



ARTICLE

## Improving Date Fruit Classification Using CycleGAN-Generated Dataset

Dina M. Ibrahim<sup>1,2,\*</sup> and Nada M. Elshennawy<sup>2</sup>

<sup>1</sup>Department of Information Technology, College of Computer, Qassim University, Buraydah, 51452, Saudi Arabia

<sup>2</sup>Computers and Control Engineering Department, Faculty of Engineering, Tanta University, Tanta, 37133, Egypt

\*Corresponding Author: Dina M. Ibrahim. Emails: d.hussein@qu.edu.sa, dina.mahmoud@f-eng.tanta.edu.eg

Received: 04 March 2021 Accepted: 11 October 2021

### ABSTRACT

Dates are an important part of human nutrition. Dates are high in essential nutrients and provide a number of health benefits. Date fruits are also known to protect against a number of diseases, including cancer and heart disease. Date fruits have several sizes, colors, tastes, and values. There are a lot of challenges facing the date producers. One of the most significant challenges is the classification and sorting of dates. But there is no public dataset for date fruits, which is a major limitation in order to improve the performance of convolutional neural networks (CNN) models and avoid the overfitting problem. In this paper, an augmented date fruits dataset was developed using Deep Convolutional Generative Adversarial Networks (DCGAN) and CycleGAN approach to augment our collected date fruit datasets. This augmentation is required to address the issue of a restricted number of images in our datasets, as well as to establish a balanced dataset. There are three types of dates in our proposed dataset: Sukkari, Ajwa, and Suggai. After dataset augmentation, we train our created dataset using ResNet152V2 and CNN models to assess the classification process for our three categories in the dataset. To train these two models, we start with the original dataset. Thereafter, the models were trained using the DCGAN-generated dataset, followed by the CycleGAN-generated dataset. The resulting results demonstrated that when using the ResNet152V2 model, the CycleGAN-generated dataset had the highest classification performance with 96.8% accuracy, followed by the CNN model with 94.3% accuracy.

### KEYWORDS

Dates fruits; data augmentation; DCGAN; CycleGAN, deep learning; convolution neural networks

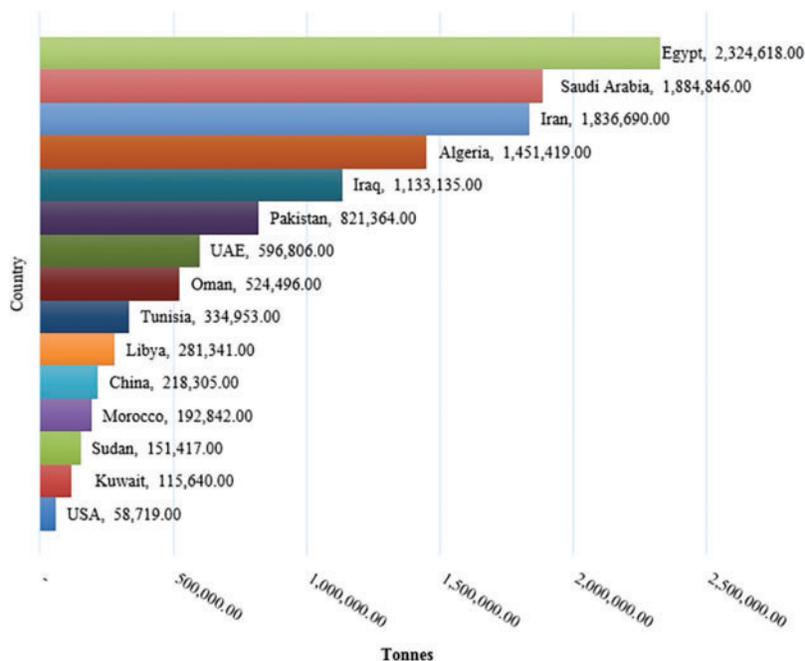
## 1 Introduction

Dates are essential for human nutrition. Dates are high in key nutrients and have several health advantages. Date fruits are also thought to protect against a variety of ailments, including heart disease and cancer [1]. Date fruit, in addition to its nutritional value, is an important agricultural crop throughout the Middle East and Northern Africa, where it plays a significant economic role. As shown in Fig. 1, Egypt and Saudi Arabia were considered as the first and second leaders in date production in the world at 2020, where the production of date fruits was 2,324,618 tones and 1,884,846, respectively.



There are over 400 different types of dates produced around the world, including Khalas, Sukkari, Rutab, Ajwa, Suggai, Berhi, and so on [2] and [3]. The National Center for Date Palm in Saudi Arabia created a quiz to show which date fruits are the most popular. Sukkari was the most popular type of date, with 69.5%, followed by Khalas with 48.4%, and Rutab with 11.1%, according to the survey, which comprised 8518 respondents. Finally, Suggai comes in fourth with 11% [4].

When utilizing a Convolutional Neural Network (CNN) to categorize images of date fruits, we recently demonstrated a reasonable level of accuracy. CNN models require a bigger amount of training data to provide improved outcomes in the classification of date fruit. Because gathering significant amounts of data in the agricultural field, especially for date fruits, is a challenging task that requires more resources and time. There is no public dataset for date fruits, which is a major limitation. Except for one study by Altaheri et al. [5], which provided a dataset containing five different types of dates, all previous studies produced their own dataset and did not make it publicly available, however the dataset contains date images in orchard contexts at various maturity stages. This problem can be solved by using data augmentation techniques. Basic manipulation and complex picture modification techniques were used in data augmentation approaches to create new data from original training data without the requirement to gather new data [6].



**Figure 1:** The top 15 largest Dates Producer countries in 2020

Because deep learning networks include an excessive number of parameters, training models require a big dataset to learn these parameters. The model training will be over-fitting if this does not happen. Data Augmentation is a good way to increase the size of the training dataset without having to acquire fresh data. It uses simple picture manipulation and complex image transformation techniques to create new data from existing training data. The Generative

Adversarial Network is the most widely used advanced approach (GAN) [7]. As a result, Data Augmentation techniques were employed to overcome dataset limitations such as a lack of data and an imbalanced dataset.

The Generative Adversarial Network (GAN) is a generative model and a deep learning network. The Generator and Discriminator neural networks make up the GAN. GAN builds new data from existing data, with the goal of producing visuals that are comparable to real data without having to view it. Discriminator, on the other hand, seeks to distinguish between produced and actual data. When the discriminator accuracy deteriorates to the point that the generated images appear to be very real, the training procedure will end.

Since the introduction of GANs in 2014 [7], a number of GAN extensions have been developed, including DCGAN (Deep Convolutional GANs) [8], PGAN (Progressively Growing GANs) [9], LAPGAN (Laplacian pyramids GANs) [10], CycleGAN [11], CGAN (Conditional GANs) [12], and StyleGAN [13]. In image data augmentation, DCGANs and CycleGAN are common GAN architectures.

