Received: 28 July 2020 | Revised: 28 October 2020 | Accepted: 1 November 2020

DOI: 10.1002/cem.3320

SPECIAL ISSUE - RESEARCH ARTICLE

Journal of CHEMOMETRICS WILEY

Varietal quality control in the nursery plant industry using computer vision and deep learning techniques

Sergio Borraz-Martínez[1,2] | **Francesc Tarrés**[5] | **Ricard Boqué**[3] | **Mariàngela Mestre**[2] | **Joan Simó**[1,4] | **Anna Gras**[1]

[1]Department of Agri-Food Engineering and Biotechnology, Universitat Politècnica de Catalunya, Campus Baix Llobregat, Esteve Terrades 8, Castelldefels, 08860, Spain

[2]Center of Initial Materials, Agromillora Iberia S.L.U, Ctra. BV-2247 km. 3, Sant Sadurní d'Anoia, 08770, Spain

[3]Department of Analytical Chemistry and Organic Chemistry, Universitat Rovira i Virgili, Campus Sescelades, Tarragona, 43007, Spain

[4]Fundació Miquel Agustí, Campus Baix Llobregat, Esteve Terrades 8, Castelldefels, 08860, Spain

[5]Signal Theory and Communications Department, UPC Barcelona Tech, Barcelona, Spain

Correspondence

Sergio Borraz-Martínez, Department of Agri-Food Engineering and Biotechnology, Universitat Politècnica de Catalunya, Campus Baix Llobregat, Esteve Terrades 8, Castelldefels 08860, Spain.
Email: sborrazma@gmail.com; sergio.borraz@upc.edu

Funding information

Spanish Ministry of Economy and Competitiveness, Grant/Award Numbers: AGL2015-70106-R, AEI/FEDER, UE, AEI/FEDER, UE, AGL2015-70106-R; Industrial Doctorates Plan of the Secretariat of Universities and Research of the Department of Economy and Knowledge of the Generalitat of Catalonia, Grant/Award Number: DI-COF 2017

Abstract

Computer vision coupled to deep learning is a promising technique with multiple applications in the industry. In this work, the potential of this technique has been assessed in the classification of two varieties of almond trees (*Prunus dulcis*), *Soleta* and *Pentacebas*. For that, a convolutional neural network named VGG16 was used. The most appropriate configuration for model training was studied, which included the comparison between two different filling modes (*reflect* and *nearest*) in the data augmentation step, the evaluation of the batch size and the analysis of the image sizes. The robustness of the model was also checked, and information was obtained about how the model extracts the information from the images. The results showed that the *reflect* fill mode was more effective than the *nearest* one. The best results were obtained using batches with 30 and 40 images, with an image size of (224 × 224) pixels. The verification of the robustness proved the capability of the technique as a promising tool for plant varietal identification.

KEYWORDS

computer vision, convolutional neural network, deep learning, nursery plant, varietal mixture

[Correction added on 30 January 2021, after first online publication: Peer review history statement has been added.]

1 | INTRODUCTION

Computer vision coupled to deep learning (CV-DL) techniques are getting more popularity in different areas such as pedestrian detection and tracking,[1] human health,[2] and structural health monitoring in civil structures,[3] especially for their great capability of object recognition. This capability has been shown in the successive Large Scale Visual Recognition Challenge (LSVRC) competitions.[4] Many studies have been developed to solve different problems and overcome challenges in agro-food quality control,[5,6] precision agriculture,[7,8] control of fruits and vegetables,[9,10] plant disease detection,[11,12] and plant phenotyping.[13,14]

We have applied CV-DL to solve the problem of varietal discrimination in the nursery plant industry. During the production process of the plants, varietal mixtures can appear in the batches due to traceability errors. When this happens and the product is commercialized, customers complain, the company must pay a fine, and its corporative image is damaged. Nowadays, the varietal identification is usually performed by means molecular methods based on DNA analysis, specifically DNA amplification by the polymerase chain reaction (PCR)[15] together with analysis of genetic variations, such as single nucleotide polymorphism (SNP).[16] Despite these molecular techniques have an excellent accuracy, they are very expensive for routine analysis of a large number of samples. Other studies have focused on the implementation of near-infrared (NIR) spectroscopy as a tool for varietal control.[17,18]

In the literature, there are studies related to plant identification using CV.[19–22] In our case, the challenge is to discriminate between two almond trees (*Prunus dulcis*) varieties, *Soleta* and *Pentacebas*, which are morphologically very similar, genetically close, and almost indistinguishable for the human eye. Figure 1 shows images of both varieties.

In other CV systems, the final classification relies on the features designed by the engineer. The programmer has to choose the features of the image that will be more representative to classify the leaves of the plant. The final quality of the system depends on this previous knowledge of the best features. But the designer does not know if the features applied are the best or another selection, or combination would produce better results. In this work, it was wanted to obtain a classification system without using any prior knowledge about the best-suited features. With DL, the programmer inputs data as well as the outputs expected from the data, and the system automatically searches the best features.[23] Therefore, CV-DL is more suitable to face complex problems, as the case of study.

