Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Detection of grapevine yellows symptoms in *Vitis vinifera* L. with artificial intelligence

Albert Cruz[a,*], Yiannis Ampatzidis[b,*], Roberto Pierro[c], Alberto Materazzi[c], Alessandra Panattoni[c], Luigi De Bellis[d], Andrea Luvisi[d]

[a] *Department of Computer and Electrical Engineering and Computer Science, California State University, Bakersfield, 9001 Stockdale Highway, Bakersfield, CA 93311, USA*
[b] *Department of Agricultural and Biological Engineering, Southwest Florida Research and Education Center, University of Florida, 2685 SR 29, Immokalee, FL 34142, USA*
[c] *Department of Biological and Environmental Sciences and Technologies, University of Salento, via Prov. le Monteroni, 73100 Lecce, Italy*
[d] *Department of Agriculture, Food and Environment, University of Pisa, Via del Borghetto, 80, 56124 Pisa, Italy*

A R T I C L E I N F O

Keywords:
Disease detection
Neural networks
Machine learning
Grapes
Symptom-based

A B S T R A C T

Grapevine yellows (GY) are a significant threat to grapes due to the severe symptoms and lack of treatments. Conventional diagnosis of the phytoplasmas associated to GYs relies on symptom identification, due to sensitivity limits of diagnostic tools (e.g. real time PCR) in asymptomatic vines, where the low concentration of the pathogen or its erratic distribution can lead to a high rate of false-negatives. GY's primary symptoms are leaf discoloration and irregular wood ripening, which can be easily confused for symptoms of other diseases making recognition a difficult task. Herein, we present a novel system, utilizing convolutional neural networks, for end-to-end detection of GY in red grape vine (cv. Sangiovese), using color images of leaf clippings. The diagnostic test detailed in this work does not require the user to be an expert at identifying GY. Data augmentation strategies make the system robust to alignment errors during data capture. When applied to the task of recognizing GY from digital images of leaf clippings—amongst many other diseases and a healthy control—the system has a sensitivity of 98.96% and a specificity of 99.40%. Deep learning has 35.97% and 9.88% better predictive value (PPV) when recognizing GY from sight, than a baseline system without deep learning and trained humans respectively. We evaluate six neural network architectures: AlexNet, GoogLeNet, Inception v3, ResNet-50, ResNet-101 and SqueezeNet. We find ResNet-50 to be the best compromise of accuracy and training cost. The trained neural networks, code to reproduce the experiments, and data of leaf clipping images are available on the internet. This work will advance the frontier of GY detection by improving detection speed, enabling a more effective response to the disease.

1. Introduction

Grapevine yellows (GY) are among the most important diseases currently studied in grapevine. Among GYs, two causal phytoplasmas are a major concern: Flavescence dorée (FD) and Bois noir (BN). FD (*Candidatus* Phytoplasma vitis) is a member of the Elm Yellows group, (Martini et al., 1999) and BN (*Ca.* Phytoplasma solani) is a member of the Stolbur group (16SrXII) (Quaglino et al., 2013). These are considered the most dangerous phytoplasmas found in all major wine-growing areas of Euro-Mediterranean countries, Chile and Asia (Gajardo et al., 2009; Belli et al., 2010; Mirchenari et al., 2015). In Europe and the Mediterranean basin, the causal agent of FD is classified as quarantine pest. FD is vectored from vine to vine by *Scaphoideus titanus*, a non-native ampelophagous leafhopper from North America (Chuche and Thiéry, 2014). Conversely, the BN etiological agent is transmitted by the polyphagous leafhopper, *Hyalesthes obsoletus*, endemic to Europe (Maixner, 1994). Current control strategies include uprooting of infected plants and vector control with pesticides. Laboratory-limited activities use antibiotics for cultivar recovery or Meristem tip culture, that can be used to generate phytoplasma-free in vitro material (Bertaccini, 2007), but they do not provide successful phytoplasma disease control. A more efficient strategy to limit phytoplasma disease diffusion is needed.

Detection protocols based on molecular assay for phytoplasmas are not completely reliable when applied to woody plants. This is due to the low concentration of the pathogen and its erratic distribution in these

* Corresponding authors.
E-mail address: i.ampatzidis@ufl.edu (Y. Ampatzidis).

https://doi.org/10.1016/j.compag.2018.12.028
Received 20 August 2018; Received in revised form 14 November 2018; Accepted 13 December 2018

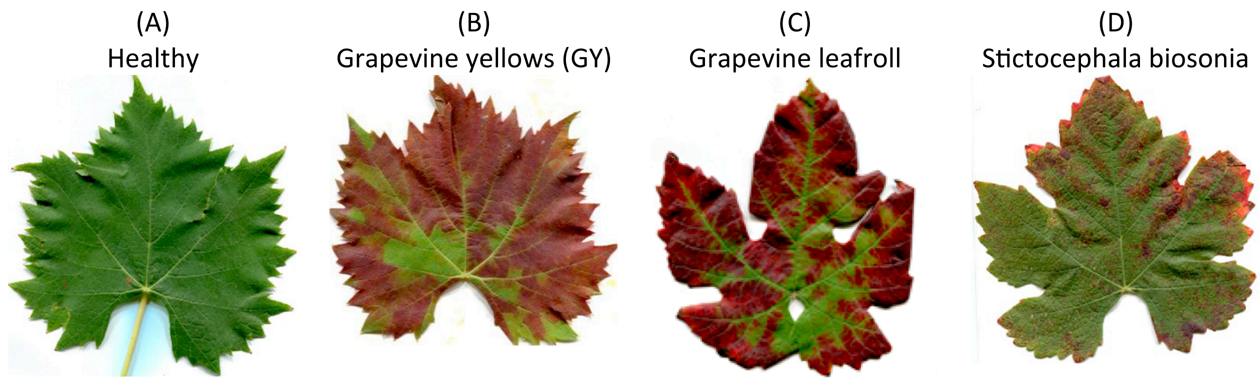


Fig. 1. Grapevine yellows (GY) caused by Flavescence dorée (FD) or Bois noir (BN) are characterized by discoloration of the leaf blade. However, some GY symptoms are common to other diseases. This figure includes examples of leaf clippings from *Vitis vinifera* L. cv. Sangiovese that illustrate how visual recognition is a challenging task: (A) a healthy control for reference, (B) GY, (C) grapevine leafroll and (D) *Stictocephala bisonia*.

hosts (Belli et al., 2010). Thus, diagnosis of phytoplasma diseases in grapevines relies on symptom identification and visual symptom recognition plays a strategic role in phytoplasma control. Further, effective sampling of plants reduces the risk of false negatives. BN and FD differ in etiology and epidemiology but are indistinguishable by symptoms: desiccation of inflorescences, reduction of growth, leaf discoloration (showing typical sectorial reddening in red cultivars), berry shrivel and irregular maturation of wood. The time at which sampling occurs (late summer) makes it difficult to distinguish GY from other grapevine disorders that may show similar symptoms. For example, GY symptoms can be similar to leafroll or direct damage due to leafhopper feeding (Belli et al., 2010). In particular, in red grapes, esca disease (caused by fungi complex) and grapevine leafroll (caused by viruses) can cause leaf discolorations which may be chromatically similar to GY. The insect *Stictocephala bisonia* causes sectorial reddening of leaves which are also quite similar to that observed in red grapes infected by GY. Thus, symptoms of GY can be confused for symptoms of other diseases, even by experts (see Fig. 1).

Supervised learning has the potential to distinguish GY from other diseases (Ampatzidis et al., 2017; Cruz et al., 2017; Sharif et al., 2018; Zhou et al., 2014). It is automatic without the need for prior expertise or skill. It can support sampling procedures by reducing false positives in large-scale vineyard monitoring. To the best of the authors' knowledge, we are the first to investigate end-to-end symptom identification of GY with deep learning.

1.1. Related work and motivation

Accurate and timely diagnosis of the disease is the most important tactic for plant disease control. But, when carried out by humans, it requires continuous monitoring and manual observation. This is prone to error, time-consuming, and costly (Ali et al., 2017). Automatic tools that identify and detect plant diseases could be an effective solution in monitoring real-time disease diffusion in crop fields (Abdulridha et al., 2018; Ampatzidis et al., 2017; Zhou et al., 2014). In recent years, various studies focused on the development of computer vision techniques to detect disease from leaf clipping images. Kaur et al. (2018) developed a system to detect disease in Glycine max with a k-means based segmentation algorithm. The system detects downy mildew, frog eye and Septoria leaf blight from images collected by the PlantVillage project (https://plantvillage.psu.edu/). A combination of color and texture features achieve an accuracy of 90.7%. Zhou et al. (2014) presented a novel representation of a leaf clipping image—also known as a feature representation, or feature—called orientation code matching. The feature overcomes differences in light source, occlusion, rotation and translation variations. The authors demonstrate good precision, recall and F-measure results when the feature is paired with a support vector machine (SVM). Sengar et al. (2018) proposed an intensity thresholding method for measuring the progression of powdery mildew in cherries. This study also used the PlantVillage dataset. Sharif et al. (2018) presented a hybrid method detecting and identifying diseases in citrus plants. The method segments citrus lesion spots with optimized weighted segmentation method uses color, texture, and geometric features and carries out feature selection with dimensionality reduction, entropy, and skewness-based covariance vector; finally, it classifies the samples with a support vector machine. The method is tested on the citrus disease image gallery dataset, plant village and a local dataset, and is capable of detecting anthracnose, black spot, canker, scab, greening, and melanose. Ali et al. (2017) proposed a method based on color difference to segment diseased parts of Kinnow mandarin leaves. Together with color histogram and texture features, the authors achieve good accuracy and area under the curve (AUC).

Recently in computer vision, there is a focus on deep learning algorithms (LeCun et al., 2015, Szegedy et al., 2015). Deep learning, also known as neural networks, are a type of supervised learning algorithm. Supervised learning is a field of artificial intelligence (AI) and machine learning (ML). In supervised learning, a system automates a domain-specific task. Some examples are face recognition, speech recognition, and object recognition. The algorithm develops a model of inference and reasoning from a set of labelled training samples. The goal is for the system to make correct decisions on a different set of test samples. It carries out automatic deduction of the sample's membership to a specific population. Supervised learning can be an effective tool for diagnosing pests and diseases. AI and ML can provide expert analysis and diagnosis of plant diseases. The end-user does not need to be an expert at identifying the disease. Deep learning is capable of even better performance than the mentioned related work (Cruz et al., 2017; Sharif et al., 2018). In previous work, supervised learning detected diseases with similar symptoms and pests, while reducing diagnosis time and cost (Ampatzidis et al., 2017; Luvisi et al., 2016). For example, Cruz et al. (2017) developed a vision-based program to detect symptoms of Olive Quick Decline Syndrome (OQDS) on leaves of *Olea europaea* L. infected by *Xylella fastidiosa*. This has great potential to distinguish symptomatic and asymptomatic leaves for sample selection in large-scale monitoring programs (Cardinale et al., 2018) and field.

