Smart Agricultural Technology 8 (2024) 100470

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Deep learning networks for olive cultivar identification: A comprehensive analysis of convolutional neural networks

João Mendes [a,b,d,*], José Lima [a,d], Lino Costa [b], Nuno Rodrigues [c,d], Ana I. Pereira [a,d]

[a] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Bragança, 5300-253, Bragança, Portugal
[b] ALGORITMI Research Centre/LASI, University of Minho, Braga, 4710-057, Braga, Portugal
[c] Centro de Investigação de Montanha (CIMO), Instituto Politécnico de Bragança, Bragança, 5300-253, Bragança, Portugal
[d] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Bragança, 5300-253, Bragança, Portugal

ARTICLE INFO

Dataset link: https://dados.ipb._347_pt/dataset.xhtml?persistentId=doi:10.34620/dadosipb/TVYE8K

Keywords:
Convolutional neural networks
CNNs
Cultivar identification
Olive leaves
Image-based identification
Precision agriculture

ABSTRACT

Deep learning networks, more specifically convolutional neural networks, have shown a notable distinction when it comes to computer vision problems. Their versatility spans various domains, where they are applied for tasks such as classification and regression, contingent primarily on the availability of a representative dataset. This work explores the feasibility of employing this approach in the domain of agriculture, particularly within the context of olive growing. The objective is to enhance and facilitate cultivar identification techniques by using images of olive tree leaves. To achieve this, a comparative analysis involving ten distinct convolutional networks (VGG16, VGG19, ResNet50, ResNet152-V2, Inception V3, Inception ResNetV2, XCeption, MobileNet, MobileNetV2, EfficientNetB7) was conducted, all initiated with transfer learning as a common starting point. Also, the impact of adjusting network hyperparameters and structural elements was explored. For the training and evaluation of the networks, a dedicated dataset was created and made available, consisting of approximately 4200 images from the four most representative categories of the region. The findings of this study provide compelling evidence that the majority of the examined methods offer a robust foundation for cultivar identification, ensuring a high level of accuracy. Notably, the first nine methods consistently attain accuracy rates surpassing 95%, with the top three methods achieving an impressive 98% accuracy (ResNet50, EfficientNetB7). In practical terms, out of approximately 2016 images, 1976 were accurately classified. These results signify a substantial advancement in olive cultivar identification through computer vision techniques.

1. Introduction

Of extreme importance, olive growing is an active part in the daily lives of the world's population, from its contribution to gastronomy to medicinal aspects [1,2]. It is estimated that olive oil consumption almost doubled from 1990/91 to 2020/21, the data indicate that around 3.2 million tonnes of olive oil were consumed in 2021/22, also, in terms of table olives, this number is relevant, with an estimated consumption of 2.7 million tonnes in 2020/21 and slightly more in 2021/22. Associated with these consumptions, imports and exports of these products prove to be essential for the economies of the producing countries, with an estimated movement of 1.23 billion dollars in world trade.[1]

In addition to its economic importance, olive growing is largely associated with the history of producing countries. It is estimated that for 6 thousand years, people have been consuming olive oil and domesticated olives, making the olive tree one of the first fruit trees to be domesticated [3]. With more than 2600 different cultivars [4], its cultivation is a recurrent practice in dozens of countries on all continents, with the exception of Antarctica.

Each of these cultivars has physical and chemical characteristics that make it more adapted to specific climatic and geographic conditions. In addition to being adapted to different climates and geographical areas, the use of different cultivars for the production of olive oil ensures a more harmonious composition of the olive oil from an organoleptic point of view, since each one of them has its own characteristics [5,6].

* Corresponding author.
E-mail address: joao.cmendes@ipb.pt (J. Mendes).

[1] International olive oil, accessed: 2023-02-08 (2022). https://www.internationaloliveoil.org/changes-in-olive-oil-consumption.

https://doi.org/10.1016/j.atech.2024.100470
Received 26 January 2024; Received in revised form 29 April 2024; Accepted 10 May 2024
Available online 15 May 2024

In addition to flavor, using different varieties also allows the improvement of the time or date of harvest by selecting varieties with varying maturation periods.

In this way, the producer must be able to distinguish his batches and know how to specify the type of varieties present, thus valuing the qualities of each one of them, which will result in a final product, be it olive oil or table olives, with added value and different from most. This identification may be facilitated when dealing with recent olive groves the farmer has planted. However, it may become a more elaborate process when dealing with olive groves of some age and where there is no certainty of the varieties present. In these cases, it is necessary to resort to identification techniques.

These techniques mainly involve genetic analysis, such as random amplified polymorphic DNA (RAPD), molecular markers, microsatellites, chemometrics, and DNA Fingerprints [7–11]. Techniques that have high associated reliability rates, but are somewhat time-consuming processes that do not allow us to identify them on site, with the need to collect samples to analyze in the laboratory and with a relevant associated cost thinking that in the same olive grove there can be dozens of different cultivars. To combat these temporal adversities, some approaches using images and artificial intelligence methods have been studied [12–14]. The authors of these studies used different algorithms to make their identifications, namely partial least squares (PLS) and principal component analysis (PCA) to classify manually extracted features, statistical analyses and neural networks are also used. Although they present high accuracy rates, this type of technique uses the fruit of the tree (olives and seeds) to carry out its classification, thus being restricted to a specific time of the year to proceed with its identification. Typically, this period aligns with the harvest season, during which producers are fully engaged in the process of fruit collection, often precluding their capacity to undertake the requisite identification tasks.

This work aims to classify cultivars through tree leaves using deep learning methods to facilitate this process. This recognition will be facilitated this way since it can be carried out on-site, without impacting the tree being identified, and without the need to wait for a specific season to collect samples since the olive tree has permanent leaves. This ensures instant identification, without associated costs to the producer.

In this way, it is possible to state that this study makes the following contributions:

- Introduction of a novel approach to cultivar classification through tree leaves using deep learning methods.
- Facilitation of on-site recognition without impacting the tree being identified, thus eliminating the need to wait for a specific season to collect samples.
- Instant identification of cultivars without associated costs to the producer, leveraging the permanent leaves of the olive tree.
- Exploration of convolutional neural networks in a new problem domain, contributing to both agricultural and academic research.
- Provision of ample data to stimulate further research in the field of cultivar classification.

Consequently, a survey of related works will be carried out, consisting of the most used methods for variety identification and the most used for image classification, which will be presented in the section 2. Then, the materials and methods used to prepare this study will be addressed in the section 3, including a brief introduction about the dataset and its composition 3.1. As well as the classification models 3.2, evaluation metrics 3.3, and methodology 3.4. The main results will be presented in section 4, and then their discussion will be elaborated in section 5. Finally, the section 6 will include the conclusion and main future works.

2. Literature review

Identifying olive tree cultivars through their leaves is a typical classification problem. In this problem, the main objective is to categorize an input never seen before according to the connections and weights learned through the training examples. Besides the dataset, it is necessary to consider the available classification algorithms and understand what best suits the problem. In this way, a preliminary bibliographical study was carried out, where the algorithms used were chosen through the most used keywords. To facilitate this process, the bibliometrix software [15] was used to analyze the papers referring to the research "leaf image identification cultivar" or "leaf image identification cultivars" or "leaf image identification species" or "leaf image identification Varieties" in the database of Web Of Science data. This resulted in approximately 850 papers, whose authors' keywords are shown in Fig. 1.

As seen in Fig. 1, since 1994, papers on image processing for the identification of different cultivars have begun to appear. With the evolution of technology, from 2013 onwards, papers involving machine learning for these tasks started to appear, with algorithms considered deep learning appearing only two years later. With these, the applications of convolutional neural networks in this field also emerged. The current trend is undoubtedly the use of deep learning algorithms, considering the advantages already mentioned above compared to machine learning algorithms, such as the ease of automatic parameter extraction.

As a starting point for deep learning models, artificial neural networks (ANNs) have changed the way of computationally solving problems. With the use of this type of techniques, the model is adapted to the observational data, formulating its own solution to the problem in question [16].

Taking advantage of all the developments in recent years, the use of this type of algorithms is widely distributed across the most varied areas of application. Among them, it is possible to highlight computer vision, using Convolutional Neural Networks (CNNs). Since its emergence in 1989 [17], this type of algorithm has attracted attention for its capabilities in the field of computer vision as well as in tasks related to segmentation, classification, and detection in images, demonstrating exemplary ability in image processing competitions [18].

This type of network has, in part, a similar functioning to ANNs, composed of neurons that self-optimize through learning. However, they were designed to correct one of the most significant limitations of ANNs, image processing, and the computational complexity required in this type of application. For this purpose, the focus of its architecture is completely directed toward image processing, in this way, some organizational differences were made, starting with the organization of neurons that thus have three dimensions (height, width, and depth). Another significant difference in relation to traditional ANNs is the network structure, which consists of three types of layers, convolutional, pooling, and fully connected. In this type of network, the initial input goes through several stages, starting with the convolutional layers that are responsible for the convolutions through the various applied filters, here, the necessary feature maps for pattern recognition are extracted. Next, activation functions are used to guarantee a non-linear expression to the network, which will improve its performance and from which the rectified feature maps used in the next step, pooling, will result. Pooling layers are used to reduce the dimensionality of feature maps, thus also reducing the number of network parameters to be trained. Finally, in the last stage, the antecedent feature maps are flattened and serve as input to the fully-connected layers, here, the neurons are all connected together, and this is where the classification takes place through connections between the input layers, the previously defined hidden layers and the output layer, which is normally activated by a softmax function which results in an input value normalized in a vector of values corresponding to the classification probabilities of each of the classes. In this way, in addition to allowing an automatic extraction of parameters, the CNNs also perform the classification for the intended problem [19].

