts for the selection of the most relevant features to discriminate diseased from healthy plants, recent developments in machine learning disclosed new possibilities toward the automatic identification of relevant features [13]. Among these, Deep Learning (DL) methods based on Convolutional Neural Networks (CNNs) are being applied in almost any image-recognition problem with unprecedentent successful results. In this framework a burst of CNNs applications to imaging based detection and identification of crop diseases has being published in the last few years [14]. For example, the authors in [15] used existing CNNs (GoogleNet and Alexnet) to classify 26 diseases over 14 crop species, using as training data 54,306 labelled RGB color images from the PlantVillage repository [16]. CaffeNet CNN was used in [17] to classify 13 diseases over various plant species with a training set of 4483 (augmented to 30,000) images downloaded from the web and submitted to preliminary manual masking and labelling. Transfer learning techniques were adopted in both [15] and [17] to specialize CNNs to the application objectives. The authors in [18] considered 5000 images of tomato leaves with manually annotated bounding boxes containing disease spots to train different architectures of Region Proposal Networks to detect diseased leaves. They also explored the use of "very deep feature extractors" such as the Visual Geometry Group (VGG) net and the Residual Network in order to obtain accurate disease classification under diverse field conditions. The authors in [19] used a Res-Net architecture on 16,415 images of diseased tomato leaf images and 1590 healthy tomato leaves from the PlantVillage dataset, upon classification of nine types of disease by human experts. The proposed algorithm automatically classified leaves into healthy or diseased and a U-net architecture was applied to semantically segment a subset of the images to recognise the disease and to estimate per-leaf its severity. Semantic segmentation was performed as well in [20] to recognize powdery mildew spots on cucumber leaves, using a U-net architecture trained with 30 annotated samples (augmented to 10,000).

A common trait of all the previously mentioned approaches is that substantial efforts have to be spent by human experts to label/annotate the images in order to prepare the typically large amount of samples required to train the chosen CNN model. Even when transfer learning techniques are used by taking advantage of already available labelled data, further labelling targeted to the specific application case is however required. This work aims to leverage the capabilities of DL methods to define unsupervised techniques for achieving a preliminary but fairly accurate automatic detection of plant diseases, including early stage symptoms. Two unsupervised approaches for disease detection are investigated, namely: a clustering approach which is intended to be a very basic prototype of this kind of conventional Machine Learning approaches and, as a preferred and more effectively choice, an unsupervised automatized feature extraction approach based on anomaly detection. As case study, we considered powdery mildew disease on cucumber (*Cucumis sativus*). Powdery mildew is a major fungal disease that mostly affects leaves in many crop plants (vegetables, fruits, cereals, etc) exhibiting common symptoms: the proliferation of hyphae filaments of the mycelium on the hosting tissue influences leaf reflectance to incident light, leading to a whitish–gray, powdery appearance. At early to middle stages of infection, these thin filamentous structures have still a low influence on the spectral signature of the leaf surface due to their small size, low density, and spatial arrangement, making the early detectability of powdery mildew a non–trivial problem. The study considered multispectral images of leaves in order to exploit the altered spectral signature of disease symptoms not only in the visible (RGB) spectrum but also in the near-infrared (NIR) band [7, 12, 21, 22].

In the following sections, after a brief recall on the relevance of leaf reflectance for disease detection purposes and the description of the measurement procedures, an in depth presentation of the unsupervised approaches of CNN clustering and anomaly detection is given, along with details on the networks architectures implemented. Finally the obtained results are discussed to highlight the most significant findings of this work.

## 2 Data acquisition

### 2.1 Biophysical background

Leaf reflectance features have a high potential in detecting deviations from the healthy status of plants linked to dysfunction of the photo-system or destruction of the photochemical pigments, modifications in plant tissue composition and structure, or to the development of pathogen spores or propagules on the leaf surface. These biophysical modifications induce significant changes in the spectral signature of plant tissue that can be detected with adequate techniques [5, 23]. Changes in the visible (VIS, 400 to 700 nm) and near-infrared (NIR, 700 to 1100 nm) spectral ranges are of particular relevance since they can be measured with common silicon-based sensors or cameras. In these bands, healthy leaves typically exhibit (a) low reflectance at VIS wavelengths owing to strong absorption by pigments; (b) high reflectance in the NIR owing to internal scattering in the leaf structure, except for weak water absorption at specific wavebands. General disease symptoms correspond to discrete structures or lesions on leaf tissue evolving from millimeter-scale size to macroscopic patches, and are characterised by an increased reflectance in VIS range, especially in the chlorophyll absorption bands in the blue (430-470 nm) and red (630-690 nm) bands. Conversely, at more advanced stages of disease, reflectance in NIR range on symptomatic areas is reduced by oxidation and senescence processes in the tissue, and at plant canopy scale by decreased biomass growth, defoliation and drying. These general features hold for the specific case of powdery mildew and upon this it relies the rationale of using multi-spectral imaging in the above indicated bands to detect regions in leaf surface exhibiting deviations from healthy spectral signatures.

### 2.2 Plant material and disease inoculation

Plants of cucumber (*Cucumis sativus*) were sown and grown in pots under controlled conditions in greenhouse at 25/22°C (day/night), 60% relative humidity. Plants were regularly watered and fertilized as needed, and no pesticide treatment was applied. At a development stage of 3 leaves, a group of plants was separately inoculated with isolates of the fungus *Podosphaera xanthii* by spraying a suspension of freshly sporulating colonies onto leaves. The rest of the plants were kept isolated under controlled conditions in order to maintain healthy conditions during the growth. Multiple lots of plants were subsequently cultivated and inoculated to provide enough samples to the aim of the experiment.

