Classification of Palm Trees Diseases using Convolution Neural Network

Marwan Abu-zanona¹, Said Elaiwat², Shayma’a Younis³, Nisreen Innab⁴, M. M. Kamruzzaman⁵

Department of Computer Sciences, College of Shari’a and Islamic Studies in Al Ahsaa, Al Imam Mohammad Ibn Saud Islamic University, Al Ahsaa, Saudi Arabia¹
Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakakah, Saudi Arabia², ³
Department Biomedical Informatics engineering, Yarmouk University, Irbid, Jordan⁴
Department of Computer Science and Information Systems, College of Applied Sciences, AlMaarefa University, Riyadh, Saudi Arabia⁵

Abstract—The palm tree is considered one of the most durable trees, and it occupies an advanced position as one of the most famous and most important trees that are planted in different regions around the world, which enter into many uses and have a number of benefits. In the recent years, date palms have been exposed to a large number of diseases. These diseases differ in their symptoms and causes, and sometimes overlap, making the diagnosing process with the naked eye difficult, even by an expert in this field. This paper proposes a CNN-model to detect and classify four common diseases threatening palms today, Bacterial leaf blight, Brown spots, Leaf smut, white scale in addition to healthy leaves. The proposed CNN structure includes four convolutional layers for feature extraction followed by a fully connected layer for classification. For performance evaluation, we investigate the performance of the proposed model and compare to other CNN-structures, VGG-16 and MobileNet, using four evaluation metrics: Accuracy, Precision, Recall and F1 Score. Our proposed model achieves 99.10% accuracy rate while VGG-16 and MobileNet achieve 99.35% and 99.56% accuracy rates, respectively. In general, the performance of our model and other models are very close with a minor advantage to MobileNet over others. In contrast, our model characterized by simplicity and shows low computational training time comparing to others.

Keywords—Palm trees diseases; convolutional neural networks; mobileNet; VGG-16

I. INTRODUCTION

The date palm is considered one of the most important fruit trees in the Arab and Islamic world, as Arab countries account for 71% of its trees in the world, and 81% of the total global production, while this percentage rises to 99% (103.95 million trees) of the number of trees. The world’s date palm, amounting to 105 million, when combining the Arab and Islamic worlds, according to the Food and Agriculture Organization of the United Nations. According to the latest statistics adopted in the world, including the statistic conducted by the Egyptian Embassy in Brazil in 1990 as shown in table I, it was found that the palm tree is one of the main agricultural products produced by the Arab countries and is considered a major element in supporting the macro economy in these countries, so it is very important to pay attention to the quality and quantity of production Palm trees, but unfortunately, the quality, quality and quantity of palm trees are greatly threatened with confinement due to the common palm diseases these days, where in general palm trees are threatened by 4 main types of diseases, which are, Bacterial leaf blight, Brown spots, Leaf smut, white scale, the nature and symptoms of these diseases are different in their form, in the area of their appearance and distribution on palm trees, so it is very important to reveal modern techniques that contribute greatly to discovering them before they cause tremendous pressure on the quality and quantity of palm trees produced.

Symptoms of leaf smut Small irregular brown to black spots occurred on the upper and bottom surfaces of rachis and fronds, ranging in size from 3 to 7 mm [1]. Bacterial Leaf blight symptoms were elongated brown to black patches that grew in size and spread across a considerable region, creating cankers on the midrib [1].

