



South valley university



Faculty of science

Detection and Classification of Plant Leaf Diseases

Using Deep Learning

Under Supervision

DR/ Ahmed Elaraby

Preparation by /

Yasser Abdelgaber Mortada

Computer science Department

Academic year

2019-2020

The second part of the research

Part 2:

- ☒ *convert the data into arrays using numpy*
- ☒ *Building CNN Model*
- ☒ *compile the model and view the layers summary*
- ☒ *Train the model*
- ☒ *Test the model*
- ☒ *predicting pictures & its predicting category*

Abstract:

Agricultural productivity is something on which Economy highly depends. This is the one of the reasons that disease detection in plants plays an important role in agriculture field, as having disease in plants are quite natural. If proper care is not taken in this area then it causes serious effects on plants and due to which respective product quality, quantity or productivity is affected.

Usually farmers or experts observe the plants with naked eye for detection and identification of disease. But this method can be time processing, expensive and inaccurate. Automatic detection using image processing techniques provide fast and accurate results. This paper is concerned with a new approach to the development of plant disease recognition model, based on leaf image classification, by the use of deep learning

Introduction:

Automated, field-based detection of plant disease symptoms would be valuable for plant breeders and growers. However, this is complicated by the “noisy” nature of field imagery. There may be many sources of dead tissue, along with obscured symptoms. This requires a computer vision approach that is specific to the target disease and insensitive to such variations. Deep learning are a class of machine learning models that can be trained to accurately detect objects in images, making them the current standard for object recognition

Machine learning:

A machine-learning system is *trained* rather than explicitly programmed. It's presented with many examples relevant to a task, and it finds statistical structure in these examples that eventually allows the system to come up with rules for automating the task.

Deep learning:

Deep learning is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive *layers* of increasingly meaningful representations. It stands for this idea of successive layers of representations. How

many layers contribute to a model of the data is called the *depth* of the model. Other appropriate names for the field could have been *layered representations learning* and *hierarchical representations learning*. Modern deep learning often involves tens or even hundreds of successive layers of representations and they're all learned automatically from exposure to training data.

In deep learning, these layered representations are (almost always) learned via models called *neural networks*, structured in literal layers stacked on top of each other. The term *neural network* is a reference to neurobiology, but although some of the central concepts in deep learning were developed in part by drawing inspiration from our understanding of the brain, deep-learning models are *not* models of the brain. There's no evidence that the brain implements anything like the learning mechanisms used in modern deep-learning models.

Convolutional Neural Networks (cnn)

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

Convolutional neural networks are neural networks used primarily to classify images (i.e. name what they see), cluster images by similarity (photo search), and perform object recognition within scenes. For example, convolutional neural networks (ConvNets or CNNs) are used to identify faces, individuals, street signs, tumors, platypuses (platypi?) and many other aspects of visual data.

The efficacy of convolutional nets in image recognition is one of the main reasons why the world has woken up to the efficacy of deep

learning. In a sense, CNNs are the reason why deep learning is famous. The success of a deep convolutional architecture called AlexNet in the 2012 Image Net competition was the shot heard round the world. CNNs are powering major advances in computer vision (CV), which has obvious applications for self-driving cars, robotics, drones, security, medical diagnoses, and treatments for the visually impaired.

Convolutional networks can also perform more banal (and more profitable), business-oriented tasks such as optical character recognition (OCR) to digitize text and make natural-language processing possible on analog and hand-written documents, where the images are symbols to be transcribed.

CNNs are not limited to image recognition, however. They have been applied directly to text analytics. And they be applied to sound when it is represented visually as a spectrogram, and graph data with graph convolutional networks

Data preprocessing

Data preprocessing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In simple words, data preprocessing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format

Why Data Preprocessing in Machine Learning?