For that, a database with images from the two varieties of interest was created, and a convolutional neural network (CNN) was used to build a classification model. The specific objectives of the study are (1) to study the training configuration of the neural network; (2) to explore what features of the plant is the model using in the modeling/training step; and (3) to check the robustness of the technique.

2 | MATERIAL AND METHODS

2.1 | Image acquisition

The red–green–blue (RGB) images used were acquired using a photo studio box and a Nikon D5300 camera. Two LED lamps (Color temperature 5500 K, color rendering index of 93%, and 13 000 lumens) were placed at the bottom of the box to achieve the correct illumination of the plant. The position of the light source was optimized in order to highlight

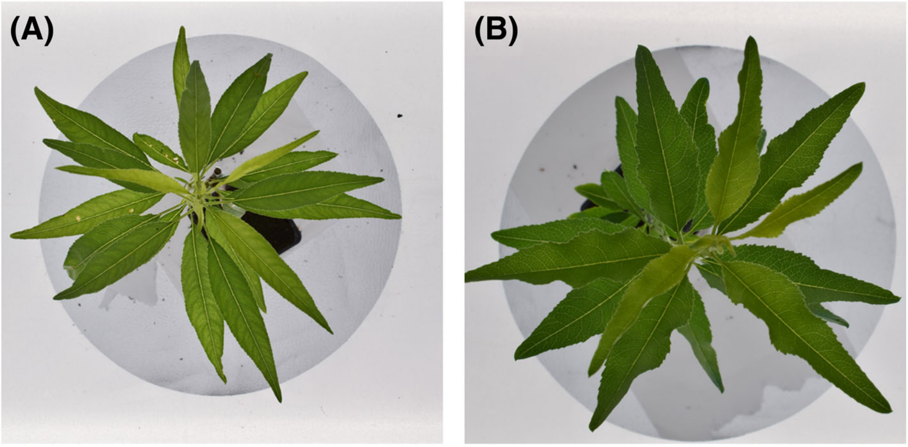

FIGURE 1 Images of almond tree plants. (A) *Pentacebas* variety; (B) *Soleta* variety

the features of the leaves: the venation and the borders. To avoid pictures with too much light intensity or reflectance effects, a textile diffuser was placed between the light source and the canopy of the plant. The images were collected from the top of the plants, thus obtaining an apical view. Before image acquisition, the setup of the camera was adjusted (diaphragm opening = *f*/29, shutter spit = 1/30, and International Organization for Standardization [ISO] sensitivity = 200). This parameter set was selected to produce a well-balanced histogram. In total, 824 images (one image per plant) were taken from two *P. dulcis* varieties, 412 images of *Soleta*, and 412 of *Pentacebas*. Table 1 shows more details about the number of images used in the study.

2.2 | Image preparation

The rectangular raw images obtained from the camera (image size of 6000 × 4000 pixels) were cropped to an image size range between 4000 × 4000 and 2500 × 2500 pixels to obtain squared images, which constituted the dataset used in this work. The model was adjusted to be fed with images of 224 × 224 pixels, so just before introducing them into the model, their resolution was reduced automatically to 224 × 224 pixels. Moreover, the pixel component levels were normalized (1.0/255) in order to adjust their values between 0 and 1. These two last modifications are indispensable to decrease the computation time.

The whole dataset was randomly split into a training set (75% of images), a validation set (15% of images), and a test set (10% of images). Table 1 shows more details about the number of samples of the three datasets.

2.3 | Convolutional neural network

A CNN is a type of network architecture commonly used in image analysis. It is characterized by having a reduced number of tunable parameters, because each neuron of the convolutional layer represents a small kernel that is applied to the previous layer using a convolution operator. Therefore, not all neurons in one layer are connected to all neurons in the previous layer. The neurons of the CNN form three dimensional layers, with height, width, and depth, also named channels.[24] Next to the convolutional layers, there are pooling layers that reduce the resolution of the output features in higher-level layers. In this way, higher layers have a global view of the image contents while lower layers are specialized in detecting local features.

The Vgg16 CNN[25] was used to perform the study. Vgg16 was developed by the Geometry Group at the University of Oxford to obtain state of the art results in the ILSVRC-2014 competition.[4,26] The convolutional base of the CNN is constituted by 16 layers, which are ordered in five blocks. Moreover, after the convolutional base, the Vgg16 contains a classifier that is composed by three fully connected layers. Blocks 1 and 2 have two convolutional layers each one and 64 and 128 filters, respectively. Blocks 3–5 contain three convolutional layers and 256, 512, and 512 filters, respectively. Every convolutional layer use filters with a receptive field of 3 × 3. The convolution stride is 1 × 1 pixel, and the spatial padding of the convolutional layer input is such that the spatial resolution is preserved after convolution.[25] In addition, one max-pooling layer performed over a 2 × 2 pixel window is located at the end of each block. The classifier is composed of three fully connected layers. Figure 2 shows a scheme of the Vgg16 architecture.