1.2. Objectives

Herein, we demonstrate the potential for deep learning to detect grapevine yellows (GY) in *Vitis vinifera* L. cv Sangiovese, a red globe cultivar sensitive to this disease (Pierro et al., 2018a,b). An overview of the proposed system is given in Fig. 2. To the best of the authors' knowledge, we are the first to approach this problem with deep

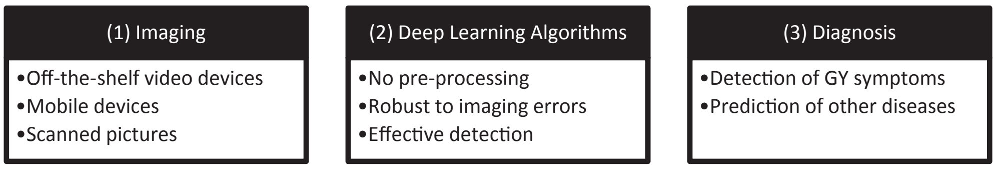

(1) Imaging
•Off-the-shelf video devices
•Mobile devices
•Scanned pictures
(2) Deep Learning Algorithms
•No pre-processing
•Robust to imaging errors
•Effective detection
(3) Diagnosis
•Detection of GY symptoms
•Prediction of other diseases

Fig. 2. A high-level description of the overall system. (1) Off-the-shelf consumer grade cameras, mobile devices and scanners can be used to take a photo of a leaf clipping image. (2) Deep learning, a machine learning (ML) and artificial intelligence (AI) algorithm, (3) provides end-to-end diagnosis of GY symptoms and other diseases.

learning. This work addresses the following research questions:

(1) Is it possible to automatically detect GY from leaf clipping images? And, can it be distinguished from other, similar looking diseases?
(2) Among the many deep learning architectures available, which algorithm is best?
(3) How does the proposed system compare to related work that does not use deep learning?
(4) How do human recognition rates compare to the machine recognition rates?

Surprisingly, few AI and ML studies compare human performance. Previous researchers were content to study AI and ML for the sake of advancing the frontier of research. Or, it was assumed that machine would be better. One study (Russakovsky et al., 2015) applied ML to predict 1000 different object categories and found that human recognition rate was roughly the same or slightly worse than AI recognition rates (1.7% better to 6.1% worse). Another study (Dodge and Karam, 2017), that also used 1000 different object categories, found that human performance is roughly the same as AI performance. However, these studies are for many different object categories. There are no studies of AI vs. human performance for plant disease diagnosis, and no studies for our specific problem (grapes and grapevine yellows).

2. Materials and methods

2.1. Data methods

Two sets of data are aggregated for this study. Section 2.1.1 discusses the first of the two data sets. Plants were surveyed, sampled, photographed and then diagnosed for GY with DNA analysis of the pathogen. With supervised learning, the ML algorithm learns from example (datasets). In training, the algorithm is fed a digital image with a correct diagnosis and learns to recognize visual cues associated with that diagnosis. The membership of a sample to a disease population must be correct or the ML algorithm will learn the wrong cues. Visual inspection is not enough to verify GY, thus samples are lab verified to have the phytoplasmas BN and FD with DNA extraction followed by real time PCR test. Section 2.1.3 discusses the second of the two datasets, PlantVillage (https://plantvillage.psu.edu/). The machine learning algorithms used in the work require on a very large data set—more than we collected in the first dataset. We use PlantVillage data to fulfill this requirement. This is called *data augmentation*. PlantVilllage data comes with labels so there is no verification. Section 2.1.5 discusses how we further augment the data by generating multiple samples from a single sample with image processing.

2.1.1. Sampled data

Field surveys were conducted in Tuscany (Central Italy) from July to October 2017 in vineyards of cv. Sangiovese localized in several districts. Healthy control (HC) samples of cv. Sangiovese were collected from the greenhouse of the Department of Agriculture, Food and Environment (DAFE, University of Pisa, Italy). Infected control (IC) leaves were collected from *V. vinifera* plants that were previously assessed by molecular tools and found to be infected by '*Ca*. P. solani' (subgroup 16SrXII-A) and Flavescence dorée phytoplasmas (subgroups 16SrV-C and -D).

To further stress the system, we also collected non-GY leaves from grapevines that showed other diseases: downy mildew, esca disease, grapevine leafroll, powdery mildew and *Stictocephala bisonia*. Recognition of non-GY diseases was carried out visually, evaluating symptoms. Images of downy mildew, grapevine leafroll, powdery mildew and *S. bisonia* were combined into a single population because there were insufficient samples to form separate populations. We refer to this population as other (OD). In total, we collected 134 healthy controls, 134 GY and 104 samples of other diseases.

2.1.2. DNA verification of sampled data

The following describes how we verify GY for collected data with DNA extraction and real time PCR tests. For each grapevine, 10–12 leaves were collected, and their fresh central midribs were dissected and stored at −20 °C until DNA extraction. DNA was extracted with 2% cetyltrimethylammonium bromide (CTAB) based buffer from leaf veins according to the protocol described by Li et al. (2008), with the modifications that follow. Briefly, 1 g of leaf veins were smoothed in plastic bags (Bioreba, Switzerland) with 5 ml of 2% CTAB buffer using Homex 6 (Bioreba, Switzerland). The homogenate was incubated at 65 °C for 15 min. DNA was extracted by one volume of chloroform: iso-amylalcohol (24:1) and precipitated with one volume of isopropanol. Pellets were washed with 70% ethanol, air-dried, suspended in 100 µl of deionized water and stored at −20 °C until use. Concentrations of the nucleic acids were determined by measuring the absorbance at 260 nm with a spectrophotometer. Purity was assessed by calculating the ratio of the absorbance at 260 nm over the absorbance at 280 nm. Specific detection of phytoplasmas associated with BN and FD was carried out by amplification of 16S ribosomal DNA through TaqMan assay using the Rotor-Gene Q (Qiagen, Germany) following reaction conditions as described by Angelini et al. (2007). The template used in the assay was a 1:10 dilution of the DNA extracted from the samples. The grapevine chloroplast chaperonin 21 gene was used as endogenous control, while DNA extracted from HC plants and ICs were used as negative and positive controls, respectively. The DNA samples, which gave no positive signal to endogenous gene, were further cleaned up and tested with real-time PCR until they yielded the specific control amplicon. Threshold cycle (Ct) $<$ 37 was associated with the presence of GY phytoplasmas (Mori et al., 2015).

2.1.3. Transfer learning

The machine learning algorithms used in the work require on a very large data set. This helps the algorithm learn discriminative features of the disease of interest and reduces the false negatives by providing the system with knowledge of cues that are irrelevant to GY. However, sourcing the required number of images can pose a challenge. As a rule-of-thumb, thousands to hundreds of thousands of images are normally expected. It would be too costly and time-consuming to collect such an order of magnitude of images, particularly for leaf clippings of specimens infected with a quarantine pathogen. Thus, we use a concept called transfer learning (Schmidhuber, 2015; Yosinski et al., 2014), described as follows:

(1) The algorithm is trained some other dataset that has a sufficient number of data samples. For example, in Karpathy et al. (2014) images from the ImageNet grand challenge are used. ImageNet has 150,000 data samples and 1000 different populations (object categories).

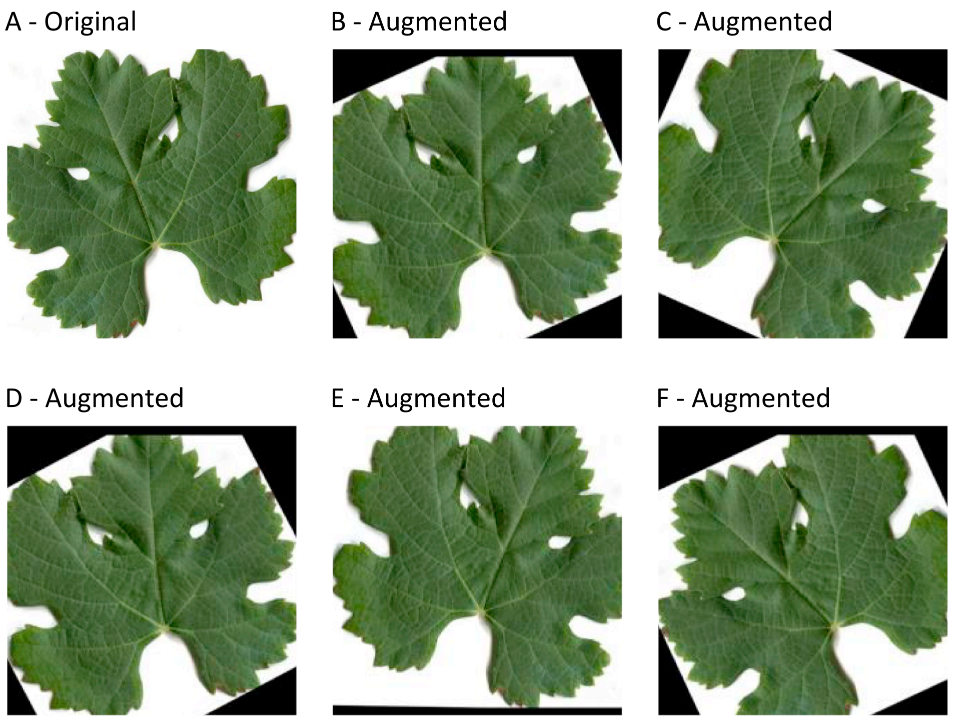


Fig. 3. The computer methods in this work require thousands to tens of thousands of data samples. Such an order of magnitude of samples would be too difficult collect and too costly to lab verify. One measure to address this is data augmentation with image processing. This figure demonstrates how a single sample can generate many samples by randomly translating, rotating and flipping the images. (A) Original. (B–F) Augmented images generated from A.

(2) The algorithm is then retrained with the data at hand.

The idea is that with a large, somewhat related dataset, the ML method can learn relevant patterns that are generally applicable for all problem domains. It is then re-trained on the data, and doing this is better than attempting to train the ML algorithm from scratch (Cruz et al., 2017; Schmidhuber, 2015; Yosinski et al., 2014). The ML algorithms in this work are publicly obtainable with step (1) completed, and this is also known as a *pre-trained network*. The reader is referred to those works for details of the training procedure (Krizhevsky et al., 2012; He et al., 2015; Szegedy et al., 2015; Iandola et al., 2016; Szegedy et al., 2016).

2.1.4. *Data augmentation from publicly available sources*

Even with transfer learning, we still need to add more data to our dataset for the ML algorithm to work. We add grapevine leaf clipping images from the publicly available PlantVillage dataset (https://plantvillage.psu.edu/). Specifically, we use the subset that is publicly available from Mohanty et al. (2016). The subset we use contains the following grapevine diseases: black rot, esca and leaf blight. This dataset does not contain any GY samples because GY is a nascent problem. Even so, we add it to our data to meet the minimum sample requirement for training a deep ML algorithm. We used all of the grapevine images provided by the dataset: 1180 images of grapevine with black rot; 1383 esca (black measles); 1076 leaf blight (Isariopsis leaf spot). The collected esca samples were merged into the augmented esca samples population. The reader is referred to the Mohanty et al. (2016) for methodology of health assessment for other diseases.

2.1.5. *Data augmentation from image preprocessing*

Before the images are processed by the neural network, we apply image processing techniques to center the leaf in the image. This procedure also removes unnecessary background information. In this step, data is further augmented with translation and rotation of original samples. First, the image is segmented from the background and noise. This automatically centers the leaf and removes the background from the canvas of the image. Second, the image is randomly flipped along the horizontal axis with a rate of 0.5. Third, the image is rotated randomly [−30°, 30°] about the center of the image and translated independently along the vertical and horizontal axes randomly by [−30, 30] pixels. Steps two and three generate many samples from an original image (Fig. 3). Perturbing the original images prevents the machine learning algorithm from overfitting to spatial image artifacts and allows a certain amount of tolerance when orienting the leaf during acquisition. As a result, the system does not need the leaves to be perfectly aligned, though the apex must be on the top and the petiole on the bottom. The pre-trained neural networks used in this work require an image of a specific size. We resize all leaf clipping image to 227×227, the input image size required by the AlexNet neural network architecture. For images of uneven aspect ratio, the horizontal axis is resized to match the required size. The data used in this study is available publicly on GitHub.[1]

In the following, we explain the segmentation procedure in detail. First, we obtain the *leaf mask*. The leaf mask is a black and white image where the white pixels correspond to the leaf. The leaf mask is obtained by:

1. Converting the original RGB image to grayscale with a YIQ transformation;
2. Blurring the grayscale image with a 3×3 Gaussian filter to remove noise;
3. Obtaining a rough leaf mask by thresholding with Otsu's algorithm (Baxi and Vala, 2013);
4. Convolving the leaf mask with a morphological *close* operation with the goal of removing the petiole from the final image—the *close* operation uses a 15×15 disc;
5. Finally, median filtering with a 11×11 filter to remove small objects due to noise.