When it comes to creating a Machine Learning model, data preprocessing is the first step marking the initiation of the process. Typically, real-world data is incomplete, inconsistent, inaccurate (contains errors or outliers), and often lacks specific attribute

values/trends. This is where data preprocessing enters the scenario – it helps to clean, format, and organize the raw data, thereby making it ready-to-go for Machine Learning models

Image Pre-processing:

Pre-processing is a common name for operations with images at the lowest level of abstraction -- both input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

Four categories of image pre-processing methods according to the size of the pixel neighborhood that is used for the calculation of a new pixel brightness: pixel brightness transformations, geometric transformations, pre-processing methods that use a local neighborhood of the processed pixel, and image restoration that requires knowledge about the entire image. Other classifications of image pre-processing methods exist.

Image pre-processing methods use the considerable redundancy in images. Neighboring pixels corresponding to one object in real images have essentially the same or similar brightness value. Thus, distorted pixel can often be restored as an average value of neighboring pixels. Do you remember the example of filtering impulse noise?

Dataset:

All the images collected for the dataset were downloaded from the Internet, searched by disease and plant name on various sources. Images in the dataset were grouped into different classes which represented plant diseases which could be visually determined from leaves. In order to distinguish healthy leaves from diseased ones, one more class was added in the dataset. It contains only images of healthy leaves. An extra class in the dataset with background images was beneficial to get more accurate classification. Thus, deep neural network could be trained to differentiate the leaves from the surrounding.

✓ Plants Diseases:

A plant diseases takes places when an organism infect a plant and disrupts its normal growth habits , symptoms can range from slight discoloration to death of the plant , diseases have many causes including fungi, bacteria ,viruses and nematodes.one of these plants is Tomato .

■ Tomato Diseases

❖ Bacterial Spot of Pepper and Tomato

Bacterial spot is caused by four species of *Xanthomonas* and occurs worldwide wherever tomatoes are grown. Bacterial spot causes leaf and fruit spots, which leads to defoliation, sun-scalded fruit, and yield loss. Due to diversity within the bacterial spot pathogens, the disease can occur at different temperatures and is a threat to tomato production worldwide. Disease development is favored by temperatures of 75 to 86 °F and high precipitation. In North Carolina, it is more prevalent in seasons with high precipitation and less prevalent during dry years.



❖ Early Blight of Tomato

Early blight is a common tomato disease caused by the fungus *Alternaria solani*. It can affect almost all parts of the tomato plants, including the leaves, stems, and fruits. The plants may not die, but they will be weakened and will set fewer tomatoes than normal. Early blight generally attacks older plants, but it can also occur on seedlings. Stressed plants or plants in poor health are especially susceptible. Early blight is also a problem with potatoes.

Early blight is a difficult enough disease to deal with, but don't confuse it with the much more problematic [late blight](#) that can quickly kill tomato plants and spread for miles.



❖ Tomato Leaf Mold

Production of tomatoes under high tunnel and plastic has increased significantly over the last few years in part due to consumers demand for “local” produce. Even though growing under these conditions can reduce the occurrence of some diseases, it can increase the occurrence of others. Tomato leaf mold disease is one that’s showing a significant increase. It’s caused by a fungus formerly known as *Cladosporium fulvum*, but now known as *Fulvia fulva* by producers and those in the seed trade, and *Passalora fulva* by mycologists. The disease is rarely seen on field-grown plants, and when it’s observed in the field, it’s due to infected greenhouse-grown transplants.



❖ Tomato Late_blight

Late blight is caused by the oomycete *Phytophthora infestans*. Oomycetes are fungus-like organisms also called water molds, but they are not true fungi. There are many different strains of *P. infestans*. These are called clonal lineages and designated by a

number code Many clonal lineages affect both tomato and potato, but some lineages are specific to one host or the other. The host range is typically limited to potato and tomato, but hairy nightshade (*Solanum physalifolium*) is a closely related weed that can readily become infected and may contribute to disease spread. Under ideal conditions, such as a greenhouse, petunia also may become infected.