The convolutional base of the Vgg16 neural network was downloaded pretrained (with the parameters adjusted) with ImageNet dataset[27] from TensorFlow.[28] ImageNet is formed by 1.4 million labeled images and 1000 different classes, and it is commonly used to pretrain CNNs in CV. Then, a new classifier was added, constituted by one flatten layer followed by two dense layers. The CNNs learn local features of the input image. For this reason, the purpose of these fully connected layers is to collect the local information and combine it through nonlinear filtering, which is learned

TABLE 1 Details of the number of samples in the datasets used

	Percentage %	*Pentacebas*	*Soleta*	Overall dataset
Training set	75	309	309	618
Validation set	15	62	62	124
Test set	10	41	41	82
Total		412	412	824

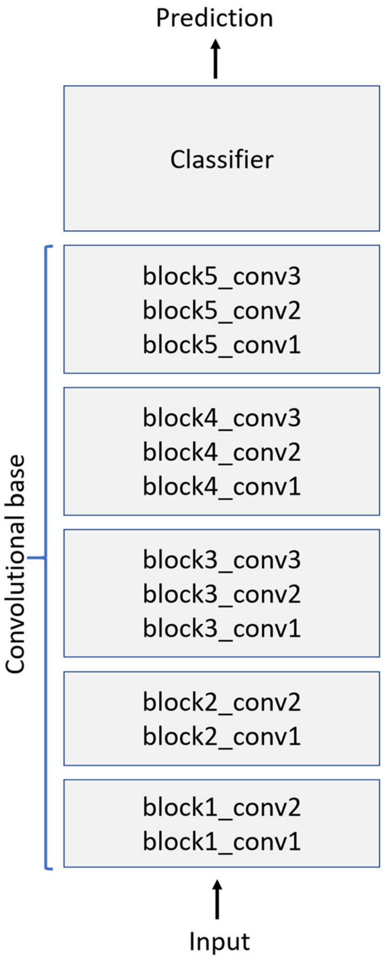

FIGURE 2 Vgg16 neural network architecture

from the training set. Usually, a minimum of two fully connected layers is used. The flatten layer allows transforming the 3-D tensor of the CNN to a 1-D tensor, which is required by the dense layer. The first dense layer included 256 neurons with rectified linear unit (ReLU) activation. This layer is necessary because it integrates all the local features learned by the last CNN layer, taking into account nonlinear combinations. Otherwise, without this inner layer, outputs would be a form of regression of the convolutional layers. Finally, the second dense layer contained one neuron with sigmoid activation. The output of the sigmoid will be an estimation of the confidence of having an input image of one or the other class.

During the training process, only the classifier and the convolutional layers of Block 5 were trained. The rest of parameters retained the values acquired from the training with the ImageNet dataset. We decided to train the Block 5 because this way, it is possible to improve the accuracy of the classification model. This method, which involves training the last CNN layer, is known as *Fine-tuning*.

2.4 | Data augmentation study

To avoid overfitting, a data augmentation technique was used. The overfitting occurs when the model fits very specific features of the training images that are not representative of the whole class. Thus, the accuracy in the classification of new images is reduced.

Data augmentation consists of obtaining multiple images from the original one. This is achieved by modifying the original image. The common modifications are rotation, shear, horizontal flip, zoom, and shift of the image. The goal is to increase the diversity of the training set, which helps to improve the robustness of the model and its prediction accuracy.

When using data augmentation, it is important to check the modified images. Misuse of data augmentation can lead to a reduction in the effectiveness of the model. An important aspect of data augmentation is what to do with pixels that are empty due to image rotation or shifting. This can be dealt with using the argument *fill_mode*. This argument defines the method for filling empty pixels. There are several strategies, the most common being *nearest* and *reflect*. The command *nearest* fills the pixels with the values of the contiguous pixels. In contrast, *reflect* generates a specular image. Both methods were compared to investigate which one generates less differences with the original images and thus leads to better results in the test.

Data augmentation was only applied to the training set because in the validation and test sets, it is preferable to use original images. In addition to the fill mode, other image modifications were applied: shear in a range of 0.2, zoom in a range of 0.4, horizontal flip, random rotation in a range of 180°, and width and height shift in a range of 0.2.

