A Deep CNN Approach for Plant Disease Detection

Fatma MARZOUGUI
Faculty of Sciences, University of Gafsa, Gafsa, Tunisia
fatma91marzougui13@gmail.com

Mohamed ELLEUCH
National School of Computer Science (ENSI), University of Manouba, Manouba, Tunisia
mohamed.elleuch@fss.usf.tn

Monji KHERALLAH
Faculty of Sciences, University of Sfax, Sfax, Tunisia
monji.kherallah@fss.usf.tn

***Abstract*— The diagnosis of the plants is carried out with a visual inspection by experts and a biological examination is the second choice if necessary. They are usually expensive and time consuming. This inspired several computer methodologies to detect plant blights based on their leaf images. We apply a computer methodology on Deep Learning systems based on artificial neural networks, this branch also allows for the early detection of plant diseases, by applying convolutional neural networks (CNNs) familiar with some of the famous architectures, notably the "ResNet" architecture, using an augmented dataset containing images of healthy and diseased leaves (each leaf is manually cut and placed on a uniform background) with acceptable accuracy rates in the research environment. This Deep Learning technique has shown very good performance for various object detection problems. The model fulfills its role by classifying images into two categories (disease-free) and (diseased). According to the results obtained, the developed system achieves better detection performances than those proposed in the state of the art. Finally, to compare their performances, we use the implementation under Anaconda 2019.10.**

Keywords: plant disease detection; Deep Learning; CNN; Data Augmentation; ResNet

I. Introduction

Globally, plant diseases have been identified as an increased threat to food security. Therefore, the detection of plant diseases is the most important step in achieving good crops. The classification of plants "with and without disease" is considered a difficult problem because of the variety and similarity of plants in nature [1] and [2].

The most successful approaches using artisanal features require object representations using a local descriptor of a point of interest provide local characterization in the form of an attribute vector. Indeed, there are algorithms SURF [3] and SIFT [4], allow to detect points of interest and to build robust descriptors with several transformations. In object recognition, a global descriptor is easier to use because it processes the entire image. And all the pixels of the image corresponding to the area of interest are taken into account in the description.

The descriptor is thus less sensitive to distortions from one image to another. Currently, the descriptor the most widely used in object recognition, the HOG gradient histogram of the makes it possible to obtain satisfactory and fast results.

After learning a discriminating machine learning model, for example the support vector machines (SVM) and K-nearest neighbor (KNN), with such representations.

The exploration of a "Deep Learning" approach for agricultural uses has intensified in research and opens the door to new uses and gains performance compared to current methods. Especially, The Deep Learning" is used to determine the extraction of characteristics in a way that synthetic and to give a clear detection on a proposed data set. This method is used to reduce the memory footprint and improve performance. In general, "CNN" is the best method for any prediction problem involving input image data and requires minimal pre-processing. It is structured to classify large-scale images.

In this context, several research works have been carried out to improve the performance of "CNN" on tasks related to computer vision, due to the detection of plant diseases. Advances in "CNN" can be classified in different ways, including activation, loss function, data enhancement, optimization algorithms.

Mohanty et al. [5] analyzed 14 plant types from the PlantVillage dataset with convolutional neural networks (CNN) and achieved over 99% classification accuracy on images in the research environment.

Wang et al. [6] applied the transfer learning technique on the same PlantVillage dataset and showed an accuracy of 90.4%. Fujita et al. [7] used their own sheet data set in the field and analyzed it with CNN. They showed an average accuracy of 82.3% under various background and photographic conditions. The authors of [8] used a CNN-based system to identify 13 disease types in five crops using images

978-1-7281-8855-3/20/$31.00 ©2020 IEEE

downloaded from the Internet and achieved 96.3% overall accuracy..

With the successful use of "CNN" for image classification, another architecture called "VGG" [9], which gives good results both for disease identification, but suffers from the need for a lot of computation.

The choice of architecture is very complex, so it is important to study and explain effective architectures to inspire our research". GoogleNet" [10] and [11] obtained the best results with accuracies of 98.33% and 97.66% [12], on the "AgrilPlant" dataset and the "LeafSnap" dataset.

However, the ResNet architecture, reaches an error rate of 3.57% (top 5 error rate) [13], where the recipe for success of this architecture to form such a deep network is that it has residual connections and get the accuracy of a much deeper network.