The original image is cropped to the minimum bounding box enclosing the leaf mask. Note that classification in later stages uses RGB and that grayscale is only used for calculating the leaf mask. This procedure is automatic. An overview of the segmentation procedure is

[1] https://github.com/DrAlbertCruz/Salento-Grapevine-Yellows-Dataset.

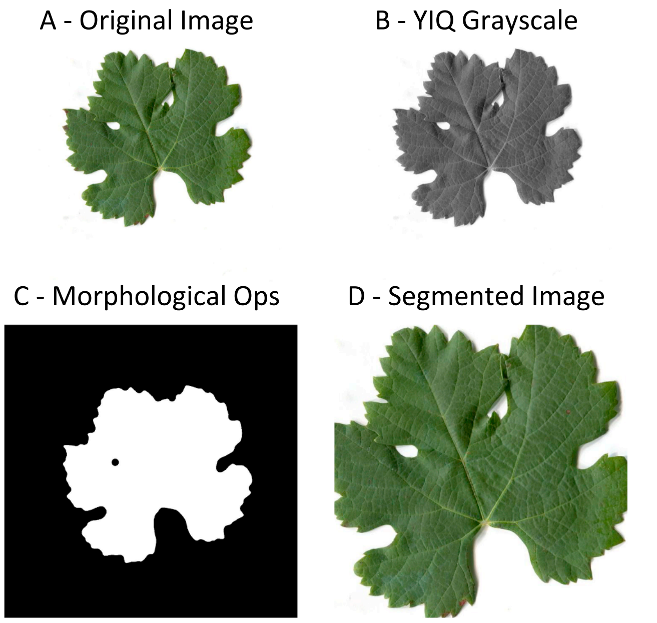


Fig. 4. The system proposed by this work is end-to-end. When presenting a leaf for inspection the user does not need to pre-segment the image or align it. The algorithm detailed in Section 2.1.5 automatically segments the leaf from the background, centers it in the frame, and crops the background. This figure includes step-by-step results of the process. (A) Original images are color JPEG images. An outline of the leaf blade is necessary to segment the leaf from the background. To accomplish this, (B) the original image is converted to grayscale with a YIQ transformation then (C) thresholded to obtain the final outline (also known as the leaf mask). (D) The original image's canvas is reduced to the minimum bounding box of the leaf mask.

given in Fig. 4.

2.2. Experimental methods and parameters

2.2.1. Background of deep learning

Deep learning was demonstrated to have high accuracy in other fields and has been successfully applied to the detection of a variety of crops and diseases (Mohanty et al., 2016; Cruz et al., 2017). Deep learning is a collection of methods that improve optimization and generalization of neural networks and allow neurons to share parameters (convolutional neural networks). Deep learning has gained popularity over the past few years, though it is not a new concept. It may have been used to describe neural networks as early as the 1980s (Dechter, 1986). Backpropagation, the training mechanism of a deep learning algorithm, has long been used to train neural networks (LeCun et al., 1989). A neural network consists of many ordered layers. In a *feed-forward* neural network, signals are propagated sequentially from the front layer to the end layer of the network. A layer consists of a set of neurons that receive input stimulus $\vec{x} = \{x_1, x_2, ..., x_n\}$. x_i is a real number that is the output of some other neuron, or—with image data and for the first layer only—an image pixel. $\vec{x}$ is the ordered output of all the neurons in the previous layer. The number of neurons, layers, and structure of the layers are an experimental parameter, and commonly used structures are given later in Section 2.2.2. The input is linearly weighted by the weight vector $\vec{w} = \{w_1, w_2, ..., w_n\}$. An overview of a single neuron is given in Fig. 5.

The result is the scalar charge X. If the charge is enough the neuron fires and sends stimulus to each neuron in the next layer of the network. The activation function defines the nature of how neurons fire. The neural networks in this work use a rectified linear activation function (ReLU), described as follows (Nair and Hinton, 2010):

$$y(X) = \max(0, X) \tag{1}$$

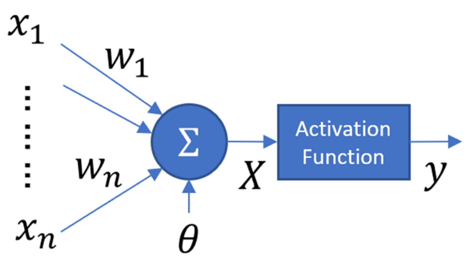


Fig. 5. A neural network consists of many layers and in each layer are neurons. This is an overview of a single neuron. A neuron receives input stimulus $\vec{x} = \{x_1, x_2, ..., x_n\}$ and holds weights $\vec{w} = \{w_1, w_2, ..., w_n\}$. θ is the bias term. The resulting charge X is the dot product of $\vec{x}$ and $\vec{w}$ plus the bias term. It is thresholded by the activation function to produce the output y.

where max(.) is the maximum value of its input arguments. It is currently the most popular activation function (Krizhevsky et al., 2012; He et al., 2015; Szegedy et al., 2015; Iandola et al., 2016; Szegedy et al., 2016) because it addresses the problem of the vanishing gradient. The neural network is a generally feed forward structure, where information propagates layer by layer through the network. At the front end of the network, the whole image is given as input. The end of the network reports the confidence of the membership of a sample to a particular population. Unlike the other layers, the end layer uses a softmax activation function to generate pseudo-probabilities. Let the layer Ψ be the set of neurons in the output layer, and y_i be the prediction for class i.

$$y_i(X) = \frac{e^{X_i}}{\sum_{\forall j \in \Psi} e^{X_j}} \tag{2}$$

Eq. (2) forces the set of outputs of Ψ to sum to 1, resembling a probability. The proper weights $\vec{w}$ are obtained by minimizing the following loss function (Girosi et al., 1995; Bishop, 1995; Nielsen, 2015):

$$L(\vec{w}) = \underbrace{-\log y_d(X)}_{Cost} + \underbrace{\frac{1}{2}\lambda \sum_{\forall w_i \in \vec{w}} w_i^2}_{Regularization} \tag{3}$$

The first term is the cross-entropy cost function, where y_d is the pseudo-probability for the correct class. The second term is a regularization term. It prevents overfitting the training data. λ is the regularization parameter. It can be shown that to minimize the loss Eq. (3) weights can be updated as follows:

$$\overrightarrow{w_{t+1}} = \overrightarrow{w_t} - \underbrace{\alpha \nabla L(\overrightarrow{w_t})}_{Gradient} - \underbrace{\beta\, \overrightarrow{\Delta w_{t-1}}}_{Momentum} \tag{4}$$

This equation is referred to as *stochastic gradient descent with momentum,* also known as the generalized delta rule. $\overrightarrow{w_{t+1}}$ is the updated weight. $\overrightarrow{w_t}$ is the current weight. In the gradient term, α is the learning rate, a small non-zero value. α is small so that a single sample cannot alter the model too much; it is useful for outliers that are not consistent with the current model. α decreases over time to allow the model to converge. $\nabla L(\overrightarrow{w_t})$ is the error gradient. The second term is called momentum. It can help the model overcome local minima by allowing the model to continue along the gradient. Otherwise, the model might find a suboptimal solution to Eq. (3) in a local minima. β is the momentum weight, typically much larger than α. $\overrightarrow{\Delta w_{t-1}}$ are the weight updates from the previous step. A more common weight update rule is *mini-batch stochastic gradient descent with momentum.* It has a more stable descent of the error gradient by splitting the training data into mini-batches. It calculates updates to the model on the set of samples in the mini-batch as a whole, then carries out the update to $\overrightarrow{w_{t+1}}$:

$$\overrightarrow{w_{t+1}} = \overrightarrow{w_t} - \frac{\alpha}{m}\sum_{i=1}^{m} \nabla L_i(\overrightarrow{w_t}) - \beta(\overrightarrow{\Delta w_{t-1}}) \tag{5}$$

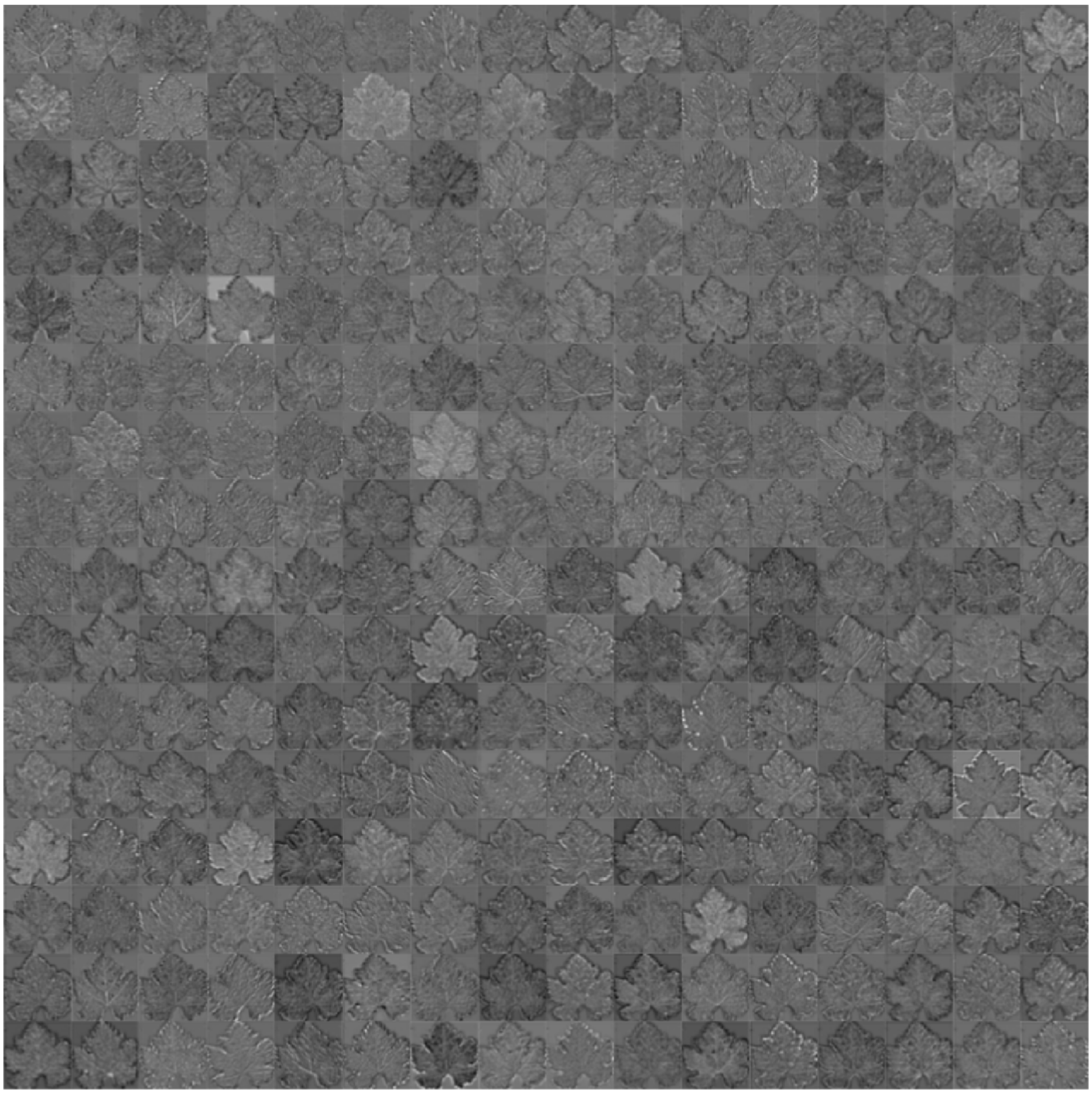


Fig. 6. Convolutional neural networks detect visual cues in an image. It does this by processing a given image with many filters. Each subfigure is the output of a single filter and highlights cues that the neural network determined to be important. The deep learning architecture AlexNet (Krizhevsky et al., 2012) was used to generate this figure.

where m is the number of samples in a mini-batch. Neural networks have thousands of weights, and the above equation must be carried out for each mini-batch.