2.5 | Batch size and optimizer

Batch size is an important aspect to consider before starting to train a DL model. A batch is a set of input images that are passed through the network with fixed weights. The loss function is computed, and gradient backpropagation is applied to modify the weights for each batch. The steps per epoch (weight updates) come from the division of the number of images of the training/validation set by the batch size assigned. The validation step consists of the division of the number of images contained in the validation set by the batch size. Both concepts show the number of iterations by epoch. If the batch size is small, the number of iterations will be high, but each learning step will be performed with few images, which means little information. In contrast, if the batch size is large, the number of iterations will be low, but each learning step of the model will be robust because it will be based on many images.

We studied different batch sizes to evaluate what ratio between number of images per batch and number of iterations per epoch was the most suitable. The batch sizes used were 20 (30 steps per epoch/6 validation steps), 30 (20 steps per epoch/4 validation steps), and 40 (15 steps per epoch/3 validation steps) samples per batch. The number of epochs was defined based on the convergence between training and validation accuracies. Moreover, the progression of the classification error (loss value) along of the training process was taken into account. The models were trained in a range between 80 and 90 epochs, because overfitting was observed above 100 epochs.

2.6 | Comparison between image sizes

The Vgg16 CNN can be fed with two image sizes, 150 × 150 or 224 × 224 pixels. The advantage of using 224 × 224 pixel images is their higher resolution. This allows keeping more fine features, which can provide useful information for the identification and the discrimination of the classes. However, this strategy has two main disadvantages; first, the number of operations to adjust the parameters of the convolutional base of the Vgg16 neural network increases; second, the number of parameters of the classifier also increases, which translates in longer computing times.

In this study, 224 × 224 pixel images were used because we decided to prioritize resolution instead of time. However, we compared the performance when using images of 224 × 224 and 150 × 150 pixels. The dataset images, with a size range between 4000 × 4000 and 2500 × 2500 pixels, were automatically reduced to images with a resolution of 150 × 150 pixels through adjusting the setup of the model. To perform the comparison, we applied the training setup resulting from the study of the previous sections.

2.7 | Model robustness assessment

The robustness of the model trained with the configuration mentioned above was studied. To perform this analysis, five models were built using different random splits of the images into training, validation, and test sets. The variability in the test accuracy obtained helped us to assess the robustness of the DL technique to classify the *Prunus* varieties.

2.8 | Visualization of the intermediate activations

It is interesting to know how the model performs the analysis of the images in order to understand how it works. To try to answer this question, an input (image) was introduced, and the output was observed in different convolutional layers. The output is the activation of the layer, which provides information about how the input is decomposed into the different filters learned by the network.[23] By doing this, feature maps were created, which are composed of channels or filters. Each channel performs a specific analysis of the input, although all the filters in a layer perform a similar decomposition of the input image. This characteristic allows us to have an overview of the features extracted by each layer of the neural network. Finally, to plot the feature maps, a 2-D image was created with an independent plot of the contents of every channel of a layer of the CNN.

3 | RESULTS AND DISCUSSION

3.1 | Data augmentation study

The images were analyzed by using the *nearest* and *reflect* fill modes. Several modified images were obtained using both methods. Figure 3 shows three examples from each method. It is important to remark that the images shown in Figure 3 are an example of the images that are feeding the neural network.

Regarding the nearest fill mode, despite the modified images showed an appearance very similar to the original ones, several anomalies or deformations were detected, which are highlighted with a circle in Figure 3. These distortions arose because some of the empty pixels were filled using the value of the leaf pixels, which resulted in an elongation of the original leaves. The presence of this deformed leaves in the training images represents a significant alteration of the original images.

Concerning the *reflect* fill mode, the modified images showed an alteration too, which is highlighted with a square in Figure 3. The *reflect* fill mode generated a deformation of the original images too but with the difference that created a specular image of the original one.

Table 2 shows the results of the DL models. Using the *nearest* fill mode, an accuracy average of 95.1% in the test set was obtained, whereas the *reflect* fill mode provided an average accuracy of 96.7%. When the results obtained using the different batch sizes, in all cases, better results were obtained with the *reflect* fill mode. This result confirms that image deformations caused by the *nearest* fill mode negatively affected the accuracy of the model. To understand better this result, it is important to remark that data augmentation is only used during the training of the model. Therefore, the

