Early Prediction of Plant Diseases using CNN and GANs

Ahmed Ali Gomaa
Management Information Systems
Madina Academy High Institute for Management and Technology, Giza, Egypt

Yasser M. Abd El-Latif
College of Computing and Information Technology
Arab Academy for Science, Technology and Maritime Transport
Faculty of Science, Ain Shams University
Cairo, Egypt

Abstract—Plant diseases enormously affect the agricultural crop production and quality with huge economic losses to the farmers and the country. This in turn increases the market price of crops and food, which increase the purchase burden of customers. Therefore, early identification and diagnosis of plant diseases at every stage of plant life cycle is a very critical approach to protect and increase the crop yield. In this paper using a deep-learning model, we present a classification system based on real-time images for early identification of plant infection prior of onset of severe disease symptoms at different life stages of a tomato plant infected with Tomato Mosaic Virus (TMV). The proposed classification was applied on each stage of the plant separately to obtain the largest data set and manifestation of each disease stage. The plant stages named in relation to disease stage as healthy (uninfected), early infection, and diseased (late infection). Classification was designed using the Convolutional Neural Network (CNN) model and the accuracy rate was 97%. Using Generative Adversarial Networks (GANs) to increase the number of real-time images and then apply CNN on these new images and the accuracy rate was 98%.

Keywords—Plants diseases; deep learning; early detection; convolutional neural network; generative adversarial networks

I. INTRODUCTION

Quality crop production is an essential feature of any country’s economic growth. The agricultural sector provides jobs for many people; in addition, it accounts for a large part of the Gross Domestic Product (GDP) in many countries around the world [1]. For example, it is clear that there is a rapid and wide agricultural development and land reclamation in Egypt, with increased application of technological advances. There has been remarkable increase made in agricultural sector, such as the 1.5 million Acres Project, the El-Alamein project, and the lake and Fayoum projects. The use of modern methods and technologies in agriculture, significantly increase crop production and yield, and increase protection of plants from insect pest infestation and disease infections at all stages of planting, harvesting, and post-harvesting till successful marketing [2]. In a fast-growing world population, although there are many improvements in large production and access to food, food security is threatened by a different set of factors such as a decreased fertility of soil and lands, decreased plant pollination efficiency, insect and other arthropod pests, and plant diseases. Plant diseases are classified as follows: bacterial diseases (bacterial speck, bacterial spot, bacterial canker) [3], fungal diseases (early blight, late blight, sectorial leaf spot, and Anthracnose fruit rot) [4], and viral diseases Tomato Mosaic Virus (TMV) [5]. The early detection and accurate diagnosis of plant diseases at every stage of the plant life cycle and the extent of infection until reaching the most infectious stage or appearance of severe disease symptoms, and easily classifying them is very important, as shown in Fig. 1.

Our Deep Learning (DL) model uses leaf images to detect diseases in plants by CNN to extract features from images, such as horizontal edges, vertical edges, Red Green Blue values, etc. A plant disease diagnosis system that uses machine learning techniques can correctly identify diseases plants healthy or unhealthy only [6]. Automatic detection of plant diseases at every age is an important research, analytical, and applied topic because it can help in monitoring large fields of crops in short time with high accuracy. Therefore, disease detection can discover symptoms visually and mechanically in the earliest time they appear on the leaves or other parts of the plant [6], as reported in numerous research publications and reports. CNN is believed to be the best DL neural network for extracting visual features [7]. The CNN-based network can be trained to discover diseases in plants by providing a large number of real-time images. In the case of lacking enough and good quality data or number of images, other techniques such as GANs can be used to generate the needed data for analysis and comparison with real-time data collected from the field. Both healthy and diseased plants and a future training model can be used to predict diseases in plants using plant leaf images [8].

II. RELATED WORK

Recent advances in agricultural technology have led to a demand for a new set of automated, non-destructive methods
for detecting plant diseases. Hence, several methods have
turned to computer visual and machine learning (ML)
techniques to create a rapid method for detecting plant
diseases when symptoms appear [9]. Classifying plant
diseases can be a very complex task because it depends mainly
on published and used classification systems and also by
experience of farmers and researchers.