To further reduce training time, computation of neural network weight updates is often carried out with GPU computing. Recent advances in computing power and the outstanding performance in the ImageNet object recognition challenge (Krizhevsky et al., 2012) have renewed the scientific community's interest in neural networks (LeCun et al., 2015). We focus on convolutional neural networks, a type of neural network that can complete recognition tasks previously thought to be too challenging. Generally, a convolutional neural network trains itself to filter an input to detect cues in an image. A convolutional layer accomplishes this by learning a bank of filters that are convolved with the input:

$$\left(h * k\right)(\vec{x}) = \sum_{\vec{j}=\left(0,0\right)}^{(m-1,n-1)} k\left(\vec{j}\right) h\left(\vec{j} + \vec{x}\right) + \theta \tag{6}$$

where h is the input image, k is a filter learned by the network. Convolution layers are followed by pooling layers which downsample the input stimulus. Though there is still much to be understood about the human visual system, convolutional neural networks resemble the process of image filtering with the human receptive field. An example of the output of convolutional layers are given in Fig. 6. These images provide examples of the cues used by the neural network.

2.2.2. *Deep learning architectures*

Six pre-trained convolutional neural networks are used in this work. The first is AlexNet (Krizhevsky et al., 2012), a feed-forward Convolutional Neural Network. AlexNet is trained to detect one thousand different objects from roughly one million images (Russakovsky et al., 2015). The second neural network architecture in this work is GoogLeNet, an improvement on deep convolutional neural networks developed by Google (He et al., 2015). It was the first departure from traditional convolutional neural networks that stacked many layers of convolution operations. GoogLeNet layers form a directed acyclic graph. It introduced a concept called *inception* modules which combine a set of convolution and pooling carried out in parallel. The operations are of varying sizes enabling it to detect cues of varying sizes. The third

Table 1
This table compares the different deep learning architectures used in this work according to the number of layer operations. Layer operations include cross channel normalization, pooling, concatenation, activation, dropout, etc. They approximate the complexity of the architecture. An architecture with more layer operations will take longer to train. It will also have more weights, requiring increasingly non-trivial hardware. AlexNet (Krizhevsky et al., 2012) and SqueezeNet (Iandola et al., 2016) represent the most "light weight" architectures considered in this work. Inception v3 (Szegedy et al., 2016) and ResNet-101(Szegedy et al., 2015) represent the most complex architectures.

Name	Reference	Type	Layer operations
AlexNet	Krizhevsky et al. (2012)	CNN	23
GoogLeNet	He et al. (2015)	Inception CNN	142
Inception v3	Szegedy et al. (2016)	Inception CNN	314
ResNet-50	Szegedy et al. (2015)	Residual CNN	175
ResNet-101	Szegedy et al. (2015)	Residual CNN	345
SqueezeNet	Iandola et al. (2016)	Residual CNN	66

neural network is an optimization of GoogLeNet, called Inception v3 (Szegedy et al., 2016). The fourth and fifth neural networks are ResNet-50 and ResNet-101 (Szegedy et al., 2015). Unlike other networks, ResNet-50 has *residual* components. The networks convolution-pooling operations are organized into sets. A set of convolution-pooling operations supplements its normal output with stimulus supplied to the beginning of the set bypassing the sets filters. ResNet-101 is a more complex improvement upon ResNet-50. The last neural network architecture is SqueezeNet (Iandola et al., 2016), a residual network without inception components. Like Inception v3, SqueezeNet is an optimization. However, it is applied to the ImageNet challenge with the goal of increasing overall accuracy while minimizing the costliness of training a network. For transfer learning, we reuse the parameters of the pre-trained network except for the last classification layers. They are replaced with a fully connected layer and a softmax layer. For a complete explanation of neural network architectures, the reader is referred to Krizhevsky et al. (2012), He et al. (2015) and Szegedy et al. (2015), Iandola et al. (2016) and Szegedy et al. (2016). Layer operations can be used to approximate the relative complexity of the network, though Table 1 does not consider the layer inter-connections.

2.2.3. *Parameters and cross-validation*

Experiment 1: For the first set of experiments, six pre-trained neural networks are obtained. The reader is referred to the respective literature for the training procedures for those networks (Krizhevsky et al., 2012; He et al., 2015; Szegedy et al., 2015; Iandola et al., 2016; Szegedy et al., 2016). After obtaining the networks, they are retrained on the augmented dataset of images we collected. We use fine-tuning (Li and Hoiem, 2017). That is, all network layers are frozen except for the final fully connected layer. Retraining parameters are as follows: minibatch size is 25, initial learning rate α is 0.001, L2 regularization λ is 0.0001. We use mini-batch stochastic gradient descent for the optimizer. The weight learning rate and bias learning rate factor is increased by a factor 20. Retraining is limited to 5 epochs for all architectures. We use fivefold random cross validation. The size of each population, after data augmentation and image processing is given in Table 2.

For a single fold, 2680 images are selected randomly from each population, resulting in a data subset of 16,800 images. By ensuring that all populations have an equal number of samples, we avoid an apriori bias. With each fold, 70% of the images are used for training and the remaining 30% of the images are for testing. This is repeated five times to obtain the testing results given in the 'Results and Discussion' section. Experiments were coded in MATLAB 2018A and executed on a Dell Precision Rack 7910 with the following specifications: Intel Xeon E5-2630 v3 microprocessor operating at 2.40 GHz, 128 GB of DDR3 memory operating at 2133 MHz, Ubuntu 16.04 operating system, and a Nvidia Quadro K6000. Code is publicly available on GitHub.[2]

Experiment 2: With the second set of experiments, we compare a deep learning setup to one without deep learning. The convolutional neural network (deep learning) setup is described as follows: (1) the image is processed by a pre-trained convolutional neural network, AlexNet trained on ImageNet (Krizhevsky et al., 2012). (2) However, the image is not fully processed by AlexNet, as we harvest the activations of the convolutional part of the network. (3) The harvested activations are classified by a support vector machine (SVM) (Chang and Lin, 2011) for final prediction.

The system without deep learning, the baseline, is described as follows: (1) Local binary patterns (Ojala et al., 1994) and color histogram features are extracted from the image. (2) The local binary patterns and color histogram features are classified with a SVM for final prediction.

For both systems, the SVM has the following parameters: a C-SVM is used; C is set to 1; a radial basis function kernel is used; and gamma is set to the inverse of the number of pixels in the image. The baseline system we compare our work to performs so poorly on the dataset used in Experiment 1, that we had to reduce the complexity of the problem. We reduce the complexity by reducing the problem to a binary classification problem (GY or non-GY); not including data from publicly available sources that would introduce more disease populations; and augmenting the data with image processing that would perturb the data with misalignment. GY is considered positive (P) and non-GY is considered negative (N). Only GY and non-GY leaf clipping images that we collected are used for this experiment (from Section 2.1.1).

Experiment 3: While the overall goal of our work is to develop a ML system, we would also like to know how well a human compares to the performance of our diagnostic test. We conduct a test where human participants are presented with at least 100 images of grape leaves, presented one at a time. For each leaf, the participants guess if the leaf is positive for grapevine yellows (GY), or negative for GY. There were 7 trained novices and 5 experts for this study. Participants processed 149.08 ± 45.67 images. For this purpose a web-based software were developed to collect participants responces (Fig. 7).

Prior to beginning any data collection, informed consent was obtained from each volunteer participant.[3] Participants received training to detect GY from sight. First, the participants spent 30 min reading about GY and identification guide. Second, participants received an additional 30 min of training where they were presented with trial images. Participants diagnosed image-by-image at their own pace and received immediate feedback about their decisions. Their guesses were not logged at this point.

After 60 min of training, the participants were given a set of testing images. Participants made a single guess for each image, at their own pace. They received no feedback on their responses. The responses to the testing images were collected. This experiment used the same dataset as experiment 2. Participants who were already experts in plant science were not required to receive training.

2.3. *Evaluation metrics*

In the following, we explain the metrics to assess overall performance. In general, metrics are a rate and higher is better. Sensitivity, also known as true positive rate, is calculated as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} \tag{7}$$

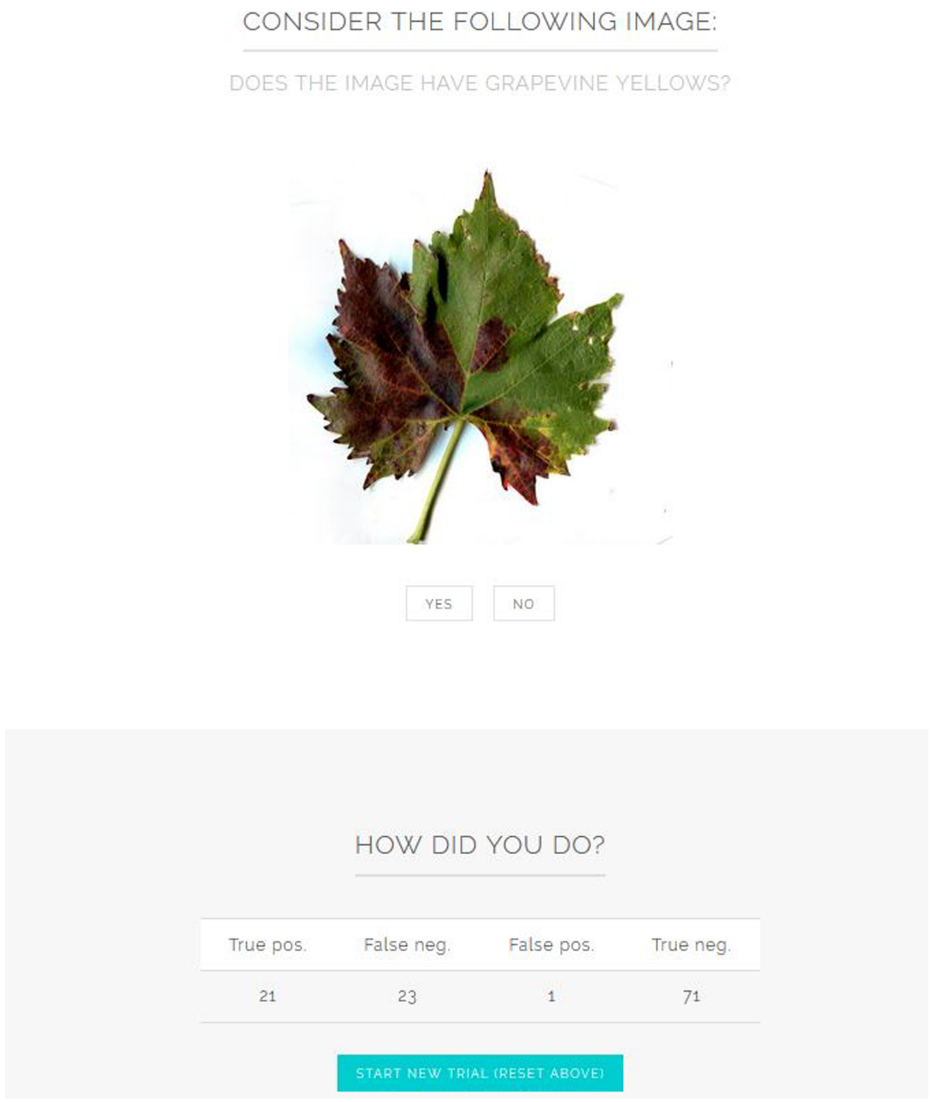
where TP is the number of true positives and P is the number of positive samples. Specificity, also known as true negative rate is calculated as follows:

[2] https://github.com/DrAlbertCruz/MATLAB-Deep-Learning.

