Journal of Food Measurement and Characterization (2022) 16:2545–2555
https://doi.org/10.1007/s11694-022-01367-5

ORIGINAL PAPER

Cultivar identification of pistachio nuts in bulk mode through EfficientNet deep learning model

Alireza Soleimanipour[1] · **Mohsen Azadbakht**[1] · **Abbas Rezaei Asl**[1]

Received: 6 September 2021 / Accepted: 11 March 2022 / Published online: 24 March 2022

Abstract

Robust and readily accessible identification systems for agricultural products have received increasing attention in inspection processes. The conventional visual inspection methods rely on human experts and could not be automated efficiently. This paper presents the application of the state-of-the-art Convolutional Neural Network (CNN) model to identify the top 4 most popular Iranian pistachio cultivars, including 'Fandoghi', 'Kaleh-Ghouchi', 'Ahmad-Aghaei', and 'Akbari'. An image dataset was collected by acquiring pistachio samples images in the bulk mode under the condition of placing perpendicular to the camera with a fixed distance and natural lightening. The feature extraction block of EfficientNet-B3 was combined with a custom classification block. The proposed model was fine-tuned using the pistachio image dataset and adjusted to identify the various cultivars. The pistachio identification system was implemented using Python programming and the Keras API with TensorFlow Machine Learning framework. The results show that the classification accuracy of the best model evaluated on a hold-out test dataset is 98.00%. On the other hand, the proposed CNN model using EfficientNet-B3 presents average precision, recall, and F1-score values of 96.73%, 96.70%, and 96.67%, respectively. These results confirmed that the implementation of these methods as a mobile application and/or on an automated processing line could be a useful and effective approach to develop fast and robust food processing and inspection systems.

Keywords Pistachio · Visual inspection · Image classification · Deep learning · EfficientNet

Introduction

Pistachio is one of the oldest flowering nuts which have been described as being native to Iran, Central Asia, and Western Asia. Among the various species of the genus Pistacia which is referred to as pistachio, the fruits of *P. vera* attain a large enough size to be acceptable to consumers as edible nuts [1]. This delicious nut contains a lot of healthy fats and is an excellent source of fiber, protein, and antioxidants which makes it one of the world's most abundant and most delicious tree nuts [2]. Pistachio is cultivated in the Middle East, the United States, and Mediterranean countries [3]. Iran is the major pistachio producer in the world with an annual production of more than 57 tones (49%) [4]. The Iranian pistachio industry relies mainly upon a limited number of cultivars, which include 'Ohadi' or 'Fandoghi' (Round), 'Ahmad-Aghaei' (Long), 'Akbari' (Super Long), and 'Kaleh-Ghouchi' (Jumbo) and cover more than 95% of the cultivation area [5].

Cultivar identification of pistachio nuts is a difficult task for non-expert persons; however, it is a crucial issue especially for customers in global markets. On the other hand, separation of the mixed nuts and grading them are desirable for producers to provide a uniform product for the customers [6]. In convention, pistachio nuts classification and grading are performed by trained experts. This is requiring dedicated personnel and the results are mostly variable dependent on the experience of the person. Besides this traditional method, some more accurate methods are conducted in labs via content analysis techniques. For instance, determination of origin country of pistachio by DSC (differential scanning calorimetry) and HPLC [7]; detection of the geographic growing origin of pistachio using chemical profiling [8]; authentication of pistachio varieties using chemical partitioning and DNA fingerprinting [9]; classification of pistachio cultivars using multi-elemental fingerprinting [10].

✉ Alireza Soleimanipour
asoleimani@gau.ac.ir

[1] Department of Biosystem Engineering, Gorgan University of Agricultural Sciences and Natural Resources, Gorgan, Iran

Even though the marker-based methods have good accuracy, there are some pitfalls in employing them, including invasive, time-consuming and high-cost; which make these approaches unsuitable for routine and ready-to-use applications [11]. Therefore, there is a growing demand for rapid, non-invasive, and unbiased techniques to do automate food processing procedures [12]. In this perspective, some research works were carried out to classify pistachio nuts based on the physical properties of various cultivars. The majority of these works focused on using the image and/or signal processing, as well as machine learning techniques. Kouchakzadeh and Brati [13] developed a neural network (NN) to discriminate five major cultivars of Iranian pistachio nuts, in which different physical properties of pistachio samples were fed to the NN model as input. Omid et al. [14] developed a sorting system for pistachio nuts upon the combination of acoustic signals, principal component analysis, and artificial neural network. Ghazanfari et al. [15] employed histogram features of gray-scale images and Fourier descriptors to classify pistachio nuts into one of three USDA size grades or as having closed shells. Omid et al. [16] implemented an algorithm based on image processing, and machine learning techniques of ANN and SVM, for grading peeled pistachio kernels into five classes.

Some vision-based cultivar identification and quality inspection systems involve computer recognition of their attributes such as length, width, volume [17]. Alternatively, through model-based object detection methods, mainly deep learning boosted models, these processes can be implemented as an end-to-end system without the need to manually definition of features. In recent years, a growing number of research articles have been published which try to apply CNN and deep learning algorithms to classify various kernel/seeds. In Table 1, some most recent applications of CNN models for seed/kernel cultivar identification have been summarized. So far, the potential of identifying Iranian commercial pistachios (Jumbo, Long, Round, and Super long) trees via fine-tuning the EfficientNet-B1 deep learning model has been investigated [18]. To the best of the authors' knowledge, the present study is the first to investigate the potential of a pre-trained CNN model to classify pistachio nuts in the bulk mode based on their cultivars. This research aims to develop a low-cost and readily accessible system to identify popular pistachio cultivars using only a usual image. The proposed method can be readily implemented as a mobile application to identify pistachio cultivars in markets, as well as employed in an industrial system to do the crop processing operations more automated.