### 2.3 Multispectral images acquisition and preprocessing

In order to obtain a wide range of severity in powdery mildew symptoms, the inoculated plants were sampled at different dates, *i.e.* after 5, 10, 15 days from the inoculation, and imaged together with age-companion healthy plants. Healthy and diseased cucumber leaves were imaged via a QSi640 ws-Multispectral camera (Atik Cameras, UK) equipped with a Kodak 4.2 Mp micro-lens image sensor and 8 passband spectral filters operating at wavebands from 430 to 740 nm. For the purpose of this experiment, leaves were imaged singularly on a dark background, under controlled diffuse illumination conditions (see Fig. 1). Images were acquired in the single spectral channels 430 nm (blue, B), 530 nm (green, G), 685 nm (red, R) and 740 nm (near–infrared, NIR). A set of RGB images of the same leaves in standard CIE color space were also acquired for reference. Camera parameters were set and image collection was performed via an *in–house* developed acquisition software written in MATLAB. Reflectance calibration of the grey-level intensity of the pixels at different acquisitions was carried out by including in each image 3 reflectance references targets (Spectralon R = 0.02, R = 0.50 and R = 0.99; Labsphere, USA). The obtained dataset consisted in 97 images of healthy leaves and 114 images of diseased leaves exhibiting a wide range of symptoms, from early to severe. Starting from an original resolution of $2048 \times 2048$, each image was cropped and resized in order to get the resolution down to $512 \times 512$, which was more manageable for processing purposes. The resized images were preprocessed with min-max normalization in order to obtain pixels values within the interval $[0,1]$. By using the NIR channel, where high reflectance of leaf tissue allows immediate discrimination from the background, binary masks indicating foreground pixels belonging to leaves were computed.

## 3 Anomaly detection: clustering and deep learning techniques

Anomaly detection, also known as novelty detection, is the process of detecting data instances that do not conform to a model of "normal" behavior. Feature extraction is closely related to dimensionality reduction: this technique consists in the transformation of data from a high-dimensional space into a low-dimensional space, so that the low-dimensional

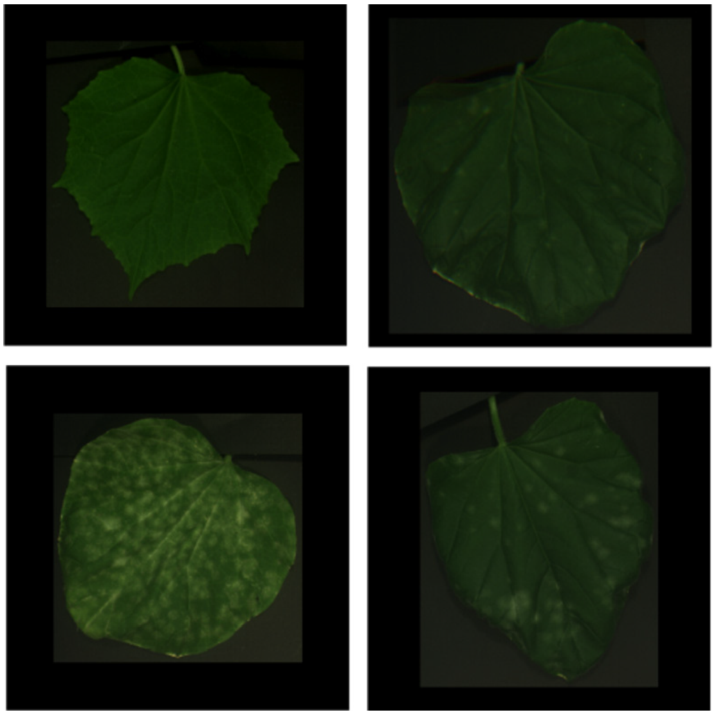

**Figure 1** RGB images of four samples of cucumber leaves from the acquired dataset. The leaves show different symptoms of powdery mildew, from very mild (top left) to severe, in clockwise order

representation is an informative encoding of the original data. In the following, we first provide a brief background on autoencoders and then we discuss more in detail the two considered approaches.

### 3.1 Background on autoencoders

Autoencoders (AE) play a fundamental role in unsupervised learning and in deep architectures for transfer learning and other machine learning tasks and are basically a form of compression. As used in this work, an AE is a neural net which has the primary purpose of learning an encoding of reduced dimensionality with respect to the input that can be used for different applications by learning to reconstruct a set of input observations well enough [24, 25]. Typically, an AE of this type is composed by an encoder, which is an operator $\phi_e$ depending on the parameters $\Theta_e$ that maps the input $x$ into the so called hidden representation $z = \phi_e(x; \Theta_e)$, a meaningful representation of the input with a reduced dimensionality attained at the bottleneck of the net, and by a decoder, which is an operator $\phi_d$ depending on the parameters $\Theta_d$ that decodes the hidden representation into an estimate of the input $\widehat{x} = \phi_d(z; \Theta_d)$. Both encoder and decoder operators are composed by series of linear filtering operations (convolutions), optionally followed by non linear activation functions (*e.g.*, sigmoid, hyperbolic tangent, ReLU function) and they can be trained with backpropagation algorithms. This architecture learns the structure of the mappings $\phi_e$ and $\phi_d$ by estimating the set of network parameters that minimize a distance metric, called loss function $\ell$, between the input and its reconstruction

$$\{\Theta_e, \Theta_d\}^* = \arg\min_{\Theta_e,\Theta_d} \sum_{x \in train} \ell\bigl(\phi_d\bigl(\phi_e(x; \Theta_e); \Theta_d\bigr)\bigr), \tag{1}$$

where $x$ belongs to a subset of the collected data called train set. The loss function $\ell$ is chosen in this work as the standard mean square error (MSE). Alternative forms of loss functions can also be used in specific contexts, using metrics different from the $L^2$ one, weighted metrics, or different functionals as for example cross-entropy. The minimization procedure can be carried out through well-known iterative techniques, for example gradient descent methods [26–28]. Our AE formulation then reads:

$$\begin{aligned} &z = \phi_e(x;\Theta_e), \qquad \widehat{x} = \phi_d(z;\Theta_d), \\ &\{\Theta^*, W^*\} = \underset{\Theta,W}{\arg\min} \sum_{x\in train} \big\| x - \phi_d\big(\phi_e(x;\Theta_e);\Theta_d\big)\big\|^2. \end{aligned} \tag{2}$$

## 3.2 Feature extraction and clustering

Cluster analysis is the process of finding "natural" groupings by gathering objects together according to some similarity measure. Clustering is a well known hard task, whose outcomes depend on a number of factors, among which data dimensionality. Since not all the original features are relevant for clustering, preprocessing strategies such as dimensionality reduction allow clustering to perform better. In this work, we use the AE framework to extract a reduced feature set on which to carry out clustering. Clustering is then performed using a classic $k$-means approach where cluster centroids are sought by an iterative optimization process that minimizes the Euclidean distance between data points in the reduced feature space and their nearest centroid. Despite the fact that more sophisticated approaches have been proposed in literature (see, *e.g.*, [29]), in this work we stick to this simple algorithm since our aim is limited to probe the discriminating potential embedded in the features representation space.

