



# Evolutionary conditional GANs for supervised data augmentation: The case of assessing berry number per cluster in grapevine

Salvador Gutiérrez<sup>a,\*</sup>, Javier Tardaguila<sup>b</sup>

<sup>a</sup> Department of Computer Science and Artificial Intelligence (DECSAI), Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada (UGR), Granada, 18014, Spain

<sup>b</sup> Televitis Research Group, University of La Rioja, Logroño, 26006, Spain

## ARTICLE INFO

### Article history:

Received 5 February 2023  
Received in revised form 16 August 2023  
Accepted 28 August 2023  
Available online 3 September 2023

### Keywords:

cGAN  
Synthetic images  
Genetic algorithms  
Evolutionary computation

## ABSTRACT

In viticulture, training a deep learning model to assess the number of berries in a cluster image requires labeling by manually counting the berries on each grape cluster. This repeating, demanding task directly opposes the need for large amounts of data of deep learning methods. The objective of this work is the development of evolutionary conditional generative adversarial networks (GANs) for supervised data augmentation in the task of assessing berry number per cluster in grapevine. Ninety-seven grape cluster images were collected and labeled by manually counting the berries, making up the original dataset, and it was used for the training of a conditional GAN using evolutionary schemes involving a population of generators competing within a common and dynamic environment: the discriminator. After generative networks training, the best performing generator was used for supervised data augmentation: generation of labeled images conditioned to a berry number, making up the augmented dataset. Two models were trained with one dataset each, original and augmented, both having approximately the same number of samples, around 400. The original dataset was enriched with traditional image transforms, while the augmented dataset had incorporated images from the trained GAN generator. The two models were trained using the same convolutional regression network architecture, and then tested on an external image dataset, with more than 1300 images not used in any training, to compare the performance of GAN data augmentation. Results showed that the model augmented with GANs yielded lower error values, with a validation error of 43 berries, than those from the original model, with a validation error of 65 berries. In the test dataset, the augmented model obtained an error of 39 berries ( $R^2 = 0.75$ ), surpassing again the original model, that obtained an error of 57 berries ( $R^2 = 0.65$ ). These results evidence that evolutionary conditional GANs generate synthetic labeled images that lead to higher performance deep learning regression models for assessing berry number from cluster images.

© 2023 Elsevier B.V. All rights reserved.

## 1. Introduction

The application of machine and deep learning solutions is greatly extended in many disciplines, but virtually all of them are conditioned to a requisite that is frequently hard to fulfill: large amounts of quality data properly labeled. It is here where generative learning highlight as potential mechanisms for obtaining quality labeled data with much less effort. One of the most successful representations of DL mechanisms for synthetic generation of data are generative adversarial networks (GANs) [1], especially used in computer vision for their quality in image synthesis [2]. The base concept of GANs was originally presented by Goodfellow et al. [3], proposing the deployment of two

models—generator and discriminator—trained in an adversarial environment. Generative adversarial networks, while powerful, are hard to train since their conception, and this is known by many of their users. The three main problems that occur in the development of GANs are: (i) instability in convergence when searching the needed equilibrium between both networks; (ii) mode collapse, in which the generator learns the mechanisms to map all the latent space into a very reduced set of outputs that deceive the discriminator by mapping (in the case of image generator, all created images are always very similar); and (iii) vanishing gradients, that occur when the discriminator learns too well in early iterations to distinguish between real and synthetic samples, thus the generator is stuck in a low quality state that does not allow to effectively improve due to very small gradients (a consequence of the linked adversarial training).

Research in GANs have brought several new training mechanisms that improved the performance of the original proposal,

\* Corresponding author.

E-mail address: [salvaguti@decsai.ugr.es](mailto:salvaguti@decsai.ugr.es) (S. Gutiérrez).

addressing the aforementioned problems, among others. Some of these improvements are mainly focused on modified objective functions, like: Least Squares GANs [4], based on least squares, that allow for better quality images and a more stable training; the hinge loss, used in Geometric GANs [5]; or Fisher GANs [6]. One of the most successful variations to address unstable convergence and mode collapse are Wasserstein GANs [7] and their improvement, adding a gradient penalty at discriminator's update [8]. While the better performance of numerous GAN variants are strongly supported by their reported metrics and usage, they still have their own pros and cons, and selecting one or another could yield very divergent performance outcomes, depending on the problem configuration, input data, etc.

Evolutionary computing seeks to solve optimization problems taking inspiration from natural evolution [9]. The general approach of an evolutionary algorithm involves a population of solutions (each gathering the features or parameters that alters the output of the problem/function to optimize) with a computed associated quality (a *fitness* value) that breed, mutates and compete during several generations to produce solutions that desirably converge to an optimal state (ideally, the global minimum). This scheme, widely used and adapted in many fields [10], including agriculture [11,12], could address the problem of which GAN variant to test and select for a specific problem, as it could be conceived as an evolutionary GAN scenario in which different individuals compete and evolve as generations pass to reach increasing performances. Some works have explored this possibility (or similar ones inspired in evolutionary algorithms) in the field of adversarial networks [13–16], but Wang et al. [17] pioneered the proposal of a complete evolutionary approach in GAN training (previously published as a preprint in 2018). The authors conceived an evolutionary scenario in which a population of generators comply with the evolutionary operations in a dynamic environment, that is represented by a single discriminator, that updates based on the population's outputs.

The accurate estimation of yield components is critical in the grape and wine industries [18,19]. The variability in cluster size, shape and berry number is high within the same plant or block [20], making difficult the application of generalization or extrapolating methodologies for assessing grape production in vineyards. Therefore, the monitoring of individual grape clusters would help for a better characterization. While cluster-related features (weight, compactness, berry size, etc.) have shown useful for the yield assessment in grapevine [21,22], the counting of cluster components (flowers [23], berries [24] or grapes [25]) is one of the inputs exhibiting the highest correlation with yield measures in production. Given this, the main obstacle for counting berries is not only acquiring large amounts of images, but also the labeling with ground truth by manually counting grapes for each cluster, a laborious and repetitive task. Considering the exposed context, to the best of our knowledge, we have not found in the literature any solution for the synthetic creation of data to help yield prediction in grapevines, so there is an opportunity to explore the use of GANs for improving deep learning models for yield prediction, presenting an application in viticulture.

The objective of this work is the train and use of evolutionary conditional GANs for supervised data augmentation to improve the performance of deep learning regression models in the task of assessing berry number per cluster in grapevine.

## 2. Related works

Considering the demonstrated strength of GANs for image synthesis, these has been used recently in many agricultural applications [26]. Disease and pest detection is important in the management of agricultural operations, especially if the it is done

early during the incidence (ideally, pre-symptomatic). For this, adversarial networks have been applied for the development of an online data augmentation platform for whitefly pests [27], incorporated in the management of hyperspectral images in tomato spotted wilt [28], leaf symptoms with computer vision again in tomato [29], or used for data augmentation for disease detection in grapevine leaves [30–32]. Due to its impact in crop growth, weed monitoring has also been attempted using GANs. Synthetic images were used to help for the automated detection and segmentation of weeds from robotic platforms [33–35].

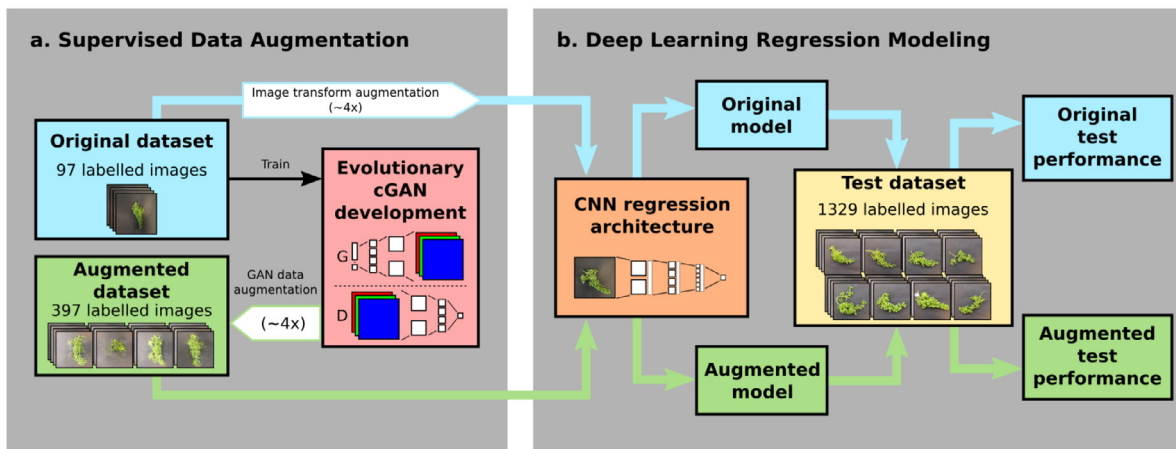
Yield estimation and fruit monitoring has also benefited from GAN image generation. Boundary Equilibrium GANs were employed as a data augmentation module in the detection of pine cone [36], combining the Wasserstein distance with an autoencoder-based GAN. A conditional GAN was trained for the completion of kiwi images and 3D models from partial data, configuring the adversarial networks in a image-to-image translation problem [37]. Orchard images from proximal unmanned aerial vehicle were used as input for a Cycle-Generative adversarial network as part of a major process of yield estimation in almond and apple trees [38]. Finally, automated monitoring assisted with GANs has also been reported for grapevine yield components: for the reconstruction of clusters due to occlusions [39] or the generation of grapevine canopy 3D models for improved fruit counting [40].