The goal of this paper is to use Deep Convolutional Generative Adversarial Networks (DCGAN) and CycleGAN approaches to augment our gathered date fruit datasets. This augmentation is required to address the issue of a restricted number of images in our datasets, as well as to establish a balanced dataset. There are three types of dates in our proposed dataset: Sukkari, Ajwa, and Suggai. We train our produced dataset using ResNet152V2 and CNN models after dataset augmentation to analyze the classification process for our three categories in the dataset. We begin by using the original dataset to train these two models. The models are then trained using the DCGAN-generated dataset, and finally using the CycleGAN-generated dataset. We analyze the performance of the results when we finish the training step.

The remainder of this paper is arranged in the following manner. [Section 2](#) discusses the work-related issues. [Section 3](#) explains the methods and materials, including the original dataset, the DCGAN and CycleGAN-generated datasets, as well as the training models. The experimental results and discussions are presented in [Section 4](#). Finally, in [Section 5](#), we draw our conclusion.

## 2 Related Work

Different date augmentation approaches have been employed in prior studies and applied to plant and leaf datasets. Which share characteristics in common with the date fruit dataset. Gandhi et al. [14] proposed a system that automatically categorized plant kinds and diseases based on the leaf to assist the Indian farmers. The authors used the PlantVillage dataset to collect around 56,000 photos (38 classes and 19 crops). Crops cultivated locally, as well as associated illnesses, are featured in this collection of photos. To supplement the limited number of datasets, the authors employed the Deep Convolutional Generative Adversarial Networks (DCGAN) algorithm, followed by the MobileNet and Inception-v3 models for classification. As a result, the model's accuracy for Inception-v3 and MobileNet was 88.6% and 92%, respectively.

Similarly, Zhang et al. [15] employed DCGAN to augment citrus canker images in order to address the issue of a limited number of images available. 800 photos were enhanced six times before being fed into generative models. They used a DCGAN generating model with modified mute layers and trained the models for 2,000 epochs. Then, based on human specialists' diagnoses, divide the dataset into two categories: positive and negative. For classification tasks, AlexNet classifiers were employed using 100 epochs, and the accuracy achieved was 93.7%.

Suryawati et al. [16] suggested an unsupervised feature learning architecture derived from DCGAN for automatic Gambung tea clone detection. The Gambung Clone dataset was used, which consisted of 1,297 leaf images from two different types of Gambung tea clones. To identify Gambung tea clones, the proposed feature learning architecture adds an encoder to the DCGAN network as a feature extractor. The model's accuracy was 91.74%, with a loss of 0.32%.

Image augmentation techniques and deep learning-based image augmentation approaches were utilized by Arun Pandian et al. [6] to overcome data shortage in agricultural fields. Plant leaf disease detection and diagnosis, in particular, were difficult undertakings. The plant leaf disease dataset was used, which includes 54305 plant leaf photos and 38 distinct healthy and sick classifications. To validate the performance of data augmentation strategies, state-of-the-art transfer learning techniques like as VGG16, ResNet, and InceptionV3 were used. InceptionV3, the best-performing model, outperformed all other models in terms of accuracy across all datasets.

There have been numerous studies on date fruits in general. Nonetheless, there are few researches on date fruit classification, particularly those that use CNN as a classifier. Furthermore, there are just a few datasets that are open to the public. For classifying date fruits, Altaheri et al. [17] suggested a deep learning-based machine vision framework. They produced a dataset consisting of 8072 date photos taken in an orchard setting. The dataset contains five different types of dates in various stages of maturity (Naboot Saif, Khalas, Barhi, Meneifi, and Sullaj). This data set is open to the public. AlexNet and VGGNet are two CNN architectures employed by the authors. As a consequence, the proposed model attained accuracy of 99.01%, 97.25%, and 98.59%.

Faisal et al. [2] presented a system for classifying date fruits in orchards based on ripeness for an intelligent harvesting decision system, as did the authors (IHDS). The suggested technology can detect seven different stages of date maturity in orchards (Immature Stage 1, Immature Stage 2, Pre-Khalal, Khalal, Khalal with Rutab, Pre-Tamar, and Tamar). They worked with a dataset created by Altaheri et al. [5]. VGG-19, Inception-v3, and NASNet were the three CNN architectures used. The suggested IHDS's maximum performance metrics were 99.4% in testing. Nasiri et al. [3], on the other hand, employed CNN to classify the (Shahani dates) into mature phases (Khlal, Rutab and Tamar). They generated their own dataset and have so far collected over 1300 photos. However, the dataset is unavailable. The VGG-16 model with fine-tuning network was used to create the CNN model. As a result, the average categorization accuracy per class ranged from 96 to 99%.

In Pakistan, Magsi et al. [18] proposed using deep learning and computer vision techniques to distinguish date fruits. They used CNN to classify three types of date fruits (Aseel, Karbalain, and Kupro) based on the features after preprocessing and feature extraction (Color, Shape and Size). They produced a 500-image dataset that is only available to members of the public. The model, on the other hand, was 97% accurate.

The authors of [19] proposed a deep learning system based on enhanced Faster R-CNN for multi-class fruit detection. Fruit image library building, data argumentation, enhanced Faster RCNN model generation, and performance evaluation are all part of the proposed methodology. Using 4000 real-world images, this study created a multi-labeled and knowledge-based outdoor orchard image library. In addition, the convolutional and pooling layers have been improved, resulting in a more accurate and faster detection. The test results reveal that the suggested algorithm outperforms traditional detectors in terms of detecting accuracy and processing time,

indicating that it has great potential for developing an autonomous and real-time harvesting or yield mapping/estimation system.

With respect to [20], the authors use several types of similarity measure methods to evaluate the effectiveness of deep learning features in view-based 3D model retrieval on four popular datasets (ETH, NTU60, PSB, and MVRED). The performance of hand-crafted and deep learning features is compared in detail, and the robustness of deep learning features is evaluated. Finally, the distinction between single-view and multi-view deep learning features is assessed. It reveals that multi-view deep learning outperforms single-view deep learning features in terms of computational complexity.

According to [21], the author provides a non-destructive and cost-effective approach for automating the visual examination of fruits' freshness and appearance based on state-of-the-art deep learning techniques and stacking ensemble methods. To determine the best model for grading fruits, they trained, tested, and compared the performance of various deep learning models such as ResNet, DenseNet, MobileNetV2, NASNet, and EfficientNet. The performance of the developed system was shown to be superior to that of earlier methods used on the same data sets. Furthermore, accuracy was found to be 96.7% for apples and 93.8% for bananas during real-time testing on actual samples, indicating the usefulness of the devised system.

In [22], they presented date fruits categorization utilizing texture descriptors and shape-size features. A color image of a date is dissected into its color components, the texture pattern of the date is then encoded using a local texture descriptor such as a local binary pattern (LBP) or a Weber local descriptor (WLD) histogram applied to each of the components. To characterize the image, the texture patterns from all of the components are combined. The suggested approach classifies dates with greater than 98% accuracy.