The increase in data plays a crucial role in increasing the number of training images, which often contributes to improving the classification performance of deep learning techniques for computer vision problems [14] and [15]. The results show that CNN methods with datasets augmented with specific data give the highest accuracies. The main objective of this work is to identify, from an image, the healthy leaves and sick leaves of a given dataset.

The following sections of this document are organized as follows: Section 2 details the proposed method. Section 2 details the proposed method. Section 3 presents the experimental results. In this section, the proposed deep learning architecture is validated on our own dataset. Finally, the results are discussed and our own data set is provided using a pre-trained deep neural network.

II. Proposed Method

A CNN convolutional neural network is perhaps the most widely applied method for extracting reasonable information from huge datasets.

The architecture of CNN is illustrated in Figure 1, which allows efficient processing of image data. A deep CNN architecture consists of several layers of different types. Typically, it begins with one or more convolutional layers followed by one or more grouping layers, activation layers, and ends with one or more fully connected layers. In the convolution layer, the convolution operation is performed to extract features, and the output is passed to the activation function. As long as, the clustering layer is generally used to reduce the size of the feature map and provides robust learning results for the input data.

The convolution and pooling layers are then passed through in several steps to obtain global features from the input data. Finally, the extracted characteristics are passed to the fully connected layer where classification is performed in this layer.

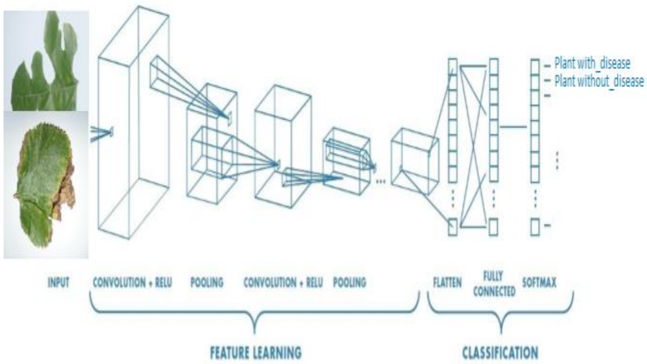

Fig. 1. Convolution Neural Network (CNN) Architecture

Our approach is to detect and classify plant diseases, for this we have used deep learning. Hence, the following figure shows our model for a plant management system.

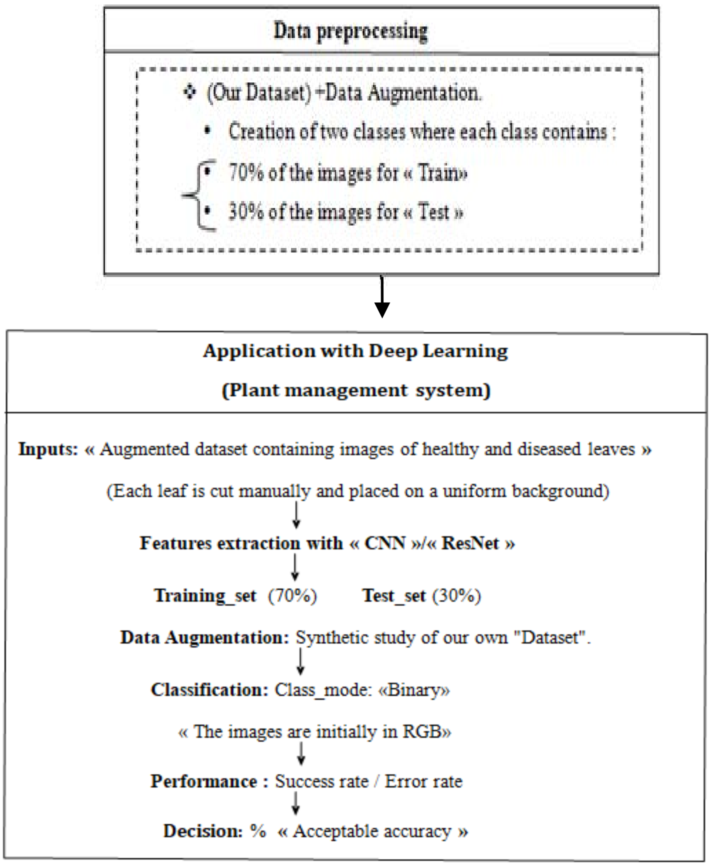

Fig. 2. Overview of Proposed Framework

The development of the model is also carried out, to be implemented at a later stage. In the "Train" learning process in our "dataset", in order to extract from the patterns or characteristics of an image and characterize it using our "CNN" model. So that he can identify plant diseases on the leaves.