Materials and methods

Pistachio samples

The top 4 most popular and commercially prominent pistachio cultivars, including 'Fandoghi' (Round Pistachio), 'Kaleh-Ghouchi' (Jumbo Pistachio), 'Ahmad-Aghaei' (Long Pistachio), and 'Akbari' (Super Long Pistachio) were evaluated. Pistachio samples were prepared from Tehran dried fruits and nuts market stall, Tehran main market, Iran. Pistachio images were acquired in bulk mode under the condition of perpendicular to the camera with a fixed distance (20 cm), and natural lightening. An image dataset containing a collection of 600 images (150 images per cultivar) was prepared by randomly placing the samples under the camera (Fig. 1). A mobile phone camera (Samsung Galaxy A12, 48 megapixels, and aperture f/2.0; Samsung Electronics Co., Ltd., South Korea)

Table 1 Some recent applications of CNN models for seed/kernel cultivar identification

Seed/Kernel	No. of classes	Algorithm	Classification accuracy (%)	References
Barley	11	CNN (two Conv[a] + FC[b])	> 93	[19]
Canola	8	CFS[c] + ANN classifier	98	[20]
Corn	9	CNN + ANN classifier	98.1	[21]
Corn	4	CNN (two Conv + FC)	93.3	[22]
Hazelnut	17	CNN (four Conv + FC)	98.63	[23]
Pepper	4	CNN + SVM classifier	99.02	[24]
Rice	4	CNN (four Conv + FC)	87	[25]
Soybean	3	pixel-wise CNN	> 90	[26]
Sunflower	4	GoogleNet	95	[27]
Wheat	30	CNN-ATT[d]	93.01	[28]

[a]Convolutional layer

[b]Fully connected layer

[c]Correlation-based feature selection

[d]Convolutional neural network with attention

Fig. 1 Pistachio cultivars in bulk mode including; **A** Ahmad-Aghaei, **B** Akbari, **C** Fandoghi, and **D** Kaleh-Ghouchi

was employed for pistachio image acquisition. Images were acquired in the resolution of 2992×2992 pixels, then in a preprocessing step, resized based on the CNN model input resolution.

Image augmentation

Many computer vision tasks have been resolved via the application of deep convolutional neural networks. However, these networks are heavily reliant on big data to avoid overfitting. Overfitting will occur when a network learns a function with very high variance and low bias. In this situation, the network perfectly models the training data but is weak in generalization [29]. There are some functional solutions such as transfer learning, batch normalization, dropout regularization to overcome overfitting. A complete survey of regularization techniques in deep learning has been compiled by Kukacka et al. [30]. In contrast to the methods mentioned, image augmentation artificially creates new training images from existing ones through different ways of processing or combination of multiple processing, such as zoom in/out, flip, shift, random rotation, do change in brightness, and contrast, etc. Image augmentation assists the model to better generalize the validation dataset by minimizing the distance between the training and validation set [31].

As the image data augmentation is typically only applied to the training dataset and not to the test dataset, one-third (33%) of the whole acquired images was set aside for testing the models (200 images). The remained images as the training set were augmented six times (by four rotations in 0, 90, 180, and 270°, and two horizontally and vertically flips). Also, a random and up to $\pm 10\%$ change in brightness and contrast was done in all augmented images. By image augmentation, a total number of 2400 (400×6) images were used for training and validation of the pre-trained deep learning model.

Transfer learning

CNNs constitute a class of deep feed-forward ANN. A CNN model can successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The CNN architectures are constructed from different combinations of three main kinds of layers, including the convolutional layer, the pooling layer, and the fully connected layer. The convolution layers hierarchically extract nonlinear features of images. As more general features such as edges and blobs, extracted in the first larger layers, becoming more specific at the deeper layers. They may be understood as banks of filters (so-called kernels) that highlight specific patterns in images through transforming an input image into feature maps [32]. Pooling layers then decrease the dimension of the feature maps to reduce the number of network parameters. The 2D feature maps resulted from the combination of convolution and pooling layers are converted to 1D vectors in fully connected layers. The classification task is complemented by feeding the output on the feature extraction part of a CNN model to a custom classification block, which consists of various layers such as dropout, batch normalization, etc. [33].

A deep CNN model needs a lot of data to have tremendous performance in comparison to traditional and regular machine learning techniques. The pistachio nuts images are roughly not sufficient to use for training the deep models from scratch with random weights. This is a comprehensive issue in deploying deep learning approaches. Therefore, there is a need to create high-performance learners trained with more easily obtained data from different domains [34]. In this regard, there exist various pre-trained CNN models which were effectively trained by the ImageNet dataset, comprised approximately 1 million images and 1000 object classes [35]. Thus, transfer learning with the concept of fine-tuning the pre-trained models is utilized to overcome the restriction of the insufficient quantity of training data [36]. During the training procedure, the pistachio images were

used to update the models' parameters. It will make model training possible with less training data and speed up the process considerably.

From 2012, as a breakthrough year for deep learning, various deep CNN architectures were introduced. Commonly employed pre-trained CNNs include AlexNet [37], VGGNet [38], Inception Net [39], ResNet [40], Xception [41], and EfficientNet [42] as the more recent one. Through initial trial and error of applying the common models on the existing dataset, as well as referring to the graph of "Model Size vs. ImageNet Accuracy" for various models [42], depicted in, the EfficientNet-B3 pre-trained model was selected for evaluation.

EfficientNet-B3 CNN model

Model scaling is one of the main issues in designing CNN architectures. That means, how to increase the model size to prevent overfitting, besides providing better accuracy. In this way, many innovative models were introduced, but many are not effective in terms of computational load, which is caused by a high number of parameters [43]. EfficientNet model provides better accuracy and improves efficiency by reducing the parameters and FLOPS (floating-point operations per second) manifold. EfficientNet is the current state-of-the-art model by reaching 84.4% accuracy with 66 M parameters in the ImageNet classification problem. EfficientNet can be considered a group of CNN models consists 8 models from B0 to B7, which each subsequent model number referring to variants with more parameters and higher accuracy. However, by growing the model number, accuracy increases noticeably while the number of parameters does not increase much. Another difference of EfficientNet with the other CNN models is the usage of the Swish activation function in its architecture, instead of the Rectifier Linear Unit (ReLU) activation function [42].

In Fig. 2, the configuration of EfficientNet-B3 is depicted. The main building block of EfficientNet is MBConv, which is similar to the inverted residual blocks used in MobileNetV2 [44]. There is a skip connection between the beginning and end of an MBConv convolutional block. In MBConv blocks, the input activation maps are first expanded to increase the depth of the feature maps, then compressed to reduce the number of channels in the output feature map. The shortcut connections are used between bottlenecks and connect narrow layers while wider layers are between skip connections. This architecture decreases the model size and the overall number of operations required.