The work presented by [23] examined a number of automatic categorization methods using six recommended features based primarily on (color, size, and texture). Images of four different types of dates in Saudi Arabia are included in the dataset. In addition to the confusion matrix, the results of the testing confirmed the validity of the selected properties of dates based on other accuracy metrics. Furthermore, the results revealed that the Support Vector Machine model outperformed the others (Neural Network, Decision Tree, and Random Forest models), despite the tiny dataset employed. The Vector Machine had a precision of 0.738, whereas the others were between 0.6 and 0.69.

Hossain et al. [24] use 5G technology to provide a framework for picture categorization that would satisfy smart city consumers. They design an automatic date fruits classification system as a case study in order to supply the demand for date fruits. A deep learning approach is used in the suggested system, along with fine-tuning pre-trained models. The date fruits' photos are transmitted with low latency and in real time thanks to edge computing and caching. The proposed framework's viability is demonstrated by the experimental results.

To this end, the contribution of this work is divided into two steps: the first step is to augment our collected data fruit datasets using Deep Convolutional Generative Adversarial Networks (DCGAN) and CycleGAN techniques. This augmentation is essential to address the issue of our limited dataset, as well as create a balanced dataset. In our suggested dataset, we have three different sorts of dates: Sukkari, Ajwa, and Suggai. The second step is to test the generated dataset by using classification models such as ResNet152V2 and modified CNN models to classify the date fruit dataset into Sukkari, Ajwa, and Suggai. The classification and training process for these two models begin with using the original dataset. The models then are trained using

the DCGAN-generated dataset, and finally using the CycleGAN-generated dataset. We analyze the performance of the results and compare the two classification models in all the cases of the datasets.

### 3 Materials and Methods

The description of the original date fruit datasets is presented in this section. Then, using DCGAN and CycleGAN augmentation algorithms, we create augmented images. Following that, we use ResNet152V2 and CNN classification models to train these datasets.

#### 3.1 Dataset

A comprehensive image of the dataset must be built in order to design a robust vision system using deep learning. Because there is not a publicly available dataset. We generated a dataset of 628 photos of three date fruit varieties: Sukkari, Suggai, and Ajwa. We manually collected images from an online website and stored them in .jpg format using smartphones. The Maturity stage is represented by all of the photographs of various dates (Tamar stage). Each date variety, however, differs from the others in terms of shape, color, texture, and size, as shown in Fig. 2.



**Figure 2:** Samples from the original dataset for Sukkari, Ajwa, and Suggai date fruits

Table 1 displays the distribution of the dataset images between the three type classes for which the dataset was categorized. When the images are different sizes and have a distinct background. We divided the dataset into training and testing sets for data division (80% of the images for the training set and 20% for the testing set). In addition, the number of images for the training and testing sets were displayed.

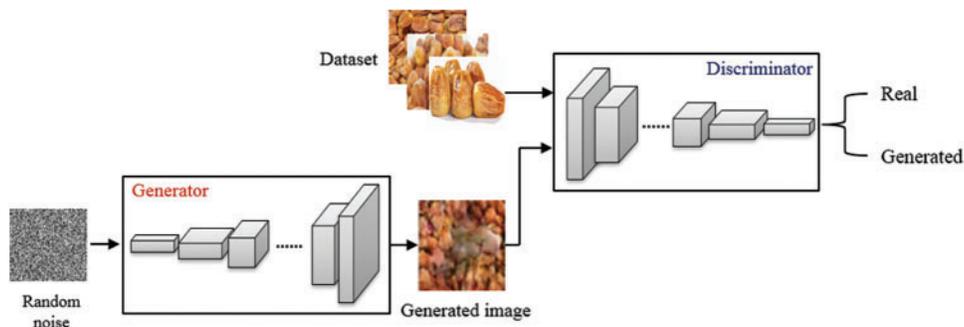
**Table 1:** Description of number of images in the original dataset

Type of class	Train set (80%)	Test set (20%)	Total (100%)
Sukkari	169	50	219
Suggai	165	53	209
Ajwa	156	44	200
Total	490	147	628

### 3.2 Data Augmentation Using Deep Convolutional Generative Adversarial Network (DCGAN)

The Deep Convolutional Generative Adversarial Network (DCGAN) is a form of GAN that builds the generator and discriminator using a convolutional neural network (CNN). DCGAN is a system that fills the gap between CNN's supervised and unsupervised learning success. DCGAN outperforms regular GAN in terms of performance. As a result, it is most widely employed in the classification of agricultural and medical images [25].

In the DCGAN architecture, stride convolutions on the discriminator and fractional-strided convolutions on the generator replaced the GAN's max pooling layers. Furthermore, batch normalization was applied to both the discriminator and the generator, ReLU activation was used in the generator for all layers except the output, which was done with tanh, and LeakyReLU activation was used in the discriminator for all layers [25]. The generator is supplied by a random noise input, while the discriminator has inputs from the original dataset and the generator's output, as shown in Fig. 3.



**Figure 3:** DCGAN model for generating Date fruit images

Table 2 shows the number of images generated and output from the DCGAN architecture. The total number of Sukkari, Ajwa, and Saggai images has increased by 200 each. These generated images were added to the original data set and divided into 80% and 20% for training and testing, respectively.

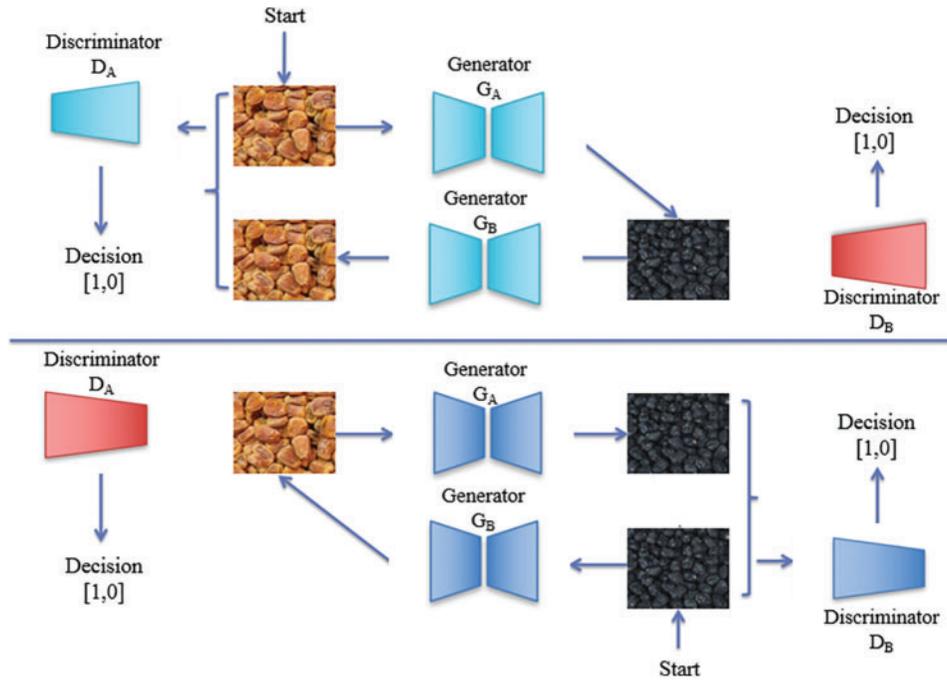
**Table 2:** Description of number of images generated by DCGAN model