- Training step: Allows you to set the initial weight to enter to the hidden layer. This step consists of two processes, namely feedback and retro-propagation.

- Testing step: It is a classification process using the weights obtained during the learning process. Recently, this process is not very different from the learning process, but during the test, no backpropagation is performed after the execution of the feedback process where the result of this process is the accuracy of the classification process.

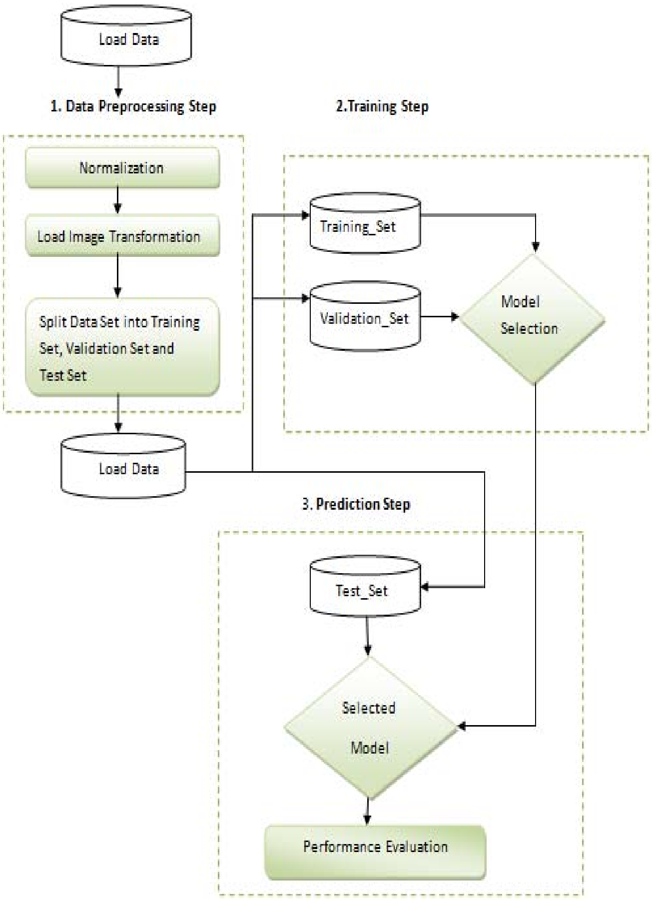

Fig. 3. The proposed structure of our model

A. *CNN Model*

The convolutional neural network uses a special mathematical operation called convolution instead of matrix multiplication in at least one of its layers. It is officially formed by a stack of layers.

- The convolution layer (CONV) which processes the data of a receiver field.
- The pooling layer (POOL); which compresses the information by reducing the size of the intermediate image.
- The correction layer (ReLU), with reference to the activation function (Linear Rectification Unit).
- The Fully Connected Layer (FC), which is a perceptron-type layer.

TABLE I. CNN MODEL

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 64, 64, 3)	0
conv2d_9 (Conv2D)	(None, 62, 62, 64)	1792
max_pooling2d_9	(MaxPooling2 (None, 31, 31, 64)	0
conv2d_10 (Conv2D)	(None, 29, 29, 64)	36928
max_pooling2d_10	(MaxPooling (None, 14, 14, 64)	0
conv2d_11 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_11	(MaxPooling (None, 6, 6, 64)	0
flatten_3 (Flatten)	(None, 2304)	0
dense_6 (Dense)	(None, 128)	295040
dense_7 (Dense)	(None, 2)	258

B. *ResNet Model*

He et al [16] rectify the layers as residual learning functions with reference layer entries, instead of learning unreferenced functions and presenting the residual learning framework.

In "ResNet", there are shortcut links present that ignore one or more of the following diapers. These shortcut connections do an identity mapping and their outputs are added at the exit of stacked layers. However, these shortcut connections do not add additional parameters or increase complexity.

The process of defining the ResNet model is given in Table 2.