In precision agriculture, evolutionary computation have been used for several applications: genetic algorithms as optimizers for spectral band selection [41–43] or yield prediction [44,45]; particle swarm optimization for crop planning [46,46,47] or disease detection [48,49]; or ant colony optimization for wavelength selection [50,51]. In the task of yield prediction, tools for grape stem detection have been developed and optimized with genetic algorithms [52], for evolutionary optimization in kiwifruit [53] or for oil palm yield prediction [54].

## 3. Proposed methodology

In this section we describe the general workflow of the proposed methodology for supervised data augmentation with Evolutionary GANs, and its particular application in the assessment of grape berry number in cluster images. Before the detailed description of the work presented in this paper, an overview of the designed pipeline is summarized in Fig. 1. This work can be conceptually divided into two stages—supervised data augmentation and deep learning modeling:

- a. From a collection of 97 grape berry cluster images, previously labeled by manual berry counting (Section 3.1), supervised data augmentation (Fig. 1a) involves the development, training and inference work of a conditional GAN with an evolutionary approach, using the original dataset as the training data (Section 3.2). We chose the training to be addressed as an evolutionary problem because, as Wang et al. [17] stated, this allows to minimize the weaknesses of different GAN objectives and optimize the networks jointly with different metrics. Therefore, the training stability improves for better generative performance. After Evolutionary GAN training, synthetic images are conditionally generated (setting berry numbers at discretion) for the supervised construction of an augmented dataset, four times larger than the original one (Section 3.3). This supervised synthetic sample generation is only possible if the GAN training and optimization is conditioned, and this is the reason we propose the use of conditional GANs [55].



**Fig. 1.** Diagram of the methodology developed and described in this paper, conceptually divided in a. supervised data augmentation (by means of a full development and training of an evolutionary conditional GAN), and b. deep learning regression (training models with CNNs under the same conditions for the results to be fully comparable).

- b. In order to rigorously prove that the GAN data augmentation really improves the performance of models developed with the generated images, regression models under the same conditions are trained using the original and augmented datasets (Fig. 1b), collecting training and validation metrics (Section 3.4). The original model was trained with data from the original dataset enlarged four times the original size with traditional image transformations, while the augmented dataset was trained with the original dataset augmented four times with synthetic images from the best generator after Evolutionary GAN training. Additionally, a real application of the models is tested, so both trained neural networks are used for the prediction of 1329 labeled cluster images not used in any training, reporting performance metrics on this test dataset (Section 3.5).

The following subsections describe in detail the phases summarized and displayed in Fig. 1. All models and algorithms in this work were implemented using Python 3.9 and PyTorch 1.13, training on a NVIDIA GeForce RTX 4090 with 24 GB of vRAM.

### 3.1. Data collection

As the final objective is the assessment of berry number from cluster images, a collection of approximately 100 cluster pictures was built and labeled (“original dataset”). A hundred grape clusters at pea-size (growth stage 31 according to Coombe [56]) of cv. Tempranillo were collected from a vineyard located in La Rioja, Spain. Clusters were individually marked and transported to the laboratory for their imaging and manual total berry counting, linking each image with its corresponding number.

Images were taken using two RGB cameras: Canon EOS 5D Mark IV (full-frame CMOS sensor, 30.4 Mpx), equipped with a Canon EF 20 mm F/2.8 USM lens; and Sony  $\alpha 7$  II digital mirrorless camera (CCD sensor, 24.0 Mpx, 5-axis image stabilization system), equipped with a Zeiss 24/70 mm lens with optical stabilization. At image acquisition, the camera was placed onto a tripod and pointing downwards to a dark paper background in which the samples (grape clusters) were presented (constant distance between camera and target of 50 cm). Pictures were taken without flash and with indirect artificial illumination. Fig. 2 shows some images from this dataset grouped by four berry number ranges (a. to d.), illustrating the different cluster shapes depending on size and berry total number. While the general image acquisition parameters were maintained in terms of distances, illumination, background, etc., a complete capture homogeneity was not

**Table 1**

Statistical summary of the original dataset.

$N$	$min$	$max$	$mean$	$std$	$median$
97	47	540	174	85	157

$N$ : number of samples (images);  $std$ : standard deviation.  $Mean$  and  $std$  values are rounded to the nearest integer. Except for  $N$ , units are total berry number of the grape cluster imaged.

sought, as illustrated in Fig. 2 by the visual slight differences in conditions. This was intended for the sake of reproducibility, to make the different AI models more robust, as the only conditions for the potential end user of the model is to make the captures approximately at the same distance, with a dark background and clear, indirect illumination (as in the case of the test dataset, described in Section 3.5). All images were cropped around the cluster with a square of constant dimensions, to preserve features in terms of cluster and berry sizes.

The statistical description of the dataset is presented in Table 1. The number of samples (cluster images) was reduced from the original 100 to 97 due to inaccuracies in the counting of three clusters, hence their images were discarded.

### 3.2. Evolutionary conditional GAN development and training

The evolutionary scheme described and used in this work is based in the work by Wang et al. [17], with slight adaptations and modifications accordingly to the problem to solve. The authors proposed an evolutionary environment in which the GAN learning occurs, transitioning from the original static adversarial training to a dynamic stage. As in any evolutionary algorithm, a population of individuals is generated and let evolve in a competitive scenario, in which the best adapted individuals survive to form a next generation. The cycle is then repeated, improving the population iteratively. If we represent each individual as a GAN generator that seek to optimize a specific function, and the dynamic environment as a single discriminator that improves from the best fitted individuals, this allows to consider and evaluate different adversarial objectives and overcome many disadvantages that are commonly found in individual objectives, like mode collapse or vanishing gradients.

In the Evolutionary GAN proposal [17], adapted here with conditional GANs, the population is composed of a set of  $\mu$  generators  $G = \{G_{\theta_1}, G_{\theta_2}, \dots, G_{\theta_\mu}\}$  that are constantly evolving, evaluated and competing in a dynamic environment, the discriminator  $D$ .



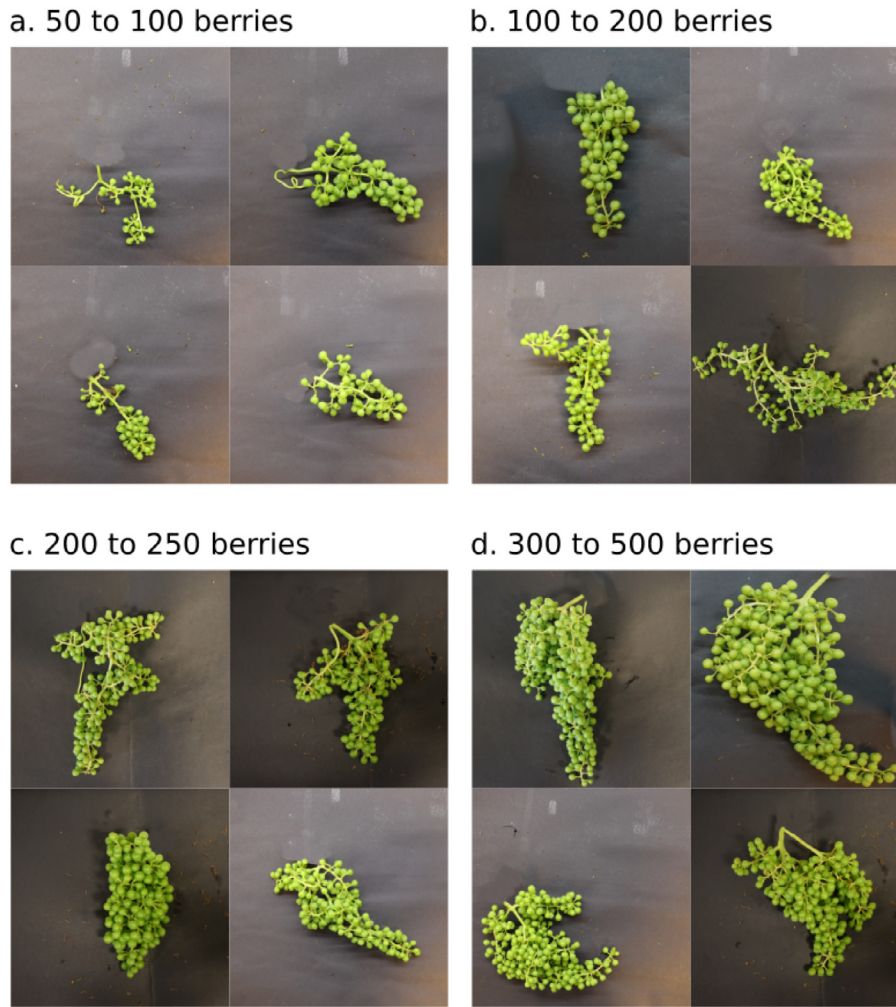


Fig. 2. Labeled images from the original dataset ( $N = 97$ ) grouped by berry number.