Brown spot disease is characterized by the appearance of non-specific dark spots, and as the infection progresses, the center of the spot turns to a pale color, but the edges remain brown to gray. The spots appear on the leaves, thorns, and the middle vein (the leaves). The size of the spots ranges from one to several centimeters, but their size and color may vary according to the fungus that causes them. Another serious risk is a lethal pest called white scale. White palm Scale is a species of armoured scale insect. This means that they produce a hard outer coating that covers the body, which protects them from pathogens. They’re also well protected from topical pesticides. White Scale insects attack palm by sucking the sap through the fine, thin feeding-tubes. Infestations rarely kill plants but can impact vigour [2]. The methods usually used by farmers depend mainly on observing the affected foliage with the naked eye by experts. Unfortunately, this method is not effective because of the distance of the palm tree from the ground and at the same time due to the somewhat similarity between the symptoms of the four palm diseases mentioned previously. In addition to that, the manual examination of palm leaves is time-consuming, especially in the case of large farms. In this paper, a CNN-based model is proposed to detect four of the most frequent palm illnesses is suggested. These illnesses are, Bacterial leaf blight, brown spots, Leaf smut, white scale. The proposed model model characterized by simplicitys
The proposed work by [5] investigated the ability to identify chimera and anthracnose infections in palm trees. The detection procedure began with the use of a digital camera to capture photographs of the damaged palm trees, which were then subjected to a series of image processing processes following segmentation using the k-mean method. The Gray Correlation Matrix was created by applying the characteristics extraction phase on the segmented pictures using gyro compatibles (GLCM). The accuracy of the findings proved the effectiveness of the support vector machine (SVM) in such circumstances, with Chimera having a 97 percent accuracy and anthracnose having a 95 percent accuracy.

In [6], The authors used a program called Therma CAM Researcher to analyze pictures collected with an uncooled infrared thermal camera linked to a microbolometer sensor in this work. To obtain credible maps of leaf temperature, this software was enhanced with local climatic variables and leaf emission. Each experiment had 4-5 duplicates of control trees and 8-10 replicates of infected trees in each group. The Crop Water Stress Index (CWSI) was produced, and the damaged trees were identified with up to 75% accuracy using this value. In[7], The authors of this article wanted to see if they could tell the difference between healthy palm trees and palm trees afflicted with the red palm weevil by using water stress and temperature rates. Aerial photos were taken using an uncooled infrared thermal camera and a microbolometer connected to begin the procedure. The palm canopy was separated from the soil using a set of image processing techniques, and the watershed method was used to build the palm canopy diagram using the photos. This study yielded no numerical data, however it did show that various palm plants had variable water stress levels.

### III. System Architecture

#### A. Data Preprocessing

Since the resolution and size of acquired 2D images are usually inconsistent, using them directly as training data is not applicable. In addition, using very high resolution images is time-consuming and may lead to high computational cost. Therefore, all images are resized to 60 x 60 resolution. Furthermore, The correlation between data within the image can slow down the learning process. To address this, we apply image whitening [8] based on Zero Component Analysis (ZCA) to reduce the correlation between data and make key features such as edges and curvatures more prominent, and thus easy to detect by CNN. To achieve that the pixel values in each image is normalized to be between 0 and 1 as the following:

$$\hat{X} = X/255$$  

(1)

where $X$ and $\hat{X}$ are the image before and after normalization, receptively. After that the image is mean normalized by defining the mean value along each feature dimension (pixel position) of training images and then subtracted from the image as the following:

$$\bar{X} = X - \mu$$  

(2)

where $\mu$ represents the mean vector across all the features of $X$ while $\bar{X}$ represents the mean normalized image. Finally, the whitened image $X_{ZCA}$ is defined based on Singular Value

### II. Related Work

This section shows the previous studies that dealt with the problem of palm tree diseases. In [1], The researchers used various classifiers to detect the three most frequent palm illnesses today: leaf spot, blight, and red palm weevil. CNN was used to distinguish between Leaf Spots and Blight Spots illnesses, and SVM was used to distinguish between the Red Palm Weevil pest and the Leaf Spots sickness. The Kaggle dataset (about 90k photos) was utilized as-is, resulting in a 65 percent accuracy for the first CNN model. The data set was split into two parts: 70% for training and 30% for testing. The accuracy ratio success rates for the CNN and SVM algorithms were 97.9% and 92.8%, respectively.

In [3], the illnesses in tomato fruit were detected using multiple classifiers, the first of which was CNN and LVQ. Leaf blight, bacterial spot, late blight, sitoria leaf spot, and yellow curved leaf disease are among the illnesses he classified. The dataset utilized in this experiment was divided into 400 photos to train the model and 100 images to assess the quality and efficiency of the model, which had a size of 512 x 512 and an accuracy of 86%. The only difficulty this study faces is the degree of symptom similarity among the illnesses that affect tomato fruit.