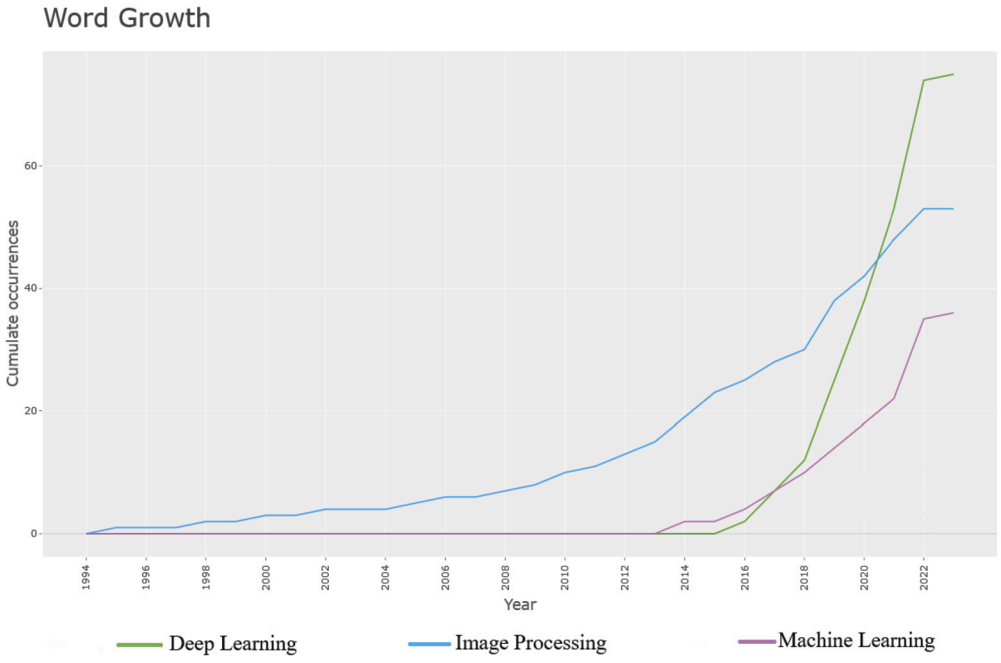

Fig. 1. Evolution of used keywords over time.

Recalling the case study presented in this paper, the identification of different cultivars of olive trees through the images of their leaves is a theme in development, being one of the first works of this kind presented in 1999 by S. Mancuso and F.P. Nicese in the paper [20], here, the authors present an identification using Artificial Neural Networks. Sampling consists of 800 leaves of 10 different cultivars, dividing 50% for training and testing. Not using a direct input in the ANN, 17 phallometric parameters were extracted (area, perimeter, centroid, etc.). After extraction, they were introduced into the neural network using 17 input layers, corresponding to the extracted parameters. Subsequently, several tests were carried out to answer the questions proposed by the authors: the optimal number of output layers, which optimization function should be used, and how many units should be part of the hidden layer. As main conclusions, the authors presented excellent results when using a network with 20 neurons in the hidden layer, ten neurons as output, and the sigmoid function as the activation function. Achieving this combination error rates of 0%.

Also, in the paper [21] a system based on Multilayer feedforward Neural Networks for identifying different olive tree varieties through their leaves is proposed. The presented system uses images from a scanner with a resolution of 300 d.p.s RGB compressed to a resolution of 50×20, normalized and segmented. After this pre-processing, the canny edge function is used to obtain the edges of the leaf from which most of the resources evaluate the variety resulted, from which ten parameters are collected (area, perimeter, centroid, etc.). Once the main parameters have been obtained, the authors resort to an ANN is feedforward with backpropagation, the structure of the network consists of 10 input neurons corresponding to the parameters collected from the leaves, followed by two hidden layers of ten neurons, ending with an output layer of 12 neurons which is the number of varieties to be identified. The set of images used to train and test this ANN consists of 120 photographs of 12 different varieties, the training consists of 60 examples, and the remaining 60 are used for testing, dividing the dataset by 50%. As the study's main conclusions, the authors present a recognition rate of about 91% in the test set.

Observing the results presented by the authors as mentioned above, it is expected that the use of Convolutional Neural Networks will result in similar results but without the need to carry out all this manual preprocessing of the leafs. This facilitates its use on a large scale. Since no examples of these applications were found in the field of olive growing, samples of other cultures where encouraging results are emerging were analyzed, such as those presented by the authors of the paper [22], where a new approach for the identification of grapevine cultivars is presented. The authors suggest using a pre-trained convolutional neural network, the VGG16. To improve the behavior of this network in the identification of vines, the authors exchanged the last three Dense layers for a classification block built through a global average pooling layer, a dense layer with the ReLU function as the activation function, batch normalization, dropout as regularization method, and a final dense layer with SoftMax classifier. To validate the structure presented, the authors resorted to data augmentation techniques in their dataset consisting of 240 training images and 60 validation images. The results presented are encouraging, achieving a general accuracy above 99%, thus obtaining a quick and inexpensive method for identifying cultivars.

Still in the field of grapevine cultivar identification, the authors of [23] mention as the main contribution of their work the creation of a dataset of 21 vine species with approximately 5000 images, where five pre-trained CNNs will be applied, namely AlexNet, ResNet, GoogleLeNet, DenseNet, and VGG, to understand which are the most suitable. Finally, the authors develop an Android application to identify the variety of vines using cloud processing automatically. The dataset was pre-processed, using a complementary image, and later was resized for the CNNs input. Several data augmentation techniques were also used, including geometric transformations such as scaling, transposing, rotation and flipping. Gaussian white noise was also used during this process to improve the algorithm's robustness. The authors used a split of 70% for training (1600 images per category) and 30% for validation (480 images per category) s per category) and ten images from each class were used to test the application. Two standard metrics were used to validate the results obtained: accuracy and recall rate. The results obtained by the authors were auspicious, reaching accuracy values above 94% for all CNNs used, managing to get a value of 99.91% with the optimization of the parameters of the GoogleLeNet network.

For the problem of identifying apple cultivars, the authors present in the work [24] a proposed solution, referring to the existence of more than 8000 cultivars of this fruit. It was used a Deep Convolutional Neural Network to classify apple cultivars using their leaf as the

only input. The presented structure is composed of an input layer, six convolution layers, each followed by a max-pooling layer, one standard one-dimensional dense, full connection layer, one dropout process, and an output layer. The Dataset used to evaluate this network it is composed of 10634 images from 14 different categories in the training set, 1181 in the validation set, and 620 in the test set. As a result, the authors present an overall accuracy of 97.11% in the classification of the 14 varieties.

A similar approach was proposed in the work [25], with a dataset of approximately 4800 images of 30 different varieties in two growth stages. In this case, the authors used data augmentation, through functions such as random cropping, level-flipping, median filtering, Gaussian noise reduction, gamma transformation, and brightness and contrast. They are managing to expand the dataset to 24000 images, of which 14400 were used for training, 2400 for validation, and the remaining 7200 for testing. In this work, the authors present a new network structure, composed by a convolution layer, in the middle are four convolutional modules with three convolutional structures in each module, where the convolutional modules are distributed in a symmetric design, and finally, in a fully connected layer with a total of 26 convolutional modules. The authors resort to six metrics to validate the presented structure, proving an average accuracy of 98.14%.

As can be seen, the application of CNNs for the identification of different cultivars is still a recent and developing process. Thus, the number of published papers is still restricted to some species and specific algorithms. Therefore, there is the possibility of exploring the behavior in other species and also with other types of algorithms to improve identification efficiencies and thus make this type of process increasingly reliable.

3. Materials and methods

To validate the possible use of convolutional networks for olive cultivar identification, it is necessary to understand which algorithms best fit the problem. Guaranteeing a sufficiently concise dataset and a well-consolidated structure allows a fair comparison between the different algorithms is essential. Therefore, this section is divided into subsections: Dataset, Classification Methods, Evaluation Metrics, and a summary of the methodology used.

3.1. Dataset

It is generally agreed upon that dataset selection plays a crucial role in determining the success or failure of an application that utilizes supervised artificial intelligence algorithms. This connection is proven if it is observed the operation of this type of algorithms, to optimize weights and connections to generalize from a set of data to the real world. In this way, the dataset made available to our work will be its baseline for the intended generalization and consequent use in the real world. Since this specific case is an innovative approach and no similar datasets are available, there was a need to create a large set of images to guarantee the functioning of the supervised classification algorithms.

Creating a dataset is not always an easy job and may not be a complex task, but it is always time-consuming and requires the availability of both human and material resources. When referring to datasets in the agricultural sector, monitoring crops' growth and recession stages that require considerable time intervals to complete production cycles is often necessary. In the case of the olive tree, which has an annual development cycle, it spends a large part of the winter in a state of lethargy in spring, summer, and autumn, when most of the development stages of the tree take place. More specifically, its leaf usually starts its development in March, which it just completed in November, being in constant change during this period [26]. In addition to the duration of their cycle, another factor that hinders the creation of this type of dataset is the need to collect samples from monovarietal olive groves only or, on the other hand, for this collection to be carried out by certified technicians in the area, thus guaranteeing the reliability of the collected samples.

3.1.1. Field and sampling method

In the case presented, this collection was carried out in olive groves where other studies have already been conducted and in partnership with certified technical personnel in the area to guarantee that only samples of the desired cultivars were collected. This collection process was planned to follow all the seasons of the year and consequently the growth and recession of the leaves, as well as other factors that have an impact on their development, such as water or nutrient stress, thus guaranteeing a correct identification regardless of the stage and the conditions in which the tree is found. After stipulating the duration of the collections, the collection method was also defined, being as random as possible, collecting leaves of the year, as well as older leaves, from the inside and outside of the canopy in an average of 20-25 leaves per tree of the selected cultivars. Emphasizing that this sampling process was used to not restrict the composition of the data and to enable the replicability of the results for any leaf of the tree. For this initial study, only four cultivars were chosen, the choice falling on the cultivars: *Cobrançosa*, *Madural*, *Negrinha de Freixo*, and *Verdeal Transmontana*. This choice is due to the fact that all of them are indigenous to Trás-os-Montes (Portugal), and are included in the classification of protected designation of origin (Azeite de Trás-os-Montes DOP), thus being essential for their identification for a consequent valorization of the olive oil produced in these olive groves. Having selected the collection methods and the desired cultivars, the last step was to select the area that would guarantee the necessary conditions, namely, having the desired cultivars, ease of travel, and knowledge of the area as the main olive oil producers in the region. The choice fell to the parish of Sucães (41.49°N, 7.26°W), with an average altitude of 350 meters, belonging to the district of Bragança and the municipality of Mirandela.