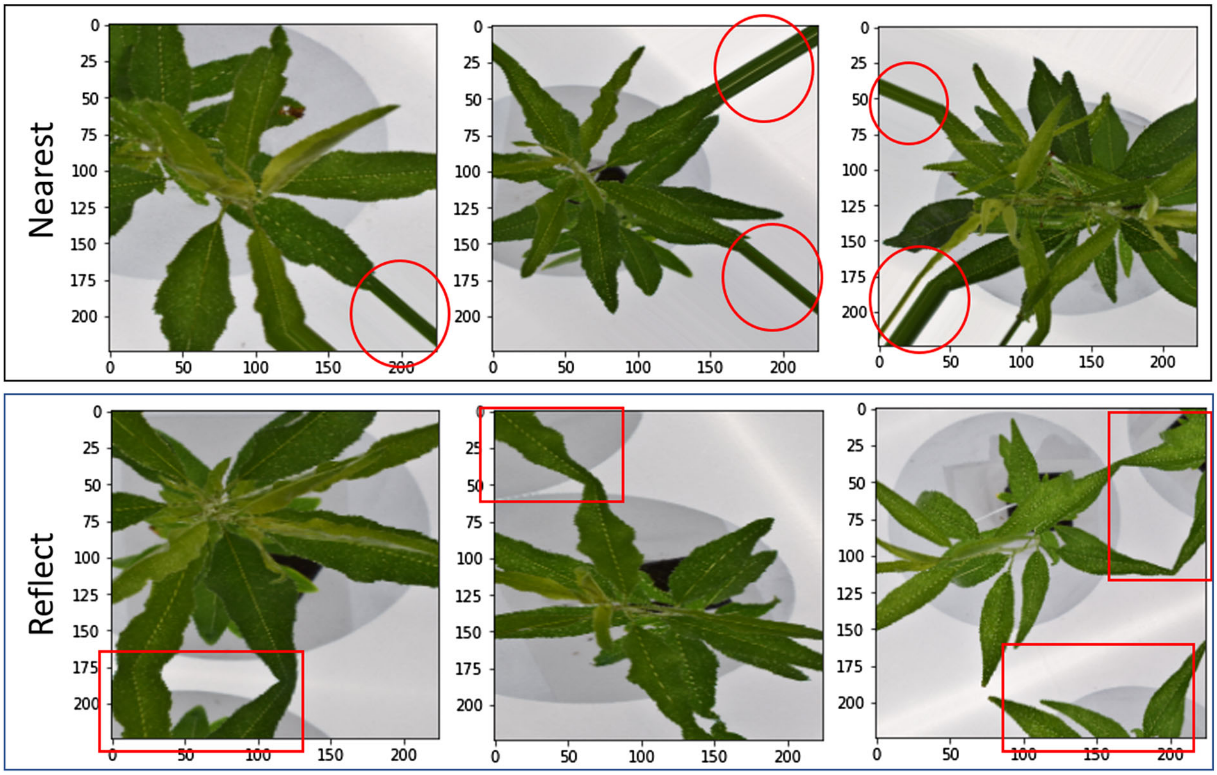

FIGURE 3 Study of the image modifications performed by *nearest* and *reflect* fill modes

TABLE 2 Result obtained in data augmentation and batch size studies

Data augmentation	Batch size	Test accuracy
Nearest	20	93.9
	30	96.4
	40	95.1
Reflect	20	95.1
	30	**97.6**
	40	**97.6**

TABLE 3 Comparison between images with 150 × 150 and 224 × 224 pixels

Size of the images	Test accuracy	Training time (s)
150 × 150	95.1	**230**
224 × 224	**97.6**	275

model will learn any alteration generated by data augmentation, even if the features are false. In addition, these deformations will not be present in the validation and test sets because data augmentation was not applied there. Thus, the *nearest* fill mode increased the gap between the training set and the test set.

3.2 | Batch size and optimizer study

Three different batch sizes, 20, 30, and 40, were studied. As it can be seen in Table 2, the best result of this comparison was achieved using the *reflect* fill mode together with a batch size of 30 and 40, reaching a 97.5% accuracy in the test set. In contrast, the worst result was obtained by applying the *nearest* fill mode together with a batch size of 20. Although a batch size of 20 allows a greater number of iterations, each learning step is performed based on a smaller number of images, and this affects the test accuracy. It seems more adequate to use a batch size of 30 or 40 images. Both provided the same result using the *reflect* fill mode, but the batch size of 30 achieved a better accuracy using the *nearest* fill mode.

3.3 | Comparison of image sizes

Table 3 shows the results of the comparison between the different image sizes. With 150 × 150 pixel images, an accuracy of 95.1% was obtained in the test set, which is the same accuracy obtained using 224 × 224 pixel images, but with a batch size of 20. This indicates that using images of 150 × 150 pixels, it is possible to obtain results comparable with those obtained using images of 224 × 224 pixels. However, a better comparison is performed with the results obtained using a batch size of 30. In this case, the model trained with 224 × 224 pixel images provided a better result but took an average of 16.4% more computing time than the model trained with 150 × 150 pixel images. When using 224 × 224 pixel images, the Vgg16 had 13.5 million trainable parameters, while using 150 × 150 pixel images the network had 9.1 million of trainable parameters, that is, a 32.6% less parameters. The increase in parameters is higher than the increase in training time. The differences in accuracy are not so great, but the time could be important depending on the requirements of the application.

3.4 | Assessment of model robustness

To assess the robustness of the model and discard the possibility that a single choice of the test set was favorable to obtain good results, six models were built with different training, validation, and test sets, which were randomly split. The results are shown in Table 4.