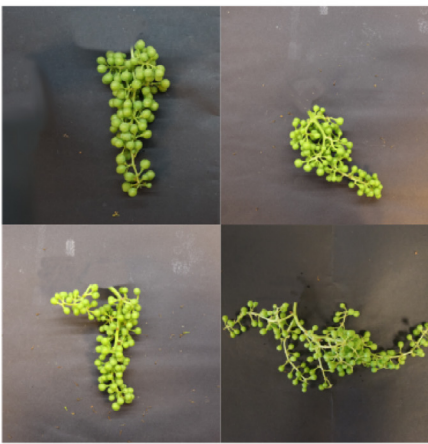
After the evolutionary stage, the discriminator is updated attending to the outputs of all the individuals from the new population (synthetic images from the new generators).

At each evolutionary step, the authors from Wang et al. [17] defined the following three operators:

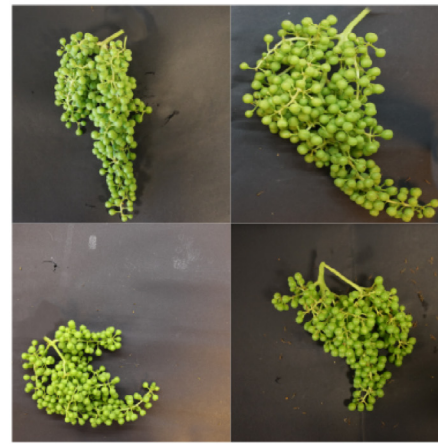
1. *Variation*: For each individual  $G_{\theta_i}$  (*parent*), its offspring is created by asexual reproduction, i.e., making copies and inducing genetic variability by means of different *mutations*. If  $n_m$  mutations are considered, then the total number of *children* is  $\mu \times n_m$ . The optimization of standard GANs is computationally intensive, and this evolutionary adaptation multiplies the updating process of the generator by  $\mu \times n_m$ , so the inclination is to not set too large values for those parameters. In fact, in Wang et al. [17], most of the models were trained with a population size  $\mu = 1$ . We set  $\mu = 2$ , with  $n_m = 4$  different mutations, yielding a total of 8 children per evolutionary step.
2. *Evaluation*: The performance of each child is evaluated using a *fitness* function  $\mathcal{F}$  that is to be maximized. We implement the same fitness definition, described later in this section.
3. *Selection*: The new population is created by selecting the  $\mu$  children with the highest fitness score, discarding the remaining.

In Wang et al. [17], the authors proposed three generator training objectives that act as mutations in the variation phase.

b. 100 to 200 berries



d. 300 to 500 berries



Two of them were used as base for our implementation. Given a generator  $G_{\theta} \in G$  and a discriminator  $D$  (the current environment), the mutations used (functions to be minimized) are defined as:

1. *Modified*: A modified objective proposed in the original GAN paper to avoid discriminator saturation:

$$\mathcal{M}_{G_{\theta}}^{\text{modified}} = -\mathbb{E}_{z \sim p_z} \left[ \log(D(G_{\theta}(z))) \right], \text{ and} \quad (1)$$

2. *Least-Squares*: Based on Least Squares GANs [4]:

$$\mathcal{M}_{G_{\theta}}^{\text{least}} = -\mathbb{E}_{z \sim p_z} \left[ (D(G_{\theta}(z)) - 1)^2 \right]. \quad (2)$$

Additionally to these mutations originally proposed, two more mutations are included in our work to further increase the search space:

3. *Wasserstein*: Based on Wasserstein GANs [7]:

$$\mathcal{M}_{G_{\theta}}^{\text{wasser}} = -\mathbb{E}_{z \sim p_z} [D(G_{\theta}(z))], \text{ and} \quad (3)$$

4. *Hinge*: Based on the hinge loss from Geometric GANs [5]:

$$\mathcal{M}_{G_{\theta}}^{\text{hinge}} = \mathbb{E}_{z \sim p_z} \left[ \min(0, 1 - D(G_{\theta}(z))) \right]. \quad (4)$$

The offspring evaluation is performed by computing a fitness value  $\mathcal{F}$  for each child. The fitness value accounts for two desirable properties in each evolved individual (child  $G'_{\theta}$ , after

mutation from a parent  $G_\theta$ ): the quality of the generated images  $\mathcal{F}_{G_\theta}^q$ ; and, to avoid extreme convergence problems (i.e., mode collapse), the diversity of the generated images  $\mathcal{F}_{G_\theta}^d$ . These are defined in (5) and (6), respectively.

$$\mathcal{F}_{G_\theta}^q = \mathbb{E}_{z \sim p_z} [D(G'_\theta(z))] \tag{5}$$

$$\mathcal{F}_{G_\theta}^d = -\log \|\nabla_D - \mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G'_\theta(z)))]\| \tag{6}$$

Eq. (5) is a raw account of the quality of the generator (its capacity of deceive the discriminator), while Eq. (6) considers the gradients from optimizing  $D$  as a measure of diversity. The final fitness value is the sum of both quality values (5) and (6), the latter weighted with a value  $\gamma \geq 0$ :

$$\mathcal{F}_{G'_\theta} = \mathcal{F}_{G_\theta}^q + \gamma \mathcal{F}_{G_\theta}^d. \tag{7}$$

After offspring evaluation, the  $\mu$  children with highest fitness are selected and treated as the new population for the next evolutionary iteration. The full Evolutionary GAN training is described in Algorithm 1. The criterion (objective function) for updating  $D$  used is the one from WGANs with gradient penalty calculation [8].

**Algorithm 1** Training of the evolutionary conditional GAN.

**Require:** Initial parameters for the discriminator.  $w_0$ ; initial parameters for the generators in the population  $\{\theta_0^1, \theta_0^2, \dots, \theta_0^\mu\}$

**Require:** The batch size  $m$ ; the number of updates per epoch to apply to the discriminator per epoch  $k$ ; the population size  $\mu$ ; the number of mutations  $n_m$ ; the weight in the fitness function  $\gamma$ ; hyperparameters for the Adam optimization  $\alpha, \beta_1, \beta_2$ ; the gradient penalty coefficient  $\lambda$ .

**for** number of epochs **do**

**for**  $c = 0, \dots, \mu$  **do**

**for**  $l = 0, \dots, n_m$  **do**

      Sample a batch of  $\{z^{(i)}\}_{i=1}^m \sim p_z$

$g_{\theta c, l} \leftarrow \nabla_{\theta_j} \mathcal{M}_{G_c}^l(z^{(i)}, y^{(i)})$

$\theta_{child}^{c, l} \leftarrow \text{Adam}(g_{\theta c, l}, \theta^c, \alpha, \beta_1, \beta_2)$

$\mathcal{F}_{c, l} \leftarrow \mathcal{F}_{c, l}^q + \gamma \mathcal{F}_{c, l}^d$

**end for**

**end for**

$\{\theta^1, \theta^2, \dots, \theta^\mu\} \leftarrow$  the  $\mu$  children with the highest  $\mathcal{F}$

**for** each batch  $\{x^{(i)}, y^{(i)}\}_{i=1}^m$  from the training data **do**

**for**  $d = 0, \dots, k$  **do**

      Sample a batch of  $\{z^{(i)}\}_{i=1}^{m/\mu} \sim p_z$

$g_w \leftarrow \nabla_w \left[ \frac{1}{m} \sum_{i=1}^m D_w(x^{(i)}, y^{(i)}) - \right.$

$\left. \frac{1}{m} \sum_{j=1}^{\mu} \sum_{i=1}^{m/\mu} D_w(G_j(z^{(i)}, y^{(i)}), y^{(i)}) + \right.$

$\left. \lambda (\|\nabla_{x^{(i)}} D_w(x^{(i)}, y^{(i)})\|_2 - 1)^2 \right]$

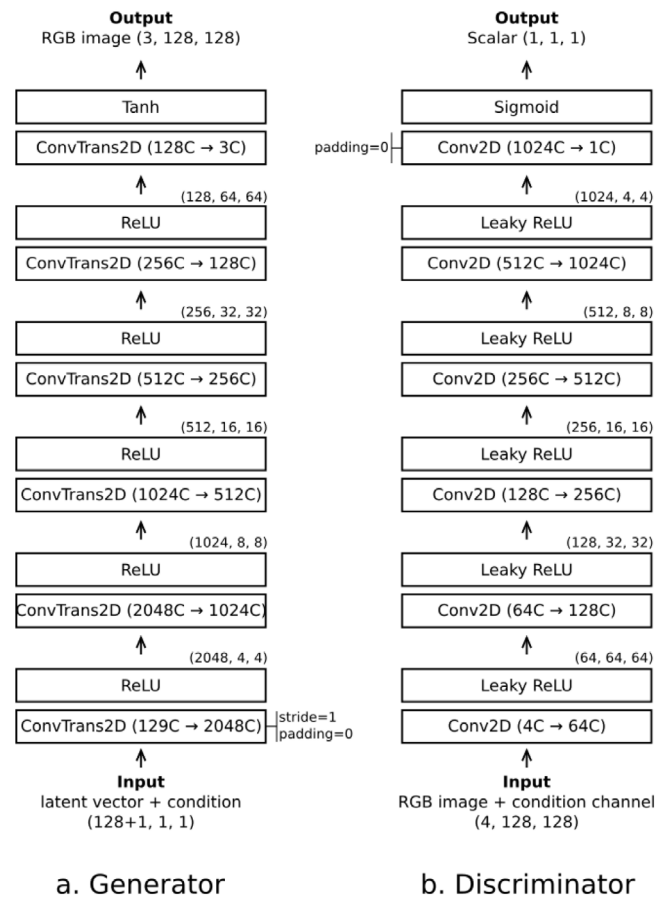
$w \leftarrow \text{Adam}(g_w, w, \alpha, \beta_1, \beta_2)$