Having established the collection parameters as well as the location, four collections have been carried out so far:

- 1st collection, carried out on 2021-10-11 Initial harvest stage. Consisting of 149 leaves of the *Cobrançosa* variety, 89 of *Madural*, 101 of *Negrinha de Freixo*, and 106 of *Verdeal Transmontana*;
- 2nd collection, carried out on 2021-12-15 Final harvest stage. Consisting of 169 leaves of the *Cobrançosa* variety, 231 of *Madural*, 219 of *Negrinha de Freixo*, and 214 of *Verdeal Transmontana*;
- 3rd collection, carried out on 2022-06-30 Fruit growth. Consisting of 380 leaves of the *Cobrançosa* variety, 237 of *Madural*, 384 of *Negrinha de Freixo*, and 402 of *Verdeal Transmontana*;
- 4th collection, carried out on 2022-10-3 Initial harvest stage. Consisting of 348 leaves of the *Cobrançosa* variety, 392 of *Madural*, 362 of *Negrinha de Freixo*, and 473 of *Verdeal Transmontana*.

Once the leaves were collected, they were taken to the laboratory where they were photographed, this process is carried out in the shortest possible time to preserve the leaf's physical characteristics. To scan the leafs, the camera on the smartphone was used to simulate the future conditions of the use of the application. The leaves of the first two collections were photographed with a digital camera with a 2610×4640 pixels resolution (Xiaomi Redmi Note 9 Pro). The remaining two collections were photographed with a similar camera but with a 3000×4000 pixels resolution (Xiaomi 11T). All samples were photographed on a white background in RGB format, as its possible to see in the Fig. 2, to facilitate their treatment and application in the algorithms. Emphasizing that the entire dataset is available in the link or go to the next url: https://dados.ipb.pt/dataset.xhtml?persistentId=doi:10.34620/dadosipb/TVYE8K

3.1.2. Dataset pre-processing

After completing the previous process, some pre-processing techniques were applied, to reduce the size and the resolution of the images. Therefore, taking into account a large number of images, some tech-

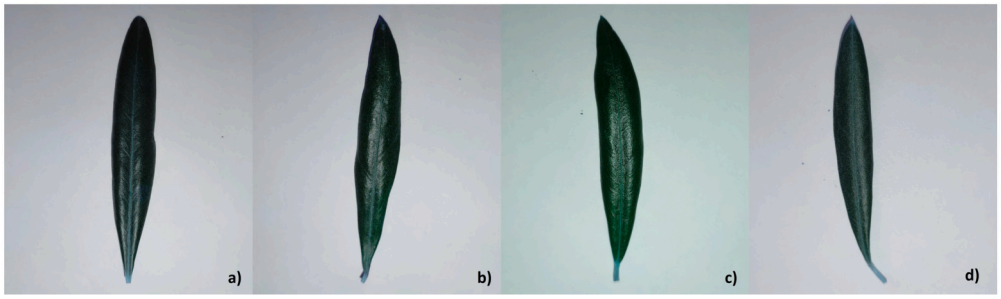

Fig. 2. Dataset cultivars: a) *Cobrançosa*; b) *Madural*; c) Negrinha; d) Verdeal.

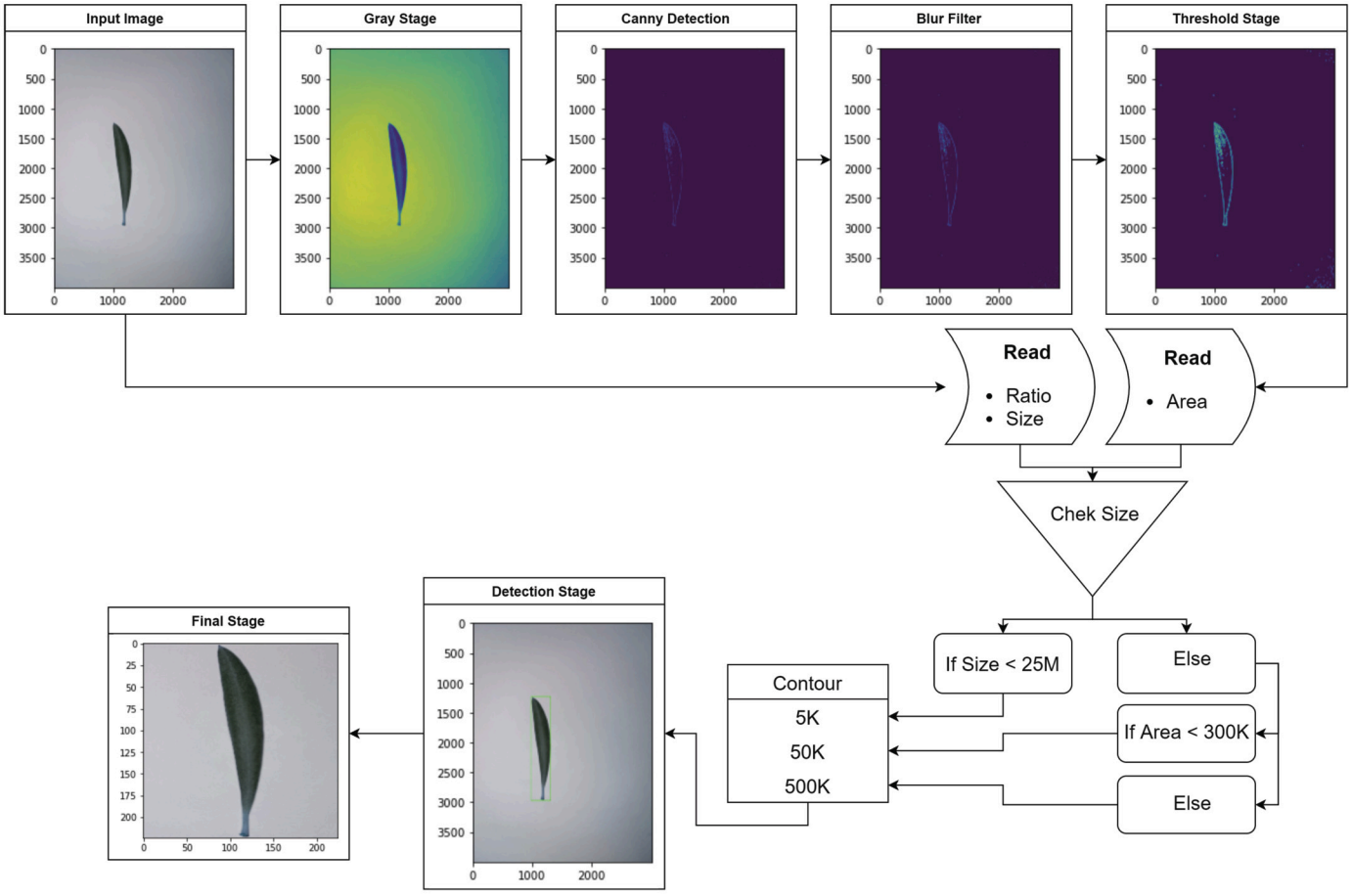

Fig. 3. Proposed system methodology.

niques were adopted to carry out this process autonomously, noting that all pre-processing was carried out using python version 3.8.12 and the OpenCV library [27] (version 4.0.1).

As seen in Fig. 3, after reading the image, its ratio, and original size are calculated, essential parameters for preserving the leaf's proportions when cutting. The image then passes through various stages and filters used to enhance the structure of the leaf. The first one is converting the image to its Gray format, thus facilitating the calculations carried out from now on. In the second step, edge detection is used through the John F. Canny algorithm [28]. In this algorithm implemented through the OpenCV library, it is necessary to provide five parameters, including the image to use, the low and the high threshold value of intensity gradient, the order of Kernel (matrix) for the Sobel filter and the L2gradient that specifies the equation for finding gradient magnitude. In the case presented, two constant values were defined as threshold values 50 and 90, when the kernel order chosen was 3×3, and finally the equation $Edge\_Gradient(G) = |G_x| + |G_y|$ was used. Also, a blur function was used to remove any noise from the image, this process is achieved by convolving the image with a low-pass filter kernel, in our case, a 5×5 kernel. In the fifth step, an Adaptive Thresholding function was used, using a Gaussian-weighted sum of the neighborhood values function to determine the threshold value. With the matrix resulting from the fifth step, the area values were calculated, counting all the pixels different from 0. After obtaining the three parameters (area, ratio, size), it was possible to define different sizes of contours, as shown in Fig. 3, serving this differentiation of contours for a generalization, being able to receive images of any dimensions, and thus ensuring that are not selecting only a part of the sheet. With the differentiation of the contours, one more condition was required for the pre-processing algorithm, where only objects with dimensions more significant than the presented contours are selected. This results in our penultimate stage, the leaf detection stage. From here, the coordinates of the bounding

box are chosen, from where, later, the coordinates to perform the crop are created. These coordinates are calculated to maintain the ratio of the initial image and guarantee a margin of 0.01% of the initial area in relation to the yy axis, resulting in our last stage.

Once pre-processing was completed, the result obtained was a dataset with approximately 4200 images from the four most representative categories of the region.

3.2. Classification models

Considering the large number of convolutional networks available, a comparison was made between ten different models, using the transfer learning technique as a common starting point for all of them.

In this context, transfer learning is taking advantage of the training already carried out by a model as a starting point for training another. This technique is usually used using pre-trained models in the ImageNet image bank, which consisted of 14197122 images (2014) [29]. This learning results in representations such as edges, visual shapes, and lighting changes, which are helpful for the generalization of any model [19]. In this way, the best methods of CNNs were selected, trying to follow the history and the evolution's achieved in the ImageNet Challenge from the year 2014 to the year 2019, making a selection of ten of which achieved better results in the chosen interval.