Type of class	Train set (80%)	Test set (20%)	Total (100%)
Sukkari	329	90	419
Suggai	325	93	409
Ajwa	316	94	400
Total	970	277	1228

### 3.3 Data Augmentation Using Cycle Generative Adversarial Network (CycleGAN)

CycleGAN is a method for automatically training image-to-image translation models without using paired samples. Translating an apple to an orange or vice versa is an example of image to image translation. CycleGAN eliminates the need for paired data by employing a cycle consistency loss. It can translate from one domain to another without the source and target domains having a one-to-one mapping. The purpose of the image-to-image translation issue is to use a training

set of matched image pairs to learn the mapping between an input image and an output image. For two domains X (i.e., Sukkari) and Y (i.e., Ajwa), CycleGAN learns a mapping to generate Sukkari from Ajwa and to generate Ajwa from Sukkari. This mapping is done between Sukkari and Ajwa, as well as between Suggai and Ajwa. The CycleGAN model for producing Date fruit images is shown in Fig. 4.



**Figure 4:** CycleGAN model for generating Date fruit images

The number of produced images created by the CycleGAN architecture is shown in Table 3. The total number of Sukkari, Ajwa, and Suggai images each increased by 500. These images were then added to the original dataset and separated into 80% training and 20% testing, respectively.

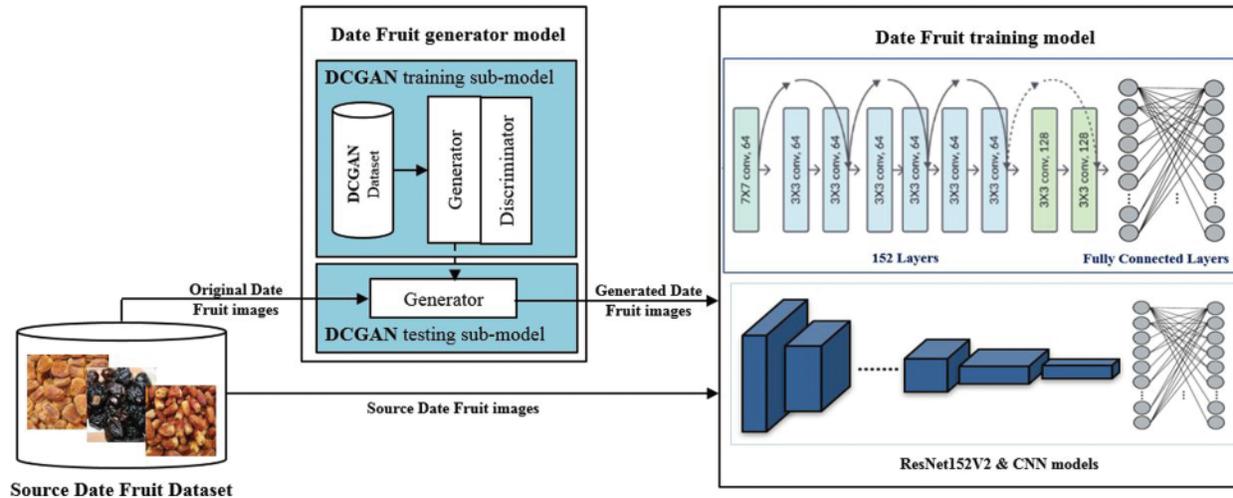
**Table 3:** Description of number of images generated by CycleGAN model

Type of class	Train set (80%)	Test set (20%)	Total (100%)
Sukkari	569	150	719
Suggai	565	153	709
Ajwa	556	144	700
Total	1690	447	2128

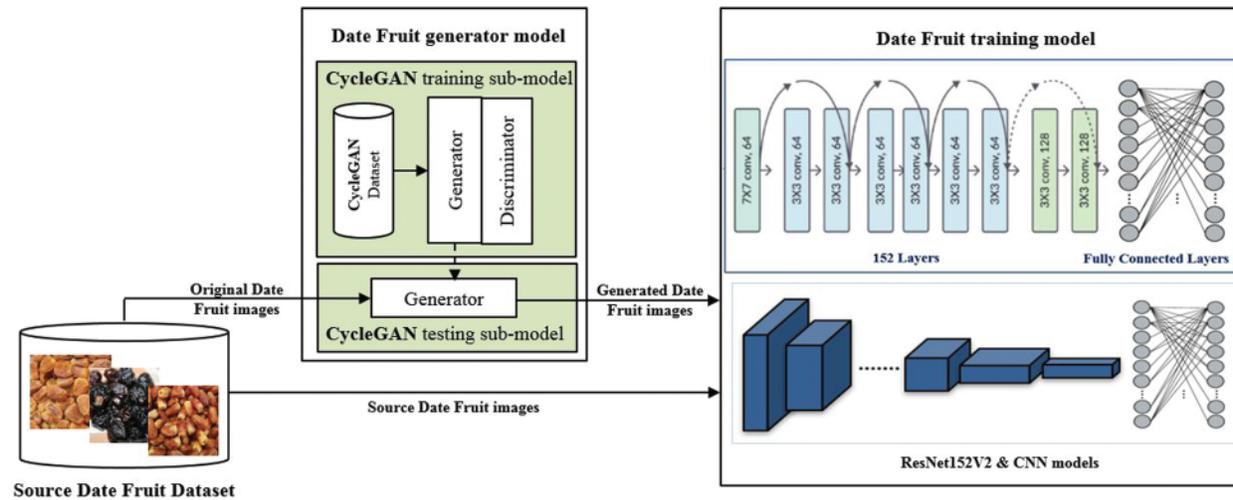
### 3.4 Methodology

Our methodology block diagrams are shown in Figs. 5 and 6. As clearly seen from the figure, the methodology consists of two main stages: date fruit generated model which generates the augmented dataset using DCGAN as in Fig. 5 and using CycleGAN as in Fig. 6. Then, the date fruit training model that applies deep learning models, ResNet152V2 and CNN, for feature

extraction, and classification. The proposed model uses the generated images as well as the original dataset as inputs, and the final output is the classification of the input image into one of the three classes: Sukkari, Ajwa, and Suggai.