In this research, the EfficientNet-B3 model has been used for the transfer learning process. The top layers of the model are replaced by a classification block to customize the model for the present case study. In the architecture of the classifier, A global average pooling layer is used to reduce the total number of parameters. In addition, 2 dense layers with RELU activation functions are added. The first dense layer contains 128 units and follows by a batch normalization layer and a dropout layer, while the second one contains 64 units and follows by only a dropout layer (Table 2). The batch normalization layer is included for holding the inputs of layers in the same range. Also, the dropout layers (with the rate of 20%) decrease the overfitting risk in the training process. The last layer is an output dense layer with a softmax activation function and contains four output units, which have been added to create a 4-class pistachio identification system. The total number of parameters of the proposed model is 10,989,043. Table 1 presents a summary of the structure of the proposed model (layers order), number of parameters, and output shape for each layer. It is worth noting that the EfficientNet-B3 model takes input images of shape (300, 300, 3), and the input data should range [0, 255] unlike the other CNN models. Normalization is included as part of the model.

Experimental setup

The EfficientNet-B3 model used in this research is recently developed by Google and is available from Github repositories. This model is one of the EfficientNet model groups that

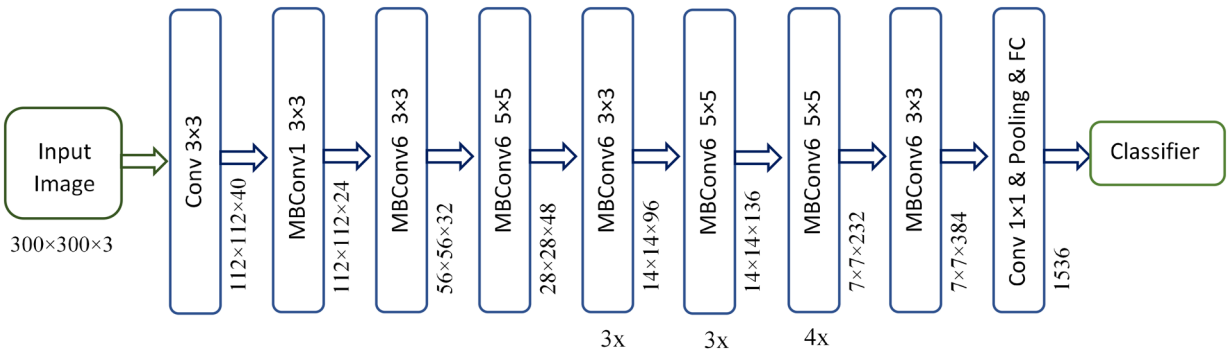

Fig. 2 Schematic representation of EfficientNet-B3

Table 2 Layer's type and the number of parameters for the proposed model

Layer (Type)	Output shape	Param #
input (Input layer)	[(None, 300, 300, 3)]	0
EfficientNetB3 (Functional)	(None, 10, 10, 1536)	10,783,535
global_average_pooling2d	(None, 1536)	0
dense (Dense)	(None, 128)	196,736
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
batch_normalization	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 4)	260
Total params: 10,989,043		
Trainable params: 10,901,612		
Non-trainable params: 87,431		

were pre-trained using the ImageNet dataset and scaled for use in transfer learning based on the complexity, resource, and purpose of the problems. The model with a custom classifier was compiled using Google Colab notebook in GPU run time type. Colab notebook is a web-based interactive environment and offers some GPUs (K80s, T4s, P4s, and P100s) without costs since it is provided by Google for research activities. All codes are realized in Python programming language and the Keras API with TensorFlow framework as tensor backend engine.

Evaluation metrics

The raw imagery data, before any further processing, was split into two parts. One-third of collected images were set aside as the final test set, and the remaining images were preprocessed and augmented for training and initial validation of the model. So, the model validation has been conducted in two phases. For the first phase, the fivefold cross-validation method was employed and the model is validated using the augmented image dataset. In the second phase, the hold-out dataset is used to evaluate the performance of the model. The test dataset contains images that have not been augmented and used during the training phase.

Given the all datasets are manually labeled data and the category of each image are known. Four independent groups of pistachio nuts with labels have been created from the testing data, which are named ground-truth data. After running the trained EfficientNet-B3 classifier on the test dataset, a predicted class was obtained for each image that exists in the dataset. The model performance is measured via assessing how well the predicted classes match with the ground-truth data, using accuracy, precision, recall, and F1-score. As well, the confusion matrix has been extracted to analyze the sources of errors in classification.

For more explanation, each row in a confusion matrix represents an actual class, while each column represents a predicted class. A perfect classifier would have only true positives and true negatives, so its confusion matrix would have nonzero values only on its main diagonal (top left to bottom right). If we consider a class, in particular, one instance may be distinguished as one of the following groups. True positive when the instance considered correctly as identified; false positive when the instance considered incorrectly as identified; true negative when the instance considered correctly as rejected; and false negative when the instance considered incorrectly as rejected. If we consider the number of instances for true positive, false positive, true negative, false negative as #TP, #FP, #TN, #FN, respectively; then the metrics will be as follows [45].

$$accuracy = \frac{\#TP + \#TN}{\#TP + \#TN + \#FP + \#FN} \times 100 \tag{1}$$

$$precision = \frac{\#TP}{\#TP + \#FP} \times 100 \tag{2}$$

$$recall = \frac{\#TP}{\#TP + \#FN} \times 100 \tag{3}$$

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \times 100 \tag{4}$$

The metrics of the models trained using each data fold have been computed for each pistachio class, then the average values for each fold have been calculated. Also, the receiver operating characteristic (ROC) curve of the test dataset has been drawn for the classifier, and its area under the ROC curve (AUC) was calculated.

Results and discussion

Evaluation of training process

Training of EfficientNet-B3 CNN model using pistachio nuts images was carried out using stratified fivefold cross-validation method. The training was conducted in 2 phases. In the first phase, the trainability of the base model of EfficientNet was set to False, so its weights had been frozen and they were no longer trainable. In the other words, the weights of the proposed classifier block had been tuned. For all folds, the first phase was last in 4 training epochs. Then in the second training phase, all layers of the model were set to trainable. The number of epochs for the second phase was set to 20. But, to avoid overfitting of the models, the early stopping method was used. Early stopping is a regularization technique in deep learning. This method stops training at

the point when performance on the validation dataset starts to degrade. The method monitored the validation loss and its *patience* hyperparameter was set to 4 with the ability to restore the best weights. Moreover, the mini-batch size was 16. So, training in each epoch was done in 120 steps, which was calculated by dividing the number of training images by the mini-batch size. It is worth noting that the experiments were conducted using GPU run time type of Google Colab notebook.