3.2.1. VGG

The VGG structure [30] was presented in 2014 for the ImageNet Challenge. For this purpose, the authors presented three different structures, namely, the structure that became known as VGG16, a structure composed of thirteen interconnected convolutional layers with five layers of max pooling and three fully-connected layers with a softmax activation function at the end. With an input of $224 \times 224 \times 3$, and a stride of one being followed by ReLU non-linearity. The second structure presented by the authors was known as VGG19 and had a similar organization but with three more convolutional layers. In terms of results obtained, the VGG 19 guaranteed a TOP 1 accuracy of 74.5% with 144 million parameters. The VGG 16 was only 0.1% behind with a Top 1 accuracy of 74.4% with 138 million parameters.

3.2.2. Inception

The inception-type algorithms originated from the need to solve some problems found in CNNs until then, namely, finding the right size for the kernel of the convolutional layers, solving the overfitting problem that was happening due to the use of very deep networks, and also the high computational cost associated with this type of networks. To solve the announced problems, the concept of "inception" was introduced by Szegedy et al. in [31], using three types of filters in the convolution (1×1, 3×3, and 5×5) and a Max pooling layer (3×3) in parallel. The result of these processes is later concatenated and sent to the next inception module. In the specific case of the version used in this work, InceptionV3 proposed by [32], some features were added to the original algorithm to improve its behavior, namely, to give a more active role to the auxiliary classifiers, using a BatchNorm for this purpose, RMSProp was also used optimizer, and to improve the behavior of the network in relation to overfitting, label smoothing was used. The result is a network 42 layers deep, which secured 78.8% in the TOP 1 accuracy of the ImageNet Challenge 2015.

3.2.3. ResNet

So far, convolution merging has been working well, getting good results up to approximately 30 layers, but authors He, Kaiming, et al. managed to prove in [33] that when using more than 30 layers the problem of vanishing/exploding was real, not existing any improvement with the superimposition of more layers. In this way, the authors presented a new structure, the ResNet. The proposed solution is composed of several blocks (stacked residual units) and starts from the principle that it is easier to optimize the residual mapping of the convolutional layers than to optimize the original mapping, in this way, "shortcuts" are created between layers the identity shortcut, where there is a bypass of the input volume for the addition operator, and the projection shortcut, in these a convolution operation is performed to ensure that the volumes are similar to those found in addition. Ensuring that even the most complex structure of the ResNet's presented (ResNet-152) has a lower complexity (11.3 billion floating point operations (FLOPs)) than the VGG16/19 structure (15.3/19.6 billion FLOPs) despite having a depth eight times greater the one found in VGG19. In addition to the "traditional" ResNet-50, the ResNet-152V2 version will also be used in this study, a version that features a batch normalization before each weight layer, compared to the first version (ResNet-152).

3.2.4. Inception ResNetV2

Observing the good results obtained either by the inception structure or by the ResNet structure, the question was raised of whether combining the two structures is beneficial. In this attempt to build a hybrid model, the authors Szegedy et al. present in [34] clear evidence of how training significantly accelerates when introducing residual connections between inception modules. This hybrid structure, Inception ResNetV2 with 164 layers deep, secured 80.1% in the TOP 1 accuracy of the ImageNet Challenge 2016.

3.2.5. Xception

With the inception structure as a basis, the author F. Chollet suggests in [35] a new structure. Xception, as the author calls it, replaces the inception modules with depthwise separable convolutions. This replacement aims to improve performance, but mainly the necessary computational cost, the concept of depthwise separable convolutions comes to fill the very high cost associated with traditional convolution, these are divided into two main stages the Depthwise Convolution where instead of computing the convolution being performed on all channels is performed only one by one. The second phase is the Pointwise Convolution operates a classical convolution, with size $1 \times 1 \times N$, where N corresponds to the total number of kernel/filter used. This simplification makes it possible to reduce the number of operations carried out by a factor of $1/N$. The results achieved by this network in the ImageNet Challenge 2016 contest were 79% in TOP 1 accuracy.

3.2.6. MobileNet

Developed to be used in mobile and embedded systems, MobilNet [36] was designed to allow computer vision applications in systems with less computational capacity. To guarantee the ideal optimization between latency and accuracy, the authors present two simple global hyper-parameters, allowing these to give the programmer the choice of the right-sized model for their application based on the constraints of the problem. This results in a network with slightly lower performance than those previously presented, with 70.6% in the TOP 1 accuracy of ImageNetCallange 2017, but using only 4.2 million parameters. This paper will also use the second version of this network, MobileNetV2, presented by Sandler, Mark, et al., in [37]. In this version, the non-linearities in narrow layers were removed, and a better module was introduced with an inverted residual structure, thus managing to reduce further the number of parameters used (3.4M) and, consequently, the execution time and maximize the accuracy in TOP 1, reaching 72% in ImageNet Challenge 2018.

3.2.7. EfficientNet

To optimize the models of convolutional networks, the authors Tan, Mingxing, and Quoc Le presented in [38] presented the EfficientNet. This structure is a convolutional network that uses a composite coefficient to uniformly scale the network's width, depth, and resolution, leaving aside the typical arbitrary choice of these parameters. This composite coefficient is justified by taking into account the size of the image, a larger image will need a network with more layers to increase

Table 1
Summary table of CNNs to use.

Method	Top 1 Accuracy ImageNet	No. Parameters (millions)	Year
VGG16	74.4	138	2014
VGG19	74.5	144	2014
Inception V3	78.8	24	2015
ResNet 50	75.3	25	2015
ResNet-152V2	78	60	2015
Inception ResNetV2	80.1	56	2016
XCeption	79	23	2016
MobileNet	70.6	4.2	2017
MobileNetV2	72	3.4	2018
EfficientNetB7	84.4	66	2019

the receptive field and more channels to capture finer patterns. All EfficientNet have the same baseline, a model based on the mobile inverted bottleneck MBConv [37,39] with the addition of squeeze-and-excitation optimization. From here, the compound coefficients were varied, giving rise to all EfficientNets, from EfficientNet-B0 to EfficientNet-B7. The results obtained in ImageNet Challenge 2019 by the model used in this study (EfficientNet-B7) were 84.4%.

In short, Table 1 summarizes the models that will be used in this study, as well as the number of parameters they use and the Top 1 accuracy obtained in the ImageNet Challenge contest of the respective year.

3.3. Metrics

When dealing with a classification problem, the metrics to be used will be based on the confusion matrix generated for each CNNs [40]. Based on the predefined constitution for any binary problem, in this case, the confusion matrix will also be composed of: True Positive (TP), which represents the number of images that were correctly classified in the positive class, True Negative (TN) (correctly classified as negative), False Positive (classified as positive but actually negative) and False Negative (classified as negative but actually positive). But in this specific case, it will have a confusion matrix depending on the four olive tree cultivars, being necessary to add some values to arrive at the desired values. From these, the metrics will be calculated:

3.3.1. Accuracy

The accuracy is the most used metric in this type of problem its formula describes the number of correct classifications as a function of all inferences made and can be calculated as (Eq. (1)):

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

3.3.2. Precision

The precision is used to measure positive patterns that are correctly classified from the total classifications patterns in a positive class and is calculated using (Eq. (2)):

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3.3.3. Recall

The recall (Eq. (3)) is used to measure the fraction of True Positives that were correctly classified:

$$Recall = \frac{TP}{TP + TN} \tag{3}$$

3.3.4. F-score

The traditional F-score (or F1) is the harmonic mean of precision and recall and is defined by (Eq. (4)):

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Table 2
Number of photographs for each cultivar in each collection.

	Cobraçosa	*Madural*	*Negrinha de Freixo*	*Verdeal Transmontana*
1st Collection (2021/10/11)	149	89	101	106
2nd Collection (2021/12/15)	169	231	219	214
3rd Collection (2022/06/30)	380	237	384	402
4th Collection (2022/10/03)	348	392	362	476

3.4. Methodology

All algorithms presented here were tested, trained, and implemented on a computer equipped with an AMD Ryzen Threadripper 3970X 32-Core Processor, using an NVIDIA TITAN V graphics unit with 12 GB RAM with a version of CUDA 11.7. The python version used was 3.6.13, using the TensorFlow-GPU machine learning library in version 2.4.0 [41]. The methodology used is shown in Fig. 4:

After collecting and pre-processing the dataset, it was necessary to divide the images into three sets, Training, Validation, and Test. This division was carried out in a totally random way, guaranteeing that the sets were composed of approximately the same amount of data from each collection and ensuring that there was the same percentage of each one of the collections in the different sets. Afterward, a function of the Python random module was used to fill these sets.

Observing Table 2, it is easier to understand, taking the example of the first collection, 89 images of each of the varieties were selected, and of these 89 images, 70% were dedicated to training, 20% to validation, and the remaining 10% were used for the models testing. In the initial set, this resulted in around 2360 images for model training, 676 for validation, and 336 for testing. The amount of "wasted" images is notorious, but it is the only way to guarantee that the dataset is balanced between collections and cultivars.

From this initial set, a new set of larger dimensions was created using data augmentation techniques for the purpose, this type of procedure has as its main objective to increase the generalization capacity of the model by creating variations of real images through geometric transformations and/or coloring. In this case, two position augmentation methods (Flipping and Rotation) and a color augmentation method (Brightness) were chosen. With its use, the final result was a dataset consisting of 14,160 training images, 4056 validation, and 2016 test images. It should be noted that this division process ensured that no images from data augmentation techniques were in different sets, to avoid training and testing with similar images.

Regarding the training process of the models, and considering that pre-trained models are being used with the weights of ImageNet, some changes were made to ensure that all the models had the same conditions at the start. For this purpose, specific pre-processing for each model was used, namely the normalization of images according to their expected input. Some changes were also made to the structure of the selected models, starting with normalizing the input dimensions for all models and choosing an input of $224 \times 224 \times 3$. Still in the network structure, all fully-connected layers present in the models were removed, and a new fully-connected custom set was added, as shown in Fig. 5.

This is composed by the flattened result of the last convolutional block of each model and later by two dense layers, activated through the function ReLU [42], one with 512 neurons and one with 256, with a dropout layer between them of 0.4, finally, another dense layer activated through the softmax function was used, which allows us to obtain a final output in percentage form for the four different varieties. Several values were tested to choose the number of neurons in the layers, from the traditional 4096 present in the VGG structures to the minimum of only four neurons. This combination guarantees the best results in most models, the same happening with the chosen Dropout.