**Figure 5:** The proposed methodology based on dataset augmented by DCGAN model

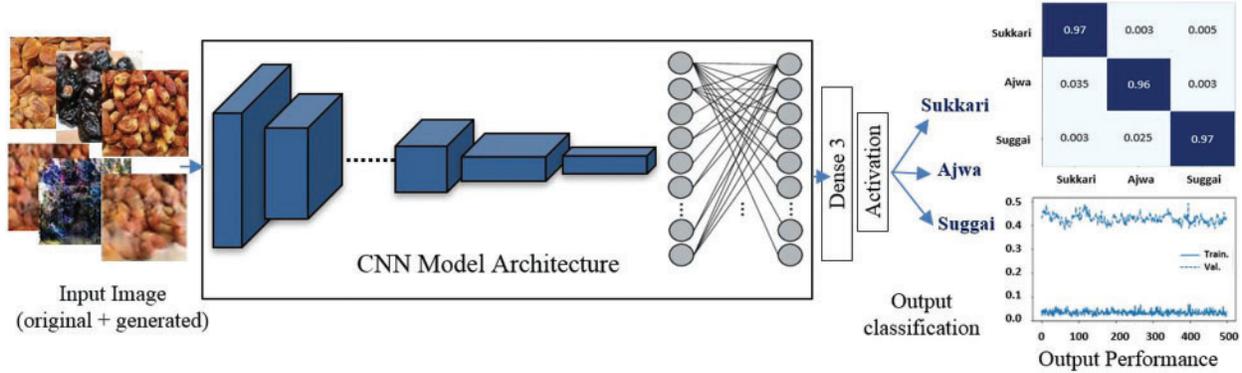


**Figure 6:** The proposed methodology based on dataset augmented by CycleGAN model

### 3.4.1 Convolutional Neural Network (CNN) Model Architecture

Convolutional neural networks (CNNs) are a mathematical construct made up of three types of layers: convolutional, pooling, and fully connected layers. The convolution and pooling layers extract visual features, whereas a fully connected layer turns those characteristics into final output, such as classification. As shown in Fig. 7, the standard CNN design consists of one or more stacks of multiple convolution and pooling layers, followed by one or more fully linked layers.

The output of one layer feeds the next layer, resulting in hierarchically more complex retrieved features [26].



**Figure 7:** The proposed CNN model architecture

To extract features, a convolution layer uses a combination of linear and nonlinear processes such as convolution and activation functions. Convolution is a form of linear operation that extracts features from an input image by applying a limited array of parameters called kernel. Multiple kernels are used in the convolutional process to create a number of feature maps, each of which extracts various features. To construct the activation maps, the linear operation result is processed over a nonlinear activation function, with only the activated features being transmitted to the next layer [26,27]. The most common used is the rectified linear unit (ReLU), where it is simply calculated (Eq. (1)), the function returns 0 if it receives any negative input, but for any positive value  $x$ :

$$f(x) = \max(0, x) \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (1)$$

To reduce the number of learnable parameters, a pooling layer minimizes the dimension of the feature maps. The most common type of pooling operation is max pooling, which collects patches from the input feature maps, extracts the largest value in each patch, and discards the remaining values [27]. Eq. (2) represents this mathematically:

$$f_{max} = \max_{n \times m}(A_{n \times m}) \quad (2)$$

The final convolution or pooling layer creates feature maps that are flattened. That is, the mean is converted to a one-dimensional numerical array and linked to one or more completely connected layers. Each input is coupled to each output through weights in the completely connected layer, also known as the dense layer. The final fully connected layer uses the SoftMax activation function and contains a number of output nodes equal to the number of classification classes. SoftMax is the most known function used in the classification tasks and given by the Eq. (3) where  $n$  neurons for  $n$  classes,  $p_i$  is the prediction probability value,  $a_i$  is the softmax input for Class  $i$  and  $i, j \in \{1, 2, \dots, n\}$ .

$$p_i = \frac{\exp(a_i)}{\sum_{j=1}^n \exp(a_j)} \quad (3)$$

The final output is the classification of the input image into one of the three classes: Sukkari, Ajwa, and Suggai. The model architecture is outlined in [Table 4](#). We used Python 3.6 using Keras library to build and train the CNN model.

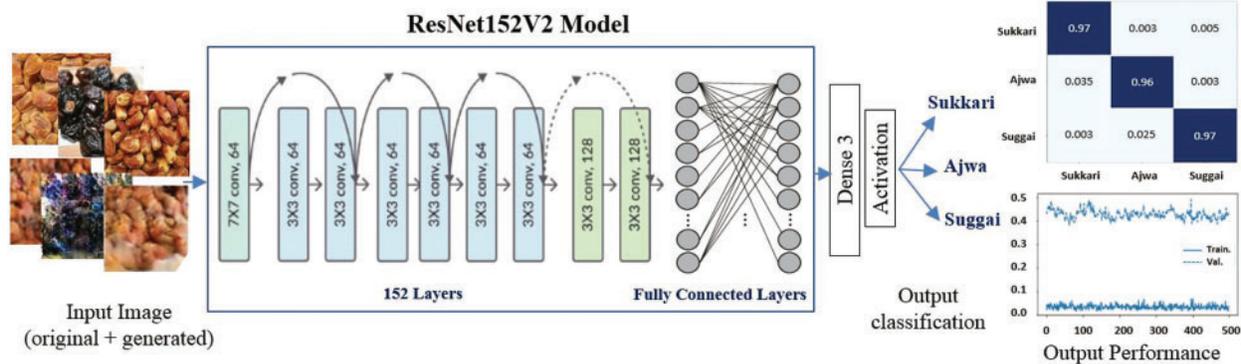
**Table 4:** The summary of the proposed CNN model architecture

Layer (type)	Output shape	Parameters
conv2d_6 (Conv2D)	(None, 98, 98, 200)	5600
conv2d_7 (Conv2D)	(None, 96, 96, 150)	270150
max_pooling2d_3 (MaxPooling2)	(None, 24, 24, 150)	0
conv2d_8 (Conv2D)	(None, 22, 22, 120)	162120
conv2d_9 (Conv2D)	(None, 20, 20, 80)	86480
conv2d_10 (Conv2D)	(None, 18, 18, 50)	36050
max_pooling2d_4 (MaxPooling2)	(None, 4, 4, 50)	0
flatten_2 (Flatten)	(None, 800)	0
dense_5 (Dense)	(None, 120)	96120
dense_6 (Dense)	(None, 100)	12100
dense_7 (Dense)	(None, 50)	5050
dropout_2 (Dropout)	(None, 50)	0
dense_8 (Dense)	(None, 3)	153
Total params: 673,823		
Trainable params: 673,823		
Non-trainable params: 0		

### 3.4.2 ResNet152V2 Model Architecture

Residual networks (Resnet) [28] are a class of deep neural networks that have similar structures but differ in depth. To combat the degradation of deep neural networks, Resnet introduces a structure known as the residual learning unit. The structure of this unit is a feedforward network with a shortcut connection that adds new inputs and generates new outputs. The main advantage of this unit is that it improves classification accuracy without adding to the model's complexity. We select ResNet152V2 as it achieves the best accuracy among Resnet family members [29,30].

In the second model, instead of the CNN model, ResNet152V2 is utilized as a feature extraction model, as illustrated in [Fig. 8](#). Because it is a pre-trained model, the model includes initial weights, which can help it achieve acceptable accuracy faster than a regular CNN. The ResNet152V2 model is followed by a reshape layer, a flatten layer, a dense layer with 128 neurons, a dropout layer, and finally a dense layer with SoftMax activation function to categorize the image into its appropriate class in the model architecture. [Table 5](#) shows the architecture in full. The ResNet152V2 has 71,177,348 total parameters, which are divided into two categories: the trainable parameters and the non-trainable parameters, which are 71,033,604 and 143,744, respectively.