Developing a reliable system that can be applied to many
plant classes is a difficult task. To date, most automatic plant
disease classification methods have depended on ML
algorithms and basic feature engineering. These methods
usually focus on specific environments and are suitable for a
smaller number of categories, as some small changes in the
system can lead to a severe drop in resolution. Recently,
CNNs have shown impressive results in many image
classification tasks that have allowed researchers to improve
the classification of agriculture and plant diseases [10]. CNN
is a technology that mixes artificial neural networks (ANNs)
and up to date DL strategies [11].

In deep learning, CNN is at the center of spectacular
advances. This ANN has been applied to several image
recognition tasks for decades and has attracted the eye of the
researchers of many countries in recent years; as CNN has
shown promising performances in several computer visual and
ML tasks [12]. This paper describes the underlying
architecture and different applications of the CNN.

In Y. Kawasaki, ET. al. [13], the authors introduce a novel
plant disease detection system based on CNN. Using only
training images, CNN can automatically acquire the requisite
features for classification and achieve high classification
performance. A total of 800 cucumber leaf images are used to
train CNN using the proposed techniques. Under the 4-fold
cross-validation strategy, the proposed CNN-based system
(which also extends the training dataset by generating
additional images) achieves an average accuracy of 94.9 % in
classifying cucumbers into two typical disease classes and a
non-diseased class. In this study, the authors proposed a novel
plant viral disease detection system using CNN and confirmed
its effectiveness. They also asserted that the strategy for
training CNN has significantly improved the accuracy of its
classification. This work will free system users from paying
extra attention to the details of plant shooting conditions.

In Y. Kawasaki, et. al. [13], future the system makes a
large contribution in the training field. Data augmentation
is an essential part of the training process applied to DL
models. The motivation is that a robust training process for
DL models depends on large annotated datasets, which are
expensive to be acquired, stored and processed. Therefore, a
reasonable alternative is to be able to automatically generate
new annotated training samples using a process known as data
augmentation [14]. A GAN model consists of two important
factors: the discriminator (D), and the generator (G). The
generator and discriminator have opposite objectives during
training. The discriminator is trained toward distinguishing
between synthesized and real-time data while the generator is
trained to fool the discriminator with synthesized data, as
shown in Fig. 2.

In D. Farm. [15], the authors propose a synthetic sampling
solution is presented at the data level to identify them from
small and unbalanced data sets using GANs. The reason for
using GANs is that the challenges in different fields as they deal
with small data sets and volatile amounts of samples per
category [16]. As a result, GANs offer an approach that can
improve learning regarding data distributions, reduce bias
resulting from class imbalance, and change classification.
Resolution limits towards more accurate results. The method
of [16] was trained on a small dataset of 2789 images of
highly perishable tomato plant diseases with a class imbalance
in 9 disease categories. Moreover, they evaluated their results
in terms of different measures and compared the quality of
these results for stratified excellence. GANs are an exciting
and quickly changing field, delivering on the deal of
generative models in their capacity to generate realistic
examples across a range of problem domains. In 2014,
conditional GANs was extended to a conditional model if both
the generator and discriminator are conditioned on some extra
data. They can perform the conditioning by data feeding into
both the discriminator and generator as additional input layer
[17].

In 2016, the Auxiliary Classifier GAN (AC-GAN) has
received much interest due to easy and extensibility to
different applications. AC-GAN integrates the conditional
information (label) by training the GAN discriminator with an
additional classification loss. AC-GAN is able to generate
high-quality images and has been extended to different
learning problems. However, the difference between the
generated samples by AC-GAN going to decrease as the
number of classes increases; hence limiting its power on large-
scale data [18].

In 2016, the Information Maximizing GAN (Info-GAN)
integrated the output of the generator to a component of its
input called the hidden codes. Uncovering some successful
and unsuccessful configurations for generating images using
Info-GAN [19] are shown in Fig. 3.
III. METHODOLOGY
This methodology works on three steps:

First step: divide the plants into three class’s generation (G1–G2–G3) with respect to the age stage.

Second step: each generation contains three Phases (Pi) of plants according to its pre-symptoms (uninfected P1 - early infection P2 – late infection P3).

Third step: Prediction and early detection of diseases to apply it to each generation, as shown in Fig. 4.