Jiang et al. [4] proposed a system to identify apple leaf illnesses using deep learning algorithms. To identify and categorize alternate leaf spot, brown spot, mosaic, spot Gray, rust, and eventually brown apple leaf disease, the proposed system employed a new CNN based technique called INAR-SSD based on a framework called Caffe on the GPU platform. Set of data The data utilized is made up of 26377 photos that were separated into two groups: one for training the model and the other for verifying the model. The model speed was 23.13 frames per second, and the accuracy of the findings was 78.80 percent mAP.

### Table 1. Statistics Done by the Embassy of Egypt Economic And Commercial Office in Brazil: March 18, 2019 [1]

<table>
<thead>
<tr>
<th>Rank</th>
<th>Country</th>
<th>Production (1000 Metric Tons)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>Egypt</td>
<td>1,927.35</td>
</tr>
<tr>
<td>2nd</td>
<td>Saudi Arabia</td>
<td>1,122.82</td>
</tr>
<tr>
<td>3rd</td>
<td>Iran</td>
<td>1,016.61</td>
</tr>
<tr>
<td>4th</td>
<td>United Arab Emirates</td>
<td>900.00</td>
</tr>
<tr>
<td>5th</td>
<td>Algeria</td>
<td>690.00</td>
</tr>
<tr>
<td>6th</td>
<td>Iraq</td>
<td>619.18</td>
</tr>
<tr>
<td>7th</td>
<td>Pakistan</td>
<td>557.28</td>
</tr>
<tr>
<td>8th</td>
<td>Oman</td>
<td>268.01</td>
</tr>
<tr>
<td>9th</td>
<td>Tunisia</td>
<td>180.00</td>
</tr>
<tr>
<td>10th</td>
<td>Libya</td>
<td>105.95</td>
</tr>
</tbody>
</table>
Decomposition (SVD) of the covariance matrix of \( \bar{X} \) as the following

\[ X_{ZCA} = U \cdot \text{diag}(1/\sqrt{\text{diag}(S) + \varepsilon}) \cdot U^T \cdot \bar{X} \quad (3) \]

The function \( \text{diag}(\cdot) \) returns the diagonal matrix of the input matrix. The variable \( U \) and \( S \) represent the Eigen vector and Eigen value of the SVD of covariance matrix while the hyper-parameter \( \varepsilon \) represents whitening coefficient.

### B. Convolutional Neural Networks

The Convolutional Neural Network (CNN) [9], [10] is a deep learning model capable of achieving both feature coding (feature extraction) and classification in a single coherent architecture. In contrast, traditional neural networks are limited for classification task and require well-defined feature (engineered features) to achieve high performance [11], [12]. In general, CNNs are widely applied in many computer vision applications such as face recognition [13], object detection [14], Natural language processing [15], medical image analysis and others [16]. Fig. 1 shows the basic architecture of the CNN which involves three different set of layers, namely, ‘Input Layer’, ‘Feature Learning Layers’ and ‘Classification Layers’. The first set is a single layer, includes the input image (usually colored image). The second set is a sequence of successive Convolution and Pooling layers which results in extracted features. The last set usually involves fully connected layers to achieve classification tasks. The main advantage of CNN is the ability to define local features such as horizontal edges through convolution layers by maintaining the same structure of the input image whilst using a small number of parameters. For example, regardless of the size of the input image, the number of parameters depends on the size and the number of filters (e.g., 10 filters of [3 x 3] size).

In the recent years, numerous attempts have been made to enhance CNN’s initial design (by LeCun et al. [9]) in order to achieve better performance. A popular example of CNN architecture are AlexNet [17], VGG-16 [18], ResNet [19] and MobileNet [20]. In this work, a simple CNN architecture is firstly proposed for the problem of Palm Trees Diseases classification and then the performance of our CNN is compared to two well-known CNN structures: “VGG-16” and “MobileNet”.