## 3.3 Feature extraction and anomaly detection

Anomaly detection, also known as novelty detection, is the process of detecting data instances that deviate from a given set of samples (observations in the train set). Anomaly detection can be carried out via a neural network by training the net on normal samples so to build a feature representation of "normality". The idea that we pursue to use of AEs as anomaly detectors is inspired by [30] (albeit if this work belongs to a very different context) and is the following: an AE tailored to encode and decode a specific kind of data, fails in encoding and decoding correctly other kinds of data, revealing an anomaly. If we feed the AE with a train set only consisting in images of healthy leaves, the net learns to represent not showing any disease trace. If the trained AE is then fed with the image of a leaf presenting disease spots, it will fail in encoding the presented input and the error can be used as anomaly indicator. Namely, an anomaly score can be introduced to quantify the discrepancy of a sample from its reconstruction performed by the net, so that normal (healthy leaves) samples will yield a low anomaly score, while anomalous samples (diseased leaves) will yield a higher anomaly score. We define the score of a sample $x$ (not included in the train set) as

$$s_x = \frac{\|x - \widehat{x}^*\|^2}{\|x\|^2}, \tag{3}$$

where $\widehat{x}^* = \phi_d^*(\phi_e^*(x))$ represents the reconstruction of the input sample $x$ performed by the net with the optimized parameter set $\theta^*$. If the score is larger than a set threshold, then the sample is classified as an anomaly. One can also profitably use the anomaly score based on the discrepancy between the compressed representation $z^* = \phi_e^*(x)$ of the sample and the compressed representation of its reconstruction $\widehat{z}^* = \phi_e^*(\widehat{x}^*)$, given by

$$s_z = \frac{\|z - \widehat{z}^*\|^2}{\|z\|^2}. \tag{4}$$

We found that this latter choice provided enhanced results due to the informative content of the encoded representation. This is a common finding in these architectures, where the latent representation offers a more effective space where to evaluate score metrics or to perform operations like regularization procedures (see, *e.g.*, [31, 32] in different applications).

## 4 Convolutional autoencoder networks architectures and implementation

We describe in this section the convolutional autoencoder architectures considered for the two studied approaches.

### 4.1 Clustering approach

#### *4.1.1 Clu-AE network*

For clustering approach the encoder part of the network was composed of 4 blocks, each made of a convolutional layer, a batch normalization layer and a ReLU activation layer. The blocks are connected through max pooling layers in order to decrease the resolution of the image. The number of features for each block was 8, 16, 32, 64, respectively, going from the shallowest block to the deepest one. At the deepest level, a dropout layer was inserted to both reduce over–fitting and training time. Our experiments also showed that the drop–out layer also positively affects the network to learn diverse, non–redundant features. The decoder part symmetrically mirrored the encoder, except for the last block. In fact it was composed of 3 blocks, with 32, 16, 8 filters, connected through upsampling layers in order to increase the resolution of the image, eventually restoring its original size. After the last decoder block, a convolutional layer with a 1 × 1 kernel and a number of filters equal to the channels of the original image is considered, coupled with a logistic activation function, providing as final output a reconstruction of the input image (see Fig. 2). For brevity, this architecture is hereby referred as Clu-AE.

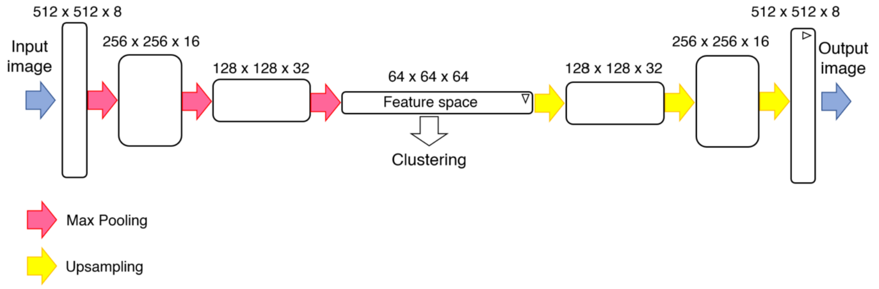

**Figure 2** Structure of the convolutional autoencoder used in the clustering approach (Clu-AE network). Each block is composed of a convolutional layer, a batch normalization layer and a ReLU activation. The block marked with the symbol ∇ includes a dropout layer at its end. The block marked with the symbol ▹ is followed by a 1 × 1 kernel convolutional layer with a number of features equal to the number of channels of the input image

As input for the experiments on Clu-AE network, the spectral index NIR/R obtained as the pointwise ratio between the NIR channel and the R channel (NIR/R) was also included, in addition to regular R, G, B channels. The dataset of images was split in training (90%), validation (5%) and test (5%). The training was performed via Adam optimizer, with learning rate $\eta = 10^{-3}$, default hyperparameters and He initialization of the parameters. The maximum number of epochs to train the model was set to 500 and early stopping was implemented, monitoring the validation loss with a patience parameter of 20 epochs. The batch size was set to 8. After each epoch, the whole training set was shuffled. For clustering of features, we used the `sklearn.cluster.KMeans` module of the `sklearn` library package in Python [33] using k-means++ with 20 initializations. We have also tried random initialization but, in general, this gave us worse results (not reported here). We choose 2, 3 or 4 clusters in the attempts to first separate healthy leaves from diseased ones and then to obtain clusters according to the severity of the disease. We found than using more than 4 clusters did not yield improvements, especially in view of recognizing early stages of the disease.