TABLE 4 Study of the robustness of the model

Models	Test accuracy
1	96.3
2	96.3
3	97.6
4	96.3
5	97.6
Average	**96.8**

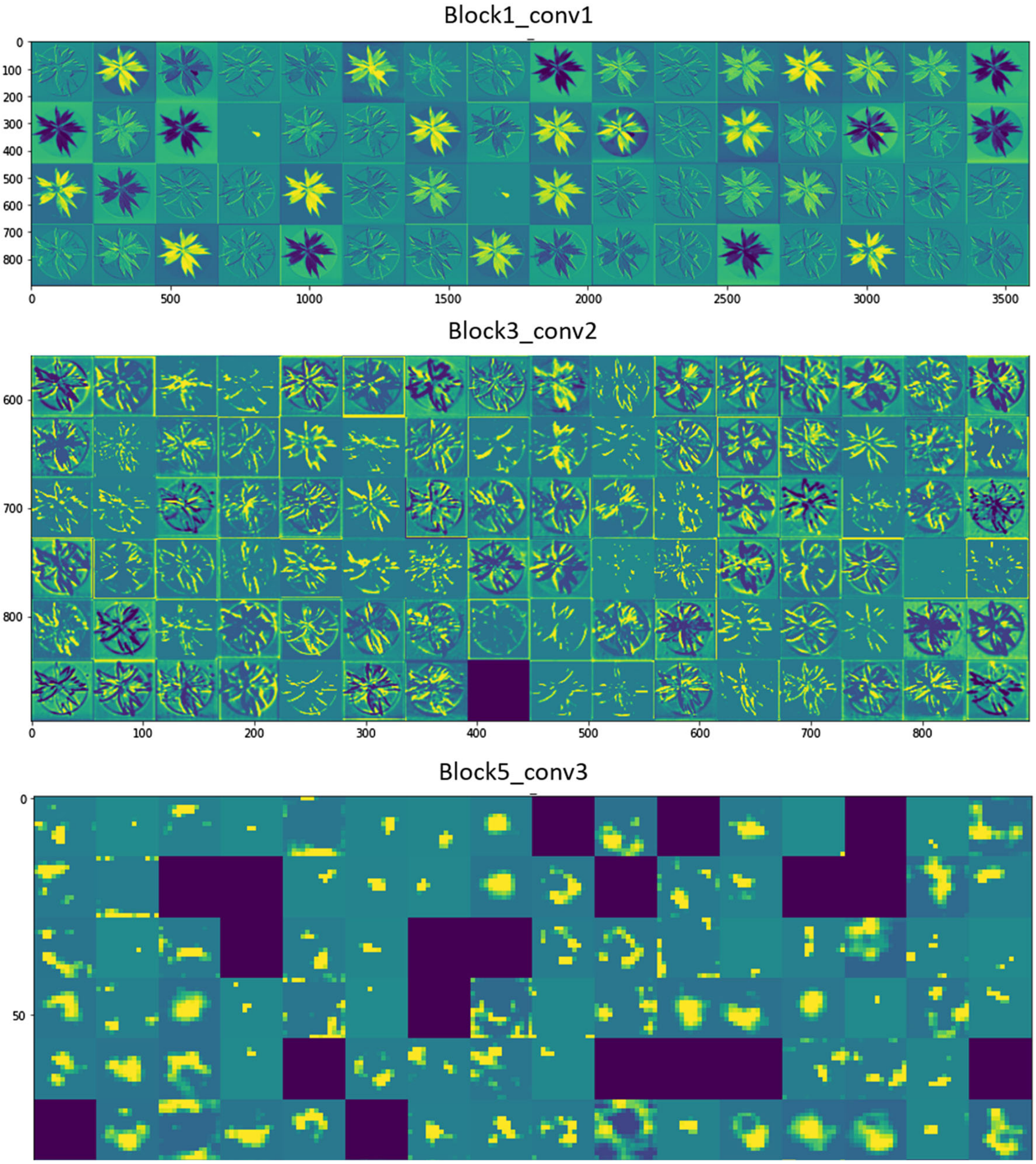

FIGURE 4 Intermediate activations of the neural network

The average accuracy obtained was 96.8%, which was a little lower than the best result achieved, 97.6% accuracy, which was reached twice. It is important to note that the difference between 96.3% and 97.6% corresponds to just a mistake in one image. Therefore, all the models showed similar behavior.

It is important to highlight that this robustness study was developed under specific illumination conditions, which are described in the Section 2. Therefore, a change in these conditions could yield different results. For this reason, it will be important to keep the same illumination conditions in future implementations.