[3] Human subject testing authorized (CSUB IRB #19-75).

Table 2
Number of samples for each population after data augmentation. On each fold, 2680 samples are randomly selected from each population.

Black Rot (BR)	Healthy Control (HC)	Esca Disease (ED)	Grapevine Yellow (GY)	Leaf Blight (LB)	Other (OD)
9440	8924	11,444	2680	8576	4250


CONSIDER THE FOLLOWING IMAGE:

DOES THE IMAGE HAVE GRAPEVINE YELLOWS?

HOW DID YOU DO?

True pos.	False neg.	False pos.	True neg.
21	23	1	71

Fig. 7. Human diagnostic test; web-based software to collect human participant responces and to evaluate their performance on detecting GY disease, based on leaf symptoms.

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} \tag{8}$$

where TN is the number of true negatives and N is the number of negative samples. Precision, also known as positive predictive value (PPV) is calculated as follows:

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{9}$$

where FP is the number of false positives. Negative predictive value (NPV) is calculated as follows:

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN+FN}} \tag{10}$$

where FN is the number of false negatives. False negative rate (FNR) and false positive rate (FPR) are calculated as follows:

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{P}} \tag{11}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} \tag{12}$$

Accuracy is calculated as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{P+N}} \tag{13}$$

It is often used as a measure of overall performance. Another metric for performance is the F1 score:

$$\mathrm{F1}=2\left(\frac{\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV+TPR}}\right) \tag{14}$$

It is the harmonic mean of the precision and sensitivity. It provides an alternative to accuracy, which does not account for false positives.

Table 3
A summary of results for automatic prediction of grapevine yellows (GY) from leaf clipping images of *Vitis vinifera* L. cv. Sangiovese with six different deep learning architectures. Higher is better for all metrics except FNR and FPR. Results are presented as average across five trials with variance, given in parenthesis in terms of 10^{-3}. A summary of the best and worst methods for each metric are given in the bottom two rows. ResNet-101 is the clear best performer across all deep learning architectures and has the least variance of results. The worst performers are AlexNet and SqueezeNet. TPR: true positive rate. TNR: true negative rate. PPV: positive predictive value. NPV: negative predictive value. FNR: false negative rate. FPR: false positive rate. ACC: accuracy. F1: F1-score. MCC: Matthew's correlation coefficient.

	TPR	TNR	PPV	NPV	FNR	FPR	ACC	F1	MCC
AlexNet	0.9754 (0.6725)	0.9765 (0.1952)	0.8962 (2.9528)	0.9951 (0.0267)	0.0246 (0.6725)	0.0235 (0.1952)	0.9763 (0.0732)	0.9328 (0.4632)	0.9207 (0.5934)
GoogLeNet	0.8512 (4.5200)	0.9861 (0.0893)	0.9280 (1.5836)	0.9709 (0.1636)	0.1488 (4.5200)	0.0139 (0.0893)	0.9636 (0.0287)	0.8855 (0.4590)	0.8668 (0.5173)
Inception v3	0.9363 (3.3712)	0.9939 (0.0142)	0.9692 (0.2981)	0.9875 (0.1263)	0.0637 (3.3712)	0.0061 (0.0142)	0.9843 (0.0601)	0.9514 (0.6831)	0.9430 (0.8486)
ResNet-50	0.9851 (0.2406)	0.9932 (0.0071)	0.9668 (0.1541)	0.9970 (0.0096)	0.0149 (0.2406)	0.0068 (0.0071)	0.9918 (0.0035)	0.9757 (0.0318)	0.9710 (0.0461)
ResNet-101	0.9896 (0.0616)	0.9940 (0.0050)	0.9709 (0.1078)	0.9979 (0.0025)	0.0104 (0.0616)	0.0060 (0.0050)	0.9933 (0.0014)	0.9801 (0.0121)	0.9761 (0.0170)
SqueezeNet	0.7134 (52.0459)	0.9825 (0.1769)	0.9082 (3.7837)	0.9466 (1.4966)	0.2866 (52.0459)	0.0175 (0.1769)	0.9377 (0.8328)	0.7723 (26.4523)	0.7603 (16.9321)
Best method	ResNet-101	ResNet-101	ResNet-101	ResNet-101	ResNet-101	ResNet-101	ResNet-101	ResNet-101	ResNet-101
Worst method	SqueezeNet	AlexNet	AlexNet	SqueezeNet	SqueezeNet	AlexNet	SqueezeNet	SqueezeNet	SqueezeNet

Though, it does not account for false negatives. A further alternative is the Matthew's Correlation Coefficient (MCC):

$$\mathrm{MCC} = \frac{\mathrm{TP}{\times}\mathrm{TN} - \mathrm{FP}{\times}\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{15}$$

It accounts for both false positives and false negatives. It is like the chi-square statistic for the confusion matrix. MCC is the only metric that is not a rate; $\mathrm{MCC} \in [-1, 1]$ and higher is better. Some results are presented in terms of a confusion matrix, also known as a contingency table. Cells of a confusion matrix are calculated as:

$$C_{ij} = \frac{1}{n_i}\sum_{k=0}^{n_i-1} [\mathrm{Prediction}(k) = i \cap \mathrm{Label}(k) = j] \tag{16}$$

where i is the row (a population label), j is the column (also a population label), k is an iterator (an index of the current test sample), n_i is the number of test samples belonging to population i, $\mathrm{Prediction}(k)$ references the k-th prediction, $\mathrm{Label}(k)$ references the true label of the k-th test sample. [.] is an Iverson bracket that evaluates to the number 1 if the premise of the argument is true:

$$[X] = \begin{cases} 1 & X \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

For this work we enumerate the population labels as follows: black rot (BR) is 0, healthy control (HC) is 1, Esca disease (ED) is 2, grapevine yellows (GY) is 3, leaf blight (LB) is 4, all other diseases and pests (OD) are 5. The rows of the confusion matrix represent the samples of a population. The columns represent how each sample was predicted. An ideal system would have an identity matrix as a confusion matrix, that is, the non-diagonal elements should be zero. The diagonal elements of a confusion matrix correspond to the sensitivity for recognition of a certain population. Confusion matrix elements are a ratio.

For all experiments, many trials are carried out. We present the results of each experiment with the performance metrics in Eqs. (7)–(16), given as the average and variance across all trials. Variance is defined as follows:

$$\mathrm{Var}(\vec{m}) = E\left(\vec{m}^{2}\right) - E(\vec{m})^{2} \tag{18}$$

where $\vec{m}$ is a vector of the metric values and $E(.)$ is the expectation, or average of the metric values.

To discuss performance, we consider two more metrics:

(1) Sigma levels, to determine if an architecture is sufficient. Sigma levels are often used in manufacturing to gauge the performance of a process. We use the 68-95-99.7 rule of a normal distribution to establish one, two and three sigma levels of performance without error. That is, one sigma corresponds to an expectation success without error 68% of the time; two sigma, 95%; and three sigma, 99.7%. Generally, a process less than two sigma performance is not desirable and should not be used in production.
(2) Unpaired t-tests to determine if there are significant differences between two architectures. Because there are few samples for each population (from five validation folds) we use pooled variance. Unpaired t-tests to determine P value and statistical significance, a P value of less than 0.05 is generally considered statistically significant.

3. Results and discussion

3.1. Experiment 1 – Feasibility of GY detection by computers

The first set of experiments answer the questions: *can GY be predicted from leaf clipping images?* And, *among the many deep learning architectures available, which is best?* We answer these questions by providing recognition results for six different network architectures. They detect the following diseases from a leaf clipping images of grapes: healthy control (HC), GY, black rot (BR), esca (ED), leaf blight (LB), and other grapevine diseases (OD). These experiments demonstrate state-of-the-art performance when attempting to automatically detect GY. In this experiment, recognizing the symptoms of GY is a complex problem because detection must occur in spite of other diseases that are similar in appearance. As a further challenge, the alignment of the images (where the leaf is placed on the image canvas) was perturbed during data augmentation with image processing. Finally, the images have varying background conditions. Despite this, the system achieves outstanding performance. Some architectures begin to meet expectations for a production level system (two sigma levels of performance). A summary of results for GY detection is given in Table 3 and a full presentation of results for all populations is given in Table 4.

The best sensitivity, also known as true positive rate (TPR), is given by ResNet-101 with a TPR of 98.96% and the worst sensitivity is given by SqueezeNet with a TPR of 71.34%. All but two architectures have at least two sigma TPR and those are GoogLeNet and SqueezeNet. When considering a diagnostic test, one should consider sensitivity, specificity and predictive value, so it would be premature to dismiss GoogLeNet

Table 4
In-depth results for automatic prediction of GY in *Vitis vinifera* L. cv. Sangiovese by various deep learning algorithms. The results are given in terms of confusion matrixes measuring performance for prediction of non-GY control (HC), GY, black rot, esca, leaf blight, and other diseases/pests. Results are presented as average across five trials with variance, given in parenthesis in terms of $10^{-3\ <!--Queryid="Q7"desc="PleasecheckthelayoutofTable4,andcorrectifnecessary."/-->}$.