### 4.2 Anomaly detection approach

For the anomaly detection approach, the autoencoder architectures employed were deeper than for Clu-AE, and leveraged on the use of residual units, in which each layer feeds into the next layer and farther layers to improve over degradation problems in gradient backpropagation. The residual blocks were connected through max pooling layers in order to decrease the resolution of the image, while moving to deeper levels. Also in this case, at the deepest level, a dropout layer was included to reduce overfitting and training time, and the decoder part mirrored the encoder with three residual blocks, connected through upsampling layers.

The last decoder residual block was followed by a final convolutional layer with a 1 × 1 kernel and by a number of filters equals to the channels of the original input image, with a logistic activation function providing the output image. For the aim of this study, different AE architectures with a varying number of filters in each block and/or kernel size were experimented, namely:

- Model S3: 2, 4, 8, 16, 8, 4, 2 filters and 3 × 3 kernels
- Model S5: 2, 4, 8, 16, 8, 4, 2 filters and 5 × 5 kernels
- Model M3: 4, 6, 8, 10, 8, 6, 4 filters and 3 × 3 kernels
- Model M5: 4, 6, 8, 10, 8, 6, 4 filters and 5 × 5 kernels
- Model B3: 32, 64, 128, 256, 128, 64, 32 filters and 3 × 3 kernels.

The latter model B3 resulted to perform better, and in the following we will refer to it as Ano-AE (shown in Fig. 3). The results presented and discussed in next section were obtained by this network, unless differently specified.

As input images in the experiments on this approach, the R,G,B, NIR channels were used. Each considered model was only trained on healthy leaves. The dataset was split in: 70% of images for training, 10% for validation and the remainder for testing. Data augmentation was perfomed on the dataset by translation, rotation, reflection, and zooming. The resulting train dataset consisted of 552 healthy samples. Training was performed with Adam optimizer, learning rate $\eta = 10^{-3}$, default hyperparameters and He initialization. The maximum number of epochs to train the model was set to 500 with early stopping, with a patience parameter of 20 epochs for monitoring validation loss. The batch size was set to 4 and after each epoch, the whole training set was shuffled.

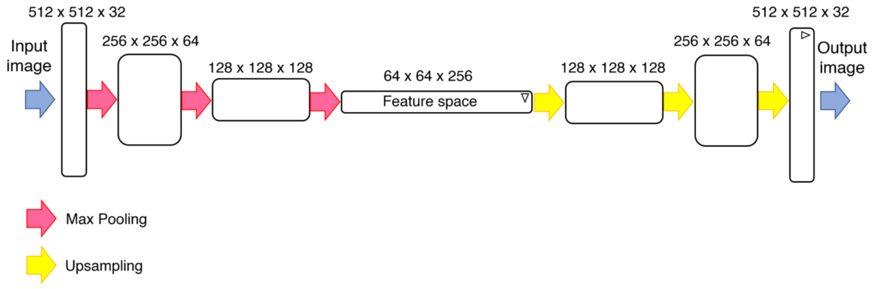

**Figure 3** Structure of Ano-AE, the convolutional autoencoder used in the anomaly detection algorithm. Filter sizes correspond to Model B3 in the body text and each block is composed of convolutional layers, two batch normalizations, ReLU activation and residual skip connections. The block marked with the symbol ▽ has a dropout layer at its end. The block marked with the symbol ▹ is followed by a 1 × 1 kernel convolutional layer with a number of features equal to the number of channels of the input image

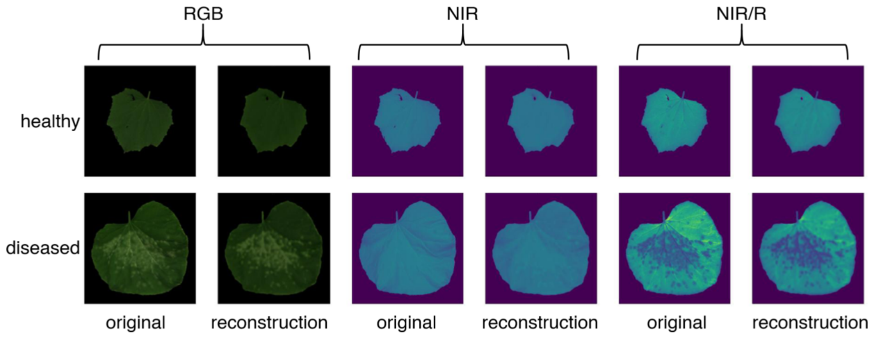

**Figure 4** Clu-AE model: comparison between original and reconstructed images of two random leaves. For each channels, RGB, NIR, and NIR/R, the original channel is shown on the left and its reconstruction on the right. The first row refers to a healthy leaf, the second row to a diseased leaf

## 5 Results and discussion

The numerical experiments were performed using the TensorFlow, Keras and scikit–learn machine learning libraries, under the Python 3 framework. All the computations have been run on the cluster INDACO owned by University of Milan on a NVIDIA Tesla K40 GPU.

### 5.1 Clustering approach results

*5.1.1 Clu-AE images reconstruction and compression*

A first indicator of the results obtained with the Clu-AE model was a qualitative assessment of its capability to reconstruct the visual features of input images. In Fig. 4 we show an illustrative example of the original datum $x$ and the reconstructed datum $\widehat{x}$ for one random healthy leaf and one random diseased leaf. The Clu-AE autoencoder resulted able to reconstruct the leaves shapes, and attributes like veins or other spots, with fair accuracy, even if a certain degree of blurriness was however present, as commonly found in applications of autoencoders. Figure 5 visualizes the 64 maps of the learned compressed features for a random diseased leaf. It is evident that many of the features are focused more on shape (for example in features: 1, 3, 4, 9, 15, 28, 30, 31, 51, 52, 61, 62) than on inner patter, by encoding the leaf area or edge highlighting different portions of the leaf shape and