The summation of time elapsed for the training of the model in both the first and second phases was 68.5 min. The models training had been completed in 2760 training iterations. In all of the training runs, the process was halted by the early stopping. ADAM optimization method and Categorical Cross Entropy were used in compiling the model. The initial learning rate was 1×10^{-5}. Also, the initial training rate was reduced as the training progresses using an adaptive learning rate method (its hyperparameters were *beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e-08, decay = 0.0*). The training and validation graphs for both loss and accuracy are shown in Fig. 3. The curves are related to the trained CNN model using the fold with the best performance. After training the CNN model on a fivefold cross-validation dataset, the best model is identified using evaluation on the test dataset, which was not used in training and validation processes.

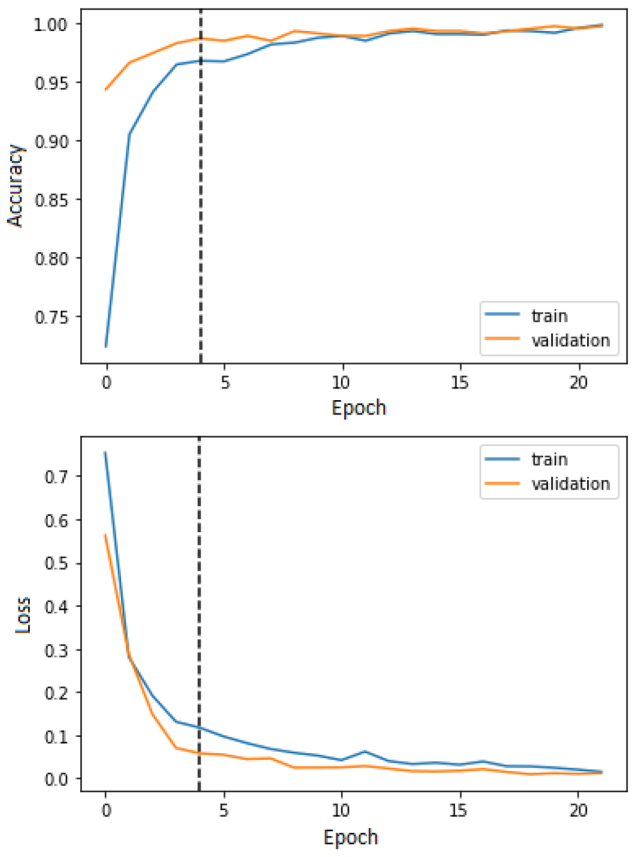

Fig. 3 Classification accuracy and cross-entropy loss for training and validation dataset (1st and 2nd training phases specified via the dashed line)

The accuracy and loss of training and validation datasets for all of the five folds were perfect. For instance, the accuracy and loss curve depicted in Fig. 3, which is related to fold 3 with the highest performance on the test dataset, shows that after epoch 10 the gap between training accuracy and validation accuracy stabilizes and there is no significant change in the values of both curves. The same happened for loss curves related to training and validation datasets. For this fold, the classification accuracy of the training and validation dataset for the last saved weights were 99.90% and 99.79%, respectively. When the training process was halted through Early Stopping, the model weights have been restored from the epoch with the best value of the monitored quantity. Also, the cross-entropy loss for training and validation datasets was achieved at 0.0152 and 0.0118, respectively.

Evaluation of the model performance

The results for the cultivar identification models of pistachio nuts have been presented in this section. In total, 600 images were collected from 4 different cultivars, one-third of them i.e., 200 images put aside for testing, by performing augmentation on the rest of the images, 2400 images were used for training. The results are reported for each model trained on the dataset folds, and the average value is also calculated. The results for evaluation metrics, including precision, recall, and F1-score are computed for each cultivar in separate tables.

The results for the 'Ahmad-Aghaei' cultivar are reported in Table 3. The maximum precision value is 97.92 occurred at the model trained using the twofold dataset. The maximum recall is 100% at the threefold and fivefold models, and the highest F1-score is 98.04 occurred in the model trained on threefold. On the other hand, the model trained on the fourfold dataset has the lowest performance concerning the 'Ahmad-Aghaei' cultivar. The lowest precision value

Table 3 Results of the identification model for 'Ahmad-Aghaei' cultivar

Fold	Precision (%)	Recall (%)	F1-score (%)
1	96.00	96.00	96.00
2	97.92	94.00	95.92
3	96.15	100	98.04
4	95.65	88.00	91.67
5	94.34	100	97.09
Average	96.01	95.90	95.74

Table 4 Results of the identification model for 'Akbari' cultivar

Fold	Precision (%)	Recall (%)	F1-score (%)
1	98.04	100	99.00
2	94.34	100	97.09
3	100	100	100
4	94.34	100	97.09
5	100	100	100
Average	97.34	100	98.64

Table 5 Results of the identification model for 'Fandoghi' cultivar

Fold	Precision (%)	Recall (%)	F1-score (%)
1	96.15	100	98.04
2	98.00	98.00	98.00
3	98.00	98.00	98.00
4	94.34	100	97.09
5	98.00	98.00	98.00
Average	96.89	98.80	97.83

Table 6 Results of the identification model for 'Kaleh-Ghouchi' cultivar

Fold	Precision (%)	Recall (%)	F1-score (%)
1	97.87	92.00	94.84
2	93.88	92.00	92.93
3	97.92	94.00	95.92
4	95.83	92.00	93.88
5	97.87	92.00	94.84
Average	96.67	92.40	94.48

Table 7 Results of the identification model between all cultivars

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
1	97.00	97.02	97.00	96.97
2	96.00	96.03	96.00	95.98
3	98.00	98.02	98.00	97.98
4	95.00	95.04	95.00	94.92
5	97.50	97.55	97.5	97.48
Average	96.70	96.73	96.70	96.67

is 94.34% occurred in the fivefold model. The minimum recall and F1-score values are 88.00% and 91.67% occurred both in the fourfold model, respectively. Moreover, the average precision, recall, and F1-score are 96.01%, 95.90%, and 95.74%, respectively.