#### 1) Proposed CNN

In this section, we introduce a simple CNN architecture for efficient learning especially with limited training data size. The proposed architecture includes four convolutional layers followed by fully connected layer and softmax as output layer as shown in Fig. 2.

The input layer includes the input image with dimensions [60, 60, 3] where the first two represent the width and the height of the image while the last value represents the number of channels (three channels for red, green and blue). The first convolutional layer includes 32 filters of each [3 x 3] size, followed by a maxpool of 3x3. In the next three convolutional layers, 64 filters [3 x 3] are used for the second layer, while 28 and 256 filters are used for the third and fourth layers, respectively. Each of these convolutional layers is followed by a maxpool of 2x2. Finally, a flatten process is applied, resulting in a layer with 25600 features, fully connected to a dense layer with activation function “softmax”.

#### 2) VGG-16

VGG-16 was originally proposed by Simonyan and Zisserman [18] in 2014. The basic idea behind VGG-16 is to increase the depth of the network by adding more convolutional layers while using very small [3 x 3] convolution filters in all layers. Fig. 3 shows the architecture of VGG-16.

During the training phase, the only preprocessing step is to subtract the mean RGB value (calculated over the training set) from each pixel. After that, the image is passed through a set of convolutional layers with small filters ([3 x 3] size). To preserve the spatial resolution the same after the convolution, the stride and the spatial padding are both set to one pixel. Some of the convolutional layers are followed by a maxpooling layer, so a total of five max-pooling layers are applied in the whole network.

#### 3) MobileNets

MobileNet is deep learning model, proposed by Howard et al. [20] of Google Research team. This
model was presented to effectively maximize the performance of CNN under limited resources which is ideal for mobile and embedded vision applications. MobileNets is characterized by light weight and hyper-parameters (small number of required weights comparing to other CNN models), which allows to trade-off between latency and accuracy. To achieve that, the standard convolution filters [Fig. 4(a)] replaced by two layers: depthwise convolution [Fig. 4(b)] and pointwise convolution [Fig. 4(a)].

The computational cost $C$ of the standard convolutional layer can be represented as:

$$C = F \cdot F \cdot M \cdot N \cdot H \cdot W$$

(4)

In contrast, MobileNets is capable to reduce the computational cost as:

$$C = F \cdot F \cdot M \cdot H \cdot W + M \cdot N \cdot H \cdot W$$

(5)

which includes the cost of the depthwise convolution plus the cost of the pointwise convolution. By comparing the ratio between Eq. 4 and 5 based on [3 x 3] depthwise separable convolutions, MobileNets can achieve 8 to 9 times less computational cost than standard convolutions [20].

IV. EXPERIMENTAL RESULTS

In this section, the performance of the proposed CNN model, in addition to VGG 16 and MobileNet, are evaluated by combining two datasets, ‘Date Palm’ [22] and ‘Leaf_disease_3’ [23].

A. Dataset

Table II gives a summary of each dataset. In our experiments, Date Palm [22] and Leaf_disease_3 [23] datasets are combined to build a new dataset (DP-disease dataset). In more details, Date Palm dataset includes two types of diseases: Brown Spots with 470 samples and White Scale with 958 samples. In addition to that, the dataset also includes 1013 samples of Healthy leaves. In contrast, Leaf_disease_3 dataset includes three different types of diseases: Bacterial Leaf Blight, Brown Spot and Leaf Smut. The dataset is partitioned into training and testing sets where 82,200 samples are used for training and 27,360 samples are used for testing. The number of samples for each disease is 27,360 in the training set and 9120 in the testing set. By combining samples from both datasets, it allows use to include more diseases (four different disease) in addition to the healthy case. As a result, the classification task in all experiments embeds five classes: 1) Bacterial Leaf Blight, 2) Leaf Smut, 3) Brown Spots, 4) White Scale and 5) Healthy leaves.

Fig. 5(a) shows samples from Bacterial Leaf Blight disease where 1,013 images with dimension 3081 x 897 are included in our experiments. Fig. 5(b) shows samples from Leaf Smut disease where 827 images with dimension 510 x 383 are included in our experiments. Samples from Brown Spots disease and White Scale disease are shown in Fig. 6(a) and (b) where 470 images and 958 images are used in our experiments for Brown spots disease and White scale disease, respectively. For the case of healthy leaves, Fig. 6(c) shows sample of Healthy leaves where 1,203 images are included in our experiments.