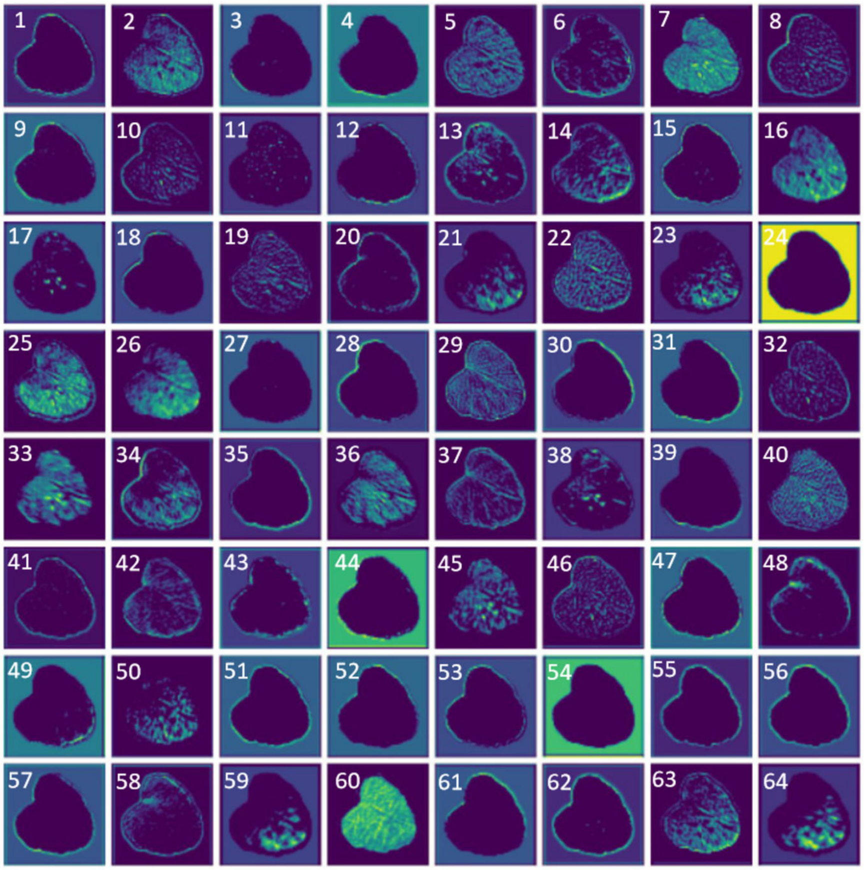

**Figure 5** Visualization of the representation space extracted by the Clu-AE model. The input image is the (diseased) leaf shown in RGB. Each of the 64 features has its own range of values; the brighter the color of a pixel in the feature image, the higher its numerical value

edge. More interestingly, other features encode the information in the interior of the leaf, highlighting leaf veins (for example, in features: 2, 29, 37), shades of healthy tissue (for example, in features: 7, 29, 37, 60) or the presence and degree of severity of disease spots (for example, features: 13, 32, 33). Not all the features are intuitive to interpret, as some of them light up without a clear meaning for a human observer.

### *5.1.2 Clustering results*

The clustering quality has been evaluated using two common metrics:

- Silhouette coefficient [34], defined as:

$$S(i) = \frac{d_s(i) - d_a(i)}{\max\{d_a(i), d_s(i)\}},$$

  where $d_a(i)$ is the average distance of point $i$ from all other points belonging to its cluster and $d_s(i)$ is the smallest average distance of $i$ to all points in any other cluster. The Silhouette coefficient measure how well each individual point fits in its cluster: if $S \simeq 0$, the point is right at the border between two clusters; if $S \simeq -1$ the point would be better assigned to another cluster, if $S \simeq 1$, the point is well-assigned to its cluster. In order to obtain a global evaluation of the clustering quality, it is customary to average the Silhouette coefficients of all the points to give the Average Silhouette coefficient (aSC);

- Davies-Bouldin index (DB) [35], defined as

$$DB = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq i}\frac{\sigma_i + \sigma_j}{d(c_i, c_j)},$$

where $n$ is the number of clusters, $d(c_i, c_j)$ is the distance between the centroid of cluster $i$ and cluster $j$, and $\sigma_i$ is the average distance of all points in cluster $i$ from its centroid $c_i$. The DB index leverages the concept that very dense and well spaced clusters constitute a good clustering result. The lower the DB value, the better the clustering performance, with a minimum score of zero corresponding to perfect clustering.

Table 1 reports the evaluation metrics of the clustering performance when all of the 64 extracted features were considered. The number of clusters obtained is considered here as an hyperparameter to be chosen. The best results indicated by the two metrics were obtained by a different number of clusters: 2-clusters for aSC and 4-clusters the for DB index. When analysing the composition of the 2-clusters case, we found that the first cluster contained the vast majority of the leaf samples, both healthy and diseased leaves, even the most severely infected leaves. The second cluster contained as well both healthy and diseased leaves. This result can be arguably explained by the fact that many of the leaves are similar in terms of their shape and size. More interestingly, when analysing the 4-clusters case, we found that the first and third clusters again contained both healthy and diseased leaves. On the other hand, the fourth cluster only contained healthy leaves, while the second cluster was instead composed of the most severely diseased leaves. Despite this encouraging result, it must be underlined that also the 4-clusters partition failed to establish if a certain leaf with no evident signs of infection was diseased or not. In fact, the dichotomy "healthy cluster" vs "diseased cluster" was lost in favour of size, shape characteristics, as it is especially evident in the first, third and fourth cluster. This failure can be mainly explained by two reasons: first, the dimensionality of the data, even if it was reduced by 80% in the encoding, it appeared to be still too high for the $k$-means algorithm; secondly, too many features identified by the Clu-AE network only detected the edge of the leaf, which explains the tendency of the model to partition the leaves based on their shape and dimension. In an attempt to improve these results, we tried to cluster a compressed version of our data, only considering a restricted number of features as they appeared more relevant for the objective of the study. We obtained the best results by selecting only one feature, *i.e.* feature no.33 (see Fig. 5). In Fig. 6, we report some illustrative responses of this feature when applied to input leaves exhibiting different levels of disease symptoms. This highlights how this feature is evidently responsive to disease spots: the more severe the disease symptoms, the higher the response.