The parameters chosen to compile the model were the most common in this type of classification problems, using the categorical cross-entropy loss function, as the selected metric was the model's accuracy,

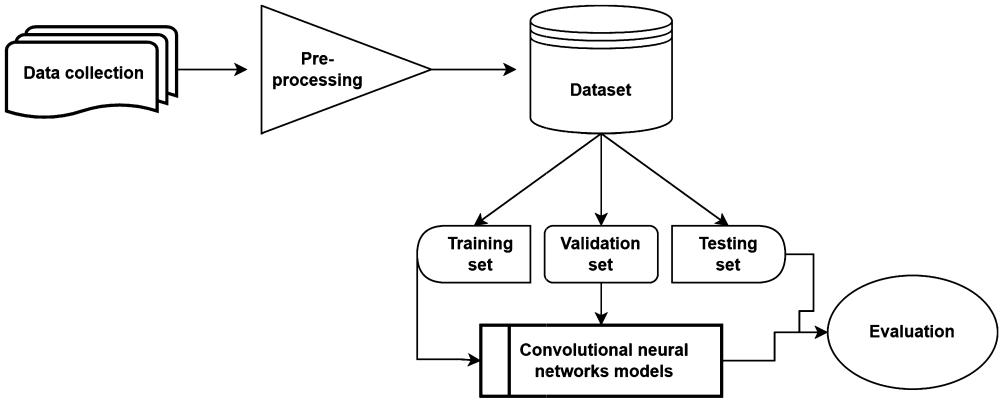

Fig. 4. Proposed system methodology.

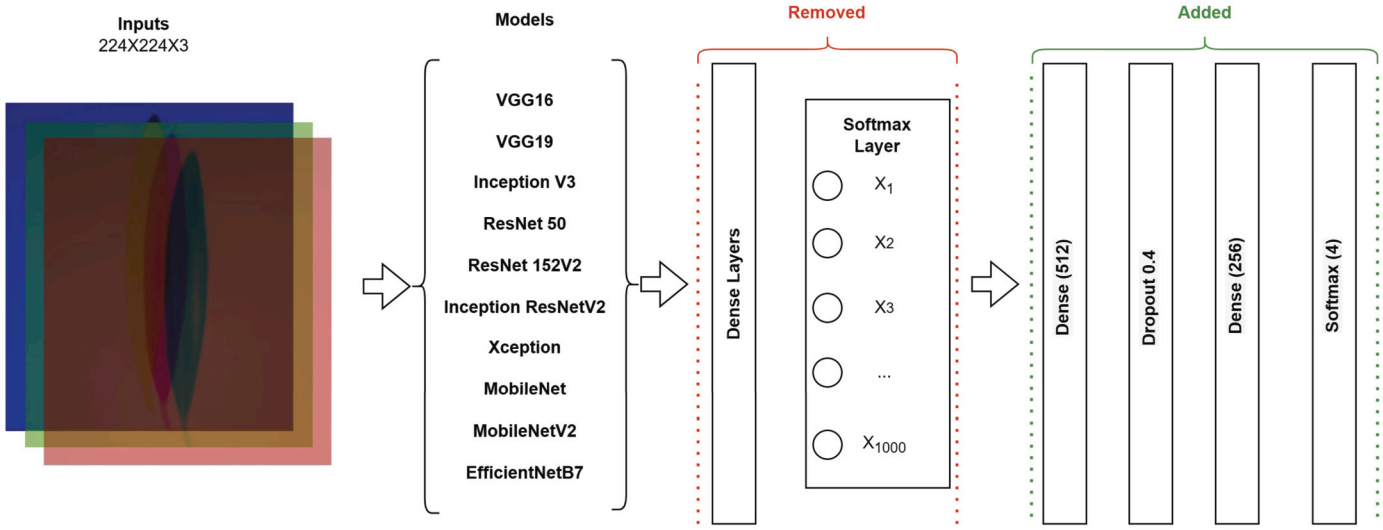

Fig. 5. Proposed structure of the networks.

and finally decided to use the Adam algorithm as optimizer. The model was trained for a maximum of 500 epochs, but with an early stopping whenever the accuracy in the validation set does not improve after 12 epochs. Also, for the learning rate, a method dependent on the validation set was used, starting the training with a learning rate of 5e-5 and decreasing this with a factor of 0.6 whenever after two epochs, the accuracy in the validation set does not increase down to a minimum of 5e-7. At the end of the training, the model always restored the best weights, and the model with the best accuracy in the validation set was always evaluated.

Various sizes were tested regarding the use of mini-batches, from the usual 64 to training without any mini-batches. Although it was not ideal, a set of 8 images was used, the maximum number achieved for the presented hardware. After finding this maximum, some hyper-parameters were varied to extend this comparison to more than one scenario. In pursuit of this, as shown in Fig. 6, eight tests were carried out for the ten models. The hyper-parameters chosen to vary have mostly to do with the structure of the models, changing between training all the layers of these, using only the structure of the model and disregarding the weights coming from the ImageNet, or, on the other hand, not training the convolutional layers of the model, using ImageNet learning and training only the fully connected custom layer. The second variation has to do with the format used after the last convolutional layer, which can be flattened, which results in a one-dimensional tensor consisting of all the values of the previous feature maps, or on the other hand, Global Average Pooling (GAP) [43], where an average pooling of all feature maps is applied, resulting in a vector that is the average of all of them, serving as input in our fully-connected custom. It should be noted that the variations presented were made in the two datasets (Starter set, Data augmentation).

Concerning the evaluation of the methods, and considering their number as well as the number of tests carried out, all of them will be initially compared in terms of their general accuracy. After this initial comparison, the best methods will be chosen, and their in-depth results will be presented and compared, using the evaluation techniques explained above.

4. Results

Considering that all training and tests were prepared using a virtual machine, that was not exclusively used for this purpose, the times referring to training and tests will be omitted at the risk of providing unreliable information. As an initial point of comparison, the accuracy of the models in the different sets will be used, as well as their convergence speed, to limit the group of models/sets to be used, since the associated computational cost is extremely high, taking approximately about 13 days and 23 hours to train the 80 models once. In this way, the results obtained from the first iteration were analyzed, however, taking into account the discrepancy between the test and validation set (approximately 7%), it was necessary to carry out a new iteration. In this way, and trying not to overextend the paper, the results for the ten models in the eight sets will be presented, but only composed by the average values of the accuracy (validation and test set) of the two iterations (two times cross-validation), for this, and making use of the

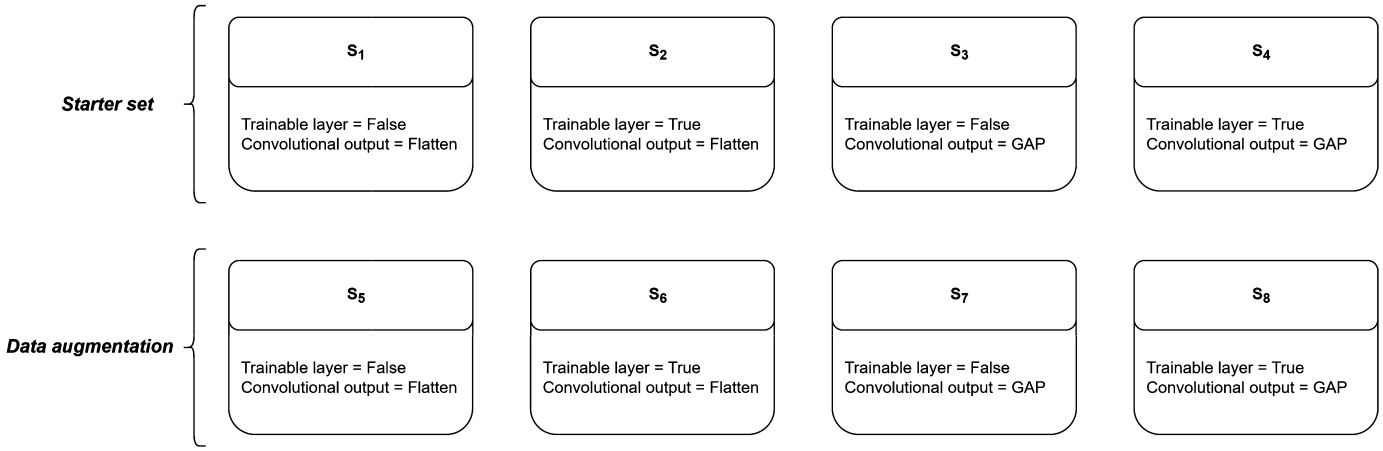

Fig. 6. Proposed change sets.

Table 3
Results of the first iteration.

Model	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	Average
VGG16	0.883	**0.928**	0.845	0.918	0.883	0.918	0.845	0.910	0.891
VGG19	0.865	0.905	0.803	0.865	0.875	0.908	0.833	0.905	0.870
Xception	0.835	0.915	0.783	0.920	0.848	**0.925**	0.818	0.928	0.871
ResNet50	**0.890**	0.915	**0.855**	**0.925**	**0.885**	0.913	**0.873**	**0.938**	**0.899**
ResNet15V2	0.818	0.903	0.790	0.910	0.845	0.915	0.800	0.920	0.863
MobileNet	0.868	0.905	0.808	0.908	0.868	0.908	0.838	0.920	0.878
MobileNetV2	0.843	0.895	0.815	0.903	0.845	0.913	0.810	0.915	0.867
InceptionV3	0.803	0.898	0.755	0.913	0.808	**0.925**	0.785	0.923	0.851
InceptionResNetV2	0.765	0.900	0.690	0.913	0.800	0.915	0.758	0.918	0.832
EfficientNetB7	0.845	0.910	0.788	0.920	0.858	**0.925**	0.845	0.933	0.878
Average	0.841	0.907	0.793	0.909	0.851	0.916	0.820	**0.921**	

methodology explained above, the results obtained are visible in Table 3.

