

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/316976487>

Deep Learning for Tomato Diseases: Classification and Symptoms Visualization

Article in *Applied Artificial Intelligence* · May 2017

DOI: 10.1080/08839514.2017.1315516

CITATIONS

44

READS

2,763

3 authors, including:



Med Brahimi

Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj

13 PUBLICATIONS 90 CITATIONS

[SEE PROFILE](#)



Boukhalifa Kamel

University of Science and Technology Houari Boumediene

84 PUBLICATIONS 338 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Research National Project "SITI" [View project](#)



ics 2017 [View project](#)

Deep learning for Tomato Diseases: Classification and Symptoms Visualization

Mohammed Brahimi^{a,b,*}, Kamel Boukhalifa^a, Abdelouahab Moussaoui^c

^a*Computer Science Department, USTHB University, Algiers, Algeria*

^b*Computer Science Department, Mohamed El Bachir El Ibrahimi University, Bordj Bou Arreridj, Algeria*

^c*Computer Science Department, Setif 1 University, Setif, Algeria*

Abstract

Several studies have invested in the Machine Learning classifiers to protect plants from diseases by processing Leaves Images. Most of the proposed classifiers are trained and evaluated with a small dataset, focusing on the extraction of hand-crafted image features to classify the leaves. In this study, we have used a large dataset compared to the state of the art method. Here, the dataset contains 14828 images of tomato leaves infected by nine diseases. To train our classifier, we have introduced the convolutional neural network (CNN) as a learning algorithm that leads to the direct use of images and to avoid hand-crafted features. To analyze the proposed deep model, we have used visualization methods to understand symptoms and to localize disease regions in the leaf. The obtained results are encouraging, reaching 99.18% of accuracy, which outperforms dramatically shallow models, and they can be used as a practical tool for farmers to protect tomato against disease.

Keywords: Tomato diseases classification, Deep Learning, Convolutional Neural Network, Symptoms Visualization

1. Introduction

Tomato occupies a prominent place in the Algerian agricultural economy (Nechadi et al., 2002). In fact, tomato harvested area is 22497 hectares

*Corresponding author

Email addresses: m_brahimi@esi.dz (Mohammed Brahimi), boukhalk@gmail.com (Kamel Boukhalifa), moussaoui.abdel@gmail.com (Abdelouahab Moussaoui)

4 with production quantity of 975075 tonnes (FAOSTAT, 2016). As regards
 5 the production quantity, tomato is ranked seventh in Algerian crops after
 6 potatoes, wheat, watermelons, barley, and onions. Also, Algeria is ranked
 7 19th among tomato producing countries (FAOSTAT, 2016). Despite this
 8 position of tomato in Algerian agriculture, the yield of tomato is 433,424
 9 quintals per hectare (FAOSTAT, 2016). This yield is very low compared to
 10 neighboring countries of Algeria like Morocco which has 922.745 quintals per
 11 hectare (FAOSTAT, 2016). Diseases are often considered as one of the major
 12 limiting factors in the cultivation of tomato. Tomato crops are highly affected
 13 by diseases, which decrease the yield and cause dramatic losses in agriculture
 14 economy (Hanssen and Lapidot, 2012). For instance, early blight represents
 15 one of the most common diseases in the world and can cause a significant
 16 decreasing of yields and many lesions in fruits (Blancard, 2012). Likewise,
 17 late blight causes serious damages in crops, and it can be very destructive in
 18 areas with humid climate (Blancard, 2012). Protecting tomato from diseases
 19 is crucial for improving the quantity and the quality of the crops. Thus,
 20 providing an early detection and a good identification of diseases are mainly
 21 helpful in choosing treatment against big damages (Al Hiary et al., 2011).
 22 For good identification, the practitioners must have an academic training and
 23 extensive knowledge of various disciplines together, along with experiences
 24 (Blancard, 2012), as well as a large number and diversity of the causes of
 25 diseases (Blancard, 2012). Thus, a good practitioner needs to be informed
 26 of all symptoms and signs caused by the disease. Furthermore, farmers must
 27 monitor plants continuously, showing a tedious task. Hence, large efforts have
 28 been devoted to coming up with a strategy that automates the classification
 29 of disease, using leaves images. The aim of these approaches is to early detect
 30 the disease, focusing on the appropriate treatment in due course (Al Hiary
 31 et al., 2011; Akhtar et al., 2013; Mokhtar et al., 2015; Sannakki et al., 2013).
 32 Also, these approaches are based on Machine Learning and Computer Vision
 33 to build a classifier of diseases using the image of just the leaf. In regards
 34 to building such classifier, features are extracted from images to facilitate
 35 the work of the classifier. The experts create these features, known as hand-
 36 crafted features to extract relevant information from images. Hence, the
 37 learning system is not fully automated because of the dependency on the
 38 hand-crafted features (Breitenreiter et al., 2015). After the extraction of
 39 features phase, the classifier is trained using labeled images. Unfortunately,
 40 collecting these labeled data is very expensive, because the labeling is done
 41 manually by an expert, who can examine the image and to label it with