**end for**

**end for**

**end for**

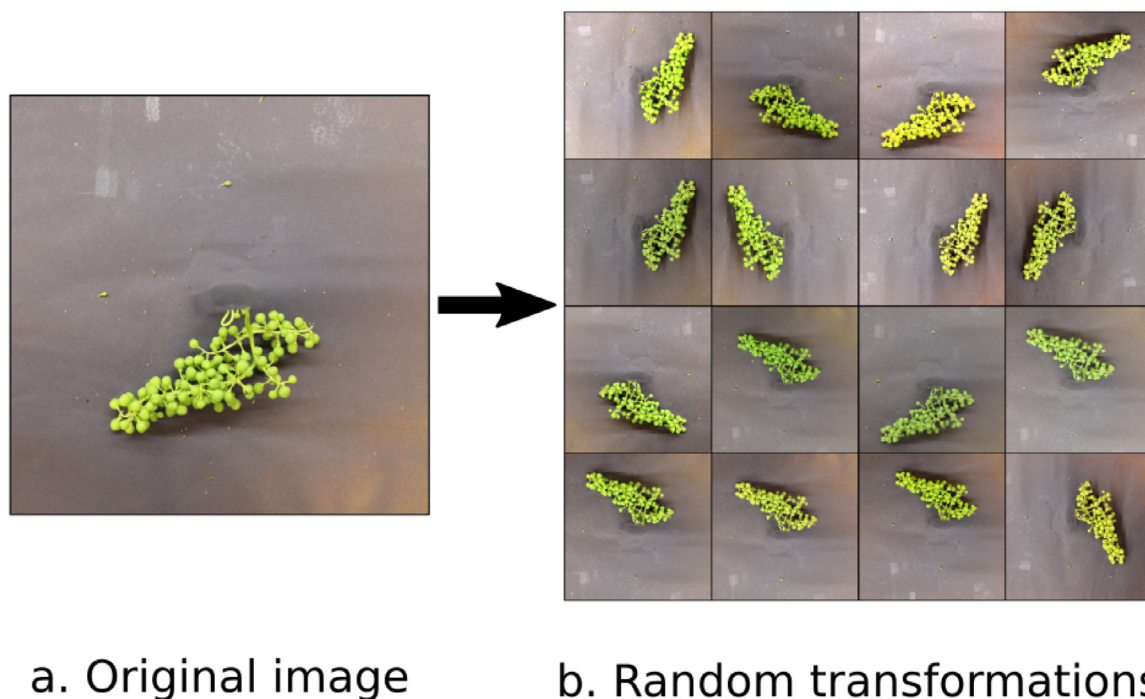
The working image size for all the models described in this paper is  $128 \times 128$  pixels, so both generator and discriminator networks were designed accordingly to that size. Fig. 3 displays both architectures. No pooling operations were included, so all dimension alterations are obtained via appropriate kernel size, striding and padding values in convolutional filters (down-sampling with two-dimensional convolutions in the generator; upsampling with two-dimensional transposed convolutions). As conditional image generation is required, the targeted value—the



**Fig. 3.** Architectures of generator (a.) and discriminator (b.) of the conditional GAN. The convention used for data structures is that the first dimension refers to the channels. Conv2D and ConvTrans2D refer to 2D convolution and transposed convolution operations, respectively. Except when indicated, all convolutions used a kernel size of  $4 \times 4$ , a stride value of 2 and a padding of 1. All leaky ReLU modules had a slope value  $\alpha$  of 0.2. The last sigmoid activation function from the discriminator was only used when corresponding, depending on the objective function used.

number of berries—is aggregated to the batch before forwarding it into the network. In the case of the generator, the condition is appended to the latent vector as a last value. For the discriminator, the value is included as a fourth image channel, repeated in all the positions. As mentioned, the value of the condition refers to the number of berries in the image (to be generated by  $G$  or to be evaluated by  $D$ ), but these ranged within values that are too large for neural networks (Table 1). For this reason, all conditions forwarded to any GAN network need to be rescaled, and in our work we computed the new scale using 30 and 600 berries as minimum and maximum values.

The training of the evolutionary conditional GAN was capped at 2000 epochs, with continuous monitoring of generated images via fixed noise and conditions, and storing checkpoints every 100 iterations. The  $\mu$  value (population size) was set to 2; the weight value  $\gamma$  in fitness calculation was set to 0.5; the gradient penalty magnitude  $\lambda$  was set to 10; and the number of discriminator updates per epoch  $k$  was set to 3. Adam optimization was set up for both networks, with a learning rate of 0.0002, and  $\beta_1$  and  $\beta_2$  values of 0 and 0.9, respectively. A summary of the Evolutionary GAN parameters used is presented in Table 2, compared to their default values from the original proposal [17]. In order to enrich the original dataset for better training, several random image transformations were applied online to the images for every batch iterated:



**Fig. 4.** Example of the different common transformation techniques applied for an original image from the dataset (a). Transformations (b) randomly alter several image features: rotations by multiples of 90, color jitter, affine transformations, and vertical and horizontal flipping.

**Table 2**  
Summary of the hyperparameters of Evolutionary GANs training originally proposed versus values used in this work.

Parameter	Default value from Wang et al. [17]	Value used in this work
Learning rate $\alpha$	0.0002	0.0004
Adam's $\beta_1$	0.5	0
Adam's $\beta_2$	0.99	0.9
Discriminator updating steps $k$	2	1
Number of parents $n_p$	1	2
Number of mutations $n_m$	3	4
Fitness weight $\gamma$	0.5	0.5

- Rotation by multiples of 90.
- Color jitter, affecting brightness, contrast, saturation and hue.
- Affine transformations, with translation not greater than 5% for both dimensions and scaling not larger than 1.2 the original size.
- Vertical and horizontal flipping.

Examples of these random transformations are displayed in Fig. 4. With this, the total size of the original dataset for training was increase by a factor of 10.

### 3.3. Supervised data augmentation

After the training of the Evolutionary GAN, the images generated at each epoch for monitoring were visually analyzed. The goal was to pick the generator from the epoch with the images most appealing compared to the real ones, in terms of realism, definition and cluster shape and size (presumably, the latest epochs).

Once the generator was selected, three batches of 100 images each were created conditionally, setting the number of berries equally spaced from 250 to 400 in the first batch, from 50 to 200 berries in the second batch, and from 350 to 500 berries

in the last batch. The 97 images from the original dataset were then augmented with the new 300 synthetic samples, having then an augmented dataset with a total of 397 grape cluster labeled images.

### 3.4. Deep learning regression modeling

To comparatively test the performance of the augmented dataset, an unique CNN regression architecture was designed to be trained separately with the two datasets (original and augmented). The neural network was configured to receive a 3 channel,  $128 \times 128$  image and forward it to a block with eight two-dimensional convolutional filters (kernel size of  $3 \times 3$ , striding and padding values of 1), activated with ReLU (rectified linear unit), downsized using two-dimensional  $2 \times 2$  max pooling, and regularized with a dropout layer ( $p = 0.6$ ). The output of this block is then flattened to a 128-units fully connected layer, with ReLU as activation function, and after connected to a single output neuron having no activation function, to map a continuous value representing the berry number from the input image.

Training was set up using Adam algorithm for optimization, with learning rate of 0.003,  $\beta_1$  and  $\beta_2$  values of 0.9 and 0.999, respectively, and the mean squared error (MSE) as loss function. Training was limited to 1000 epochs. As previously stated, two models were trained with this architecture and configuration: original model (using the original dataset) and augmented model (using the augmented dataset). Still, using the data as-is for the training would not have been equitable, as the original dataset comprises 97 labeled images versus the augmented dataset, with 397 labeled images. For this reason, training the original model involved enlarging the original dataset up to four times its original size, using traditional image transformations similar to the ones used during the Evolutionary GAN training (Fig. 4) and described in Section 3.2, giving a total of 388 images. To objectively determine that the original model—with 388 samples—outperforms a model with the source dataset, a baseline model was trained under the same conditions, but using only the 97 images, with no augmentation.