Observing the behavior of the models through, it is possible to verify an oscillation of accuracy over the eight sets, ranging from 69% (InceptioResNet15V2 in the S_3) to 93.8% (ResNet50 in the S_8). An average per set and model was added to facilitate the interpretation of the results. Starting by analyzing the models' average, it is possible to observe that the oscillations are much smaller, obtaining in the worst model (InceptionResNet15V2) an accuracy of 83.2% and in the best model (ResNet50) an accuracy of 89.9%. Reducing from an oscillation of 24.8% to 6.7%, emphasizing that the results presented here result from the average of the validation and test set of the two tests. In agreement with the results by model, also when the values are observed individually, the ResNet50 method is presented as the best, namely in the S_8 with an accuracy of 93.8%, also in the same set, it is possible to find the second-ranked of this list with 93.3% (EfficientNetB7). Likewise, one of the third best results was obtained in the S_8 by the Xception method with 92.8%, sharing this place with the VGG16 method in the S_2. The fourth place, in turn, is already found in the S_4 and S_6, being shared by three different methods in the S_6 (Xception, InceptionV3, and EfficientNetB7) and again by ResNet50 in the S_4, reaching an accuracy of 92.5%. Observing the best accuracy in perspective, they all turn out to be relatively close to each other, another point in common is their origin, as they all come from sets with all trainable layers. Similarly, evaluating the average of the ten methods for each set, it is noticeable that the methods that present a higher average are again from the sets with all trainable layers (S_2, S_4, S_6, and S_8). Observing in more detail, it is even possible to verify that the sets where data augmentation was used (S_6 and S_8), have an average accuracy of approximately 1% higher. Although more tenuous, there is also a slight advantage in using the GAP layer compared to Flatten, which guarantees an improvement of roughly 0.35%. If, on the one hand, the averages per model were more or less constant with oscillations of 6.7%, the same did not happen in the averages per set, here there is a clear disadvantage of the groups that used ImageNet weights, with oscillations of almost twice as much (12.8%). Analyzing the convergence of the models in each set, Fig. 7 shows the average of epochs necessary for the model to converge to its optimal point (12 epochs before the stop).

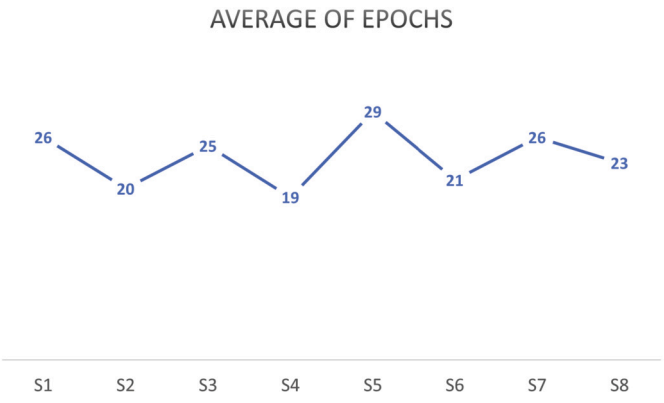

Fig. 7. Average number of epochs required until the optimal point.

Through Fig. 7, it is noticeable that the lowest number of epochs until convergence is achieved again by the methods with the trainable layers (S_2, S_4, S_6, and S_8), with the best case found in the S_4 and the worst in the S_5 with a difference in the average of ten seasons. Analyzing in more detail, it is possible to verify an increase in this average in the sets that resorted to data augmentation of approximately two epochs. When the use of the GAP layer with Flatten is compared, the best results are obtained by using the GAP layer but only for 0.75 epochs.

Table 4
Performance evaluation.

Set	Model	Cultivar	Precision	Recall	F1-score	Accuracy	Validation	Dif Test vs Val
S_2	VGG16	*Cobrançosa*	0.946	0.952	0.950	0.950	0.954	0.004
		Madural	0.950	0.938	0.944			
		Negrinha de Freixo	0.950	0.970	0.960			
		Verdeal Transmontana	0.958	0.940	0.948			
S_4	Xception	*Cobrançosa*	0.972	0.960	0.966	0.968	0.968	**0**
		Madural	0.978	0.976	0.972			
		Negrinha de Freixo	0.952	0.980	0.968			
		Verdeal Transmontana	0.970	0.954	0.906			
	ResNet50	*Cobrançosa*	0.982	0.974	0.976	**0.984**	0.982	0.002
		Madural	**0.990**	0.982	**0.984**			
		Negrinha de Freixo	0.988	0.986	**0.984**			
		Verdeal Transmontana	0.972	0.986	0.980			
	EfficienteNetB7	*Cobrançosa*	0.974	0.950	0.964	0.966	0.980	0.014
		Madural	0.970	0.972	0.972			
		Negrinha de Freixo	0.962	0.804	0.970			
		Verdeal Transmontana	0.960	0.960	0.960			
S_6	Xception	*Cobrançosa*	0.964	0.964	0.968	0.966	0.97	0.004
		Madural	0.970	0.972	0.970			
		Negrinha de Freixo	0.968	0.972	0.970			
		Verdeal Transmontana	0.966	0.958	0.962			
	InceptionV3	*Cobrançosa*	0.960	0.950	0.956	0.962	0.962	**0**
		Madural	0.972	0.974	0.970			
		Negrinha de Freixo	0.962	0.972	0.966			
		Verdeal Transmontana	0.960	0.962	0.962			
	EfficienteNetB7	*Cobrançosa*	0.974	0.978	0.976	0.980	0.982	0.002
		Madural	0.980	**0.988**	**0.984**			
		Negrinha de Freixo	0.982	0.976	0.978			
		Verdeal Transmontana	0.982	0.966	0.972			
S_8	ResNet50	*Cobrançosa*	0.978	0.976	0.976	0.974	0.980	0.006
		Madural	0.980	0.978	0.980			
		Negrinha de Freixo	0.970	0.980	0.976			
		Verdeal Transmontana	0.974	0.970	0.970			
	EfficienteNetB7	*Cobrançosa*	0.982	0.974	0.980	0.980	**0.984**	0.004
		Madural	0.986	0.980	0.980			
		Negrinha de Freixo	0.978	0.986	0.982			
		Verdeal Transmontana	0.972	0.978	0.954			

To move on to the second phase of evaluating the results, the best methods were chosen based on the results already presented. In this way, the best methods of the best sets were chosen. The choice of Sets was easy to perform, with the selection of the four Sets that presented the best results on average (S_2, S_4, S_6, and S_8). As for the methods, a margin of 0.5% was stipulated, with the best methods being chosen per set and all those within that margin. In this way, nine different methods resulted:

- VGG16 in the S_2;
- Xception, ResNet50 and EfficientNetB7 in the S_4;
- Xception, InceptionV3 and EfficientNetB7 in the S_6;
- ResNet50 and EfficientNetB7 in the S_8.

Once the methods were selected, the previously discussed metrics were applied. However, in this case, five-fold cross-validation was also used, with the results presented below (Table 4) the average of these five interactions.

Evaluating the applied metrics (Table 4), the results have improved considerably compared to those previously presented. Observing in detail, it is possible to verify that most of the methods present an accuracy in the test set superior to 95%, reaching in the best case 98.4% in the case of ResNet50 (S_4). This method also guaranteed an accuracy of 99% in the *Negrinha de Freixo* category, the best result, achieving this brand with an F-score of 98.5% in the same cultivar. Then with 98%, the EfficientNetB7 (S_6 and S_8) and in third place again the ResNet50 methods but this time in the S_8. It is important to point out that these results come from the test sets of five different iterations, also referring to the importance of the differences between the test and validation sets that proved negligible, with an average difference of 0.4%.

Table 5
Summary table.

Model	Accuracy (Test)	Epoch's
ResNet50 (S_4)	0.984	25
EfficienteNetB7 (S_6)	0.980	42
EfficienteNetB7 (S_8)	0.980	23
ResNet50 (S_8)	0.974	28
Xception (S_4)	0.968	21
Xception (S_6)	0.966	26
EfficientNetB7 (S_4)	0.966	18
InceptionV3 (S_6)	0.962	28
VGG16 (S_2)	0.950	21

Summarizing and ordering the results obtained, Table 5, shows that the model that guaranteed the best accuracy on average was the ResNet50 in the S_4, with EfficientNetB7 in second place. Evaluating the epochs needed by each method to reach the optimal training point, The best was achieved by EfficientNetB7 in the S_4 with 18 epochs on average in the five iterations performed. In general, the best records, in this aspect, were achieved by methods that did not use data augmentation, managing on average to reach the optimal point in eight times less than the others. On the contrary, when the average accuracy presented by the methods without data augmentation is evaluated, even slightly, it turns out to be 0.59% worse.

Table 6
Number of trainable parameters of each method according to the Set.

	S_1 & S_5	S_2 & S_6	S_3 & S_7	S_4 & S_8
VGG16	12 977 924	27 692 612	395 012	15 109 700
VGG19	12 977 924	33 002 308	395 012	20 419 396
Xception	51 513 092	72 320 044	1 181 444	22 042 924
ResNet50	51 513 092	75 047 684	1 181 444	24 716 036
ResNet152V2	51 513 092	109 700 996	1 181 444	59 369 348
MobileNet	25 822 980	29 029 956	657 156	3 864 132
MobileNetV2	32 245 508	34 469 380	788 228	3 012 100
InceptionV3	26 347 268	48 115 620	1 181 444	22 949 796
InceptionResNetV2	19 793 668	74 069 860	919 300	55 195 492
EfficientNetB7	64 358 148	128 145 108	1 443 588	65 230 548
Average	34 906 270	63 159 357	932 407	29 190 947

5. Discussion

After presenting the results obtained, it is necessary to carry out a small discussion to draw the necessary conclusions from the study. Starting with the time required to carry out this type of study, given the impossibility of accounting for the exact time needed for each model to go through a step, the number of epochs necessary to reach the maximum accuracy point in the validation dataset was used. This metric is not always fair, considering that the learning rate has decreased at the above-mentioned rate. Often, these points coincide with the increase in accuracy, even if in small decimal places, which leads the model to be trained for at least another 12 epochs. However, considering the conditions were the same for all methods, this comparison of the optimal number of epochs was made. As there was only data from a two-time cross-validation for the whole set, it was decided to compare the sets, thus obtaining the average of the ten different methods. This guarantees greater robustness than using data from a single model. Observing the Fig. 7, it is possible to verify that the minimum number of seasons is reached in the S_2 and S_4, namely, sets with all trainable layers and without the use of data augmentation. To identify the influence of the number of parameters, these were calculated for the eight sets and are present in Table 6.