❖ Tomato Septoria_leaf_spot

Septoria leaf spot, also called Septoria blight, is a very [common disease of tomatoes](#). It is caused by a fungus (*Septoria lycopersici*) and can affect tomatoes and other plants in the Solanaceae family, especially [potatoes](#) and [eggplant](#), just about anywhere in the world. Although Septoria leaf spot is not necessarily fatal for your tomato plants, it spreads rapidly and can quickly defoliate and weaken the plants, rendering them unable to bear fruit to maturity.



❖ Tomato Target_Spot

Target spot, or early blight is one of the most common diseases attacking leaves and stems of potatoes and tomato. The disease is caused by the fungus *Alternaria solani*. On the leaves the disease starts as small circular to oval dark brown to black spots. These spots enlarge, becoming oval to angular, and are normally confined within the main veins of the leaflets. They are up to 6 mm in diameter. Under very favourable conditions individual spots can grow to 10-12 mm. They become leathery looking, and the development of close concentric rings within each spot gives the disease its name, target spot. When the disease is severe, spots can unite and cause an upward rolling of the leaf tips and death of leaves. Spots that develop on stems are more elongated than those on the leaves. Target spot can sometimes infect the tubers. It appears as small dark, slightly sunken, circular to irregular-shaped lesions (10 to 20 mm in diameter), with slightly raised margins. A brownish, corky dry rot up to 6 mm deep develops in the tissue beneath the lesion. Affected tubers are prone to attack from other soil fungi which can cause complete rotting of the tuber.



❖ Tomato _mosaic_virus

Tomato mosaic virus is a serious and extremely contagious disease. It is also hard to identify, with symptoms varying wildly depending upon the variety and age of the infected plant, the strain of the virus, and environmental conditions. To make matters worse, it is very hard to distinguish from the closely related tobacco mosaic virus. Tomato mosaic virus symptoms can be found at any stage of growth and all parts of the plant may be infected. They are often seen as a general mottling or mosaic appearance on foliage. When the plant is severely

affected, leaves may look akin to ferns with raised dark green regions. Leaves may also become stunted. Infected plants may have a severe reduction in fruit set and those that do set may be dotted with yellow blotches and necrotic spots while the interior of the fruit is brown. Stems, petioles, leaves and fruit may all show signs of infection



❖ Tomato Yellow Leaf Curl Virus

Tomato yellow leaf curl virus (TYLCV) is one of the most well-known tomato-infecting begomoviruses and transmitted by *Bemisia tabaci*. Seed transmission has previously been reported for some RNA viruses, but TYLCV has not previously been described as a seed-borne virus. In 2013 and 2014, without whitefly-mediated transmission, TYLCV was detected in young tomato plants germinated from fallen fruits produced from TYLCV-infected tomato plants in the previous cultivation season. In addition, TYLCV-Israel (TYLCV-IL) was also detected in seeds and their seedlings of TYLCV-infected tomato plants that were infected by both viruliferous whitefly-mediated transmission and agro-inoculation. The seed infectivity was 20–100%, respectively and the average transmission rate to seedlings was also 84.62% and 80.77%, respectively. TYLCV-tolerant tomatoes also produced TYLCV-infected seeds, but the amount of viral genome was less than seen in TYLCV-susceptible tomato plants. When tomato plants germinated from TYLCV-infected seeds, non-viruliferous whiteflies and healthy tomato plants were placed in an insect cage together, TYLCV was detected from whiteflies as well as receiver tomato plants six weeks later. Taken together, TYLCV-IL can be transmitted via seeds and tomato plants germinated from TYLCV-infected seeds can be an inoculum source of TYLCV. This is the first report about TYLCV seed transmission in tomato.



❖ Tomato Spider_mites Two-spotted_spider_mite

The two-spotted spider mite is the most common mite species that attacks vegetable and fruit crops in New England. Spider mites can occur in tomato, eggplant, potato, vine crops such as melons, cucumbers, and other crops. Two-spotted spider mites are one of the most important pests of eggplant. They have up to 20 generations per year and are favored by excess nitrogen and dry and dusty conditions. Outbreaks are often caused by the use of broad-spectrum insecticides which interfere with the numerous natural enemies that help to manage mite populations. As with most pests, catching the problem early will mean easier control.