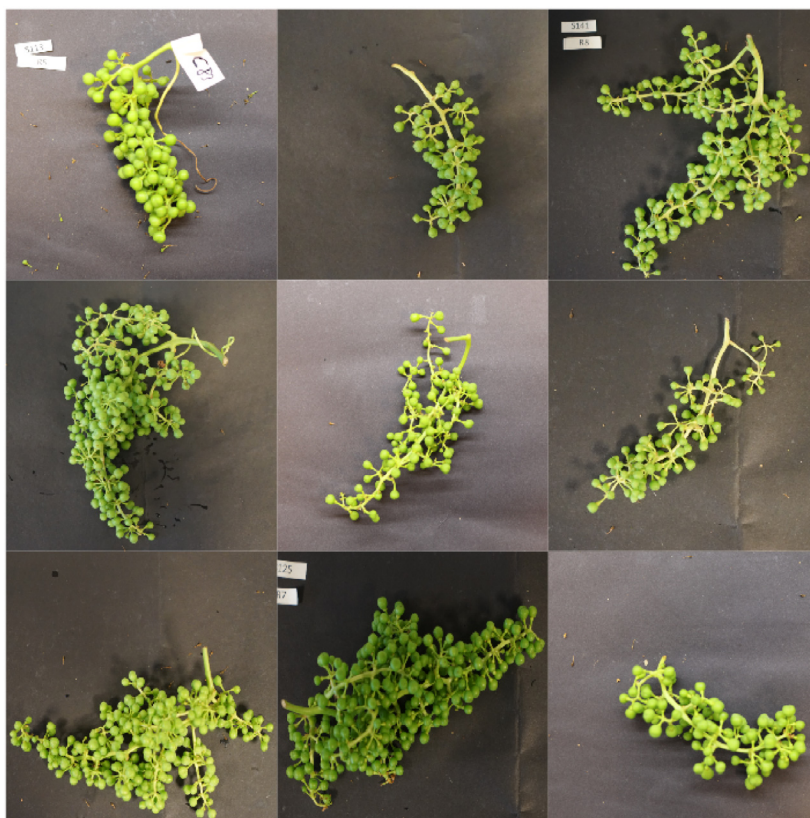


Fig. 5. Images from the test dataset ( $N = 1329$ ), composed of cluster images taken under conditions similar but not exact to those from the original dataset (Fig. 2).

### 3.5. Validation and testing

The training of the regression models – original and augmented – was carried out using the 80% of their corresponding dataset, while the remaining 20% was used for validation (hold-out validation). The division of these subsets was random, just ensuring that the same data distribution was present in each one. At each training epoch, the loss metric was registered for each subset, to supervise the performance, convergence and overfitting of the model. Specifically, the root-mean-square error (RMSE) was computed as performance metric ( $RMSE = \sqrt{\text{loss}}$ ).

To further verify the supposed added quality of the synthetic images in the training of the augmented model, a fully external dataset was prepared. Over 1300 grape cluster images were acquired under similar conditions, although not identical, than those used for the training of the neural networks. Clusters were acquired and imaged not only from grapevine of Tempranillo variety, but others too, collected from different plots in La Rioja, Spain. The considerably high number of images in this test dataset made the manual counting virtually impracticable, so the labeling of reference values (total number of berries of each image) was carried out automatically using an algorithm based on vitisBerry from the Televitis Research Group [57]. This labeling was supervised and adjusted dynamically when required depending on the conditions present on these images (color adjusting, thresholding, etc.).

The statistical description of the test dataset is shown in Table 3. It is necessary to highlight the fact that, while light and framing conditions of these images were not very different from the images used for training, many visible artifacts, such as labels, stickers, dirt, water spots, and others, were present here, but not found in the images from the original dataset, as can be seen in Fig. 5 (compared to Fig. 2). Additionally, no cropping was applied, so small deformations were implied when the images

Table 3

Statistical summary of the test dataset.

$N$	$min$	$max$	$mean$	$std$	$median$
1329	83	420	180	61	169

$N$ : number of samples (images);  $std$ : standard deviation.  $Mean$  and  $std$  values are rounded to the nearest integer. Except for  $N$ , units are total berry number of the grape cluster imaged.

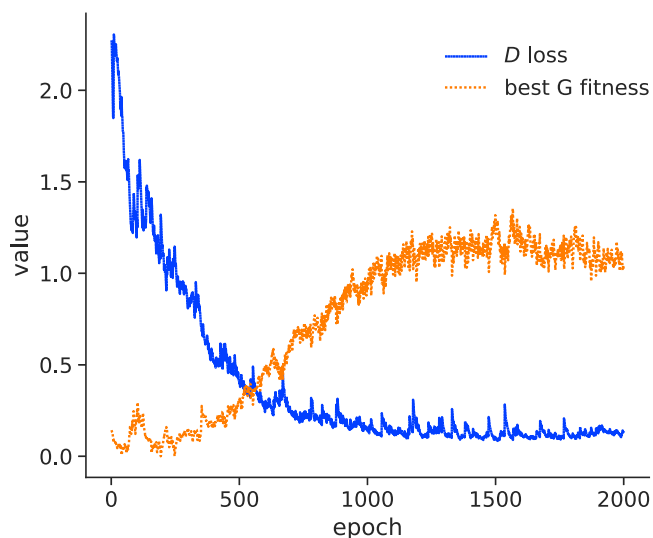
were reduced to the models' required input dimensions,  $128 \times 128$  pixels.

## 4. Results

After approximately 30 h of training of the Evolutionary GAN, with  $\mu = 2$  and  $n_m = 4$ , the metrics history of the training is displayed in Fig. 6, reporting the evolution of discriminator's loss and the fitness of the best generator at each epoch. After a slight divergence during the first hundred iterations, the behavior of the training followed an expected evolution.

The discriminator network  $D$  is considered a dynamic environment in which the population of generators evolve, and its loss (continuous blue line, Fig. 6) is decreasing asymptotically as it gets better at discerning synthetic from real images. By the nature of GANs training (and, by extension, Evolutionary GANs), the discriminator's updates feed back the training of the generators, and they constantly improve the quality of their generated images, attending to the value of the best fitness (dotted orange line, Fig. 6). The training starts to balance a few epochs after the 1000th epoch mark, when a convergence stage is reached, as no significant improvement occurs from further training.

After considering these metrics along all the training iterations, images from generators around epoch 1000 were analyzed, and we concluded that the images from epochs close to 800 were



**Fig. 6.** Evolutionary conditional GAN training plot. The continuous, descending blue line represent the loss of the discriminator network  $D$ . The dotted, ascending orange line represents the fitness from the best individual (generator) at each epoch. Fitness values were shifted over 0 for better visual comparative with  $D$  loss.

the most visually appealing and convincing. Sample synthetic images conditioned by berry number from this generator are shown in Fig. 7 (a. to d., in groups similar to those from Fig. 2). The generated images presented in all cases grapes grouped in ways consistent with real grape clusters, also in terms of color and texture, and with different shapes, always dependent on the condition added to the network's input. The selected generator was used for the supervised data augmentation step described in Section 3.3, thus building the augmented dataset with almost 400 images, 75% of them conditionally created by the selected generator.

The training of the regression models (original and augmented) required less than an hour to complete (1000 epochs). The training and validation metrics (RMSE) along all the epochs are reported in Fig. 8. For the original model—trained with almost 400 images: 97 real images augmented four times with image transformations—the losses (Fig. 8a) exhibited an expected sudden drop during a few epochs at first (as gradients start following a convergent path), and then started a finer adjusting to the training data, with a small divergence in the validation curve prior to epoch 200, finishing with a clear stabilization in both subsets (validation RMSE around 65 berries). In the case of the augmented model—trained with almost 400 images: 97 real images and the remaining synthetically created by the selected generator—the same loss descending occurs during the first epochs (Fig. 8b). The training convergence in this case needed for a few more epochs than with the original model, but a higher stability was finally reached. The descending behavior was very similar in both training and validation losses (at different scales), the latest laying slightly over an error value of 40 berries. In all cases, both models outperformed the best metrics obtained for the training of the baseline model (using only the 97 labeled images with no augmentation), which obtained RMSE values of 78.810 for training and 100.25 for validation.

The training of both original and augmented models was programmed so that the network from the epoch with the lower validation error (respectively) was stored as the definitive trained regression network—attending to the validation curves in Fig. 8, in the latest epochs. The metrics of these models are presented in Table 4. These models were used for the prediction of the

**Table 4**

Metrics for the deep learning modeling and testing on external dataset. RMSE: Root mean squared error (lower is better).

Model	Training	Validation	Test	
	RMSE	RMSE	RMSE	R <sup>2</sup>
Original	9.781	64.739	56.515	0.65
Augmented	7.674	40.169	38.774	0.75

images from the test dataset ( $N = 1329$ , Section 3.5), that in any case were not used for the training of any model. The predicted values from both models were related to the actual grape cluster berry number of the images and presented in Fig. 9, a. original model, and b. augmented model. The determination coefficient  $R^2$  was computed, along with the RMSE, given in berry number per cluster.