Observing the average of parameters by set in Table 6, it is noticeable that, the sets with the most trainable parameters on average (S_2 and S_6) are the ones that need the lowest number of epochs, approximately 20. Then, with just one more training epoch, on average, the S_4 and S_8 similarly these sets also present all their trained layers, however, in this case, the number of trainable parameters decreases to less than half taking into account the use of the GAP function. The third case arises with the S_3 and S_7 requiring 25 epochs on average in this case, however, ImageNet weights are used allied to a GAP layer which guarantees the minimum number of trainable parameters. The last case with 27 epochs on average comes from the S_1 and S_5, with the second highest number of trainable parameters. Several factors may influence this number of epochs, however, what appears to be happening, in this case, is a difficulty in framing the results from the convolutional layers when using ImageNet weights. The most likely cause for this to happen is the constitution of the ImageNet dataset, which, despite its size, does not have enough examples of leaf images [29]. Taking into account the specificity of this problem, it is possible to mention that the behavior revealed by the results was already expected since the work uses the small details of the leaves, and what is usually learned with the ImageNet dataset is coarser information such as shapes, edges, circles, lines [44]. Thus, it becomes evident that a larger number of parameters does not invariably correlate with slower convergence rates. In certain contexts, training the algorithm from scratch proves advantageous for addressing specific problems, necessitating a higher parameter count but fewer epochs to achieve convergence.

The second point of this discussion concerns the difference in values reached when comparing the accuracy values in the test set of the first division (Table 3) with those reached in the average of the following iterations (Table 4). The non-use of cross-validation explains this difference in results. What happened in the case of the first set (complete set) was that the first iteration used a random division with a test set that is extremely difficult to evaluate, this is a risk present in this type of problem, especially when using a test set of reduced dimensions as was the case (10%) [45]. This point is easily proven when the results obtained in the test set and the validation set are compared, which is 7% higher in the validation set. This problem could be synonymous with an overfitting of the model, however, when the model has been re-evaluated in the second iteration of the entire set, and in the following evaluations, the differences between the validation set and the test set are practically negligible, with the worst case presented in the EfficientNetB7 model in the S_4 with a difference of 1.4%. This difference in results is also associated with the composition of the dataset, which despite its already considerable size, is still not large enough to allow us to use it without cross-validation [46], taking into account the difficulty of the problem, emphasizing once again the similarity between the leaves of the four varieties but also mainly due to the changes shown between the leaves in their different stages.

Comparing the results obtained with the accuracy presented by the models in ImageNet (Table 1), it is possible to verify some discrepancies, starting with the best method that, in our case, turned out to be ResNet50 and that presented a top1 accuracy of only 75.3% in ImageNet challenge. This finding aligns with previous research highlighting the robustness of ResNet architectures in various image recognition tasks [47] Similarly, EfficientNetB7 and Xception demonstrated effectiveness in our problem domain, consistent with prior studies that have demonstrated their superior performance in diverse image classification tasks [48,49] One of the surprises was the traditional VGG16 architecture, despite the accuracy shown on ImageNet and its "age" managed to enter the Top 9 for our case, noting that it is a nine-year-old method and with a structure that, despite the high number of parameters is very simple, as explained earlier. This observation underscores the enduring relevance and efficacy of well-established models in certain contexts [50]. On the other hand, it was expected that better results would be obtained by other methods, namely InceptionResNetV2, since both methods that gave rise to it are in our Top 9. This discrepancy may be attributed to factors such as architectural complexity and dataset characteristics [34] However, in general, it is possible to extrapolate that if more tests were made to the general set, the final results would be better, proving this statement with the example of the chosen methods that, after a five-fold cross-validation, increased their accuracy by an average of 4.3% compared to the initial results. When looking at the results in relation to related works, the comparison becomes complex, since the approach presented here is innovative, and there is no other work available in the three databases consulted (Web of Science, Scopus, IEEE) that has used the same approach in the context of olive leaves. When comparing the results obtained with those from other trees, such as apple and grapevine, only one of the studies [22] presented results superior to those achieved by our models, while the others demonstrated lower overall accuracy [23–25]. It is important to emphasize again that the comparison is not entirely fair, since the data sets used are completely different; however, given the absence of any other similar article, this comparison is useful in providing an overview of the issue.

6. Conclusions and future works

This work presented a new approach for identifying different olive tree cultivars using deep convolutional neural network algorithms. While the experiment was conducted under laboratory conditions, simulating real-world scenarios, the method proposed here offers a promising solution to challenges inherent in genetic identification techniques. By leveraging a simple photograph of a tree leaf, our method enables rapid, cost-effective identification, potentially eliminating the need for expensive and time-consuming laboratory analyses. Although our experimentation took place in controlled settings, the applicability of

this approach extends beyond the laboratory, offering practical benefits to farmers and providing an engaging tool for enthusiasts or tourists interested in olive tree cultivation. To enable a correct identification regardless of the phase and conditions in which the tree is found, it was necessary to guarantee a Dataset of considerable dimensions, taking into account the similarity presented by the leaves of the four varieties. To meet this objective, images were collected over one year (2021-2022) for approximately 4200 images. These were collected to follow all the seasons of the year and consequently the growth and recession of the leaves, in the most random way possible, collecting leaves of the year as well as older leaves, from the inside and outside of the canopy in an average of 20-25 leaves per tree of selected cultivars. In addition to creating and pre-processing the dataset, a comparison of methods and hyper-parameters was also presented. For this, ten different methods of convolutional neural networks were selected, as well as eight training sets, where some hyper-parameters were varied, such as the use of a Flatten or Gap layer and the use or not of weights from ImageNet. In addition to this, the use or not of data augmentation was also tested. This comparison resulted in several interesting pieces of information for the study, starting with demonstrating that in the end, the choice of hyper-parameters (Sets) proved to be more important than the choice of the method itself (Models), with a standard deviation of approximately 4.6% when analyzing the means obtained by set and a standard deviation of approximately 1.8% when referring to the models. It was also possible to demonstrate the need to use cross-validation in this type of problem, mainly in cases with small test sets, which can easily lead to errors, as happened in the case studied, managing to increase the accuracy of the models by 4.3% on average first two iterations to the last five only using this technique. In general, although the best result was obtained in a set without data augmentation (98.4% accuracy) if the average is analyzed, the best results are obtained by sets with data augmentation, guaranteeing an improvement of 1% in accuracy when compared to the others. This leads us to assume that the same would happen when used in a real context since the algorithms that resorted to data augmentation were trained with greater variations in data. However, not having data to prove the statement will remain as an assumption for future work. Observing the number of epochs, on the contrary, there was an advantage in not using data augmentation in this aspect since the algorithms that did not use it converged on average three epochs faster than the ones that did, a number that rises to five epochs if the tests with five-fold cross-validation are considered.

The results obtained from this study allow us to conclude that most of the studied methods will enable this type of identification with high security, emphasizing that the results obtained by the first nine methods guarantee accuracy rates above 95%, rising to 98% for the first three methods. In practical terms, in about 2016 images 1976 were classified correctly. Emphasizing that the results presented refer to the current dataset and the four cultivars studied, not being able to extrapolate the accuracy of these data in the real context of the problem. Therefore, using this type of tools combined with experienced technicians or genetic identification techniques is always advised.

As main future works, it is possible to highlight the availability of a multi-platform application in order to test the chosen method in a real environment. As an essential part of the work, updating the dataset with more cultivars and images is always part of the objective. Finally, mentioning the possible optimization of these methods through their combination will also be the object of study, as well as the performance of more tests that allow us to increase the reliability of this type of application.

Funding

This work was carried out under the Project “OleaChain: Competências para a sustentabilidade e inovação da cadeia de valor do olival tradicional no Norte Interior de Portugal” (NORTE-06-3559-FSE-000188), an operation to hire highly qualified human resources, funded by NORTE 2020 through the European Social Fund (ESF) and was supported by international funds STEP, HORIZON-WIDERA-2021-ACCESS-03-01, n. 101078933. The authors are grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 (DOI: 10.54499/UIDB/05757/2020) and UIDP/05757/2020) (DOI: 10.54499/UIDB/05757/2020) and SusTEC (LA/P/0007/2021) (DOI: 10.54499/LA/P/0007/2020).

CRediT authorship contribution statement

João Mendes: Writing – review & editing, Writing – original draft, Software, Resources, Methodology, Investigation, Data curation, Conceptualization. **José Lima:** Writing – review & editing, Data curation, Conceptualization. **Lino Costa:** Writing – review & editing, Conceptualization. **Nuno Rodrigues:** Writing – review & editing, Methodology, Data curation, Conceptualization. **Ana I. Pereira:** Writing – review & editing, Resources, Methodology, Formal analysis, Conceptualization.

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 entire dataset is available in the url: https://dados.ipb._347_pt/dataset.xhtml?persistentId=doi:10.34620/dadosipb/TVYE8K.