(A) AlexNet

	BR	HC	ED	GY	LB	OD
Black Rot (BR)	0.9192 (0.6459)	0.0010 (0.0026)	0.0699 (0.7746)	0.0000 (0.0000)	0.0095 (0.0616)	0.0005 (0.0005)
Healthy Control (HC)	0.0025 (0.0023)	0.9363 (0.0351)	0.0002 (0.0003)	0.0493 (0.1784)	0.0000 (0.0000)	0.0117 (0.0917)
Esca Disease (ED)	0.0816 (2.4060)	0.0000 (0.0000)	0.9010 (2.6876)	0.0042 (0.0299)	0.0070 (0.0090)	0.0062 (0.0232)
Grape. Yellow (GY)	0.0000 (0.0000)	0.0032 (0.0152)	0.0000 (0.0000)	0.9754 (0.6725)	0.0000 (0.0000)	0.0214 (0.6091)
Leaf Blight (LB)	0.0124 (0.0812)	0.0017 (0.0067)	0.0097 (0.0560)	0.0000 (0.0000)	0.9759 (0.2882)	0.0002 (0.0003)
Other (OD)	0.0002 (0.0003)	0.0077 (0.0846)	0.0020 (0.0028)	0.0639 (3.1501)	0.0000 (0.0000)	0.9261 (3.5678)

(B) GoogLeNet

	BR	HC	ED	GY	LB	OD
Black Rot (BR)	0.9035 (2.0843)	0.0015 (0.0026)	0.0930 (2.1460)	0.0000 (0.0000)	0.0015 (0.0026)	0.0005 (0.0012)
Healthy Control (HC)	0.0015 (0.0026)	0.9241 (2.4953)	0.0007 (0.0028)	0.0303 (0.3485)	0.0035 (0.0142)	0.0398 (1.0357)
Esca Disease (ED)	0.0716 (2.9591)	0.0007 (0.0012)	0.9139 (3.7239)	0.0015 (0.0073)	0.0067 (0.0067)	0.0055 (0.0469)
Grape. Yellow (GY)	0.0002 (0.0003)	0.0363 (1.3470)	0.0060 (0.0243)	0.8512 (4.5200)	0.0002 (0.0003)	0.1060 (5.5091)
Leaf Blight (LB)	0.0012 (0.0015)	0.0010 (0.0026)	0.0052 (0.0212)	0.0000 (0.0000)	0.9925 (0.0410)	0.0000 (0.0000)
Other (OD)	0.0017 (0.0074)	0.0157 (0.4236)	0.0087 (0.0843)	0.0378 (1.5699)	0.0007 (0.0005)	0.9353 (3.1102)

(C) Inception v3

	BR	HC	ED	GY	LB	OD
Black Rot (BR)	0.9607 (0.6061)	0.0010 (0.0026)	0.0366 (0.5481)	0.0002 (0.0003)	0.0015 (0.0019)	0.0000 (0.0000)
Healthy Control (HC)	0.0000 (0.0000)	0.9560 (0.6633)	0.0007 (0.0012)	0.0194 (0.0840)	0.0002 (0.0003)	0.0236 (0.6644)
Esca Disease (ED)	0.0114 (0.1233)	0.0002 (0.0003)	0.9826 (0.1400)	0.0000 (0.0000)	0.0020 (0.0020)	0.0037 (0.0147)
Grape. Yellow (GY)	0.0000 (0.0000)	0.0104 (0.0670)	0.0002 (0.0003)	0.9363 (3.3712)	0.0000 (0.0000)	0.0530 (3.7952)
Leaf Blight (LB)	0.0002 (0.0003)	0.0005 (0.0005)	0.0030 (0.0360)	0.0002 (0.0003)	0.9960 (0.0382)	0.0000 (0.0000)
Other (OD)	0.0000 (0.0000)	0.0042 (0.0654)	0.0030 (0.0446)	0.0107 (0.3168)	0.0007 (0.0028)	0.9813 (1.0365)

(D) ResNet-50

	BR	HC	ED	GY	LB	OD
Black Rot (BR)	0.9779 (0.4435)	0.0007 (0.0012)	0.0192 (0.4607)	0.0002 (0.0003)	0.0020 (0.0105)	0.0000 (0.0000)
Healthy Control (HC)	0.0000 (0.0000)	0.9784 (0.0886)	0.0000 (0.0000)	0.0209 (0.0846)	0.0000 (0.0000)	0.0007 (0.0012)
Esca Disease (ED)	0.0214 (0.5967)	0.0007 (0.0028)	0.9716 (1.2333)	0.0007 (0.0012)	0.0055 (0.1180)	0.0000 (0.0000)
Grape. Yellow (GY)	0.0000 (0.0000)	0.0082 (0.0979)	0.0002 (0.0003)	0.9851 (0.2406)	0.0000 (0.0000)	0.0065 (0.0854)
Leaf Blight (LB)	0.0002 (0.0003)	0.0010 (0.0003)	0.0005 (0.0012)	0.0002 (0.0003)	0.9980 (0.0028)	0.0000 (0.0000)
Other (OD)	0.0000 (0.0000)	0.0045 (0.0538)	0.0005 (0.0005)	0.0119 (0.2000)	0.0000 (0.0000)	0.9831 (0.4615)

(E) ResNet-101

	BR	GY	ED	GY	LB	OD
Black Rot (BR)	0.9861 (0.1071)	0.0007 (0.0012)	0.0122 (0.0738)	0.0000 (0.0000)	0.0010 (0.0026)	0.0000 (0.0000)
Healthy Control (HC)	0.0000 (0.0000)	0.9741 (0.3136)	0.0000 (0.0000)	0.0192 (0.2008)	0.0002 (0.0003)	0.0065 (0.0320)
Esca Disease (ED)	0.0182 (0.1474)	0.0000 (0.0000)	0.9786 (0.1821)	0.0010 (0.0011)	0.0015 (0.0019)	0.0007 (0.0012)
Grape. Yellow (GY)	0.0000 (0.0000)	0.0182 (0.7608)	0.0005 (0.0012)	0.9704 (0.5828)	0.0000 (0.00000)	0.0109 (0.0483)
Leaf Blight (LB)	0.0000 (0.0000)	0.0007 (0.0012)	0.0002 (0.0003)	0.0000 (0.0000)	0.9990 (0.0019)	0.0000 (0.0000)
Other (OD)	0.0000 (0.0000)	0.0025 (0.0240)	0.0000 (0.0000)	0.0022 (0.0034)	0.0002 (0.0003)	0.9950 (0.0317)

(F) SqueezeNet

	BR	GY	ED	GY	LB	OD
Black Rot (BR)	0.8731 (11.2776)	0.0037 (0.0248)	0.1010 (6.9943)	0.0000 (0.0000)	0.0206 (0.6502)	0.0015 (0.0073)
Healthy Control (HC)	0.0030 (0.0221)	0.9112 (2.6636)	0.0017 (0.0074)	0.0266 (0.4166)	0.0065 (0.0459)	0.0510 (2.1449)
Esca Disease (ED)	0.1734 (92.793)	0.0020 (0.0198)	0.7910 (96.831)	0.0020 (0.0043)	0.0177 (0.1651)	0.0139 (0.1728)
Grape. Yellow (GY)	0.0017 (0.0105)	0.0562 (2.4778)	0.0117 (0.5179)	0.7134 (52.045)	0.0045 (0.0376)	0.2124 (62.669)
Leaf Blight (LB)	0.0114 (0.1202)	0.0072 (0.0312)	0.0092 (0.1753)	0.0000 (0.0000)	0.9687 (0.9927)	0.0035 (0.0606)
Other (OD)	0.0065 (0.1279)	0.0527 (4.8704)	0.0199 (1.6298)	0.0587 (2.8762)	0.0102 (0.1171)	0.8520 (18.589)

and SqueezeNet just from poor TPR performance. Comparing the best and second-best performers (ResNet-50) in terms of TPR, an unpaired t-test of the results yields a $t = 40.51$ with a P-value of less than 0.0001. Thus, ResNet-101 alone has the best sensitivity. Comparing the best and worst performers in terms of TPR, an unpaired t-test of the results yields a $t = 11.87$ with a P-value of less than 0.0001. Tests that find a statistical significance between the best and worst performers support the idea that there is a difference between architectures, and that it is important to consider which neural network to use.

The best specificity, also known as true negative rate (TNR), is given by ResNet-101 with a TNR of 99.40% and the worst specificity is given by AlexNet with a TNR of 97.65%. All architectures have at least two sigma TNR, so all architectures have sufficient specificity. Comparing the best and second-best performers (Inception V3) in terms of TNR, an unpaired t-test of the results yields a $t = 14.85$ with a P-value of less than 0.0001. Thus, ResNet-101 has the best specificity. Comparing the

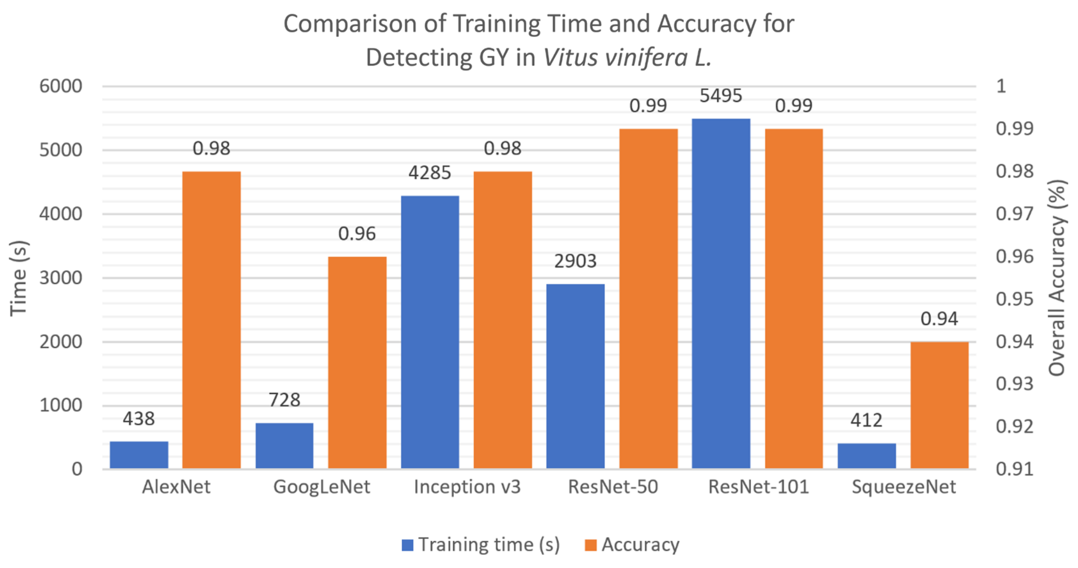


Fig. 8. Results for automatic prediction of GY in *Vitis vinifera* L. cv. Sangiovese by various deep learning algorithms, in terms of accuracy and training time. The blue column represents the average time to fine-tune the model across five folds. The orange column represents the average accuracy over five folds. AlexNet and SqueezeNet take the least amount of time to train, though only AlexNet has competitive accuracy. ResNet-50 and ResNet-101 have comparable accuracies. But, considering the training time, ResNet-50 may be the best option.

best and worst performers in terms of TNR, an unpaired t-test of the results yields a $t = 740.50$ with a P-value of less than 0.0001.

The best positive predictive value (PPV) is given by ResNet-101 with a PPV of 97.09% and the worst PPV is given by AlexNet with 89.62%. Only three architectures achieve at least two sigma PPV, the two variations of ResNet and Inception v3. PPV is better than TPR or TNR when indicating the ability of the system to detect a disease. Thus, the two variations of ResNet and Inception v3 are the best contenders thus far. ResNet-101 has only slightly higher PPV than the other two and there would appear to be no difference. But, an unpaired t-test of the of the best and second-best architectures according to PPV results yields a $t = 11.99$ with a P-value of less than 0.0001, indicating a significant difference. ResNet-101 has the best predictive value. Comparing the best and worst performers in terms of PPV, an unpaired t-test of the results yields a $t = 56.53$ with a P-value of less than 0.0001.

Negative predictive value (NPV) is the ability of a diagnostic test to determine that a sample does not have GY. The best NPV is given by ResNet-101 with a NPV of 99.79% and the worst NPV is given by SqueezeNet with a NPV of 94.66%. SqueezeNet is the only architecture that does not achieve at least two sigma NPV, and both versions of ResNet achieve three sigma NPV. Considering previous discussion of TPR, and PPV, it is easier for a deep learning algorithm to reject a GY diagnosis than it is to conclude that a sample has GY. Considering the two different ResNet variations, an unpaired t-test of the results yields $t = 0$ and a P-value of greater than 0.99999, thus there is no significant difference between the two architectures. Surprisingly, AlexNet has third-best NPV. The t-test conclusion is the same when comparing ResNet-101 to AlexNet, with a P-value of greater than 0.99999. This is surprising because AlexNet was one of the seminal methods that demonstrated the potential of convolutional neural networks in 2012. One would assume that more recent architectures would perform better, yet for NPV AlexNet performs better than GoogLeNet and Inception v3. Comparing the best and worst performers in terms of NPV, an unpaired t-test of the results yields a $t = 73.73$ with a P-value of less than 0.0001.