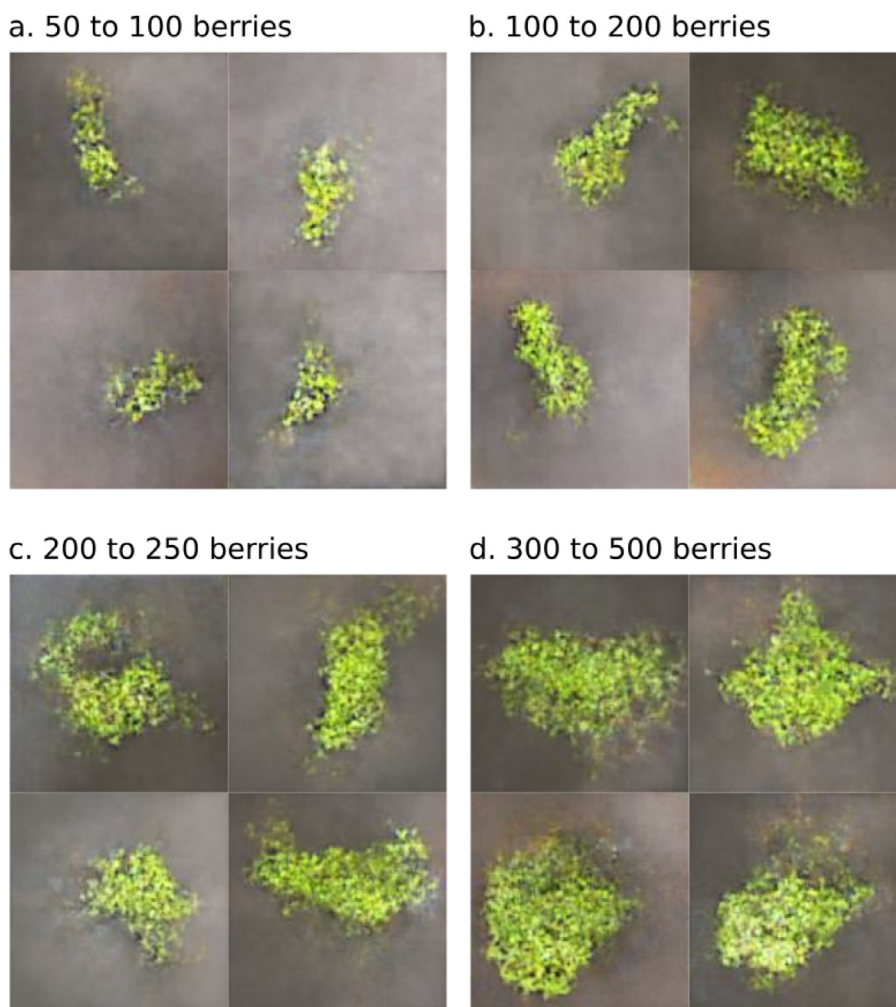
The original model obtained an  $R^2$  of 0.65 versus 0.75 from the augmented model, evidenced by a higher dispersion of the points from their corresponding regression lines (higher in the original model, Fig. 9a, especially when handling cluster images with a high number of berries). The higher precision found in the augmented model is highlighted by the tighter fit of the points to the regression line in Fig. 9b, hence the increase of  $R^2$  in 0.10 points. In terms of accuracy (average prediction closer to real values), again is the augmented model better than the original one, with an rounded error value of 39 versus 57 berries.

## 5. Discussion

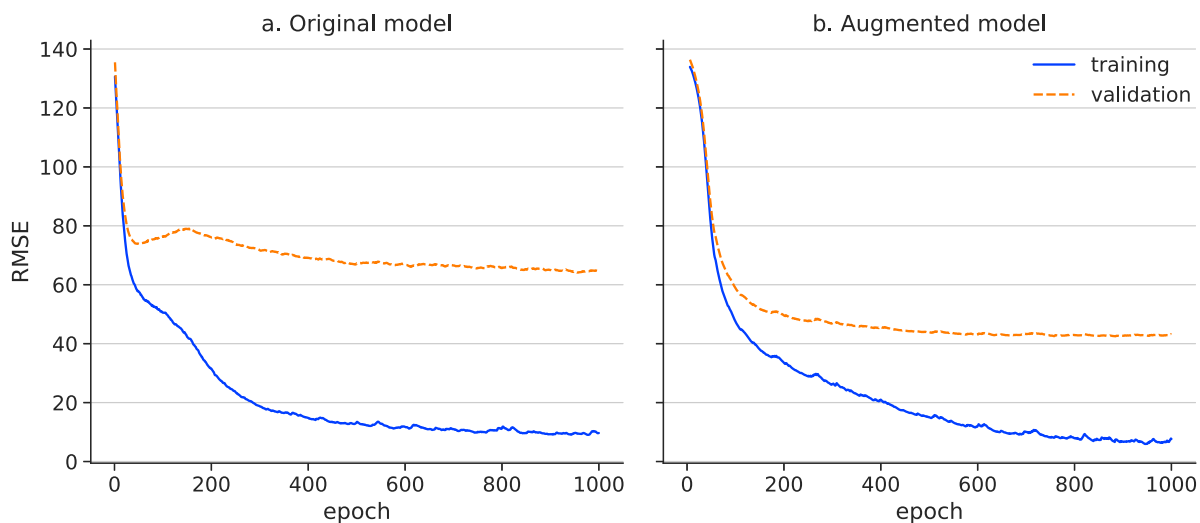
The results observed from the training of the evolutionary conditional GAN evidence that—from a quantitative point of view (Fig. 6)—a numerical stability was achieved by the networks (discriminator and best generator at each epoch) and not at the initial iterations, meaning that an successful adversarial training was carried out, and the generators from the best epochs were supposedly able to produce realistic images able to compete against the train data. Low training stability and early convergence (*i.e.*, severe imbalance towards one of the two networks that prevents further learning) are common problems when working with GANs, and the evolutionary approach proposed by Wang et al. [17] and adapted here did result in a more stable performance and, thus, actual learning from both networks. Evolutionary GANs are helpful when many training objectives are available, but testing all of them could involve the repetitive training of several models, considering the great computing requirements that GANs demand in terms of time and hardware.

From a qualitative point of view, the conditionally generated images (Fig. 7) were consistent to those from the training in terms of cluster shape, texture, surrounds, shadows, among other features. While the visual similarity of the synthetic images was significant, it was not completely consistent with the training data. A cause for this could be the working image dimensions of  $128 \times 128$ , that by nature diminish the very fine details that could expose the fake source of the image—like, for example, grape stems, Fig. 2. Other possible causes may be the reduced size of the original dataset—around a hundred images—and the relatively low complexity of the adversarial network architectures (Fig. 3), that were designed accordingly to the constrained variability from the images taken under laboratory conditions. More complex architectures may yield better quality images to the human eye, like attention mechanisms [58] or residual blocks [8]. Still, the objective of this work (*i.e.*, the improvement of deep learning models for assessing berry number) was reached, meaning that the developed regression networks did not really benefit from, and thus did not attend to, some fine details from the images. The general pipeline illustrated in Fig. 1 can certainly be generalized





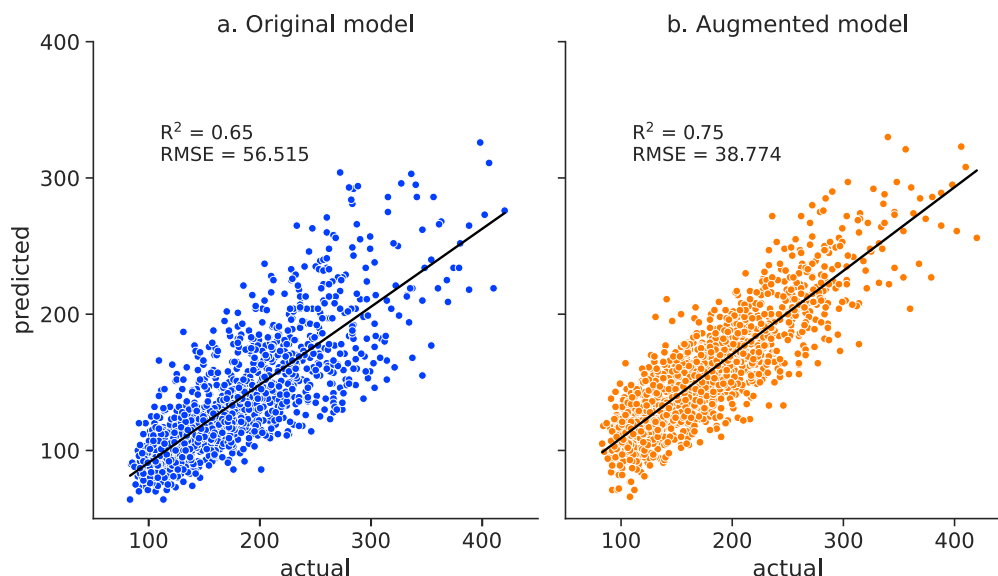
**Fig. 7.** Images created by the best generator. Each image was generated conditioned to a berry number within the ranges stated from a. to d. These ranges were selected so that image groups (grape clusters) can be compared to those from Fig. 2.