![Image of the Model of Plants Generation/Phases](image)

Description of the model of plant generation/phases:

- $p_{\text{data}}$ Represents all data set.
- Sample a noise set and a real-data set that includes classes (G1, G2, G3), each with size m.
- X represents the real sample belonging to the distribution $X \sim p_{\text{data}}$.
- Z denotes a random series belonging to the distribution $p_{z}$, which obeys a normal distribution. D and G represent the discriminator and generator respectively.
- The output of the generator is $X$ fake (Data).
- The generator to make synthetic samples G (z) extremely approach to the distribution $p_{\text{data}}$.
- To increase the data set apply the discriminator on this data and extract the global polling and hidden layer to get extra real data.
- Global polling layer is inserted in front of the discriminator network’s output layer to extract representative features with 512 dimensions.
- The discriminator input is ($X$ and G (z)) and they are compared until get the output discriminator.
- The final output of the discriminator is real/fake images.
- When data is fake, the discriminator and generator are trained alternatively. For the training process of the generator, the synthetic samples G (z) is taken into the discriminator and the produced loss value loss G is transmitted back to the generator one more time.
- When the data is real, it is entered into the data augment repository and then after that it is transferred to the CNN structure.
- The final model output is the data augmentation as a shape class to using in CNN mode.

The following results were collected from the use of these images on the CNN, as shown in Table I.

IV. EXPERIMENTS AND RESULTS
In this section, the study concludes the Experimental setup for our synthetic task generates data to detect diseases early:

1) Dataset: These images were collected from agricultural lands and it is a real data set that was used in this work to prove the growth stages of the plant and also increase the data from the original data and determine the stages of plant disease, there is a total of 5400 real images of diseased and healthy plants. These images covered all growth stages of plants and the extent of disease infection.

2) CNNs are proposed to reduce the number of parameters used and adapt the network architecture exactly to visual tasks. CNNs are usually composed of a set of layers that can be grouped by their functionalities; a CNN is typically composed of four types of layers: Convolution Layer, ReLu and sigmoid functions, Pooling, and Fully Connected Layer [20, 21, 22].

<table>
<thead>
<tr>
<th>Generation</th>
<th>Phases</th>
<th>Number of images</th>
<th>Loss</th>
<th>Vall_Loss</th>
<th>Accuracy</th>
<th>Vall_Accu</th>
<th>Max_Accu</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>P1</td>
<td>600</td>
<td>0.0936</td>
<td>2.9398</td>
<td>0.9726</td>
<td>0.4200</td>
<td>0.9726</td>
</tr>
<tr>
<td></td>
<td>P2</td>
<td>600</td>
<td>0.0725</td>
<td>2.8965</td>
<td>0.9701</td>
<td>0.3811</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P3</td>
<td>600</td>
<td>0.0664</td>
<td>2.8356</td>
<td>0.9689</td>
<td>0.2147</td>
<td></td>
</tr>
<tr>
<td>G2</td>
<td>P1</td>
<td>600</td>
<td>0.0861</td>
<td>2.7649</td>
<td>0.9601</td>
<td>0.2641</td>
<td>0.9754</td>
</tr>
<tr>
<td></td>
<td>P2</td>
<td>600</td>
<td>0.0845</td>
<td>2.8527</td>
<td>0.9754</td>
<td>0.3979</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P3</td>
<td>600</td>
<td>0.0689</td>
<td>2.8950</td>
<td>0.9742</td>
<td>0.4215</td>
<td></td>
</tr>
<tr>
<td>G3</td>
<td>P1</td>
<td>600</td>
<td>0.0621</td>
<td>2.6691</td>
<td>0.9597</td>
<td>0.2954</td>
<td>0.9748</td>
</tr>
<tr>
<td></td>
<td>P2</td>
<td>600</td>
<td>0.0859</td>
<td>2.9184</td>
<td>0.9726</td>
<td>0.4021</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P3</td>
<td>600</td>
<td>0.0746</td>
<td>2.8302</td>
<td>0.9748</td>
<td>0.4593</td>
<td></td>
</tr>
</tbody>
</table>
Table I shows that the highest percentage in healthy cases is in the first age early stage (uninfected) of plant growth. The highest percentage in cases of the first virus infection (early infection) is in the second age stage. The highest percentage in diseased cases (late infection) is in the third age stage.