False negative rate (FNR) can be calculated as: 1 – TPR. False positive rate (FPR) can be calculated: as 1 – TNR. We do not include a discussion on these metrics because they are linear functions of other metrics and would not entail conclusions different from analysis of TPR and TNR.

The best overall accuracy (ACC) is 99.33%, obtained by using ResNet-101 followed by ResNet-50, Inception v3, AlexNet, GoogLeNet and SqueezeNet in that order. ResNet-50 is a close contender, with an accuracy of 99.18%. An unpaired t-test of the two results yields $t = 0$ and a P value of greater than greater than 0.99999, so there is no significant difference. The next-best architecture to be statistically different from ResNet-50 is AlexNet, with an accuracy of 97.63%, $t = 268.79$ and a P value of less than 0.00001. AlexNet continues to surprise as a strong contender despite its lack of contemporary layer structures (residual and inception). All architectures achieve two sigma levels of accuracy except for SqueezeNet, and none are greater than two sigma performance. A comparison of best and worst (SqueezeNet) architectures according to accuracy with an unpaired t-test yields $t = 146.52$ and a P value of less than 0.00001.

The best F1-score is 98.01% obtained by ResNet-101, followed by ResNet-50, Inception v3, AlexNet, GoogLeNet and SqueezeNet. The next-best architecture to be statistically different from ResNet-101 is Inception v3 with an F1-score of 95.14%, $t = 90.75$ and a P value of less than 0.00001. Only ResNet-50, ResNet-100 and Inception v3 architectures achieve two sigma levels of F1-score. Comparing best and worst performer (SqueezeNet) with an unpaired t-test yields $t = 17.57$ and a P value of less than 0.0001. There is a significant gap between the best and worst performer. MCC is a similar metric to F1 in that it tries to balance false positives and false negatives. Because of the similarity, the conclusions would be like analysis of F1 results. MCC is not a rate so sigma levels are not applicable.

The most surprising conclusion of Table 3 is that AlexNet has competitive performance. Also, it was surprising that SqueezeNet did not do well, and it was often the worst architecture. Further, SqueezeNet has the worst variance among the various metrics. SqueezeNet was an optimization of a residual convolutional neural network with the goal of achieving the highest overall accuracy on the ImageNet grand challenge while minimizing the complexity of the network. This may explain why it does not do as well for leaf clipping images. The ResNet architectures are the only networks to achieve at least two sigma performance for all relevant metrics, and the only networks we can recommend for production-grade systems for automatic detection of GY.

Considering GY performance only, ResNet-101 is the best architecture for automatic detection. However, for NPV, accuracy and F1-

score, there is no significant difference between ResNet-50 and ResNet-101. A comparison of training time and accuracy for detecting GY is given in Fig. 8. ResNet-101 and Inception v3 take the longest time to train. Their performance may be a diminishing return considering that ResNet-50 takes half as long to train. Considering this, ResNet-50 is a good alternative to ResNet-101. It is notable that AlexNet and SqueezeNet have a similar training time. A *t*-test yields a *P* value of 0.05, so it is not quite statistically significant. However, during the training procedure, all but the final fully connected layer is frozen. It is possible that SqueezeNet may be faster if the networks were not frozen.

Table 3 contains results for only GY detection, whereas Table 4 contains confusion matrixes that describe the performance across all six populations. Considering the confusion matrixes, AlexNet has the best sensitivity when predicting leaf blight, GY, healthy controls, other diseases, black rot then esca in that order. The greatest variance is given by the other diseases population and this is not surprising; the other disease population consists of many diseases with varying symptoms. AlexNet achieves two sigma level sensitivity for leaf blight and GY. It achieves one sigma level of sensitivity for all other populations.

GoogLeNet has best sensitivity when predicting leaf blight, other diseases, healthy controls, esca disease, black rot and GY in that order. In general, GoogLeNet has a high variance of sensitivity for all populations, and leaf blight is the only population with two sigma levels of sensitivity. GoogLeNet is not as versatile and reliable as AlexNet when identifying diseases from leaf clipping images.

Inception v3 has best sensitivity when detecting of leaf blight, esca disease, other diseases, black rot, healthy controls, GY in that order. Comparing Inception v3 to GoogLeNet, variance of sensitivity is greatly reduced. GoogLeNet achieves two sigma levels of sensitivity on all populations, except for the GY population—the primary focus of our work.

ResNet-50 has best sensitivity when detecting leaf blight, GY, other diseases, healthy controls, black rot, and esca disease in that order. Generally, ResNet-50 has low variance except for esca disease, its worst performing population. ResNet-50 achieves two sigma levels of performance for all populations, and three sigma levels for leaf blight.

ResNet-101 has best sensitivity for detecting leaf blight, other diseases, black rot, esca disease, healthy controls and GY in that order. It achieves two sigma levels of sensitivity for all populations except for leaf blight where it achieves three sigma levels of sensitivity. The variance is generally low for all populations.

SqueezeNet has best sensitivity with leaf blight, healthy controls, black rot, other diseases, esca disease and GY in that order. SqueezeNet has the worst performance of all the networks when applied to our task. For GY, esca, other diseases and black rot, it has a variance that is a whole order of magnitude greater than the variances of other networks. SqueezeNet achieves two sigma levels of performance for only the leaf blight population.

Considering the confusion matrixes, leaf blight appears to be the easiest disease to detect for deep learning architectures. SqueezeNet, the worst performing architecture in our experiments, does not perform well in general, except for leaf blight recognition where it still achieves two sigma levels of sensitivity. Some architectures (such as AlexNet, ResNet-50 and ResNet-101) show promising results for a variety of diseases. Further, the highly complex architectures (Inception v3, ResNet-50, ResNet-101) do well for the other diseases population that contains many diseases with different symptoms. Considering that AlexNet is the least complex of all the architectures discussed in this work, it is possible that complexity increases the ability of a network to account for varying cues within a population. These results conclusively demonstrate that a neural network can detect GY from leaf clipping images. While the accuracy of traditional detection methods, such as PCR, is not disputed, a system using one of these neural network architectures can offer an early screen method that is used in parallel with traditional lab testing. This will accelerate responses and mitigate crop loses.

3.2. Experiment 2 – Is deep learning necessary?

For the second set of experiments, we address the question, *how does the proposed deep learning system compare to related work that does not use deep learning?* Occam's razor suggests that the best solution is the simplest one. Neural networks are difficult to implement because they are time consuming to train and they introduce the complications of data augmentation and transfer learning. Deep learning algorithms require expensive GPUs to train and there is no benefit to using deep learning if a more conventional system would suffice. We investigate if color and texture features with a support vector machine are enough to detect GY from grapevine leaf clipping images.

The baseline is similar to the approach in Kaur et al. (2018) and Sharif et al. (2018). Poor performance of the baseline system demonstrates the need for a more complex classification scheme, such as deep learning. For this experiment, the deep learning-based system takes the activations from the convolutional layer of the AlexNet architecture and pipes the activations to a support vector machine for final prediction. This also demonstrates the need for convolutional neural networks, over non-convolutional neural networks. Deep learning obtains a 92.06% overall accuracy and a Matthew's correlation coefficient of 0.832. The baseline system with local binary patterns (LBP) and color histogram with a SVM obtains only 26.79% overall accuracy and -0.1244 respectively (Table 5). The gap in performance is so great that there is no need for statistical analysis. Results indicates that the problem is indeed complex enough to outweigh the drawbacks of deep learning.

3.3. Experiment 3 – Comparison to human experts

Experiment 3 addressees the question, *among human recognition, deep learning and a baseline system without deep learning, which is better?* Results are given in Table 5. The results in this table use the data subset from experiment 2, but not the same data set from experiment 1. Experiments 2 and 3 can be compared, but they both cannot be compared to experiment 1. Deep learning is better than humans for all metrics. There is a such a large gap in performance between humans, the baseline system and deep learning that an in-depth statistical analysis is not needed. It is obvious that there is a significant difference. Human recognition does not meet the minimum expectations for a production level system (two sigma levels of performance) and supports that adage that it is hard to recognize GY from sight. Human results are in line with ML results in that specificity (TNR) and NPV are higher than sensitivity (TPR) and PPV, indicating that it is easier to reject a hypothesis of membership to GY than it is to confirm a hypothesis of membership to GY.

3.4. General discussion

GY is a serious threat due to severe symptoms and lack of healing treatments. Methods for effective detection of Grapevine Yellow diseases are of worldwide interest. Bois noir (BN) is endemic to some regions but still dangerous. Both pathogens cause similar symptoms and health monitoring programs are carried out worldwide. Diagnosis relies on effective symptom identification. Yet, a low concentration of the pathogen and its erratic distribution in the host leads to infected but asymptomatic grapes. This leads to high rates of false-negatives in detection. Further, GY symptoms such as leaf discoloration, bunch drying, and irregular wood ripening are typical among other diseases and outstanding in the late summer. This makes recognition of GY a difficult task (Belli et al., 2010). The aim of this work is to develop a tool for supporting sampling procedures.

The accuracy of lab testing is unparalleled. Yet, this tool can accelerate a response to GY when used in parallel with traditional detection methods (e.g. polymerase chain reaction (PCR), fluorescence *in-situ* hybridization (FISH), immunofluorescence (IF), enzyme-linked

Table 5
Results comparing the performance of a system with deep learning (AlexNet) with a baseline system that does not use deep learning and humans. (Top row) Deep learning, a pre-trained convolutional neural network (CNN) features with a linear support vector machine (SVM) classifier. (Middle row) Non-deep learning, a baseline system using local binary patterns (LBP) and color histograms with a SVM classifier that is similar to the approaches given in related work. (Bottom row) Human recognition on the same dataset. The deep learning algorithm achieves good performance when predicting if a sample has GY, whereas a system without deep learning performs poorly. Without deep learning, a ML algorithm cannot exceed human performance. This demonstrates the need for deep learning when automatically diagnosing the symptoms of GY from leaf clipping images. Variance is given in terms of 10^{-1}.

System	TPR	TNR	PPV	NPV	FNR	FPR	ACC	F1	MCC
System with deep learning	0.9523 (0.0063)	0.9047 (0.0036)	0.8333 (0.1200)	0.9743 (0.0042)	0.0476 (0.0063)	0.0952 (0.0036)	0.9206 (0.0039)	0.8888 (0.0201)	0.8320 (0.0270)
Baseline system without deep learning	0.4019 (0.1390)	0.2009 (0.0059)	0.4736 (0.1570)	0.4019 (0.0680)	0.5980 (0.1390)	0.2500 (0.0059)	0.2679 (0.0089)	0.4208 (0.0580)	−0.1244 (0.1900)
Human performance	0.5469 (0.1155)	0.8676 (0.1738)	0.7335 (0.3713)	0.7893 (0.0151)	0.4531 (0.1155)	0.1324 (0.1738)	0.7569 (0.0544)	0.6063 (0.0770)	0.4635 (0.2312)

immunosorbent assay (ELISA), flow cytometry (FCM), gas chromatography-mass spectrometry (GC–MS) *etc.* (Mehle et al., 2017)). The benefits of the proposed system are as follows:

- **The end-user does not need to be an expert at detecting GY.** Prior skill of the use is not required as well. There is no manual segmentation, initialization, or initial guesses. The user submits a cropped leaf clipping image to the system and receives diagnosis.
- **Expensive sensing equipment is not required.** The system accepts images of roughly 300 pixel resolution. Thus, consumer-grade video cameras, camcorders and scanners are enough. Expensive sensor equipment (thermal, multi-spectral) is not required.
- **The system does not need pre-processing of images.** Deep learning is end-to-end. We merely expect the user to center the leaf in the image. Further, the system can tolerate poor alignment during data collection.
- **The system is very accurate.** With ResNet-50, we obtain a sensitivity of 98.96% and a specificity of 99.40%.
- **It reports the chance of other diseases.** Black rot, downy mildew, esca disease, leaf blight, grapevine leafroll, powdery mildew and *Stictocephala bisonia*.