Table 4 presents the results of various models evaluated for the 'Akbari' cultivar. The highest precision value is 100% reported at the models trained using threefold and fivefold, and the lowest precision value is 94.34% reported for the twofold model. The recall value for all the models is perfect and equal to 100%. It means that the model trained on each of the folds has correctly identified samples for the 'Akbari' cultivar, so the number of false negative samples for this cultivar is zero. This could be due to the more distinguishable shape of the 'Akbari' cultivar in comparison to the other cultivars. Also, the F1-score value ranges from 97.09% at the twofold and fourfold models to 100% at the threefold and fivefold models. The average results of 97.34%, 100%, and 98.64% are reported for precision, recall, and F1-score, respectively.

The results of the models for the 'Fandoghi' cultivar are reported in Table 5. Concerning this cultivar, the precision value ranges from 94.34% at the fourfold model to 98% in the twofold, threefold and fivefold models. The minimum recall value occurred in the models trained using twofold, threefold and fivefold datasets and is equal to 98%, and the maximum recall value is 100% presented at the onefold and fourfold models. Finally, the lower value for the F1-score is 97.09% reported at the fourfold model, and the higher value is 98.04% obtained in the onefold model. The average results concerning the 'Fandoghi' cultivar are 96.89%, 98.80%, and 97.83% reported for precision, recall, and F1-score, respectively.

Finally, the results for the identification of the 'Kaleh-Ghouchi' cultivar using the proposed CNN model are reported in Table 6. The model trained using the threefold dataset has the highest performance values, which include 97.92%, 94.00%, and 95.92% for precision, recall, and F1-score, respectively. On the other hand, the lowest precision value occurred in the twofold model and is equal to 93.88, the minimum recall value is 92.00% reported in all the models except the threefold model. Also, the lowest F1-score value is 92.93 and presented in the twofold model. The average values for precision, recall, and F1-score of the models considering the 'Kaleh-Ghouchi' cultivar is 96.67%, 92.40%, and 94.48%, respectively.

Table 7 presents the average values of the metrics used to evaluate the models. These results are calculated between all of the pistachio cultivars, which are the top 4 most important ones. The average values for accuracy, precision, recall and F1-score are 96.70%, 96.73%, 96.70% and 96.67%, respectively. The highest accuracy is 98% and reported for the model trained using the threefold dataset, and the lowest value is 95% occurred at the fourfold model. The maximum average precision is presented in the threefold model, and the minimum value is 95.04 reported at the fourfold model. The worst result for the recall is reported in the fourfold model and is 95.00%, and the best recall value is reported at the threefold model and is equal to 98.00%. Finally, the minimum average F1-score is 94.92% occurred in the fourfold model, and the maximum

value is 97.98% presented in the model trained using the threefold dataset.

The results presented in Table 7 show that the proposed CNN method is efficient for cultivar identification of pistachio nuts in bulk mode. Furthermore, the results show that the model trained using the threefold dataset presents a better performance in comparison to others, while the model trained using the fourfold model has the weakest, but still acceptable, performance. These results confirm that the proposed classification approach using the state-of-the-art EfficientNet-B3 CNN model can be utilized as a robust and reliable identification system in its application cases, such as global markets. For instance. the receiver operating characteristic for the model trained using fourfold training data is presented in Fig. 4.

Analysis of classification errors

Classification purpose is assigning a true class to a survey item. The classification models try to do this task without or with the least errors, but errors are inevitable. The classification error occurs when a respondent provides a false response to a survey item, and it happens in one of two ways: a false negative response or a false positive response. "A false negative response occurs when a respondent indicates an event did not occur when it did. A false positive response occurs when a respondent indicates an event did occur when it really did not" [46]. In this section, the source of classification errors for the proposed CNN model has been analyzed. With this purpose in mind, the confusion matrix for all of the models trained on fivefold has been drawn.

The confusion matrix of the model trained on onefold is presented in Fig. 5. The classification accuracy of this model on the test dataset is 97.00%. As can be seen, the number of false negative cases for the 'Ahmad-Aghaei' cultivar is 2 (one case is falsely considered as 'Akbari' and another as 'Kaleh-Ghouchi'). Moreover, the number of false negative cases for the 'Kaleh-Ghouchi' cultivar is 4 (2 cases are falsely considered as 'Ahmad-Aghaei' and 2 cases as 'Fandoghi'). Finally, all samples for 'Akbari' and 'Fandoghi' cultivars are correctly identified.

Figure 6 presents the confusion matrix of the model trained using twofold. The classification accuracy of this model is 96.00%. Concerning the 'Ahmad-Aghaei' cultivar, 47 cases are correctly identified, while 3 cases are not (1

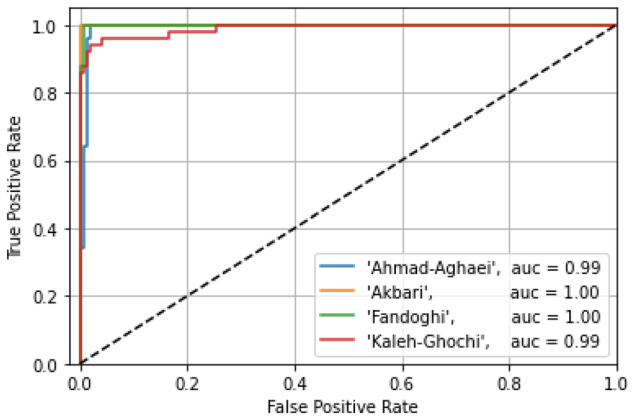

Fig. 4 ROC curve and AUC values of the cultivars evaluated using the fourfold classification model

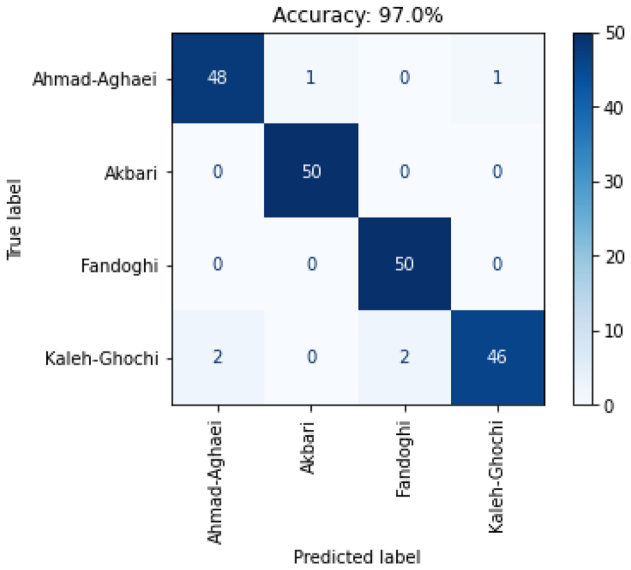

Fig. 5 Confusion matrix of the onefold classification model evaluated for the test dataset