❖ Tomato__healthy



Requirements:

- Install Anaconda for windows10 -64bit
- python3.7

lib:-

✓ **Pandas**

The *pandas* package is the most important tool at the disposal of Data Scientists and Analysts working in Python today. The powerful machine learning and glamorous visualization tools may get all the attention, but pandas is the backbone of most data projects

✓ **matplotlib.pyplot**

✓ **seaborn**

✓ **glob**

✓ **numpy**

✓ **Tensor flow**

✓ **Keras**

About Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform **Tensor Flow**. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research

Steps:-

step1: Install data set from kaggle and upload it into gdrive

step2: install all the required packages

step3: preprocessing the data set images

step4: Building CNN model


step5: Compile and train the model

Part 2:

convert the data into arrays using numpy

```
[ ] X_train = np.array(X_train)
    X_test = np.array(X_test)
    X_pred_array = np.array(X_pred)
    y_train = np.array(y_train)
    y_test = np.array(y_test)

    print(f'X_train shape is {X_train.shape}')
    print(f'X_test shape is {X_test.shape}')
    print(f'X_pred shape is {X_pred_array.shape}')
    print(f'y_train shape is {y_train.shape}')
    print(f'y_test shape is {y_test.shape}')
```

 X_train shape is (10000, 256, 256, 3)
X_test shape is (3500, 256, 256, 3)
X_pred shape is (500, 256, 256, 3)
y_train shape is (10000,)
y_test shape is (3500,)

Building The Model

now to build the CNN model by Keras , using Conv2D layers , MaxPooling & Denses

Architecture: The model uses five convolutional layers:

1. Convolutional layer with 224-3 and 3 filters
2. Convolutional layer with 150-3 and 3 filters
3. Max pooling layer with 4x4 filter
4. Convolutional layer with 120-4 and 3 filters
5. Max pooling layer with 4x4 filter
6. Convolutional layer with 80-3 and 3 filters
7. Convolutional layer with 50-3 and 3 filters
8. Max pooling layer with 4x4 filter
9. Flattening
10. Dense layer with 100 neurons
11. Dropout layer with rate 0.3
12. Dense layer with 50 neurons
13. Dropout layer with rate 0.3
14. Output layer with 10 neurons corresponding to the 10 fashion categories

We use activation padding ="sam" and activation function ="relu"

And we use activation function=" softmax"

```
KerasModel = keras.models.Sequential([
    keras.layers.Conv2D(224,kernel_size=(3,3),padding='same',activation='relu',input_shape=(s,s,3)),

    keras.layers.Conv2D(150,kernel_size=(3,3),padding='same',activation='relu'),

    keras.layers.MaxPool2D(4,4),
    ## keras.layers.BatchNormalization(),
    keras.layers.Conv2D(120,kernel_size=(3,3),padding='same',activation='relu'),
    keras.layers.MaxPool2D(4,4),
    ## keras.layers.BatchNormalization(),
    keras.layers.Conv2D(80,kernel_size=(3,3),padding='same',activation='relu'),
    keras.layers.MaxPool2D(4,4),
    ##keras.layers.BatchNormalization(),
    keras.layers.Conv2D(50,kernel_size=(3,3),padding='same',activation='relu'),

    keras.layers.MaxPool2D(4,4),
    # keras.layers.BatchNormalization(),

    keras.layers.Flatten() ,
    keras.layers.Dense(120,activation='relu') ,
    keras.layers.Dropout(rate=0.3),
    keras.layers.Dense(100,activation='relu') ,

    keras.layers.Dense(50,activation='relu') ,
    keras.layers.Dropout(rate=0.3) ,
    keras.layers.Dense(10,activation='softmax')
])
```