the appropriate disease. Therefore, most of the previous studies include the use of small labeled datasets (10 to 800 images) to train and evaluate the classification accuracy (Al Hiary et al., 2011; Akhtar et al., 2013; Mokhtar et al., 2015; Dandawate and Kokare, 2015).

Deep Learning is a new trend in Machine Learning and it outperforms state of the art in various research fields, such as Computer Vision, Drug Design and Bioinformatics (Al Hiary et al., 2011). The advantage of Deep Learning is the ability to exploit directly raw data without using the hand-crafted features (Al Hiary et al., 2011; Mokhtar et al., 2015). Recently, Deep Learning provided a good result in academic and industry world, according to the two following reasons (Al Hiary et al., 2011). Firstly, large amounts of data are generated every day. Hence, these data can be used in order to train a deep model. Secondly, the power of computing provided by Graphics Processing Unit (GPU) and High Performance Computing (HPC) makes possible the training of deep models and leveraging the parallelism of computing.

The aim of the present study is to introduce Deep Learning as an approach for classifying plant diseases, focusing on images of leaves. This study presents two main contributions in plant disease classification:

1. Improvement in classification pipeline using deep models: deep models have good results in classification and outperform the used models in machine learning. Besides, deep models have the ability to use directly raw data without features engineering. Moreover, deep models offer the possibility of transfer learning from another task by using already trained models on larger datasets.
2. Detection of disease symptoms in the infected leaf: the localization of infected region in an infected leaf helps the users of the model by giving them information about the disease. Also, this biological information is extracted without the intervention of agriculture experts.

2. Related works

To describe the state of the art, we need to give the general architecture of disease classification systems that based on image processing. These systems contain three phases: image pre-processing, features extraction, and classification (Akhtar et al., 2013).

Pre-processing : The image of a leaf is prepared using some operations. For instance, color space conversion from RGB to another space is used

for reducing the dependence on device (Prasad et al., 2016; Mokhtar et al., 2015; Semary et al., 2015; Al Hiary et al., 2011; Sannakki et al., 2013). Also, many works try to remove the background of leaf, focusing the analysis on the leaf (Mokhtar et al., 2015; Dandawate and Kokare, 2015; Sannakki et al., 2013; Semary et al., 2015; Le et al., 2015). Unfortunately, removing background is difficult, and sometimes need the intervention of the user, leading to the decreasing of the systems automation (Le et al., 2015).

Features Extraction : features proposed by experts (hand-crafted features) are extracted from the image for constructing features vectors. For example, color moments are used to extract color statistics (Semary et al., 2015), in which gabor transform (GT) and wavelet transform (WT) are combined (GWT) in (Prasad et al., 2016) for extracting multiscale features and tolerates the resolutions changes (Prasad et al., 2016). As texture features, gray level cooccurrence matrix (GLCM) based on the gray level is used in many previous works (Prasad et al., 2016; Mokhtar et al., 2015; Semary et al., 2015; Xie et al., 2015; Xie and He, 2016). GLCM is 256*256 matrix where each position in matrix counts the co-occurrences of line color and column color in the analyzed image (Dandawate and Kokare, 2015), scale invariant feature transform (SIFT) is used to analyze the shape features of leaves.