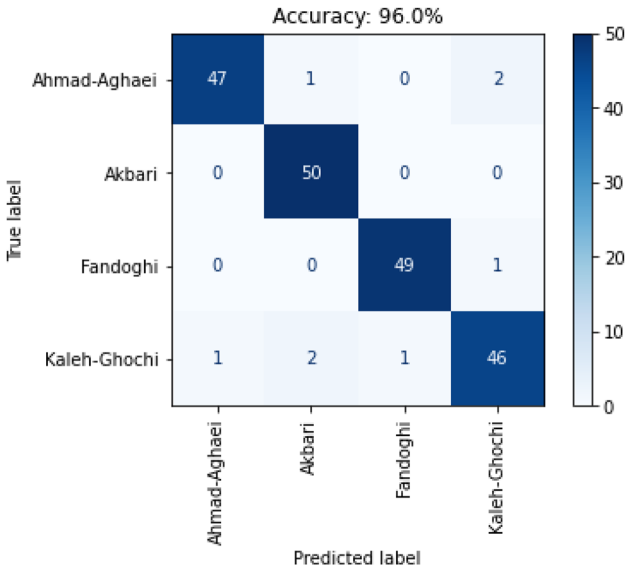

Fig. 6 Confusion matrix of the twofold classification model evaluated for the test dataset

case is falsely considered as 'Akbari' and 2 cases as 'Kaleh-Ghouchi'). Similar to the onefold results, all samples for the 'Akbari' cultivar are correctly identified. For the 'Fandoghi' cultivar only 1 case is misidentified as 'Kaleh-Ghouchi'. Moreover, the number of false negative cases for the 'Kaleh-Ghouchi' cultivar is 4 (which 1, 2, and 1 case are falsely considered as 'Ahmad-Aghaei', 'Akbari', and 'Fandoghi', respectively.

The confusion matrix related to the model trained on the threefold dataset has been shown in Fig. 7. As can be seen, this model classification accuracy is 98.00% and has the best performance among the models. The number of false negative cases for 'Ahmad-Aghaei' and 'Akbari' cultivars is zero. Only, 1 sample for the 'Fandoghi' cultivar is identified incorrectly, and falsely labeled as 'Kaleh-Ghouchi'. Finally, 3 samples of the 'Kaleh-Ghouchi' cultivar were misidentified by this model (2 samples are incorrectly considered as 'Ahmad-Aghaei' and 1 sample as 'Fandoghi').

In Fig. 8, the confusion matrix of the model trained on fourfold has been shown. As can be seen, the classification accuracy of this model is 95.00% and the lowest one among the models. The false negative cases for this model are related to 'Ahmad-Aghaei' and 'Kaleh-Ghouchi' cultivars. In the case of 'Ahmad-Aghaei', 6 samples are falsely labeled (3, 1 and 2 samples are misidentified as 'Akbari', 'Fandoghi' and 'Kaleh-Ghouchi', respectively). Also, 4 samples of the 'Kaleh-Ghouchi' cultivar are mistakenly labeled (2 cases are falsely considered as 'Ahmad-Aghaei' and 2 cases as 'Fandoghi'). Moreover, all samples related to 'Akbari' and 'Fandoghi' cultivars are truly identified.

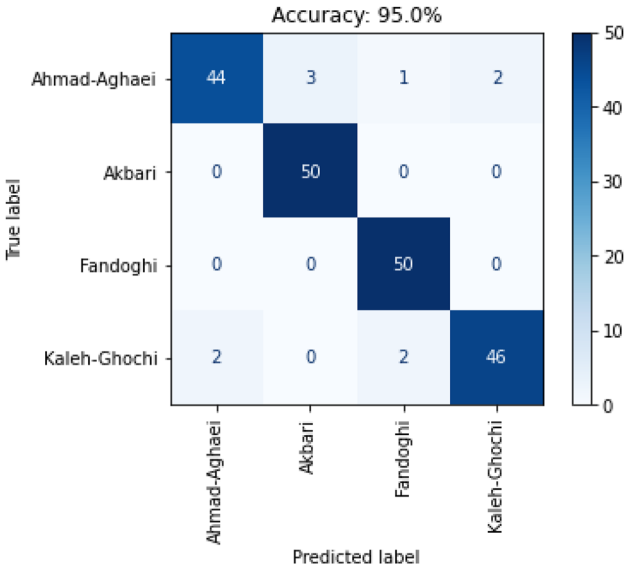

Fig. 8 Confusion matrix of the fourfold classification model evaluated for the test dataset

Finally, Fig. 9 shows the confusion matrix of the model trained using the fivefold model. The classification accuracy of this model is 97.50%. As can be seen, all samples of 'Ahmad-Aghaei' and 'Akbari' cultivars are truly labeled. In the case of the 'Fandoghi' cultivar, only 1 sample was falsely labeled as 'Kaleh-Ghouchi'. Moreover, 4 samples of the 'Kaleh-Ghouchi' cultivar were misidentified by this model (3 samples are falsely considered as 'Ahmad-Aghaei' and 1 sample as 'Fandoghi').