References

[1] V. Uylaser, G. İzli, The historical development and nutritional importance of olive and olive oil constituted an important part of the Mediterranean diet, Crit. Rev. Food Sci. Nutr. 54 (2014) 1092–1101, https://doi.org/10.1080/10408398.2011.626874.
[2] M.A. Hashmi, A. Khan, M. Hanif, U. Farooq, S. Perveen, Traditional uses, phytochemistry, and pharmacology of olea europaea (olive), in: Evidence-Based Complementary and Alternative Medicine, vol. 2015, 2015.
[3] IOC, International olive oil, https://www.internationaloliveoil.org/changes-in-olive-oil-consumption/. (Accessed 8 February 2023), 2022.
[4] V. di Rienzo, S. Sion, F. Taranto, N. D’Agostino, C. Montemurro, V. Fanelli, W. Sabetta, S. Boucheffa, A. Tamendjari, A. Pasqualone, M. Zammit Mangion, M. Miazzi, Genetic flow among olive populations within the Mediterranean basin, PeerJ 6 (2018) e5260, https://doi.org/10.7717/peerj.5260.
[5] T. Mikrou, K. Kasimati, I. Doufexi, M. Kapsokefalou, C. Gardeli, A. Mallouchos, Volatile composition of industrially fermented table olives from Greece, Foods 10 (5) (2021) 1000, https://doi.org/10.3390/foods10051000.
[6] C. Campestre, G. Angelini, C. Gasbarri, F. Angerosa, The compounds responsible for the sensory profile in monovarietal virgin olive oils, Molecules 22 (11) (2017) 1833, https://doi.org/10.3390/molecules22111833.
[7] M. Brake, H. Migdadi, M. Al-Gharaibeh, S. Ayoub, N. Haddad, A. El Oqlah, Characterization of jordanian olive cultivars (olea europaea l.) using rapd and issr molecular markers, Sci. Hortic. 176 (2014) 282–289, https://doi.org/10.1016/j.scienta.2014.07.012.
[8] S. Abdessemed, I. Muzzaupo, H. Benbouza, Assessment of genetic diversity among Algerian olive (olea europaea l.) cultivars using ssr marker, Sci. Hortic. 192 (2015) 10–20, https://doi.org/10.1016/j.scienta.2015.05.015.
[9] A. Pasqualone, C. Montemurro, V. di Rienzo, C. Summo, V.M. Paradiso, F. Caponio, Evolution and perspectives of cultivar identification and traceability from tree to oil and table olives by means of dna markers, J. Sci. Food Agric. 96 (11) (2016) 3642–3657, https://doi.org/10.1002/jsfa.7711.
[10] R. Ben Ayed, H. Ben Hassen, K. Ennouri, A. Rebai, Genetic markers analyses and bioinformatic approaches to distinguish between olive tree (olea europaea l.) cultivars, Interdiscip. Sci.-Comput. Life Sci. 8 (4) (2016) 366–373, https://doi.org/10.1007/s12539-016-0155-x.
[11] M. Aksehirli-Pakyurek, G.C. Koubouris, P.V. Petrakis, S. Hepaksoy, I.T. Metzidakis, E. Yalcinkaya, A.G. Doulis, Cultivated and wild olives in Crete, Greece-genetic diversity and relationships with major Turkish cultivars revealed by ssr markers, Plant. Mol. Biol. Report. 35 (6) (2017) 575–585, https://doi.org/10.1007/s11105-017-1046-y.
[12] S.S. Martínez, D.M. Gila, A. Beyaz, J.G. Ortega, J.G. García, A computer vision approach based on endocarp features for the identification of olive cultivars, Comput. Electron. Agric. 154 (2018) 341–346, https://doi.org/10.1016/j.compag.2018.09.017.

[13] P. Vanloot, D. Bertrand, C. Pinatel, J. Artaud, N. Dupuy, Artificial vision and chemometrics analyses of olive stones for varietal identification of five French cultivars, Comput. Electron. Agric. 102 (2014) 98–105, https://doi.org/10.1016/j.compag.2014.01.009.
[14] A. Beyaz, M. Ozkaya, D. İçen, Identification of some Spanish olive cultivars using image processing techniques, Sci. Hortic. 225 (2017) 286–292, https://doi.org/10.1016/j.scienta.2017.06.041.
[15] M. Aria, C. Cuccurullo, bibliometrix: an r-tool for comprehensive science mapping analysis, J. Informetr. 11 (4) (2017) 959–975, https://doi.org/10.1016/j.joi.2017.08.007.
[16] M.A. Nielsen, Neural Networks and Deep Learning, vol. 25, Determination Press, San Francisco, CA, USA, 2015.
[17] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551, https://doi.org/10.1162/neco.1989.1.4.541.
[18] A. Khan, A. Sohail, U. Zahoora, A.S. Qureshi, A survey of the recent architectures of deep convolutional neural networks, Artif. Intell. Rev. 53 (8) (2020) 5455–5516, https://doi.org/10.1007/s10462-020-09825-6.
[19] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[20] S. Mancuso, F. Nicese, Identifying olive (olea europaea) cultivars using artificial neural networks, J. Am. Soc. Hortic. Sci. 124 (1999) 527–531, https://doi.org/10.21273/JASHS.124.5.527.
[21] G.A. Azim, M.K. Sousow, Multilayer feed forward neural networks for olive trees identification, in: A. Press (Ed.), Proceedings of the 26th IASTED International Conference on Artificial Intelligence and Applications, AIA '08, USA, 2008, pp. 420–425.
[22] A. Nasiri, A. Taheri-Garavand, D. Fanourakis, Y. Zhang, N. Nikoloudakis, Automated grapevine cultivar identification via leaf imaging and deep convolutional neural networks: a proof-of-concept study employing primary Iranian varieties, Plants 10 (2021) 1628, https://doi.org/10.3390/plants10081628.
[23] Y. Liu, J. su, L. Shen, N. Lu, Y. Fang, F. Liu, Y. Song, B. Su, Development of a mobile application for identification of grapevine (vitis vinifera l.) cultivars via deep learning, Int. J. Agric. Biol. Eng. 14 (2021) 172–179, https://doi.org/10.25165/j.ijabe.20211405.6593.
[24] C. Liu, J. Han, B. Chen, J. Mao, Z. Xue, S. Li, A novel identification method for apple (malus domestica borkh.) cultivars based on a deep convolutional neural network with leaf image input, Symmetry 12 (2) (2020), https://doi.org/10.3390/sym12020217.
[25] J. Chen, J. Han, C. Liu, Y. Wang, H. Shen, L. Li, A deep-learning method for the classification of apple varieties via leaf images from different growth periods in natural environment, Symmetry 14 (8) (2022), https://doi.org/10.3390/sym14081671.
[26] F. Sanz-Cortés, J. Martinez-Calvo, M. Badenes, H. Bleiholder, H. Hack, G. Llácer, U. Meier, Phenological growth stages of olives trees (olea europaea), Ann. Appl. Biol. 140 (2005) 151–157, https://doi.org/10.1111/j.1744-7348.2002.tb00167.x.
[27] G. Bradski, The opencv library, Dr. Dobb's J. Softw. Tools Prof. Program. 25 (11) (2000) 120–123.
[28] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6) (1986) 679–698, https://doi.org/10.1109/TPAMI.1986.4767851.
[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252, https://doi.org/10.48550/ARXIV.1409.0575.
[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint, arXiv:1409.1556, 2014, https://doi.org/10.48550/ARXIV.1409.1556.
[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: I.C. Society (Ed.), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: I.C. Society (Ed.), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: I.C. Society (Ed.), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[34] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: I.C. Society (Ed.), Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[35] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: I.C. Society (Ed.), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[36] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, arXiv preprint, arXiv:1704.04861, 2017, https://doi.org/10.48550/ARXIV.1704.04861.
[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, in: I.C. Society (Ed.), Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[38] M. Tan, Q. Le, Efficientnet: rethinking model scaling for convolutional neural networks, in: PMLR (Ed.), International Conference on Machine Learning, 2019, pp. 6105–6114.
[39] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q.V. Le Mnasnet, Platform-aware neural architecture search for mobile, in: IEEE (Ed.), IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 2815–2823.
[40] M. Hossin, S.M. N, A review on evaluation metrics for data classification evaluations, Int. J. Data Min. Knowl. Manag. Process 5 (2015) 01, https://doi.org/10.5121/ijdkp.2015.5201.
[41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: large-scale machine learning on heterogeneous systems, software available from, http://tensorflow.org, 2015, https://doi.org/10.48550/ARXIV.1603.04467.
[42] A.F. Agarap, Deep learning using rectified linear units (relu), arXiv preprint, arXiv:1803.08375, 2018, https://doi.org/10.48550/ARXIV.1803.08375.
[43] M. Lin, Q. Chen, S. Yan, Network in network, arXiv preprint, arXiv:1312.4400, 2013.
[44] M. Huh, P. Agrawal, A.A. Efros, What makes imagenet good for transfer learning?, arXiv:1608.08614, 2016.
[45] T. Fontanari, T.C. Froes, M. Recamonde-Mendoza, Cross-validation strategies for balanced and imbalanced datasets, in: J. Xavier-Junior, R. Rios (Eds.), Intelligent Systems, PT I, 11th Brazilian Conference on Intelligent Systems (BRACIS), Univ Campinas, Campinas, Brazil, Nov 28-Dec 01, 2022, in: Lecture Notes in Computer Science, Brazilian Inst. Data Sci., vol. 13653, 2022, pp. 626–640.
[46] G. Varoquaux, Cross-validation failure: small sample sizes lead to large error bars, NeuroImage 180 (2017), https://doi.org/10.1016/j.neuroimage.2017.06.061.
[47] K. Ishihara, K. Matsumoto, Comparing the robustness of resnet, swin-transformer, and mlp-mixer under unique distribution shifts in fundus images, Bioengineering 10 (12) (2023), https://doi.org/10.3390/bioengineering10121383.
[48] A. Abirami, S. Bhuvaneswari, B.S. Gowri, S. Narayanan, N. Sharmitha, M. Sowbarnika, Mri-based brain tumour classification using efficientnetb7 model with transfer learning, J. Surv. Fish. Sci. 10 (2S) (2023) 1737–1750, https://doi.org/10.17762/sfs.v10i2S.945.
[49] X. Wu, R. Liu, H. Yang, Z. Chen, An xception based convolutional neural network for scene image classification with transfer learning, in: 2020 2nd International Conference on Information Technology and Computer Application (ITCA), 2020, pp. 262–267.
[50] Y. Gulzar, Z. Unal, H. Aktas, M.S. Mir, Harnessing the power of transfer learning in sunflower disease detection: a comparative study, Agriculture (Basel) 13 (8) (AUG 2023), https://doi.org/10.3390/agriculture13081479.