The plant in the first age stage is in the sterilization stage and healthy hybridization and not exposed to a large pesticides spraying, taking into account the appropriate weather for cultivation. After that, in later stages of growth, the plant is exposed to larger spraying with pesticides and exposed to different climate factors as well as poor workmanship (farming), its stage will be the highest complete unhealthy rate. The data was divided into 70% training and 30% testing, and determining the number of batches required for the model. The accuracy rate and the loss rate were deduced as shown in the Fig. 5.

![Model Accuracy and Loss](image)

Fig. 5. Model (Accuracy and Loss).

The number of the expected data on the actual data was clear in the following table, and it was found that the second growth stage is the most vulnerable stage to viral infection through the distribution of data by 70% training and 30% testing, as shown in Table II.

| ['Gen2Phase1', 'Gen2Phase3', 'Gen3Phase3', 'Gen3Phase2', 'Gen1Phase2', 'Gen2Phase2', 'Gen1Phase3', 'Gen3Phase1', 'Gen1Phase1'] |

Computed fusion matrix: Heterogeneous data sources can be collectively mined by data fusion. Fusion can focus on a specific target relation and exploit directly associated data together with data on the context or additional constraints [23], as shown in the Fig. 6.

![Computed Fusion Matrix](image)

Fig. 6. Computed Fusion Matrix.

3) Generative Adversarial Networks (GANs): After the CNN stage of classification and prediction was completed, the stage of increasing data into by using the (GANs), and during this stage, the experiments were done on the real data and the steps of this stage were as follows:

- Prepare Dataset:
  - import numpy as np
  - import pandas as pd
  - import os
  - print(os.listdir("TomatoDB")

- Generator and Discriminator for Dataset:

- Image Samples:
  - SAMPLES TO SHOW = 8
  - Input: (60, 64, 64, 3), as shown in the Fig. 7.

![Images Sample Complete Description](image)

Fig. 7. Images Sample Complete Description.

- Code implementation steps, as shown in Table III:

| TABLE III. RESULT ACCURACY FOR GENERATOR AND DISCRIMINATOR MODEL: "FUNCTIONAL_1" |
|----|----|----|
| Layer (type) | Output Shape | Parameter |
| input_1 (InputLayer) | [(None, 64, 64, 3)] | 0 |
| conv1 (Conv2D) | (None, 32, 32, 32) | 2432 |
| batch_norm1 (BatchNormalization) | (None, 32, 32, 32) | 128 |
| conv1_out (LeakyReLU) | (None, 32, 32, 32) | 0 |
| conv2 (Conv2D) | (None, 16, 16, 64) | 51264 |
| batch_norm2 (BatchNormalization) | (None, 16, 16, 64) | 256 |
| conv2_out (LeakyReLU) | (None, 16, 16, 64) | 0 |
| conv3 (Conv2D) | (None, 8, 8, 128) | 204928 |
| batch_norm3 (BatchNormalization) | (None, 8, 8, 128) | 512 |
| conv3_out (LeakyReLU) | (None, 8, 8, 128) | 0 |
conv4 (Conv2D) (None, 8, 8, 256) 819456
batch_norm4 (BatchNormalization) (None, 8, 8, 256) 1024
conv4_out (LeakyReLU) (None, 8, 8, 256) 0
conv5 (Conv2D) (None, 4, 4, 512) 3277312
batch_norm5 (BatchNormalization) (None, 4, 4, 512) 2048
conv5_out (LeakyReLU) (None, 4, 4, 512) 0
flatten (Flatten) (None, 8192) 0
ligit (Dense) (None, 1) 8193
Total params Trainable params Non-trainable params
4,367,553 4,365,569 1,984

MODEL: "FUNCTIONAL_3"