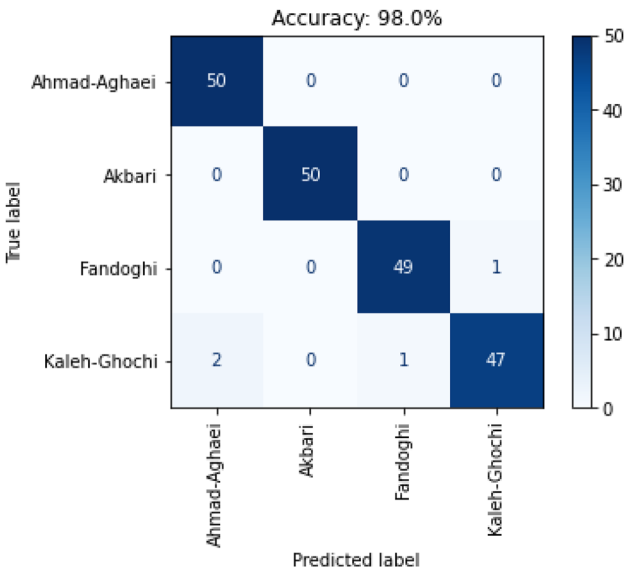

Fig. 7 Confusion matrix of the threefold classification model evaluated for the test dataset

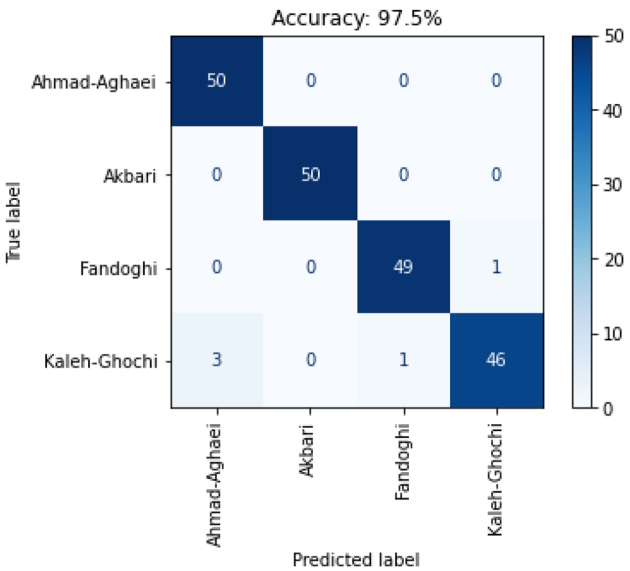

Fig. 9 Confusion matrix of the fivefold classification model evaluated for the test dataset

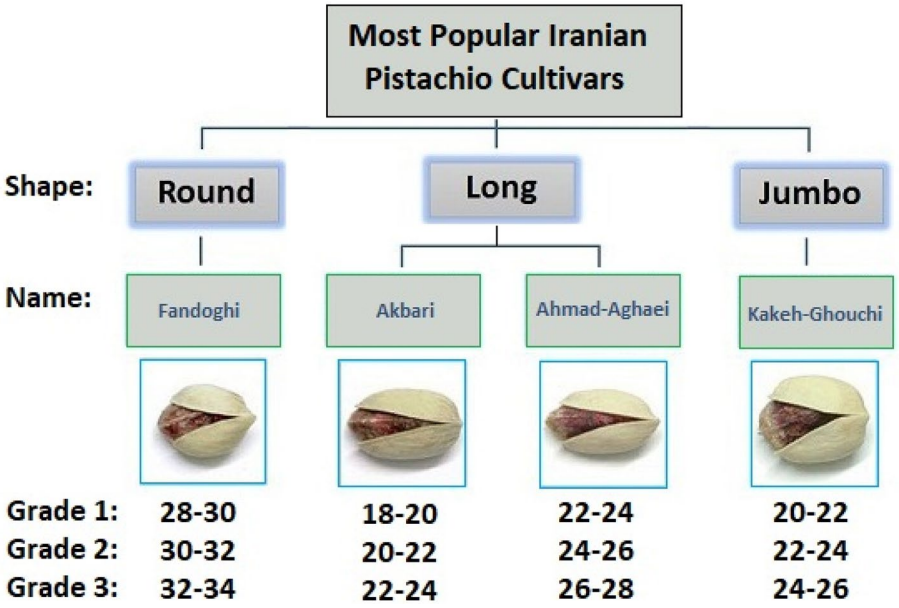

Fig. 10 Top four most popular Iranian pistachio cultivars. The grades are based on the number of nuts in one ounce

Generally, 'Kaleh-Ghouchi' is the cultivar that has the most falsely labeled cases, with a total false negative number of 19. Of those, more than 50% are misclassified as 'Ahmad-Aghaei' and more than 35% as 'Fandoghi'. Moreover, the 'Ahmad-Aghaei' cultivar with 11 false negative cases in summation, is in the second place in terms of misidentification. About 45% of falsely labeled samples of this cultivar were misclassified as 'Akbari' and 45% as 'Kaleh-Ghouchi'. 'Fandoghi' cultivar only in three cases falsely labeled and all of these cases are considered as 'Kaleh-Ghouchi'. Finally, all of the test samples related to the 'Akbari' cultivar are truly identified by the models.

The errors in the identification of the pistachio cultivars examined in this research may result from size and shape similarity between them. For a better explanation, a closed view of pistachio nut for the cultivars has been presented in Fig. 10. As the results of confusion matrices declared, the majority of misidentified samples are related to 'Kaleh-Ghouchi' and 'Ahmad-Aghaei' cultivars. With careful attention to Fig. 10, it can be realized that the shape of the 'Kaleh-Ghouchi' cultivar is somewhat similar to 'Fandoghi'; as well as from the size perspective, its nut size is close to 'Ahmad-Aghaei'. On the other hand, the error resource for misidentification of the 'Ahmad-Aghaei' cultivar may be due to its shape similarity to 'Akbari' and size equality with 'Kaleh-Ghouchi'. Sometimes, these shape and size similarities between pistachio cultivars, make identification tasks challenging for humans, even for expert persons. In this situation, kernel color and taste would be examined.

Conclusion

This paper has presented a deep learning-based approach to support the identification of pistachio cultivars in bulk mode. The top 4 most popular and important Iranian pistachio cultivars are used in experiments. The proposed approach implements EfficientNet-B3 architecture with a custom classifier and is trained and validated using fivefold cross-validation. Furthermore, a hold-out dataset is used for testing. The results show that the classification accuracy of the test dataset ranges from 95.00 to 98.00%. Concerning falsely labeled samples, it is concluded that the errors may result from size and shape similarity between pistachio cultivars, which make identification tasks sometimes challenging for humans, even for expert persons. So, the robust performance of the model, and the end-to-end nature of such methods in implementation, make it a useful and efficient approach to employ as an automated classifying system in postharvest processes, as well as an identification system implemented through a smartphone application for individuals involved in pistachio trade. Although, relying on the high classification capabilities of deep learning models and their fast inference time, such methods will attract even more demands for application, in the future.