**Table 1** Evaluation metrics for the Clu-AE approach obtained using all the 64 extracted features. In bold the best performance for each index

| | aSC | DB |
|---|---|---|
| 2 clusters | **0.17056** | 2.33102 |
| 3 clusters | 0.13617 | 2.21739 |
| 4 clusters | 0.14190 | **2.14507** |

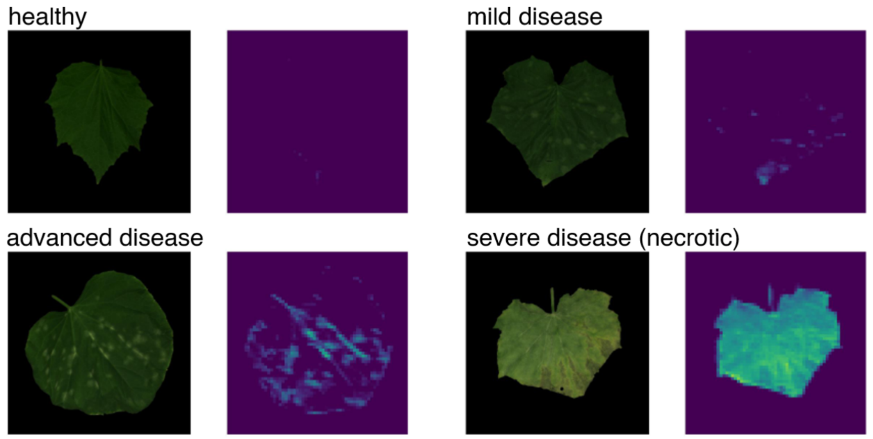

**Figure 6** Example of responses of feature 33 to different input leaves. The feature lights up in correspondence of disease symptoms (values are normalized considering all the leaves in the dataset)

**Table 2** Evaluation metrics of the Clu-AE approach when only feature no.33 of Fig. 5 is used. In bold the best performance for each index

| | aSC | DB |
|---|---|---|
| 2 clusters | **0.58295** | **0.91016** |
| 3 clusters | 0.33386 | 1.38958 |
| 4 clusters | 0.32163 | 1.87777 |

Table 2 reports the evaluation metrics of the clustering when only feature no.33 is considered. Both the aSC and DB indexes improved considerably compared to previous Table 1 referring to all the 64 features, and both the metrics indicated as best performing the 2-clusters partition. This showed that by sharply reducing the dimension of data, the performance of the clustering largely improved. Looking at the composition of the clusters, however indicated that the two clusters were not really informative yet for the aim of the study. The first cluster was composed by the majority of the samples, with both healthy and mildly diseased leaves grouped together, while the second cluster contained the most severely diseased leaves. The single extracted feature was evidently not informative to discriminate all the disease ranges, with early–mild symptoms appearing much more similar to healthy leaves than to diseased ones at the feature level. Such a problem was not solvable by increasing the granularity (number) of clusters, as our experiments showed that more clusters still group healthy and diseased leaves together.

## 5.2 Anomaly detection approach

### *5.2.1 Reconstruction results*

Looking at the capability of Ano-AE to reconstruct the visual features of input images, first we refer healthy leaves. Figure 7 shows as the Ano-AE network can reconstruct with great accuracy the pattern of leaves, together with their characteristic traits like the stem, the veins, or different shades of color. Not surprisingly, the reconstructed images show a certain blurriness compared to the original ones, even if much lesser than found in the clustering. When considering images of diseased leaves (see Fig. 8), the reconstruction power is fair enough, even if the diseased area spots are reconstructed in a more imprecise and blurred way compared to healthy tissue areas. Furthermore, the reconstructed

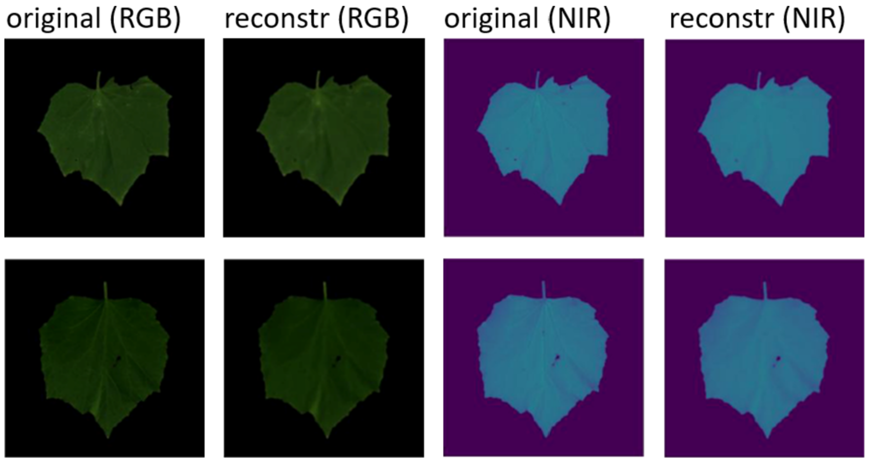

**Figure 7** Comparison between original and reconstructed channels of healthy leaves belonging to the test set

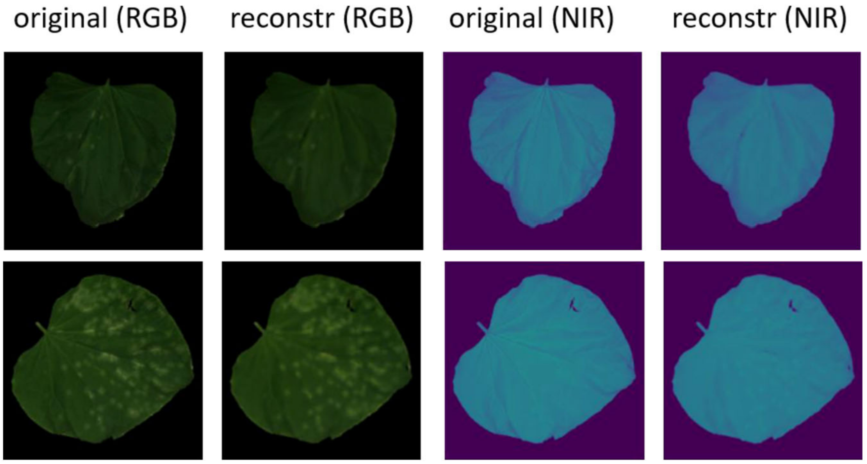

**Figure 8** Comparison between original and reconstructed channels of diseased leaves belonging to the test set

color of the disease spots deviated from the original, appearing in general more brownish and slightly darker. The capability of the Ano-AE to reconstruct symptoms spots in diseased leaves demonstrates that a number of healthy leaves in the training set included some lesions or other–than–disease spots which trained the model to reconstruct also powdery mildew spots. For example, in the leaf of the first row in Fig. 8, the presence of a whitish area with similar visual features of powdery mildew symptoms can be observed. Of course, by removing all healthy leaves with such imperfections from the training set could increase the performance of the model, but this would make the model useless in a real-life scenario, where leaves are normally riddled with imperfections due to different reasons other than disease.