There is great potential to revolutionize the detection of GY. The system provides a template for swift detection of other crops and diseases. One major drawback to the system is the complexity of deep learning algorithms. However, training the system is a one-time procedure carried out on a server and not the end-user. Transfer learning and data augmentation mitigate the data requirements of deep learning. The results of experiments 2 and 3 show the need for deep learning despite this drawback. Systems without neural networks (deep learning) and human experts do not achieve the same level of performance as the proposed system.

A drawback of our work is that the leaf images are not taken in field conditions where there are other leaves, occlusion and varying illumination. Systems that operate in the field, under very unconstrained scenarios, are still an open challenge. Detection in very constrained scenarios is still very challenging, particularly for diseases that can be easily confused for others. The purpose of our study is to show that automatic detection is now possible with deep learning, which can be applied to other pests and diseases via transfer learning. When diagnosing a plant, growers collect various plant parts for conventional testing (such as PCR). Leaf blades would be collected through this process, so it is reasonable to require growers to clip the leaf blade before imaging.

4. Conclusion

In the field, true-positive rate detection of GY was frequently overestimated (due to similar symptoms) or underestimated (due to variability in symptom expression) (Rizzo et al., 2018). Thus, the automatic tools that can help the sampler in pathogen recognition are crucial to avoid missing GY-positive plants, in particular the FD-positive ones. We demonstrate that it possible to detect GY from leaf clipping images, and that certain systems exceed expectations. ResNet-101 is the clear best performer for this application. However, ResNet-50 is often second best and both systems are at least two sigma performance for all metrics. An in-depth statistical analysis revealed that, for NPV, accuracy and F1-score, there is no significant difference between ResNet-50 and ResNet-101. We recommend ResNet-50 for leaf diagnosis systems because the increased complexity of ResNet-101 maybe a diminishing return. It was notable that AlexNet has competitive performance because of its lack of sophistication compared to the other architectures in this work. Consistently, there is a statistically significant gap between the best and worst performer, indicating the importance of selecting an appropriate neural network for this application. Deep learning has 35.97% and 22.88% better predictive value (PPV) for recognizing GY from sight, than a baseline system without deep learning and trained humans respectively. Future work in this research will focus on the development of tools that can be used in the field where there are other leaves, occlusion and varying illumination. We anticipate the need for better segmentation algorithms to facilitate this. We will also consider implementing the methods on a Nvidia Jetson so that predictions can be made without need for a remote connection to a server.

Acknowledgements

Support for this work was provided by the California Consolidated Table Grape and Pest Control District, the California State University Program for Education & Research in Biotechnology and the National Institute of Food and Agriculture, U.S. Department of Agriculture (2016-38422-25544 and 2016-67032-25008). Statements made this paper do not reflect the position or policy of the U.S. government, or the states of California or Florida.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2018.12.028.

References

Abdulridha, J., Ampatzidis, Y., Ehsani, R., de Castro, A., 2018. Evaluating the performance of spectral features and multivariate analysis tools to detect laurel wilt disease and nutritional deficiency in avocado. Comput. Electron. Agric. 155, 203–2011.

Ali, H., Lali, M.I., Nawaz, M.Z., Sharif, M., Saleem, B.A., 2017. Symptom based automated detection of citrus diseases using color histogram and textural descriptors. Comput. Electron. Agric. 138, 92–104.

Ampatzidis, Y., De Bellis, L., Luvisi, A., 2017. iPathology: robotic applications and management of plants and plant diseases. Sustainability 9 (6), 1010. https://doi.org/10.3390/su9061010.

Angelini, E., Bianchi, G.L., Filippin, L., Morassutti, C., Borgo, M., 2007. A new TaqMan method for the identification of phytoplasmas associated with grapevine yellows by real-time PCR assay. J. Microbiol. Methods 68, 613–622.

Baxi, A., Vala, H., 2013. A review on Otsu image segmentation algorithm. Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET) 2 (2), 387.

Belli, G., Bianco, P.A., Conti, M., 2010. Grapevine yellows in Italy: past, present and future. J. Plant Pathol. 92, 303–326.

Bertaccini, A., 2007. Phytoplasmas: diversity, taxonomy, and epidemiology. Front. Biosci. 12, 673–689.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press.

Cardinale, M., Luvisi, A., Meyer, J.B., Sabella, E., De Bellis, L., Cruz, A.C., Ampatzidis, Y., Cherubini, P., 2018. Specific Fluorescence in Situ Hybridization (FISH) test to highlight colonization of xylem vessels by *Xylella fastidiosa* in naturally infected olive trees (*Olea europaea* L.). Front. Plant Sci. 9, 431.

Chang, C.C., Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol 2:27:1-27:27.

Chuche, J., Thiéry, D., 2014. Biology and ecology of the Flavescence dorée vector Scaphoideus titanus: a review. Agron. Sustain. Dev. 34 (2), 381–403.

Cruz, A.C., Luvisi, A., De Bellis, L., Ampatzidis, Y., 2017. X-FIDO: an effective application for detecting olive quick decline syndrome with deep learning and data fusion. Front. Plant Sci. 8 (October), 1–12. https://doi.org/10.3389/fpls.2017.01741.

Dechter, R., 1986. Learning while searching in constraint-satisfaction-problems. In: 5th National Conference on Artificial Intelligence, pp. 178–183.

Dodge, S., Karam, L., 2017. A study and comparison of human and deep learning recognition performance under visual distortions. Computer Communication and Networks (ICCCN), 2017 26th International Conference on. IEEE.

Gajardo, A., Fiore, N., Prodan, S., Paltrinieri, S., Botti, S., Pino, A.M., Zamorano, A., Montealegre, J., Bertaccini, A., 2009. Phytoplasmas associated with grapevine yellows disease in Chile. Plant Dis. 93 (8), 789–796.

Girosi, F., Jones, M., Poggio, T., 1995. Regularization theory and neural networks architectures. Neural Comput. 7 (2), 219–269.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. ArXiv Preprint ArXiv:1512.03385.

https://plantvillage.psu.edu/, 2018. Accessed June 11, 2018.

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. arXiv preprint arXiv:1602.07360.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Li, F.F., 2014. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1725–1732.

Kaur, S., Pandey, S., Goel, S., 2018. A semi-automatic leaf disease detection and classification system for soybean culture. IET Image Proces. 12 (6). https://doi.org/10.1049/iet-ipr.2017.0822.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. Adv. Neural Inform. Process. Syst. 1–9.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1, 541–551. https://doi.org/10.1162/neco.1989.1.4.541.

Li, R., Mock, R., Huang, Q., Abad, J., Hartung, J., Kinard, G., 2008. A reliable and inexpensive method of nucleic acid extraction for the PCR-based detection of diverse plant pathogens. J. Virol. Methods 154, 48–55.

Li, Z., Hoiem, D., 2017. Learning without forgetting. IEEE Trans. Pattern Anal. Mach. Intell. https://doi.org/10.1007/978-3-319-46493-0_37.

Luvisi, A., Ampatzidis, Y., De Bellis, L., 2016. Plant pathology and Information Technology: opportunity for management of disease outbreak and applications in regulation frameworks. Sustainability 8, 831.

Maixner, M., 1994. Hyalesthes obsoletus (Auchenorrhyncha: Cixiidae). Vitis 33, 103–104.

Martini, M., Murari, E., Mori, N., Bertaccini, A., 1999. Identification and epidemic distribution of two flavescence dorée—related phytoplasmas in Veneto (Italy). Plant Dis. 83 (10), 925–930.

Mehle, N., Ravnikar, M., Žnidarič, M.T., Aryan, A., Brader, G., Dermastia, M., 2017. Detection of phytoplasmas associated to grapevine yellows diseases in research and diagnostics. Grapevine Yellows Diseases and Their Phytoplasma Agents. SpringerBriefs in Agriculture. Springer, Cham.

Mirchenari, S.M., Massah, A., Zirak, L., 2015. 'Bois noir': new phytoplasma disease of grapevine in Iran. J. Plant Prot. Res. 55 (1), 88–93.

Mohanty, S.P., Hughes, D., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1419.

Mori, N., Quaglino, F., Tessari, F., Pozzebon, A., Bulgari, D., Casati, P., Bianco, P.A., 2015. Investigation on 'bois noir' epidemiology in north-eastern Italian vineyards through a multidisciplinary approach. Ann. Appl. Biol. 166, 75–89.

Nielsen, M.A., 2015. Overfitting and Regularization in Neural Networks and Deep Learning. Determination Press.

Ojala, T., Pietikäinen, M., Harwood, D., 1994. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994), vol. 1, pp. 582–585.

Pierro, R., Passera, A., Panattoni, A., Casati, P., Luvisi, A., Rizzo, D., Bianco, P.A., Quaglino, F., Materazzi, A., 2018a. Molecular typing of 'bois noir' phytoplasma strains in the Chianti Classico area (Tuscany, central Italy) and their association with symptom severity in Vitis vinifera L. cv. Sangiovese. Phytopathology 108, 362–373.

Pierro, R., Passera, A., Panattoni, A., Rizzo, D., Stefani, L., Bartolini, L., Casati, P., Luvisi, A., Quaglino, F., Materazzi, A., 2018b. Prevalence of a 'Candidatus Phytoplasma solani' strain, so far associated only with other hosts, in Bois noir-affected grapevines within Tuscan vineyards. Ann. Appl. Biol. https://doi.org/10.1111/aab.12453.

Quaglino, F., Zhao, Y., Casati, P., Bulgari, D., Bianco, P.A., Wei, W., Davis, R.E., 2013. Candidatus phytoplasma solani: a novel taxon associated with stolbur and bois noir related diseases of plants. Int. J. Syst. Evol. Microbiol. 63, 2879–2894.

Rizzo, D., Materazzi, A., Stefani, L., Panattoni, A., Pierro, R., Marchi, G., Cinelli, T., De Bellis, L., Luvisi, A., 2018. The monitoring program of grapevine phytoplasmas in Tuscany (Italy): results and effectiveness. Adv. Horticult. Sci. 32 (in press).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. 115 (3), 211–252.

Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Networks 61, 85–117.

Sengar, N., Dutta, M.K., Travieso, C.M., 2018. Computer vision based technique for identification and quantification of powdery mildew disease in cherry leaves. Computing 1–13. https://doi.org/10.1007/s00607-018-0638-1.

Sharif, M., Khana, M.A., Iqbala, Z., Azama, M.F., Lalib, M.I.U., Javedc, M.Y., 2018. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. Comput. Electron. Agric. 150, 220–234.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Arbor, A., 2015. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, pp. 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. Proceedings of the 27th International Conference on Machinee Learning (ICML), Haifa, Israel.

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? Adv. Neural Inform. Proces. Syst. 27 (Proceedings of NIPS) 27, 1–9.

Zhou, R., Kaneko, S., Tanaka, F., Kayamori, M., Shimizu, M., 2014. Disease detection of Cercospora Leaf Spot in sugar beet by robust template matching. Comput. Electron. Agric. 108, 58–70.