**Fig. 8.** Training and validation metrics ( $RMSE = \sqrt{\text{loss}}$ , given in number of berries) from the training of the original (a.) and augmented (b.) models.

for similar agricultural problems, but in most cases different conditions need to be considered. For example, when dealing with images taken under natural conditions, the generative networks used in our experiments (Fig. 3) is likely to be inadequate, and

more sophisticated architectures would probably be necessary. This is because natural conditions introduce a high level of variability and complexity, including changes in lighting, background, and grape berry appearance. As a result, the GAN must be able



**Fig. 9.** Correlation plots between the actual values from the test dataset (number of berries in images) and the values predicted by a. the original and b. the augmented models. RMSE values are given in berry number. The continuous, black line represents the regression line within each plot.

to recognize and model these variations to accurately assess grape berry numbers. Additionally, the GAN should be able to learn from a diverse set of natural images to ensure that it can generalize to new and unseen scenarios, implying a considerable larger number of base images.

Regarding data augmentation techniques, it is worth discussing alternatives to GAN image generation, like AutoAugment [59], an automated approach for data augmentation with reinforcement to search and optimize augmentation policies based on traditional image transforms. Compared to the GAN development proposed in this work, AutoAugment has the advantage of automatically search for augmentation policies that are tailored to specific datasets and tasks. This eliminates the need for manual experimentation and tuning of hyperparameters, a hard task in GAN training. Nevertheless, AutoAugment is limited to predefined transformations and may not capture complex patterns or generate entirely new data instances like GANs. Additionally, AutoAugment alone cannot provide for supervised augmentation, *i.e.*, allowing to condition the image generation, like conditional GANs in this work do. As there are no many reported applications of AutoAugment in agriculture [60], the line is open and interesting for further research and comparison.

The behavior of the training and validation curves from both models (Fig. 8a and b) demonstrates that the selected CNN architecture for the regression model did not need for further epochs, as a stability phase was reached in both models, approximately at the same number of iterations. While the final training error was virtually the same in both models and very proximal to 0, meaning that the network captured adequately all the features and diversity of the training data, the validation error of the augmented model was comparatively lower than the original one by around 20 berries, demonstrating that the synthetic images improved the generalization capability of the model. It is important to highlight the fact that both regression models were trained with the same number of samples (the original model with basic data augmentation from image transforms— Section 3.4—*versus* data augmentation from GAN's images), so the improvement in the regression task was not a matter of number of samples for training, and can be attributed to the quality of the labeled images conditionally generated after the evolutionary training of the GAN. This was also consistent to the results obtained from the test dataset (Fig. 9), in which the improvement in error was almost

of 20 berries in favor of the augmented model, again supporting the strength of GANs for useful data augmentation. Moreover, the results from Fig. 9b can also be evaluated in absolute and isolated terms. The augmented model, from an original amount of 97 labeled images, was able to predict a collection of images more than one order of magnitude larger (1329 images) with high correlation ( $R^2 = 0.75$ ) and low error compared to the standard deviation of the dataset (Table 3). This was possible even when the test images were considerably adulterated with unmodified white labels present in the images, or with the common presence of dirt and water spots that were not present in the train dataset. This is another proof of the good capability of CNNs for the automated extraction of useful features, previous modeling their relation to the target variable.

The results obtained in this paper open new possibilities of designing similar and adapted approaches that are worth researching. In the case of yield assessment in grapevines, it would be beneficial working with images having different indoor conditions, other grape phenological stages, etc. Grape variety is also an interesting factor to consider. Although the data used for training the generative networks only came from one grape variety, the proposed pipeline is likely to be applicable to other grape varieties or datasets with higher variability. The training of the GAN involved images from a single variety, so the particular generator obtained might arguably be considered as specialized on one variety. Still, the grape phenological stage used in our experiments (so called “pea-sized berries”) makes easier for the images to be useful for the prediction of bunches from different varieties, but the same bunch state. The improved performance of CNN regression models on the augmented images generated by the GAN supports the potential generalizability of the proposed method, specially considering that the external test dataset included more grape varieties or different conditions than the ones used in training. However, higher generation/prediction quality might be expected from dataset with a larger range of variability, included several varieties. Also, while we recognize this work as a first step on the use of GAN-based data augmentation in grape yield prediction, we acknowledge that higher performance levels of deep learning models could be achieved by comparing different architectures other than the CNN used here. For this reason, future work can explore the impact of other deep learning models, such as different CNN architectures or other state-of-the-art

mechanisms like Vision Transformers. Finally, data augmentation with Evolutionary GANs could be very useful for other important aspects in viticulture, like disease detection and assessment, or in-field grapevine monitoring.

## 6. Conclusion

Evolutionary conditional generative adversarial networks were able to produce synthetic images that improved considerably the performance of the augmented regression model when compared under the same conditions to the original model, both based on convolutional networks. The development presented in this paper demonstrated that the evolutionary training of GANs resulted in a stable behavior that produced satisfactory images, qualitative and quantitatively, thus benefiting the mechanisms of evolutionary computing and reducing the need for separately testing different GAN training objectives. The augmented model by itself was able to successfully assess the berry number per cluster in an external dataset ten times larger than the base dataset, with images containing noise visible elements not present during training. This demonstrates the suitability of generative models for the assessment of berry number in viticulture with deep learning.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors do not have permission to share data.

## Acknowledgments

This work has been supported by the Spanish State Research Agency through project PID2019-105381GA-I00 (iScience). The authors want to thank Ignacio Barrio and Rubén Ñíguez for their technical support and help in data acquisition, labeling and curation.