B. Experiment Configurations

In our experiments, 4023 samples from different type of diseases were used to train each CNN model while 448 samples from different type of diseases were used to test the performance of each CNN model. The size of input images are unified in all experiments to be 60 x 60 x 3, where the first two dimensions represent the width and height while the last
one represent the number of color channels (R G B). During the training phase, Batch Gradient Descent were applied in all experiments. In addition to that, the number of epochs sets to 10 which compromises between sufficient training process and over-fitting. More details about the hyper-parameters of each CNN model are shown in Table III, IV and V.

C. Performance Evaluation

The performance evaluation of each model was analyzed using four known evaluation metrics: Accuracy, Precision, Recall and F1 Score. The Accuracy of each CNN model at each type of disease is defined as the following:

\[
\text{Accuracy} = \frac{TP + TN}{T}
\]

, where \(TP\) and \(TN\) represent number of True Positive and True Negative classifications, respectively, while \(T\) represents the total number of samples. Table VI reports the performance of each model based on accuracy. We can notice that MobileNet achieved 99.56% accuracy rate in average which is the best performance. In contrast, our model and VGG16 achieved 99.10% and 99.35% accuracy rate, respectively. It is clear that the performance of the three models are relatively very close with slight variations less than 0.5% .

The Precision of each CNN model at each type of disease is defined as the following:

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

Recall and F1 Score. The Accuracy of each CNN model at each type of disease is defined as the following:

\[
\text{Accuracy} = \frac{TP + TN}{T}
\]
Table VI. The Performance of each Model based on Accuracy

<table>
<thead>
<tr>
<th>Diseases</th>
<th>Our model</th>
<th>MobileNet model</th>
<th>Vgg16 model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown Spots</td>
<td>99.35%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>White Scale</td>
<td>99.35%</td>
<td>99.75%</td>
<td>99.35%</td>
</tr>
<tr>
<td>Bacterial Leaf Blight</td>
<td>98.75%</td>
<td>98.98%</td>
<td>98.99%</td>
</tr>
<tr>
<td>Leaf Smut</td>
<td>99.20%</td>
<td>98.88%</td>
<td>99.5%</td>
</tr>
<tr>
<td>Healthy</td>
<td>98.90%</td>
<td>99.90%</td>
<td>98.90%</td>
</tr>
<tr>
<td>Average</td>
<td>99.10%</td>
<td>99.56%</td>
<td>99.35%</td>
</tr>
</tbody>
</table>

Table VII. The Performance of each Model based on Recall

<table>
<thead>
<tr>
<th>Diseases</th>
<th>Our model</th>
<th>MobileNet model</th>
<th>Vgg16 model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown Spots</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>White Scale</td>
<td>98.55%</td>
<td>96.55%</td>
<td>99.75%</td>
</tr>
<tr>
<td>Bacterial Leaf Blight</td>
<td>96.27%</td>
<td>96.20%</td>
<td>97.27%</td>
</tr>
<tr>
<td>Leaf Smut</td>
<td>98.73%</td>
<td>98.73%</td>
<td>98.86%</td>
</tr>
<tr>
<td>Healthy</td>
<td>99.15%</td>
<td>99.67%</td>
<td>99.55%</td>
</tr>
<tr>
<td>Average</td>
<td>98.54%</td>
<td>98.63%</td>
<td>99.08%</td>
</tr>
</tbody>
</table>