TABLE II. RESNET MODEL

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 64, 64, 3)	0	
conv2d_12 (Conv2D)	(None, 32, 32, 64)	36928	activation_10[0][0]
batch_normalization_11	(BatchNor (None, 32, 32, 64)	256	conv2d_12[0][0]
add_5 (Add)	(None, 32, 32, 64)	0	activation_9[0][0] batch_normalization_11 [0][0]
activation_11 (Activation)	(None, 32, 32, 64)	0	add_5[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 64)	36928	activation_11[0][0]
batch_normalization_12	(BatchNor (None, 32, 32, 64)	256	conv2d_13[0][0]
activation_12 (Activation)	(None, 32, 32, 64)	0	batch_normalization_12 [0][0]
conv2d_14 (Conv2D)	(None, 32, 32, 64)	36928	activation_12[0][0]

batch_normalization_13	(BatchNor (None, 32, 32, 64)	256	conv2d_14[0][0]
add_6 (Add)	(None, 32, 32, 64)	0	activation_11[0][0] batch_normalization_13 [0][0]
activation_13 (Activation)	(None, 32, 32, 64)	0	add_6[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 128)	73856	activation_13[0][0]
batch_normalization_14	(BatchNor (None, 16, 16, 128)	512	conv2d_15[0][0]
activation_14 (Activation)	(None, 16, 16, 128)	0	batch_normalization_14 [0][0]
average_pooling2d_1	(AveragePooling (None, 2, 2, 128)	0	activation_19[0][0]
flatten_1 (Flatten)	(None, 512)	0	average_pooling2d_1[0] [0]
dense_1 (Dense)	(None, 2)	1026	flatten_1[0][0]

III. EXPERIMENTS AND RESULTS

In this part, we present our implementation in which we describe the operation of "CNN" for the purpose of detecting and classifying plant diseases on the leaves.

For all methods, we used separate training and testing packages. All details are reported on the test set. We used Python 3.7, the programming language and the OpenCV computer vision library for all deep learning models.

A. Dataset

To evaluate the proposed system, we are preparing our own set of data that will be used to composed of images captured by a camera / Samsung-Intelligent LCD / 12.2 M /Pixels, we avoid unwanted background.

Each sheet is cut manually and placed on a uniform background: (white paper, same capture space, etc.) and we determine the whole base in the same way. We create our dataset containing images of healthy and diseased leaves, consisting of 500 images from two classes:

- "with_disease.1"..."with_disease.250"
- "without_disease.1"..."without_disease.250".

The dimensions of each image are: [target_size =(64,64), batch_size=3].

Then we apply the data augmentation technique. We form the model on a subset of 70% and test it on another 30% subset. In the adopted distribution, some examples of 12 "Train" folder images are shown in Fig.4.

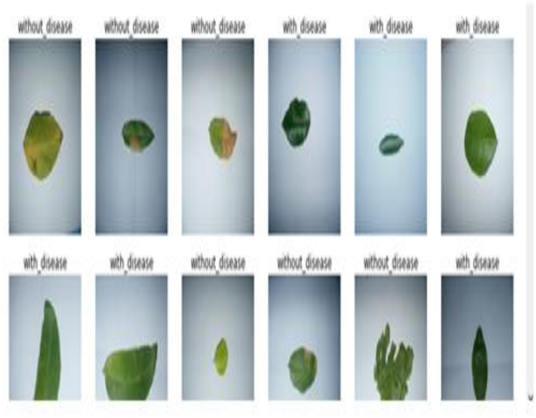

Fig. 4. Samples of plant leaf images

B. Data Augmentation

Research on disease detection on leaves using deep learning requires a huge amount of data but is difficult to obtain. So, we choose to apply a more efficient data augmentation technique to solve this problem.

The increase in the whole "train" also helps to reduce the problem of over-adjustment. The "Keras Image Data Generator" class was used to increase data and best results were obtained by setting the parameters to the following:

(a) Rotation range: 30
(b) Width and height shift ranges = 0.2
(c) Horizontal flip: True
(d) Fill mode: Nearest

The following figure illustrates an example of original photos and their preprocessed and augmented versions are presented.