## References

- [1] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: An overview, *IEEE Signal Process. Mag.* 35 (2018) 53–65.
- [2] P. Shamsolmoali, M. Zareapoor, E. Granger, H. Zhou, R. Wang, M.E. Celebi, J. Yang, Image synthesis with adversarial networks: A comprehensive survey and case studies, *Inf. Fusion* 72 (2021) 126–146.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, *Commun. ACM* 63 (2020) 139–144.
- [4] X. Mao, Q. Li, H. Xie, R.Y. Lau, Z. Wang, S. Paul Smolley, Least squares generative adversarial networks, in: *Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.
- [5] J.H. Lim, J.C. Ye, Geometric gan, 2017, arXiv preprint arXiv:1705.02894.
- [6] Y. Mroueh, T. Sercu, Fisher GAN, in: *Advances in Neural Information Processing Systems*, Vol. 30, 2017.
- [7] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 214–223.
- [8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein GANs, in: *Advances in Neural Information Processing Systems*, Vol. 30, 2017.
- [9] N. Siddique, H. Adeli, *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*, John Wiley & Sons, 2013.
- [10] H. Al-Sahaf, Y. Bi, Q. Chen, A. Lensen, Y. Mei, Y. Sun, B. Tran, B. Xue, M. Zhang, A survey on evolutionary machine learning, *J. R. Soc. New Zealand* 49 (2019) 205–228.
- [11] S. Jain, D. Ramesh, D. Bhattacharya, A multi-objective algorithm for crop pattern optimization in agriculture, *Appl. Soft Comput.* 112 (2021) 107772.
- [12] E. Pazouki, Optimizing an irrigation treatment using an evolutionary algorithm and a knowledge discovery framework based on Deep models, *Appl. Soft Comput.* 133 (2023) 109940.
- [13] C. He, S. Huang, R. Cheng, K.C. Tan, Y. Jin, Evolutionary multiobjective optimization driven by generative adversarial networks (GANs), *IEEE Trans. Cybern.* 51 (2020) 3129–3142.
- [14] V. Costa, N. Lourenç, J. Correia, P. Machado, Exploring the evolution of GANs through quality diversity, in: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 2020, pp. 297–305.
- [15] F. Liu, H. Wang, J. Zhang, Z. Fu, A. Zhou, J. Qi, Z. Li, EvoGAN: An evolutionary computation assisted GAN, *Neurocomputing* 469 (2022) 81–90.
- [16] Y. Liang, Z. Han, X. Nie, K. Ohkura, Improving generative adversarial network with multiple generators by evolutionary algorithms, *Artif. Life Robot.* 27 (2022) 761–769.
- [17] C. Wang, C. Xu, X. Yao, D. Tao, Evolutionary generative adversarial networks, *IEEE Trans. Evol. Comput.* 23 (2019) 921–934.
- [18] J.A. Taylor, B. Tisseyre, C. Leroux, A simple index to determine if within-field spatial production variation exhibits potential management effects: Application in vineyards using yield monitor data, *Precis. Agric.* 20 (2019) 880–895.
- [19] J. Tardaguila, M. Stoll, S. Gutiérrez, T. Proffitt, M.P. Diago, Smart applications and digital technologies in viticulture: A review, *Smart Agric. Technol.* 1 (2021) 100005.
- [20] R. Bramley, R. Hamilton, Understanding variability in winegrape production systems: 1. Within vineyard variation in yield over several vintages, *Aust. J. Grape Wine Res.* 10 (2004) 32–45.
- [21] F. Palacios, M.P. Diago, J. Tardaguila, A non-invasive method based on computer vision for grapevine cluster compactness assessment using a mobile sensing platform under field conditions, *Sensors* 19 (2019) 3799.
- [22] S. Nuske, S. Achar, T. Bates, S. Narasimhan, S. Singh, Yield estimation in vineyards by visual grape detection, in: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 2352–2358.
- [23] A. Aquino, B. Millan, S. Gutiérrez, J. Tardaguila, Grapevine flower estimation by applying artificial vision techniques on images with uncontrolled scene and multi-model analysis, *Comput. Electron. Agric.* 119 (2015) 92–104.
- [24] L. Zabawa, A. Kicherer, L. Klingbeil, R. Töpfer, H. Kuhlmann, R. Roscher, Counting of grapevine berries in images via semantic segmentation using convolutional neural networks, *ISPRS J. Photogramm. Remote Sens.* 164 (2020) 73–83.
- [25] D. Font, T. Pallejà, M. Tresanchez, M. Teixidó, D. Martínez, J. Moreno, J. Palacín, Counting red grapes in vineyards by detecting specular spherical reflection peaks in RGB images obtained at night with artificial illumination, *Comput. Electron. Agric.* 108 (2014) 105–111.
- [26] Y. Lu, D. Chen, E. Olaniyi, Y. Huang, Generative adversarial networks (gans) for image augmentation in agriculture: A systematic review, *Comput. Electron. Agric.* 200 (2022) 107208.
- [27] C. Karam, M. Awad, Y. Abou Jawdah, N. Ezzeddine, A. Fardoun, GAN-based semi-automated augmentation online tool for agricultural pest detection: A case study on whiteflies, *Frontiers in Plant Science* 13 (2022) 813050.
- [28] D. Wang, R. Vinson, M. Holmes, G. Seibel, A. Bechar, S. Nof, Y. Tao, Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (OR-AC-GAN), *Sci. Rep.* 9 (2019) 1–14.
- [29] A. Abbas, S. Jain, M. Gour, S. Vankudothu, Tomato plant disease detection using transfer learning with C-GAN synthetic images, *Comput. Electron. Agric.* 187 (2021) 106279.
- [30] B. Liu, C. Tan, S. Li, J. He, H. Wang, A data augmentation method based on generative adversarial networks for grape leaf disease identification, *IEEE Access* 8 (2020) 102188–102198.
- [31] C. Zhou, Z. Zhang, S. Zhou, J. Xing, Q. Wu, J. Song, Grape leaf spot identification under limited samples by fine grained-GAN, *IEEE Access* 9 (2021) 100480–100489.
- [32] H. Jin, Y. Li, J. Qi, J. Feng, D. Tian, W. Mu, GrapeGAN: Unsupervised image enhancement for improved grape leaf disease recognition, *Comput. Electron. Agric.* 198 (2022) 107055.
- [33] M. Fawakherji, C. Potena, A. Pretto, D.D. Bloisi, D. Nardi, Multi-spectral image synthesis for crop/weed segmentation in precision farming, *Robot. Auton. Syst.* 146 (2021) 103861.
- [34] N. Chebrolov, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, C. Stachniss, Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields, *Int. J. Robot. Res.* 36 (2017) 1045–1052.
- [35] B. Espejo-García, N. Mylonas, L. Athanasakos, E. Vali, S. Fountas, Combining generative adversarial networks and agricultural transfer learning for weeds identification, *Biosyst. Eng.* 204 (2021) 79–89.
- [36] Z. Luo, H. Yu, Y. Zhang, Pine cone detection using boundary equilibrium generative adversarial networks and improved YOLOv3 model, *Sensors* 20 (2020).
- [37] J.R. Olatunji, G.P. Redding, C.L. Rowe, A.R. East, Reconstruction of kiwifruit fruit geometry using a cgan trained on a synthetic dataset, *Comput. Electron. Agric.* 177 (2020).



- [38] E. Bellocchio, G. Costante, S. Cascianelli, M.L. Fravolini, P. Valigi, Combining domain adaptation and spatial consistency for unseen fruits counting: a quasi-unsupervised approach, *IEEE Robot. Autom. Lett.* 5 (2020) 1079–1086.
- [39] J. Kierdorf, I. Weber, A. Kicherer, L. Zabawa, L. Drees, R. Roscher, Behind the leaves: Estimation of occluded grapevine berries with conditional generative adversarial networks, *Front. Artif. Intell.* 5 (2022).
- [40] Z. Fei, A.G. Olenskyj, B.N. Bailey, M. Earles, Enlisting 3D crop models and GANs for more data efficient and generalizable fruit detection, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1269–1277.
- [41] S. Gutiérrez, A. Wendel, J. Underwood, Spectral filter design based on in-field hyperspectral imaging and machine learning for mango ripeness estimation, *Comput. Electron. Agric.* 164 (2019) 104890.
- [42] K. Nagasubramanian, S. Jones, S. Sarkar, A.K. Singh, A. Singh, B. Ganapathysubramanian, Hyperspectral band selection using genetic algorithm and support vector machines for early identification of charcoal rot disease in soybean stems, *Plant Methods* 14 (2018) 1–13.
- [43] X. Li, Y. Wei, J. Xu, X. Feng, F. Wu, R. Zhou, J. Jin, K. Xu, X. Yu, Y. He, Ssc and ph for sweet assessment and maturity classification of harvested cherry fruit based on nir hyperspectral imaging technology, *Postharvest Biol. Technol.* 143 (2018) 112–118.
- [44] S. Gutiérrez, A. Wendel, J. Underwood, Ground based hyperspectral imaging for extensive mango yield estimation, *Comput. Electron. Agric.* 157 (2019) 126–135.
- [45] L. Bi, G. Hu, A genetic algorithm-assisted deep learning approach for crop yield prediction, *Soft Comput.* 25 (2021) 10617–10628.
- [46] C.-C. Lin, D.-J. Deng, J.-R. Kang, W.-Y. Liu, A dynamical simplified swarm optimization algorithm for the multiobjective annual crop planning problem conserving groundwater for sustainability, *IEEE Trans. Ind. Inform.* 17 (2020) 4401–4410.
- [47] J. Luus, D. Els, C. Poblete-Echeverría, Automating reference temperature measurements for crop water stress index calculations: A case study on grapevines, *Comput. Electron. Agric.* 202 (2022) 107329.
- [48] A. Chaudhary, R. Thakur, S. Kolhe, R. Kamal, A particle swarm optimization based ensemble for vegetable crop disease recognition, *Comput. Electron. Agric.* 178 (2020) 105747.
- [49] S. Sengupta, A.K. Das, Particle swarm optimization based incremental classifier design for rice disease prediction, *Comput. Electron. Agric.* 140 (2017) 443–451.
- [50] X. Sun, L. Yang, Q. Shen, L. Ni, B. Zhang, A new ant colony optimization algorithm based band selection method, in: *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, IEEE, 2014, pp. 1–4.
- [51] T. Liu, T. Xu, F. Yu, Q. Yuan, Z. Guo, B. Xu, A method combining elm and plsr (elm-pls) for estimating chlorophyll content in rice with feature bands extracted by an improved ant colony optimization algorithm, *Comput. Electron. Agric.* 186 (2021) 106177.
- [52] T. Kalampokas, E. Vrochidou, G.A. Papakostas, T. Pachidis, V.G. Kaburlasos, Grape stem detection using regression convolutional neural networks, *Comput. Electron. Agric.* 186 (2021) 106220.
- [53] J. Massah, K.A. Vakilian, M. Shabanian, S.M. Shariatmadari, Design, development, and performance evaluation of a robot for yield estimation of kiwifruit, *Comput. Electron. Agric.* 185 (2021) 106132.
- [54] Y.Y. Hilal, W. Ishak, A. Yahya, Z.H. Asha'ari, Development of genetic algorithm for optimization of yield models in oil palm production, *Chilean J. Agric. Res.* 78 (2018) 228–237.
- [55] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, arXiv preprint arXiv:1411.1784.
- [56] B.G. Coombe, Growth Stages of the Grapevine: Adoption of a system for identifying grapevine growth stages, *Aust. J. Grape Wine Res.* 1 (1995) 104–110.
- [57] A. Aquino, I. Barrio, M.P. Diago, B. Millan, J. Tardaguila, vitisBerry: An Android-smartphone application to early evaluate the number of grapevine berries by means of image analysis, *Comput. Electron. Agric.* 148 (2018).
- [58] J. Chen, D. Zhang, M. Suzaiddola, A. Zeb, Identifying crop diseases using attention embedded mobilenet-v2 model, *Appl. Soft Comput.* 113 (2021) 107901.
- [59] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, Autoaugment: Learning augmentation policies from data, 2018, arXiv preprint arXiv:1805.09501.
- [60] B. Li, J. Tang, Y. Zhang, X. Xie, Ensemble of the deep convolutional network for multiclass of plant disease classification using leaf images, *Int. J. Pattern Recognit. Artif. Intell.* 36 (2022) 2250016.