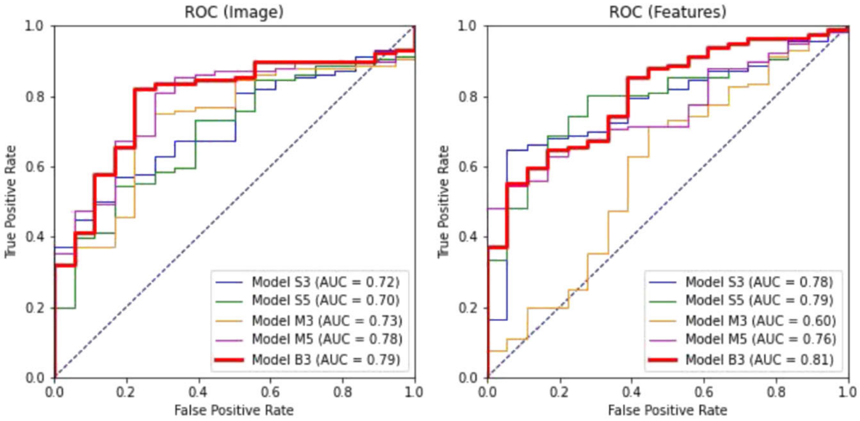

**Figure 9** ROC curves and AUC values for the different neural architectures proposed for anomaly detection. The bold ROC corresponds to the best performing model (Ano-AE), the dashed curve corresponds to random guess. The panel on the left refers to the performance of the models when considering the image reconstruction error as anomaly score, while the panel on the right considers feature reconstruction error as anomaly score

### *5.2.2 Anomaly detection results*

The performance of the Ano-AE model in unsupervised detection of anomalies, *i.e.* disease symptoms, was quantified by using the Receiver Operating Characteristic (ROC) curve, which represents the probability of detection of false positives vs true positives by spanning all possible values of the threshold $\Gamma$. By lowering the anomaly threshold, more samples are classified as positive (anomalies), resulting in an increase of both false positives and true positives. The overall capability of detection can be indicated by the AUC, *i.e.* the area under the ROC curve (AUC). The AUC can be interpreted as the probability that a classifier gives a higher probability of being an anomaly to a randomly chosen abnormal (diseased) sample than to a randomly chosen normal (healthy) sample. AUC values range between 0 (classification estimated labels are always wrong) and 1 (classification estimated labels are always correct), with mid scale 0.5 (random guess). Figure 9 shows the ROC curves obtained for the different neural architectures introduced above for anomaly detection. Each curve shows the diagnostic ability of the model while varying the threshold $\Gamma$ applied in the scoring system. Results are reported both the image reconstruction error $s_x$ as anomaly score (Fig. 9, left) and the feature reconstruction error as anomaly score $s_z$ (Fig. 9, right). As anticipated, it is evident that model B3 (*i.e.*, Ano-AE) performed better than the other experimented models, with the highest overall AUC for both the scores considered.

## 6 Conclusions

This study developed two deep learning approaches based on autoencoder networks for the automatic recognition of powdery mildew disease in multispectral images of cucumber leaves. The specific objective was to explore unsupervised techniques to overcome the need of large training set of manually labelled images, typical of CNN applications. To this aim, autoencoder networks architectures were implemented to obtain:

i) a clusterization of the features in a compressed space. This approach showed a limited capability to provide accurate disease detection, even if it was able to highlight relevant information contained in compressed features. This prospects potential of improvements when using feature clustering with supervised preliminary filtering in order to further filter the relevant features;

ii) an anomaly detection approach. This approach showed superior capabilities in unsupervised detection in diseased leaves, with a significant potential of applications in reducing the need of manual labelling of leaves images. Indeed, the developed approach could be used as an unsupervised classifier trained on a large dataset integrated with a supervised neural network trained with a limited number of manually labelled samples. Beside the binary classification healthy versus diseased leaves, this approach may provide a reliable quality check on synthetic leaves [36], artificially generate by a GAN architecture, which can be employed in a data augmentation strategy for Neural Network training.

**Acknowledgements**
Author PC acknowledges the support of the Italian National project MIUR PRIN 2017, Numerical Analysis for Full and Reduced Order Methods for the efficient and accurate solution of complex systems governed by Partial Differential Equations (NA-FROM-PDEs).

**Funding**
Not applicable

**Abbreviations**
DL, Deep Learning; CNN, Convolutional Neural Network; PA, precision agriculture; VI, vegetation index; NIR, near-infrared; VIS, visible; AE, autoencoder; Clu-AE, clustering approach; Ano-AE, anomaly detection approach.

**Availability of data and materials**
Data and materials will be made available upon request

## Declarations

**Ethics approval and consent to participate**
Not applicable

**Consent for publication**
Not applicable

**Competing interests**
The authors declare no competing interests.

**Author contributions**
AB, PC and RO conceptualized the work and designed the experiments. RO provided access to plant material and performed disease inoculation; AB, PC and GS perfomed data collection and carried out numerical simulations; AB, PC and RO performed analysis of the data and wrote the paper. All authors read and approved the final manuscript.

**Authors' information**
AB and PC belong to the GNCS group of INdAM (Istituto Nazionale di Alta Matematica)

**Author details**
[1]Dept. of Environmental Science and Policy, University of Milan, Milan, Italy. [2]Dept. of Mathematics, University of Milan, Milan, Italy. [3]Dept. of Agricultural and Environmental Sciences - Production, Landscape, Agroenergy, University of Milan, Milan, Italy.