Table VIII. The Performance of each Model based on Precision

<table>
<thead>
<tr>
<th>Diseases</th>
<th>Our model</th>
<th>MobileNet model</th>
<th>Vgg16 model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown Spots</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>White Scale</td>
<td>98.55%</td>
<td>99.75%</td>
<td>98.75%</td>
</tr>
<tr>
<td>Bacterial Leaf Blight</td>
<td>99.55%</td>
<td>99.55%</td>
<td>98.91%</td>
</tr>
<tr>
<td>Leaf Smut</td>
<td>99.33%</td>
<td>99.33%</td>
<td>99.63%</td>
</tr>
<tr>
<td>Healthy</td>
<td>98.73%</td>
<td>97.73%</td>
<td>98.88%</td>
</tr>
<tr>
<td>Average</td>
<td>99.23%</td>
<td>99.21%</td>
<td>99.23%</td>
</tr>
</tbody>
</table>

Table IX. The Performance of each Model based on F1 Score

<table>
<thead>
<tr>
<th>Diseases</th>
<th>Our model</th>
<th>MobileNet model</th>
<th>Vgg16 model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown Spots</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>White Scale</td>
<td>98.55%</td>
<td>98.87%</td>
<td>98.98%</td>
</tr>
<tr>
<td>Bacterial Leaf Blight</td>
<td>99.27%</td>
<td>99.27%</td>
<td>99.65%</td>
</tr>
<tr>
<td>Leaf Smut</td>
<td>98.94%</td>
<td>98.94%</td>
<td>98.54%</td>
</tr>
<tr>
<td>Healthy</td>
<td>99.15%</td>
<td>99.67%</td>
<td>99.65%</td>
</tr>
<tr>
<td>Average</td>
<td>99.18%</td>
<td>99.35%</td>
<td>99.36%</td>
</tr>
</tbody>
</table>

The performance of each CNN model based Recall and F1 Score are defined as the following:

\[
\text{Recall} = \frac{TP}{TP+FN} \\
F1\text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where TP and FN represent number of True Positive and False Negative classifications, respectively. Table VII reports the performance of each model based on Recall. The highest average precision rate is 99.08% reported by MobileNet, followed by VGG16 (98.63%) and our model (98.54%). It is clear that the performance of the three models are relatively very close with slight variations less than 1%.

The performance of each model including accuracy, Precision, Recall and F1 Score are defined as the following:

\[
\text{Recall} = \frac{TP}{TP+FN} \\
F1\text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where TP and FN represent number of True Positive and False Negative classifications, respectively. Table VIII and IX report the performance of each model based on precision and F1 Score, respectively. The proposed model showed more capability to deal with limited resources such as training data and training time. Fig. 7 shows the average training time of each model. Note that our model achieved the lowest average training time of 20.048s/epoch while MobileNet achieved 343.85s/epoch average training time followed by VGG16 with a huge rise (2559s/epoch). Thus, the proposed model is capable compromise between the performance (high) and the complexity (simple structure with low training time).

V. Conclusion

This paper presented an efficient CNN-model for detecting and classifying palm tree frequent diseases, including Bacterial leaf blight, Brown spots, Leaf smut and white scale. The proposed model consists of three stages. Firstly, data preprocessing was applied on all images by normalizing them using Whitening Transform to reduce the correlation among data within the image. The second stage included four convolutional layers with max pooling to extract distinctive features. Finally, the classification stage was defined using fully connected layer with softmax to detect the type of palm disease. Furthermore, this work investigated the performance of two well-known CNN models, namely MobileNet and VGG-16, for classifying palm tree diseases. To evaluate the performance of proposed CNN-model and other models, a new dataset (DP-disease dataset) was built by combining data samples from Date Palm and Leaf_disease_3 datasets. This combination allowed us to detect four deferent diseases in addition to healthy cases. A set evaluation metrics was then applied to evaluate the performance of each model including accuracy, Precision, Recall and F1 score. Our experimental results showed that the performance of the three models were very close. However, our model characterized by simplicity and low computational training cost (20.048s/epoch ) comparing to others (343.85s/epoch and 2559s/epoch for MobileNet and VGG-16, respectively).

Future avenues of work include integrating the proposed model with object segmentation methods such R-CNN for accurately identifying the affected parts of palm trees.

References

“White Scale,” [https://candidegardening.com/GB/insects/ace6662f-61ca-4806-9987-81f4b040faed].


[22] “Date Palm Dataset,” [https://www.kaggle.com/hadjerhamaidi/date-palm-data].