3.5 | Feature maps

The feature maps of the 18 layers of the convolutional base of the Vgg16 (convolutional and max pooling layers included) were studied. From these 18 layers, we chose three representative layers of the CNN, because of the large size of the images. Figure 4 shows the progress of the image along the neural network by visualizing the intermediate activations. It is important to note that due to the high number of filters in each layer, Figure 4 only shows a subset of them.

In the first layers, which were represented by the block1_conv1 layer, it is easy to identify the plant in the image. This indicates that this layer works with the whole image and retains almost all the information present in the original image. For this reason, the activation of the filters showed images like the input image. The channels of this first layer process low-level information such as contours, edges, corners, and therefore extract information about the leaf shape and margins. As we move into the neural network, the plant is increasingly more difficult to recognize. In the block3_conv2, which is located in the middle of the network, it was only possible to identify small parts of the plants because the feature map was less visually interpretable. This layer encodes higher-level features, such as parts of the leaves, which implies less information about the visual contents, but more information related to the class of the input image. When observing the activation of the last layer of the convolutional base, block5_-conv3, the abstraction level is very high, and no part resembling the plant can be identified. As the block5_conv3 layer encodes higher-level concepts than block3_conv1 layer, the visual interpretation of the feature maps is even more abstract.

When comparing the resulting image in each filter of the same layer, we were able to observe the activation similarities between them, which reaffirms the idea that each layer works as an analysis block, where each channel performs an own extraction of information but all of them in the same analysis level.

It is also remarkable that, as we go into the CNN layers, the number of filters in blank increases. In the first layer, all filters are activated by the input image, but in the block3_conv2, one filter in blank was found, and in the last layer, the number of filters in blank was 19. This means the pattern encoded by the filter is not found in the input image.[23]

4 | CONCLUSIONS

In this study, we investigated the most suitable training configuration of a CNN to obtain the best results in the classification of two almond plant varieties of *P. dulcis*, *Soleta* and *Pentacebas*, and the potential of CV together with DL models was evaluated.

The importance of the argument *fill_mode* in the configuration of the data augmentation has been demonstrated. The results indicate that the *reflect* mode is more suitable than the *nearest* mode, because the latter generated some distortions of the images, which affected the accuracy of the model. In addition, the study of the batch size showed that batches with 30 or 40 images are the best option. The advantages and disadvantages of working with 224 × 224 or 150 × 150 pixel images were also analyzed. We found that the test accuracy was better for 224 × 224 pixel images but with an increase of the computing time. The robustness of the technique has also been proven. Finally, the study of how the neural network works allowed us to have an idea of the image analysis done by the model and what features of the plant are used to discriminate the varieties.

The results of this study show the capacity of CV-DL to discriminate *P. dulcis* varieties. Future works will try to expand the dataset and the number of varieties and to assess the use of specific images from different structures of the plant, such as leaves and their venation.

FUNDING INFORMATION

In addition, the authors thank the financial support of the Industrial Doctorates Plan of the Secretariat of Universities and Research of the Department of Economy and Knowledge of the Generalitat of Catalonia (project DI-COF 2017) and of the Spanish Ministry of Economy and Competitiveness (project AGL2015-70106-R, AEI/FEDER, UE).

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/CEM.3320.

ORCID

Sergio Borraz-Martínez https://orcid.org/0000-0002-5607-9462
Francesc Tarrés https://orcid.org/0000-0003-0920-4782
Ricard Boqué https://orcid.org/0000-0001-7311-4824
Joan Simó https://orcid.org/0000-0002-2853-3381
Anna Gras https://orcid.org/0000-0003-0111-7246