Compile and train the model

display the architecture of our model so far

```
[ ] KerasModel.compile(optimizer = 'adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])

[ ] print('Model Details are : ')
    print(KerasModel.summary())
```

Model Details are :
Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 256, 256, 224)	6272
conv2d_2 (Conv2D)	(None, 256, 256, 150)	302550
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 150)	0
conv2d_3 (Conv2D)	(None, 64, 64, 120)	162120
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 120)	0
conv2d_4 (Conv2D)	(None, 16, 16, 80)	86480
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 80)	0
conv2d_5 (Conv2D)	(None, 4, 4, 50)	36050
max_pooling2d_4 (MaxPooling2D)	(None, 1, 1, 50)	0
flatten_1 (Flatten)	(None, 50)	0
dense_1 (Dense)	(None, 120)	6120
dropout_1 (Dropout)	(None, 120)	0
dense_2 (Dense)	(None, 100)	12100
dense_3 (Dense)	(None, 50)	5050
dropout_2 (Dropout)	(None, 50)	0
dense_4 (Dense)	(None, 10)	510
Total params: 617,252		
Trainable params: 617,252		
Non-trainable params: 0		

training the model with 45 epochs

```
[ ] epochs = 45
    ThisModel = KerasModel.fit(X_train, y_train, epochs=epochs, batch_size=32, verbose=1)
```

```
10000/10000 [-----] - 147s 15ms/step - loss: 0.1066 - accuracy: 0.9714
Epoch 40/45
10000/10000 [=====] - 147s 15ms/step - loss: 0.0661 - accuracy: 0.9808
Epoch 41/45
10000/10000 [=====] - 147s 15ms/step - loss: 0.1075 - accuracy: 0.9714
Epoch 42/45
10000/10000 [=====] - 147s 15ms/step - loss: 0.0980 - accuracy: 0.9721
Epoch 43/45
10000/10000 [=====] - 147s 15ms/step - loss: 0.1213 - accuracy: 0.9668
Epoch 44/45
10000/10000 [=====] - 147s 15ms/step - loss: 0.0953 - accuracy: 0.9769
Epoch 45/45
10000/10000 [=====] - 147s 15ms/step - loss: 0.1338 - accuracy: 0.9673
```

❖ Accuracy

```
[ ] ModelLoss, ModelAccuracy = KerasModel.evaluate(X_test, y_test)

print('Test Loss is {}'.format(ModelLoss))
print('Test Accuracy is {}'.format(ModelAccuracy ))
```

3500/3500 [=====] - 14s 4ms/step
Test Loss is 0.387799259477111
Test Accuracy is 0.9154285788536072

predict X test

```
[ ] y_pred = KerasModel.predict(X_test)

print('Prediction Shape is {}'.format(y_pred.shape))
```

Prediction Shape is (3500, 10)