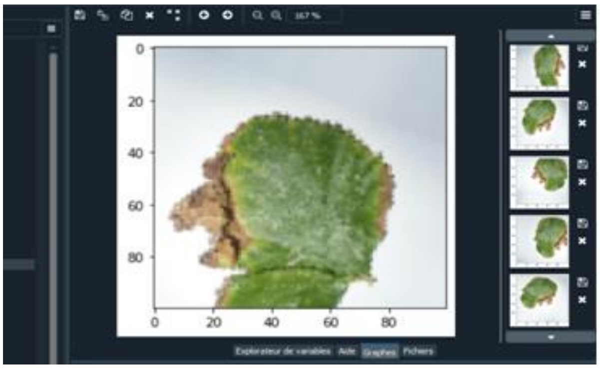

Fig. 5. . Samples of plant leaf images after using Data Augmentation

C. Experimental Settings

Our model is implemented with Python's TensorFlow library, which is a symbolic mathematical library used to create 'deep learning' models. For the validation of our proposed system, based on the deep convolutional neural network (CNN), we use our own dataset containing images of healthy and unhealthy plant leaves. It is divided into two parts: Learning and testing.

This network is recycled to detect plant diseases according to our own sample images. The configuration of our CNN model is characterized by: RMSProp Optimizer which represents a faster learning optimizer, a BatchNorm and label smoothing to decrease the loss function and avoid over-adjustment.

The learning parameters of our model are enumerated as follows: Lot size = 32 using RMSProp Optimizer and Adam with a learning rate “LR” of 0.001 and a decay rate of 0.3. Finally, the training is given for 10 epochs, which ensures the convergence.

D. Results and Discussion

After experimentation, we find that our trained model achieves an accuracy of 94.80% after training for 10 epochs. Then, using the "Data Augmentation" method on our own data set to improve the accuracy of the model without reducing the learning efficiency, we also obtain an accuracy of 97.2%. Finally, after implementation with the "ResNet" architecture, the complete detection performance reaches 98.96%.

Therefore, the following table shows the final success rates and error rates of our deep learning approach to plant disease detection.

TABLE III. PERFORMANCE OF OUR PROPOSED METHOD

Datasets	**Method**	**Epochs**	**Loss**	**Accuracy**
Our Dataset (Using Data Augmentation)	CNN (without DA)	1…..10 epochs	0.12	94.80%
	CNN (with DA)		**0.06**	97.2%
	ResNet (with DA)		0.34	98.96%

From Table III, it is obvious that the ResNet architecture gives better results than CNN, and requires less time for testing.

TABLE IV. ACCURACY RATE AND LOSS RATE OF OUR PROPOSED METHOD

Methods	**Our Approach** (CNN and ResNet) Using Data Augmentation		AlexNet [13]	GoogleNet [12]
Accuracy	97.2%	98.96%	89.51%	97.66%
Test/Time (ms)	1.5ms	1.7 ms	2.9 ms	1.7 ms

Our proposed system shows its reliability and speed with a satisfactory accuracy of 98.96%. This represents an improvement of 1.3 % compared to GoogLeNet approach and an improvement of 9.5 % over the AlexNet network.

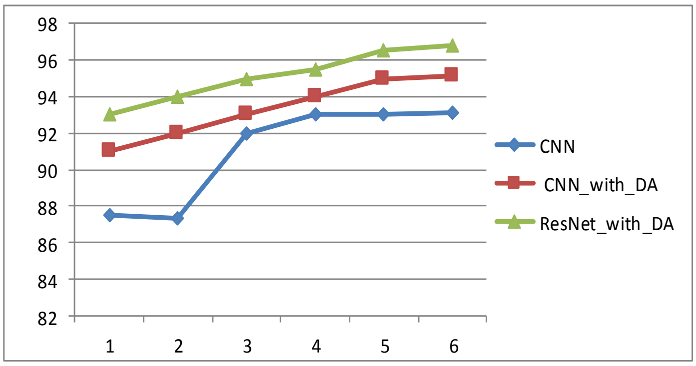

Fig. 6. Graph showing our result

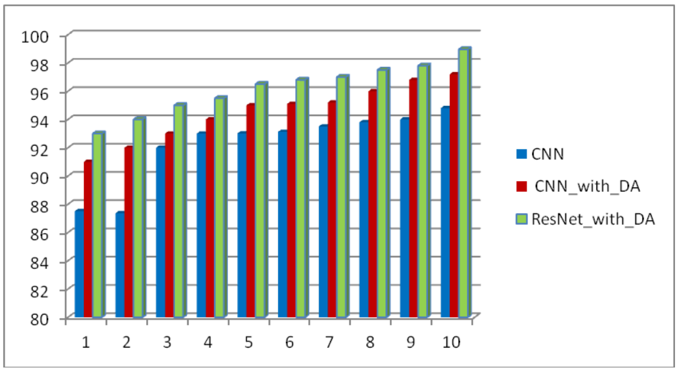

Fig. 7. Accuracy rate for each model

From the results observed (see Table III and Fig. 6), we notice that the architecture ResNet recorded a higher rate of 98.96% (see Fig.7), and the comparison with other results is presented in Table IV. We have noted that the performance of our system is achieving a promising result, further research work due to detecting plant diseases, based on traditional methods [12], [16] and [17] and in-depth learning approaches [12] and [18].