REFERENCES

1. Brunetti A, Buongiorno D, Trotta GF, Bevilacqua V. Computer vision and deep learning techniques for pedestrian detection and tracking: a survey. *Neurocomputing*. 2018;300:17-33. https://doi.org/10.1016/j.neucom.2018.01.092
2. Yeung S, Rinaldo F, Jopling J, et al. A computer vision system for deep learning-based detection of patient mobilization activities in the ICU. *Npj Digit Med*. 2019;2(1):1-5. https://doi.org/10.1038/s41746-019-0087-z
3. Bao Y, Tang Z, Li H, Zhang Y. Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Struct Heal Monit*. 2019;18(2):401-421. https://doi.org/10.1177/1475921718757405
4. ImageNet Large Scale Visual Recognition Competition 2014 (ILSVRC2014) http://www.image-net.org/challenges/LSVRC/2014/ (accessed May 7, 2020).
5. Ma J, Sun DW, Qu JH, et al. Applications of computer vision for assessing quality of Agri-food products: a review of recent research advances. *Crit Rev Food Sci Nutr*. 2016;56(1):113-127. https://doi.org/10.1080/10408398.2013.873885
6. Nyalala I, Okinda C, Nyalala L, et al. Tomato volume and mass estimation using computer vision and machine learning algorithms: cherry tomato model. *J Food Eng*. 2019;263:288-298. https://doi.org/10.1016/j.jfoodeng.2019.07.012
7. Patrício DI, Rieder R. Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review. *Computers and Electronics in Agriculture*. Elsevier B.V. October 1, 2018, pp 69–81. https://doi.org/10.1016/j.compag.2018.08.001
8. Zheng Y-Y, Kong J-L, Jin X-B, Wang X-Y, Su T-L, Zuo M. CropDeep: the crop vision dataset for deep-learning-based classification and detection in precision agriculture. *Sensors*. 2019;19(5):1058-1079. https://doi.org/10.3390/s19051058
9. Arakeri MP. Lakshmana. Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry. In *Procedia Computer Science*; Elsevier BV, 2016; Vol. 79, pp. 426–433. https://doi.org/10.1016/j.procs.2016.03.055
10. Bhargava A, Bansal A. Fruits and vegetables quality evaluation using computer vision: a review. *J King Saud Univ - Comput Inform Sci*. 2018;5:1-15. https://doi.org/10.1016/j.jksuci.2018.06.002
11. Wang G, Sun Y, Wang J. Automatic image-based plant disease severity estimation using deep learning. *Comput Intell Neurosci*. 2017; 2017:1-8. https://doi.org/10.1155/2017/2917536
12. Chouhan SS, Singh UP, Jain S. Applications of computer vision in plant pathology: a survey. *Arch Comput Meth Eng*. 2020;27(2):611-632. https://doi.org/10.1007/s11831-019-09324-0
13. Jiang Yu, Li Changying. Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping: A Review. *Plant Phenomics*. 2020;2020:1-22. https://doi.org/10.34133/2020/4152816
14. Bauer A, Bostrom AG, Ball J, et al. Combining computer vision and deep learning to enable ultra-Scale aerial Phenotyping and precision agriculture: a case study of lettuce production. *Hortic Res*. 2019;6(1):1-12. https://doi.org/10.1038/s41438-019-0151-5
15. Mullis K, Faloona F, Scharf S, Saiki R, Horn G, Erlich H. Specific enzymatic amplification of DNA in vitro: the polymerase chain reaction. *Cold Spring Harb Symp Quant Biol*. 1986;51(Pt 1):263-273. https://doi.org/10.1101/SQB.1986.051.01.032
16. Gupta PK, Roy JK, Prasad M. Single nucleotide polymorphism: a new paradigm for molecular marker technology and DNA polymorphism detection with emphasis on their use in plant. *Curr Sci*. 2001;80(4):524-535. https://doi.org/10.2307/24104242
17. Borraz-Martínez S, Simó J, Gras A, Mestre M, Boqué R. Multivariate classification of Prunus Dulcis varieties using leaves of nursery plants and near infrared spectroscopy. *Sci Rep*. 2019;9(1):1-9. https://doi.org/10.1038/s41598-019-56274-5
18. Borraz-Martínez S, Boqué R, Simó J, Mestre M, Gras A. Development of a methodology to analyze leaves from *Prunus Dulcis* varieties using near infrared spectroscopy. *Talanta*. 2019;204:320-328. https://doi.org/10.1016/j.talanta.2019.05.105
19. Barré P, Stöver BC, Müller KF, Steinhage V. LeafNet: a computer vision system for automatic plant species identification. *Eco Inform*. 2017;40:50-56. https://doi.org/10.1016/j.ecoinf.2017.05.005
20. Hemming J, Rath T. Computer-vision-based weed identification under field conditions using controlled lighting. *J Agric Eng Res*. 2001; 78(3):233-243. https://doi.org/10.1006/jaer.2000.0639

21. Luckow A, Cook M, Ashcraft N, Weill E, Djerekarov E, Vorster B. Deep Learning in the Automotive Industry: Applications and Tools. In *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*; Institute of Electrical and Electronics Engineers Inc, 2016; pp. 3759–3768. https://doi.org/10.1109/BigData.2016.7841045
22. Sun Y, Liu Y, Wang G, Zhang H. Deep learning for plant identification in natural. *Environ*. 2017;2017:1-6. https://doi.org/10.1155/2017/7361042
23. Chollet F. Deep learning with Python. Manning Publications, 2017.
24. Buduma N, Locascio N. Fundamentals of deep learning; 2019; Vol. 29.
25. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition; 2015.
26. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. *Int J Comput Vis*. 2015;115(3):211-252. https://doi.org/10.1007/s11263-015-0816-y
27. ImageNet. http://www.image-net.org/ (accessed May 7, 2020).
28. TensorFlow. https://www.tensorflow.org/ (accessed May 7, 2020).

How to cite this article: Borraz-Martínez S, Tarrés F, Boqué R, Mestre M, Simó J, Gras A. Varietal quality control in the nursery plant industry using computer vision and deep learning techniques. *Journal of Chemometrics*. 2021;e3320. https://doi.org/10.1002/cem.3320