Acknowledgements This research has been supported by Gorgan University of Agricultural Sciences and Natural Resources, Gorgan, Iran.

References

1. E.H. Shokraii, A. Esen, J. Agric. Food Chem. **36**, 425 (1988)
2. M. Kashaninejad, A. Mortazavi, A. Safekordi, L.G. Tabil, J. Food Eng. **72**, 30 (2006)
3. A. Menevseoglu, D.P. Aykas, E. Adal, J. Food Meas. Charact. **15**, 1075 (2021)
4. FAO, (FAOSTAT Online Stat. Serv. 2017). Food Agric. Organ. United Nations. (2017)
5. L. Pazouki, M. Mardi, P.S. Shanjani, M. Hagidimitriou, S.M. Pirseyedi, M.R. Naghavi, D. Avanzato, E. Vendramin, S. Kafkas, B. Ghareyazie, Conserv. Genet. **11**, 311 (2010)
6. A. Ghazanfari, J. Irudayaraj, A. Kusalik, Trans. ASAE **39**, 2319 (1996)
7. S.M. Dyszel, B.C. Pettit, J. Am. Oil Chem. Soc. **67**, 947 (1990)
8. K.A. Anderson, B.W. Smith, J. Agric. Food Chem. **53**, 410 (2005)
9. G. Mannino, C. Gentile, M.E. Maffei, Phytochemistry **160**, 40 (2019)
10. M. Esteki, E. Heydari, J. Simal-Gandara, Z. Shahsavari, M. Mohammadlou, Food Control **124**, 107889 (2021)
11. A. Taheri-Garavand, A. Nasiri, D. Fanourakis, S. Fatahi, M. Omid, N. Nikoloudakis, Plants **10**, 1406 (2021)
12. A. Soleimanipour, G.R. Chegini, J. Massah, Agric. Eng. Int. CIGR J. **20**, 219–228 (2018)
13. A. Kouchakzadeh, A. Brati, Int. J. Emerg. Sci. **2**, 259 (2012)
14. M. Omid, A. Mahmoudi, M.H. Omid, Expert Syst. Appl. **36**, 11528 (2009)
15. A. Ghazanfari, J. Irudayaraj, A. Kusalik, M. Romaniuk, J. Agric. Eng. Res. **68**, 247 (1997)
16. M. Omid, M.S. Firouz, H. Nouri-Ahmadabadi, S.S. Mohtasebi, Eng. Agric. Environ. Food **10**, 259 (2017)
17. A. Soleimanipour, G.R. Chegini, J. Food Meas. Charact. **13**, 571 (2019)
18. A. Heidary-Sharifabad, M.S. Zarchi, S. Emadi, G. Zarei, Br. Food J. **39**, 107478 (2021)
19. M. Kozłowski, P. Górecki, P.M. Szczypiński, Biosyst. Eng. **184**, 155 (2019)
20. S. Qadri, S. Furqan Qadri, A. Razzaq, M. Ul Rehman, N. Ahmad, S.A. Nawaz, N. Saher, N. Akhtar, D.M. Khan, Int. J. Food Prop. **24**, 493 (2021)
21. S. Javanmardi, S.H. Miraei Ashtiani, F.J. Verbeek, A. Martynenko, J. Stored Prod. Res. **92**, 101800 (2021)
22. J. Zhang, L. Dai, F. Cheng, J. Food Meas. Charact. **15**, 484 (2021)
23. A. Taner, Y.B. Öztekin, H. Duran, Sustainability **13**, 6527 (2021)
24. K. Sabanci, M.F. Aslan, E. Ropelewska, M.F. Unlersen, J. Food Process. Eng. **2021**, e13955 (2021)
25. Z. Qiu, J. Chen, Y. Zhao, S. Zhu, Y. He, C. Zhang, Appl. Sci. **8**, 212 (2018)
26. S. Zhu, L. Zhou, C. Zhang, Y. Bao, B. Wu, H. Chu, Y. Yu, Y. He, L. Feng, Sensors **19**, 4065 (2019)
27. F. Kurtulmuş, J. Food Meas. Charact. **15**, 1024 (2021)
28. L. Zhou, C. Zhang, M.F. Taha, X. Wei, Y. He, Z. Qiu, Y. Liu, Front Plant Sci. **11**, 575810 (2020)
29. C. Shorten, T.M. Khoshgoftaar, J. Big Data **6**, 1 (2019)
30. J. Kukačka, V. Golkov, D. Cremers, (2017) https://arxiv.org/abs/1710.10686
31. J.J. Saenz-Gamboa, M. de la Iglesia-Vayá, J.A. Gómez, Proc. Int. Conf. Pattern Recognit. **12**, 5214 (2020)
32. A. Kamilaris, F.X. Prenafeta-Boldú, Comput. Electron. Agric. **147**, 70 (2018)
33. M. Farooq, E. Sazonov, in *Int. Conf. Bioinforma. Biomed. Eng.* (Springer, 2017), pp. 464–472.
34. K. Weiss, T.M. Khoshgoftaar, D. Wang, J. Big Data **3**, 1 (2016)
35. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Int. J. Comput. Vis. **115**, 211 (2015)
36. S.H. Miraei Ashtiani, S. Javanmardi, M. Jahanbanifard, A. Martynenko, F.J. Verbeek, IEEE Access **9**, 100380 (2021)
37. A. Krizhevsky, I. Sutskever, G.E. Hinton, Adv. Neural Inf. Process. Syst. **25**, 1097 (2012)
38. K. Simonyan and A. Zisserman, (2014) https://arxiv.org/abs/1409.1556
39. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (2015), pp. 1–9.
40. K. He, X. Zhang, S. Ren, J. Sun, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (2016), pp. 770–778.
41. F. Chollet, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (2017), pp. 1251–1258.
42. M. Tan, Q. Le, in *Int. Conf. Mach. Learn.* (PMLR, 2019), pp. 6105–6114.
43. Ü. Atila, M. Uçar, K. Akyol, E. Uçar, Ecol. Inform. **61**, 101182 (2021)
44. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (2018), pp. 4510–4520.
45. A. Beyaz, D.M. Martínez Gila, J. Gómez Ortega, J. Gámez García, Postharvest Biol. Technol. **150**, 129 (2019)
46. M. Berzofsky, P. Biemer, W. Kalsbeek, Sect. Surv. Res. Methodes-JSM **2008**, 3667 (2008)