TABLE V. PERFORMANCE COMPARAISON

Authors	**Methods**		**Accuracy Rate**	
Our system	CNN (Using DA)	ResNet (Using DA)	**97.2%**	**98.96%**
Pawara et al. (2017) [12]	HOG with KNN	GoogleNet	58.51%	97.66%
Rumpf et al. (2010) [16]	SVM		86.4%	
Xie et al . (2017) [17]	KNN		67.4 %	
Suma et al. (2019) [18]	CNN (Using DA)		98.62%	

IV. CONCLUSION

The Convolutional neural network (CNN) is an important branch of deep learning. Because of its strong feature extraction capability, CNN models have been introduced to identify plant diseases.

In this research, we used pre-trained weights as a starting point to avoid a very long treatment. Subsequently, we compared the proposed approach with several artisanal shallow structure approaches based on machine learning. The proposed system achieves promising precision results on our set of data on plant leaves, demonstrating the effectiveness of this proposed approach for detection of diseases.

REFERENCES

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," Frontiers in Plant Science, European Conference on Computer Vision, 2016.

[2] H. Cap, K. Suwa, E. Fujita , S. Kagiwada, H. Uga and H. Iyatomi : A Deep Learning Approach for on-site Plant Leaf Detection. European Conference on Computer Vision, 2018.

[3] A. P. Adewale Owolawi and T. Mapayi: Deep Learning Based on NASNet for Plant Disease Recognition Using Leave Images. European Conference on Computer Vision, 2019.

[4] M. Zisserman and Nilsback; Automated flower classification over a large number of classes. European Conference on Computer Vision**,** pages 722– 729, 2008.

[5] S. P. Mohanty, D. P. Hughes, and M. Salathé: "Using Deep Learning for Image-Based Plant Disease Detection," Frontiers in Plant Science, European Conference on Computer Vision, 2016.

[6] G. Wang, Y. Sun, and J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning," Computational Intelligence and Neuroscience, pp. 1-8, 2017.

[7] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic Investigation on a Robust and Practical Plant Diagnostic System," IEEE Proc. on Machine Learning and Applications, pp. 989-992, 2016.

[8] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," Computational Intelligence and Neuroscience, vol. 2016, pp. 1-11, 2016.

[9] K. Zisserman and Simonyan: Very deep convolutional networks for large-scale image recognition. European Conference on Computer Vision,2014.

[10] Vanhoucke and V. Rabinovich: Going deeper with convolutions. Conference on Computer Vision and Pattern Recognition, pages 1–9 2015.

[11] C. Szegedy and al: Going Deeper with Convolutions. European Conference on Computer Vision, 2014.

[12] P. Pawara, E. Okafor, O. Surinta, L. Schomaker and M. Wiering1: Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition. European Conference on Computer Vision, 2017.

[13] He, X. Zhang, S. Ren, J. Sun: Deep Residual Learning for Image Recognition. European Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[14] A. Sutskever , G.E and I. Hinton: Data Augmentation for Plant Classification . European Conference on Computer Vision, September, 2017. 14

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton: ImageNet classification with deep convolutional neural networks. European Conference on Computer Vision ,3–6 December ,2012.

[16] Rumpf and al: Detection of symptoms of sugar beet diseases. European Conference on Computer Vision and pattern recognition, 2010.

[17] Xie and al : Early detection and monitoring of symptoms of gray mold on tomato culture. European Conference on Computer Vision and pattern recognition, 2017.

[18] V. R. Suma, A. Shetty, F. Rishab Tated, S. Rohan, S Pujar and Triveni: CNN based Leaf Disease Identification and Remedy Recommendation System.European Conference on Computer Vision and pattern recognition, 2019.