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>input_2 (InputLayer)</td>
<td>([None, 100])</td>
<td>0</td>
</tr>
<tr>
<td>Dense (Dense)</td>
<td>(None, 8192)</td>
<td>827392</td>
</tr>
<tr>
<td>leaky_re_lu (LeakyReLU)</td>
<td>(None, 8192)</td>
<td>0</td>
</tr>
<tr>
<td>reshape (Reshape)</td>
<td>(None, 4, 4, 512)</td>
<td>0</td>
</tr>
<tr>
<td>trans_conv1 (Conv2DTranspose)</td>
<td>(None, 8, 8, 512)</td>
<td>6554112</td>
</tr>
<tr>
<td>batch_trans_conv1 (BatchNormalization)</td>
<td>(None, 8, 8, 512)</td>
<td>2048</td>
</tr>
<tr>
<td>trans_conv1_out (LeakyReLU)</td>
<td>(None, 8, 8, 512)</td>
<td>0</td>
</tr>
<tr>
<td>trans_conv2 (Conv2DTranspose)</td>
<td>(None, 16, 16, 256)</td>
<td>3277056</td>
</tr>
<tr>
<td>batch_trans_conv2 (BatchNormalization)</td>
<td>(None, 16, 16, 256)</td>
<td>1024</td>
</tr>
<tr>
<td>trans_conv2_out (LeakyReLU)</td>
<td>(None, 16, 16, 256)</td>
<td>0</td>
</tr>
<tr>
<td>trans_conv3 (Conv2DTranspose)</td>
<td>(None, 32, 32, 128)</td>
<td>819328</td>
</tr>
<tr>
<td>batch_trans_conv3 (BatchNormalization)</td>
<td>(None, 32, 32, 128)</td>
<td>512</td>
</tr>
<tr>
<td>trans_conv3_out (LeakyReLU)</td>
<td>(None, 32, 32, 128)</td>
<td>0</td>
</tr>
<tr>
<td>trans_conv4 (Conv2DTranspose)</td>
<td>(None, 64, 64, 64)</td>
<td>204864</td>
</tr>
<tr>
<td>batch_trans_conv4 (BatchNormalization)</td>
<td>(None, 64, 64, 64)</td>
<td>256</td>
</tr>
<tr>
<td>trans_conv4_out (LeakyReLU)</td>
<td>(None, 64, 64, 64)</td>
<td>0</td>
</tr>
<tr>
<td>logits (Conv2DTranspose)</td>
<td>(None, 64, 64, 3)</td>
<td>4803</td>
</tr>
<tr>
<td>out (Activation)</td>
<td>(None, 64, 64, 3)</td>
<td>0</td>
</tr>
<tr>
<td>Total params</td>
<td>Trainable params</td>
<td>Non-trainable params</td>
</tr>
<tr>
<td>11,691,395</td>
<td>11,689,475</td>
<td>1,920</td>
</tr>
</tbody>
</table>

- Result plot loss curve (Fig. 8):

The loss result it's almost 2%, and it's the complement the accuracy rate, and also the result rate of generator little some extent and the result rate of discriminator high some extent, this means that the capacity of the model is high, this is what is aimed to be achieved:

![Losses](image.png)

- Result Accuracy and Losses (Table IV):

<table>
<thead>
<tr>
<th>TABLE IV. GANs Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>GANs:</td>
</tr>
<tr>
<td>Losses: 2.1%</td>
</tr>
<tr>
<td>Accuracy: 97.9%</td>
</tr>
</tbody>
</table>

V. CONCLUSION AND FUTURE WORK

To summarize, DL was used in early prediction to detect diseases in different plant growth stages using the CNN algorithm for classification and prediction. Here, using the tomato infected with TMV as a model, the accuracy rate of TMV infection was 97%. The GANs used to increase the size of data and prediction accuracy rate by 98% when compared to the original data. For each plant growth phase, it became clear that the most growth stage group is vulnerable to viral infection is the second group. Therefore that determining the growth stages in this paper helped at obtaining results that prove the age group most susceptible to Unhealthy by determining the stages of Unhealthy also (healthy - first infection - Unhealthy). Thus, the study has concluded the previous results by applying to a set of real data that was collected manually from one of the farms in Egypt. Future work will include several DL models for early detection and classification of plant diseases due to using the rapid progress and improvements in DL models, transfer learning techniques, and CNN frameworks. Larger real-time dataset of TMV-infected tomato plants, and other important plant-disease system will be tested for attaining highest prediction accuracy. Building a robust and accurate digital and computer-based plant pest-infestations and microbial disease-infections early-detection and warning system, will significantly help plant protection in early stages, with increased yield, quality, local marketing, and international exporting competitiveness.

REFERENCES