**Figure 8:** The proposed ResNet152V2 model architecture

**Table 5:** The summary of the proposed ResNet152V2 model architecture

Layer (type)	Output shape	Parameters #
ResNet152v2	(None, 7, 7, 2048)	58331648
reshape (Reshape)	(None, 7, 7, 2048)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 128)	12845184
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	516
Total params: 71,177,348		
Trainable params: 71,033,604		
Non-trainable params: 143,744		

#### 4 Results and Discussions

As mentioned before, the main problem of our collected dataset is the limited number of date fruit images and imbalance through each data type in the dataset. Moreover, the date fruit classification becomes a challenge due to the similarities among date types [31]. In order to overcome these issues and based on the previous research, we found that DCGAN and CycleGAN are suitable solutions. It will augment the limited number of date fruit images, generate high-resolution images, and increase the accuracy as well. Besides, according to our knowledge, none of the previous research used DCGAN or CycleGAN in the augmentation of date fruit datasets.

The experiment started with date images augmentations. We use the DCGAN and CycleGAN augmentation method that can be effectively used in supervised and unsupervised learning [32,33]. First of all, to augment the date fruit images we trained the DCGAN in our date fruit dataset by using the Colab platform. We start with a few numbers of iterations increased lately to 2000 iterations. After each iteration, the model learns and generates high-resolution images. We apply DCGAN for each date type separately which are Sukkari, Suggai, and Ajwa and add the generated images to the dataset. Additionally, after we apply the DCGAN augmentation method on the same dataset by 2000 iterations we see the results have been improved.

With CycleGAN, we repeat the same steps. Fig. 9 shows a combination of images created by the CycleGAN augmentation approach for Sukkari, Ajwa, and Suggai date types. The loss graph depicting the loss between the generator and discriminator models in the CycleGAN architecture

indicates the improvement of the generated images. The loss reduces as the number of iterations increases.

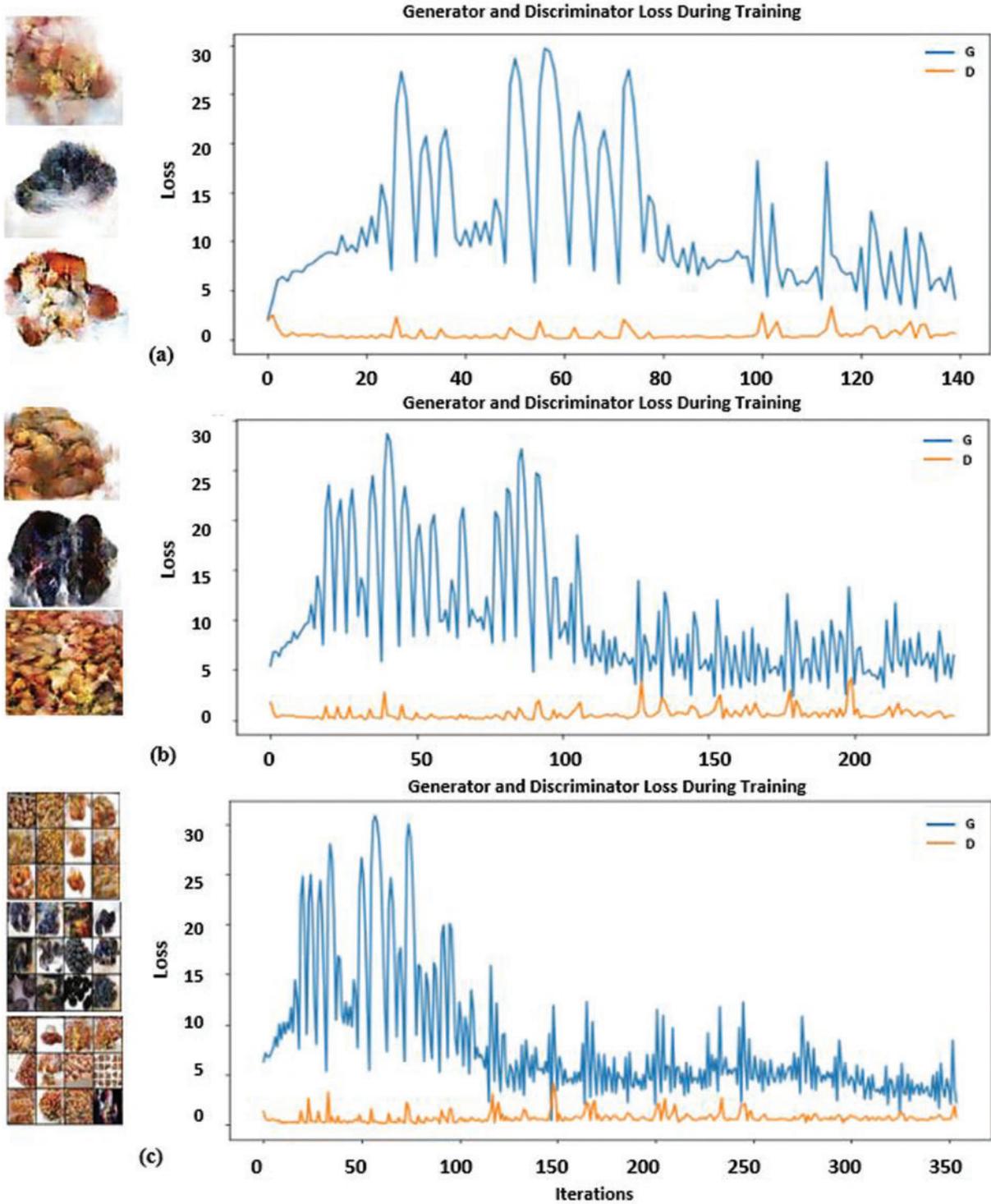


Figure 9: CycleGAN-generated images and the loss values with different numbers of iterations

Second, we use deep learning CNN and ResNet152V2 models to train our generated datasets to classify the date fruit dataset in three cases: the first case uses only the original dataset, the second case uses the DCGAN-generated dataset in addition to the original ones, and the third case uses both the original and DCGAN-generated datasets. Finally, we combine the original dataset with the CycleGAN-generated images.