Received: 29 April 2022 Accepted: 29 March 2023 Published online: 30 May 2023

**References**
1. Waggoner PE, Aylor DE. Epidemiology: a science of patterns. Annu Rev Phytopathol. 2000;38(1):71–94.
2. Oberti R, Schmilovitch Z. Robotic spraying for precision crop protection. In: Innovation in agricultural robotics for precision agriculture. Berlin: Springer; 2021. p. 117–50.
3. Cisternas I, Velásquez I, Caro A, Rodríguez A. Systematic literature review of implementations of precision agriculture. Comput Electron Agric. 2020;176:105626.
4. Martinelli F, Scalenghe R, Davino S, Panno S, Scuderi G, Ruisi P, Villa P, Stroppiana D, Boschetti M, Goulart LR et al. Advanced methods of plant disease detection. A review. Agron Sustain Dev. 2015;35(1):1–25.
5. Sankaran S, Mishra A, Ehsani R, Davis C. A review of advanced techniques for detecting plant diseases. Comput Electron Agric. 2010;72(1):1–13.

6. Barbedo JGA. A review on the main challenges in automatic plant disease identification based on visible range images. Biosyst Eng. 2016;144:52–60.
7. Mahlein A-K. Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. Plant Dis. 2016;100(2):241–51.
8. Gulhane VA, Gurjar AA. Detection of diseases on cotton leaves and its possible diagnosis. Int J Image Process. 2011;5(5):590–8.
9. Pixia D, Xiangdong W et al. Recognition of greenhouse cucumber disease based on image processing technology. Open J Appl Sci. 2013;3(1):27–31.
10. Delalieux S, Somers B, Hereijgers S, Verstraeten W, Keulemans W, Coppin P. A near-infrared narrow-waveband ratio to determine leaf area index in orchards. Remote Sens Environ. 2008;112(10):3762–72.
11. Vigier BJ, Pattey E, Strachan IB. Narrowband vegetation indexes and detection of disease damage in soybeans. IEEE Geosci Remote Sens Lett. 2004;1(4):255–9.
12. Oberti R, Marchi M, Tirelli P, Calcante A, Iriti M, Borghese AN. Automatic detection of powdery mildew on grapevine leaves by image analysis: optimal view-angle range to increase the sensitivity. Comput Electron Agric. 2014;104:1–8.
13. Zhang S, Wu X, You Z, Zhang L. Leaf image based cucumber disease recognition using sparse representation classification. Comput Electron Agric. 2017;134:135–41.
14. Boulent J, Foucher S, Théau J, St-Charles P-L. Convolutional neural networks for the automatic identification of plant diseases. Front Plant Sci. 2019;10:941.
15. Mohanty SP, Hughes DP, Salathé M. Using deep learning for image-based plant disease detection. Front Plant Sci. 2016;7:1419.
16. PlantVillage Dataset. https://www.kaggle.com/emmarex/plantdisease.
17. Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. Deep neural networks based recognition of plant diseases by leaf image classification. Comput Intell. 2016;2016:3289801.
18. Fuentes A, Yoon S, Kim SC, Park DS. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors. 2017;17(9):2022.
19. Wspanialy P, Moussa M. A detection and severity estimation system for generic diseases of tomato greenhouse plants. Comput Electron Agric. 2020;178:105701.
20. Lin K, Gong L, Huang Y, Liu C, Pan J. Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. Front Plant Sci. 2019;10:155.
21. Lowe A, Harrison N, French AP. Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress. Plant Methods. 2017;13(1):1–12.
22. Saleem MH, Potgieter J, Arif KM. Plant disease detection and classification by deep learning. Plants. 2019;8(11):468.
23. West JS, Bravo C, Oberti R, Lemaire D, Moshou D, McCartney HA. The potential of optical canopy measurement for targeted control of field crop diseases. Annu Rev Phytopathol. 2003;41(1):593–614.
24. Bank D, Koenigstein N, Giryes R. Autoencoders. 2020. arXiv preprint. arXiv:2003.05991.
25. Michelucci U. An introduction to autoencoders. 2022. arXiv preprint. arXiv:2201.03898.
26. Bonettini S, Benfenati A, Ruggiero V. Primal-dual first order methods for total variation image restoration in presence of Poisson noise. In: 2014 IEEE international conference on image processing (ICIP). 2014. p. 4156–60.
27. Zhang Z. Improved ADAM optimizer for deep neural networks. In: 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS). 2018. p. 1–2. https://doi.org/10.1109/IWQoS.2018.8624183.
28. Bonettini S, Benfenati A, Ruggiero V. Scaling techniques for $\epsilon$-subgradient methods. SIAM J Optim. 2016;26(3):1741–72.
29. Piernik M, Morzy T. A study on using data clustering for feature extraction to improve the quality of classification. Knowl Inf Syst. 2021;63:1771–805.
30. Picetti F, Testa G, Lombardi F, Bestagini P, Lualdi M, Tubaro S. Convolutional autoencoder for landmine detection on GPR scans. In: 2018 41st international conference on telecommunications and signal processing (TSP). IEEE; 2018. p. 1–4.
31. Hadjeres G, Nielsen F, Pachet F. Glsr-vae: geodesic latent space regularization for variational autoencoder architectures. In: 2017 IEEE symposium series on computational intelligence (SSCI). IEEE; 2017. p. 1–7.
32. Osada G, Ahsan B, Bora RP, Nishide T. Regularization with latent space virtual adversarial training. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part I. vol. 16. Berlin: Springer; 2020. p. 565–81.
33. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30.
34. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53–65. https://doi.org/10.1016/0377-0427(87)90125-7.
35. Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell. 1979;PAMI-1(2):224–7. https://doi.org/10.1109/TPAMI.1979.4766909.
36. Benfenati A, Bolzi D, Causin P, Oberti R. A deep learning generative model approach for image synthesis of plant leaves. 2021. CoRR, 2111.03388. arXiv:2111.03388.

## Publisher's Note