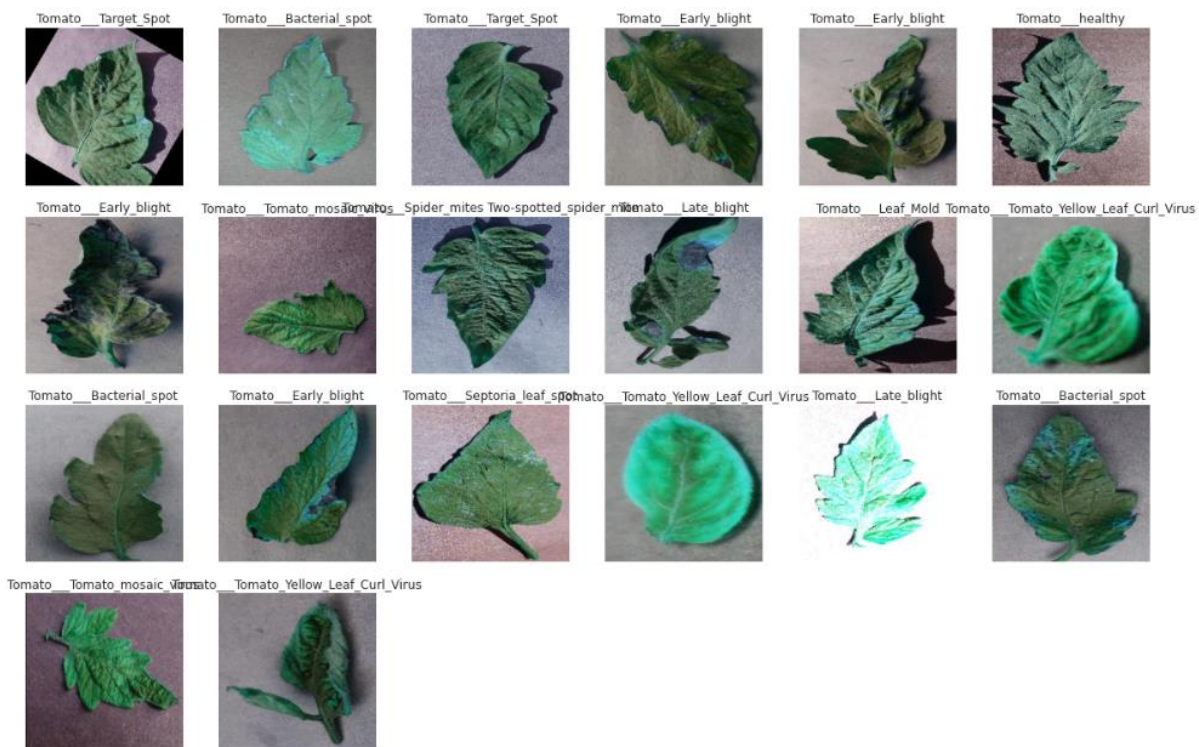
predict X Predic

```
[ ] y_result = KerasModel.predict(X_pred_array)

print('Prediction Shape is {}'.format(y_result.shape))
```

Prediction Shape is (500, 10)

```
[ ] plt.figure(figsize=(20,20))
    for n , i in enumerate(list(np.random.randint(0,len(X_pred),20))) :
        plt.subplot(6,6,n+1)
        plt.imshow(X_pred[i])
        plt.axis('off')
        plt.title(getcode(np.argmax(y_result[i])))
```



References

1. Dataset:
<https://www.kaggle.com/noulam/tomato>
2. code link:
<https://colab.research.google.com/drive/1oVI6uiuftv92kdt2RDrdtZZOD2CAG1pW#scrollTo=IQ-EkanJH8tU>
3. tomato diseases:
<https://hgic.clemson.edu/factsheet/tomato-diseases-disorders/>
4. Deep learning & Machine learning:
<https://parsers.me/deep-learning-machine-learning-whats-the-difference/>
5. Plant Leaf Disease Detection using Deep Learning and Convolutional Neural Network
Anandhakrishnan MG Joel Hanson¹ , Annette Joy² , Jerin Francis³ Department of Computer Science Engineering SCET, India
6. Khirade SD, Patil AB. Plant disease detection using image processing. In: IEEE International Conference on Computing Communication Control and Automation; Pune, India; 2015. pp. 768–771
7. Arivazhagan S, Shebiah RN, Ananthi S, Varthini SV. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. Agricultural Engineering International CIGR Journal 2013
8. Koike, S.T., Gladders, P., Paulus, A.O.: Vegetable Diseases: A Color Handbook. Academic Press, San Diego (2007)
9. Brahimi, M., Boukhalfa, K., Moussaoui, A.: Deep learning for tomato diseases: classification and symptoms visualization. Appl. Artif. Intell. 31(4), 1–17 (2017)
10. Compendium of Tomato Diseases and Pests, 2nd Ed. 2014. J. B. Jones, T. A. Zitter, M. T. Momol and S. A. Miller, eds. APS Press, The American Phytopathological Society, St. Paul, Minnesota.