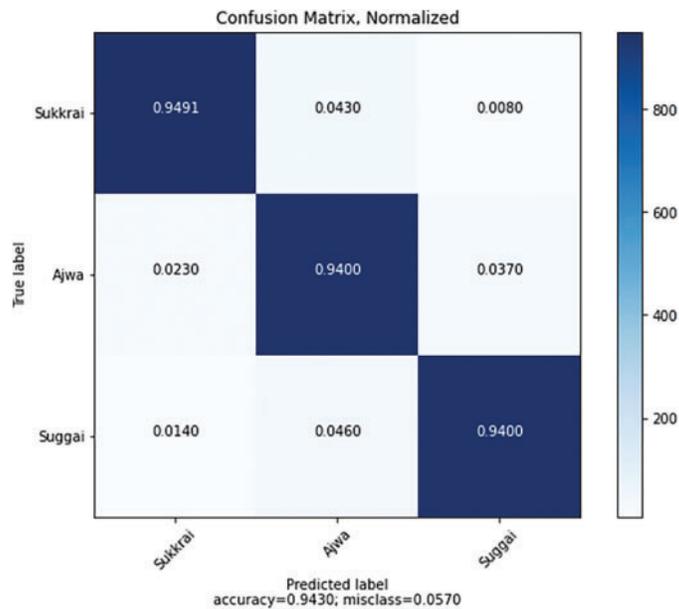
Python 3 and the Keras framework were used to create the models. These were created on a Google Colab pro edition [34] with 2 TB of storage, 25 GB of RAM, and a P100 GPU processor. The photos of the input classes were enhanced using an API Keras Augmentor [35] in the initial use of the dataset to increase the number of images in each class in order to achieve the statistical findings. The findings were generated by applying the performance metric equations to the validation data outputs, with the registered results representing the greatest validation values attained. The (Adamax) [36] optimizer was employed for our proposed models.

The performance of the classification models was evaluated based accuracy and the confusion matrix for each model. Accuracy, given in Eq. (4), is the number of examples correctly predicted from the total number of examples.

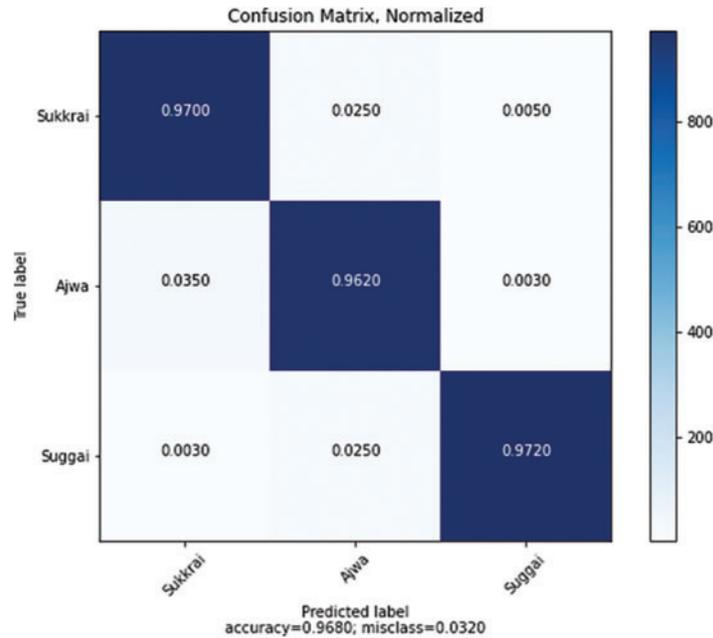
$$Accuracy(ACC) = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \quad (4)$$

where Tp and Tn are the true positive and negative parameters, respectively. Fp and Fn are the false positive and false negative values.

Correspondingly, the confusion matrix of the CNN model and ResNet152V2 model are presented in Figs. 10 and 11, respectively. The figures demonstrate the successful classification of the three date fruits categories (Sukkari, Ajwa, and Suggai). The two proposed models were compared according to accuracy. As listed in Table 6, the obtained results show that the CycleGAN-generated dataset gave the best classification performance with 96.8% accuracy when using ResNet152V2 model, followed by CNN model, with 94.3% accuracy.



**Figure 10:** Confusion matrix for the proposed CNN model using CycleGAN image generated

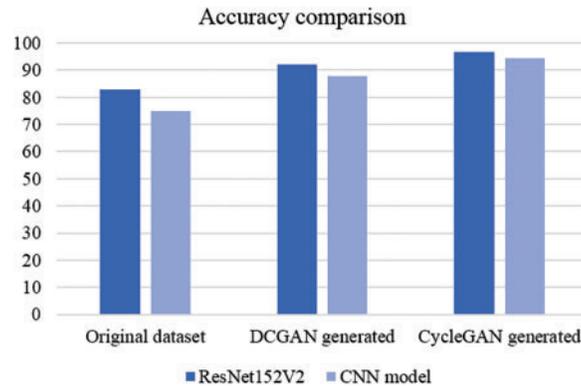


**Figure 11:** Confusion matrix for the proposed ResNet152V2 model using CycleGAN image generated

**Table 6:** The evaluation metrics for different models with different datasets

Model	Dataset	Accuracy
ResNet152V2	Original dataset	83%
	DCGAN generated dataset	92%
	CycleGAN generated dataset	96.8%
CNN model	Original dataset	75%
	DCGAN generated dataset	88%
	CycleGAN generated dataset	94.3%

Contrarily, applying the original dataset are the lowest compared to the other architectures, since the ResNet152V2 and CNN models have obtained 83% and 75% accuracy, respectively. From the table, it seems that highest results were achieved when we augment our date fruit dataset using CycleGAN architecture. Fig. 12 illustrates the performance metric between ResNet152V2 and CNN models in case of using original dataset, DCGAN generated dataset, and CycleGAN generated dataset. The figure demonstrates that the CycleGAN-generated dataset outperform the DCGAN-generated dataset. Moreover, the ResNet152V2 model architecture is more effective than the CNN model. Our source code with the dataset of Date fruit types were uploaded to the Github website and are available on [https://github.com/Dr-Dina-M-Ibrahim/Date\\_Fruit\\_Classification](https://github.com/Dr-Dina-M-Ibrahim/Date_Fruit_Classification).



**Figure 12:** Performance metric between ResNet152V2 and CNN models in case of using original dataset, DCGAN generated dataset, and CycleGAN generated dataset

## 5 Conclusion and Future Work

Due to a limitation of publicly available datasets, date fruit classification based on type had significant limitations. Furthermore, deep learning classification models require a large amount of training data in order to learn all of its parameters. As a result, Data Augmentation was employed to overcome the limitations and minimize overfitting during training. As a result, we used DCGAN and CycleGAN to maximize the date dataset, which includes three types: Sukkari, Ajwa, and Suggai.

For detecting Sukkari, Ajwa, and Suggai date types from an augmented dataset, two classification deep learning models, ResNet152V2 and CNN, were built and assessed. DCGAN and CycleGAN architectures provide the augmentation. To our knowledge, this is the first time a date fruit dataset has been enhanced with DCGAN or CycleGAN. The collected results demonstrate that when employing the ResNet152V2 model, the CycleGAN-generated dataset had the best classification performance of 96.8% accuracy, followed by the CNN model with 94.3% accuracy. Using the original dataset, on the other hand, is the least accurate of the designs, with the ResNet152V2 and CNN models achieving 83% and 75% accuracy, respectively.

Ongoing work aims to improve the suggested model's performance by increasing the number of images in the datasets utilized, increasing the training epochs, and using other GAN architectures in both classification and augmentation.

**Acknowledgement:** The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Aiadi, O., Kherfi, M. L. (2017). A new method for automatic date fruit classification. *International Journal of Computational Vision and Robotics*, 7(6), 692–711. DOI 10.1504/IJCVR.2017.087751.

2. Faisal, M., Alsulaiman, M., Arafah, M., Mekhtiche, M. A. (2020). IHDS: Intelligent harvesting decision system for date fruit based on maturity stage using deep learning and computer vision. *IEEE Access*, 8, 167985–167997. DOI 10.1109/Access.6287639.
3. Nasiri, A., Taheri-Garavand, A., Zhang, Y. D. (2019). Image-based deep learning automated sorting of date fruit. *Postharvest Biology and Technology*, 153, 133–141. DOI 10.1016/j.postharvbio.2019.04.003.
4. Aiadi, O., Khaldi, B., Kherfi, M. (2016). An automated system for date fruit recognition through images = un système automatique pour la reconnaissance des dattes a partir des images. *Revue des Bioressources*, 257(5757), 1–8. DOI 10.12816/0045893.
5. Altaheri, H., Alsulaiman, M., Muhammad, G., Amin, S. U., Bencherif, M. et al. (2019). Date fruit dataset for intelligent harvesting. *Data in Brief*, 26, 104514. DOI 10.1016/j.dib.2019.104514.
6. Pandian, J. A., Geetharamani, G., Annette, B. (2019). Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques. *IEEE 9th International Conference on Advanced Computing*, Tamilnadu, India. IEEE.
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D. et al. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 139–144. DOI 10.1145/3422622.
8. Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434. DOI 10.1007/978-3-319-71589-6\_9.
9. Wen, F., Jiang, J., Fan, J. A. (2020). Robust freeform metasurface design based on progressively growing generative networks. *ACS Photonics*, 7(8), 2098–2104. DOI 10.1021/acsp Photonics.0c00539.
10. Denton, E., Chintala, S., Szlam, A., Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. arXiv:1506.05751v1.
11. Almahairi, A., Rajeshwar, S., Sordoni, A., Bachman, P., Courville, A. (2018). Augmented cyclegan: learning many-to-many mappings from unpaired data. *International Conference on Machine Learning*, Stockholm, Sweden, PMLR.
12. DeVries, T., Romero, A., Pineda, L., Taylor, G. W., Drozdal, M. (2019). On the evaluation of conditional gans. arXiv:1907.08175v2.
13. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. et al. (2020). Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA.
14. Gandhi, R., Nimbalkar, S., Yelamanchili, N., Ponkshe, S. (2018). Plant disease detection using cnns and gans as an augmentative approach. *IEEE International Conference on Innovative Research and Development*, Bangkok, Thailand, IEEE.
15. Zhang, M., Liu, S., Yang, F., Liu, J. (2019). Classification of canker on small datasets using improved deep convolutional generative adversarial networks. *IEEE Access*, 7, 49680–49690. DOI 10.1109/Access.6287639.
16. Suryawati, E., Zilvan, V., Yuwana, R. S., Heryana, A., Rohdiana, D. et al. (2019). Deep convolutional adversarial network-based feature learning for tea clones identifications. *3rd International Conference on Informatics and Computational Sciences*, Semarang, Indonesia, IEEE.
17. Altaheri, H., Alsulaiman, M., Muhammad, G. (2019). Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access*, 7, 117115–117133. DOI 10.1109/Access.6287639.
18. Magsi, A., Mahar, J. A., Danwar, S. H. (2019). Date fruit recognition using feature extraction techniques and deep convolutional neural network. *Indian Journal of Science and Technology*, 12(32), 1–12. DOI 10.17485/ijst/2019/v12i32/146441.
19. Wan, S., Goudos, S. (2020). Faster r-cnn for multi-class fruit detection using a robotic vision system. *Computer Networks*, 168, 107036. DOI 10.1016/j.comnet.2019.107036.
20. Gao, Z., Li, Y., Wan, S. (2020). Exploring deep learning for view-based 3D model retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(1), 1–21. DOI 10.1145/3377876.
21. Ismail, N., Malik, O. A. (2021). Real-time visual inspection system for grading fruits using computer vision and deep learning techniques. *Information Processing in Agriculture*, 2, 1–14. DOI 10.1016/j.inpa.2021.01.005.

22. Muhammad, G. (2015). Date fruits classification using texture descriptors and shape-size features. *Engineering Applications of Artificial Intelligence*, 37, 361–367. DOI 10.1016/j.engappai.2014.10.001.
23. Abi Sen, A. A., Bahbouh, N. M., Alkhodre, A. B., Aldhawi, A. M., Aldham, F. A. et al. (2020). A classification algorithm for date fruits. *7th International Conference on Computing for Sustainable Global Development*, New Delhi, India, IEEE.
24. Hossain, M. S., Muhammad, G., Amin, S. U. (2018). Improving consumer satisfaction in smart cities using edge computing and caching: A case study of date fruits classification. *Future Generation Computer Systems*, 88, 333–341. DOI 10.1016/j.future.2018.05.050.
25. Yamashita, R., Nishio, M., Do, R. K. G., Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*, 9(4), 611–629. DOI 10.1007/s13244-018-0639-9.
26. Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C. et al. (2020). A review of convolutional neural network applied to fruit image processing. *Applied Sciences*, 10(10), 3443. DOI 10.3390/app10103443.
27. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W. et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.
28. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA.
29. Sustika, R., Subekti, A., Pardede, H. F., Suryawati, E., Mahendra, O. et al. (2018). Evaluation of deep convolutional neural network architectures for strawberry quality inspection. *International Journal of Engineering & Technology*, 7(4), 75–80. DOI 10.14419/ijet.v7i4.40.24080.
30. Siddiqi, R. (2020). Comparative performance of various deep learning based models in fruit image classification. *Proceedings of the 11th International Conference on Advances in Information Technology*, Bangkok Thailand.
31. Zhang, J., Chen, L., Zhuo, L., Liang, X., Li, J. (2018). An efficient hyperspectral image retrieval method: Deep spectral-spatial feature extraction with dcgan and dimensionality reduction using t-SNE-based NM hashing. *Remote Sensing*, 10(2), 271. DOI 10.3390/rs10020271.
32. Ketkar, N., Santana, E. (2017). *Deep learning with python*, vol. 1. Berkeley, CA: Apress.
33. Qin, Z., Zhang, Z., Chen, X., Wang, C., Peng, Y. (2018). Fd-mobilenet: Improved mobilenet with a fast downsampling strategy. *25th IEEE International Conference on Image Processing*, Athens, Greece, IEEE.
34. Bisong, E. (2019). *Building machine learning and deep learning models on google cloud platform*. New York City: Apress.
35. Gulli, A., Pal, S. (2017). *Deep learning with Keras*. Birmingham, UK: Packt Publishing, Ltd.
36. Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980v9.