Classification : The last phase determines which disease is present in the leaf using classification model. This model should be trained using learning algorithms and examples with a known disease (labeled examples). The SVM (Mokhtar et al., 2015; Semary et al., 2015; Dandawate and Kokare, 2015; Raza et al., 2015; M Schikora, 2014), KNN (Prasad et al., 2016; Xie and He, 2016) and ANN (Al Hiary et al., 2011; Sannakki et al., 2013; Xie et al., 2015; M Schikora, 2014) represent the most used learning algorithms in the literature. The support vector machine (SVM) algorithm maximizes the margin between classes in linearly separable cases. Nevertheless, in the case where examples are not linearly separable, kernel trick is used to transform examples to another space where they will be linearly separable. The k-nearest neighbors algorithm (KNN) classifies an image by voting between the K closest examples in the features of vector representation. The artificial neural network (ANN) is a model organized in layers, in which each

Table 1: Studies in plant diseases classification

<i>Study</i>	<i>Features</i>	<i>Classifier</i>	<i>Dataset</i>	<i>Accuracy</i>
(Prasad et al. 2016)	GWT GLCM	KNN	297 images 5 diseases	93.00
(Mokhtar et al. 2015)	GLCM	SVM	800 images 2 diseases	99.80
(Semary et al. 2015)	Color moments GLCM Wavelets decomposition	SVM	177 images 12 diseases	92.00
(Dandawate et al. 2015)	SIFT	SVM	120 images 2 diseases	93.79
(Raza et al. 2015)	Local statistics Global statistics Depth information	SVM	71 plants 2 diseases	89.93

113 layer is connected to the next one starting from the input to output.
114 ANN represents the old version of deep learning algorithms used in this
115 paper.

116 Table 1 shows the results of some recent studies in plant diseases classi-
117 fication. These studies have two principal issues. The first one is the small
118 number of examples in datasets (between 71 and 800). The second one is the
119 performance of classification measured by accuracy (Table 1) that is located
120 between 89.93 % and 93.79 % of the most cases. Although the accuracy
121 of Mokhtar in (Mokhtar et al., 2015) is 99.83 %, the used dataset contains
122 only two classes (healthy and infected). In other words, the proposed sys-
123 tem (Mokhtar et al., 2015) can detect the presence of disease in leaf, with
124 no information about the type of this disease. In summary, the previous
125 works are based on several pre-processing operations and hand-crafted fea-
126 tures. The choice of pre-processing operations and features is difficult and
127 time-consuming which makes classification system not fully automated. In
128 the present paper, we propose to use deep learning and specifically Convo-
129 lutional Neural Network (CNN) as an alternative approach for building a
130 model of disease classification. Thus, features are constructed in a fully au-
131 tomated way and learned from the data in the training phase. Moreover, we

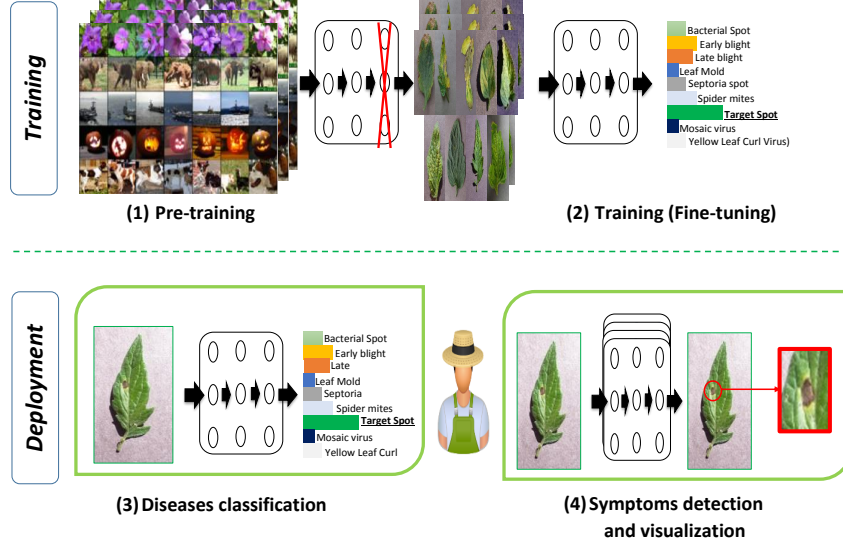


Figure 1: Overview of the proposed deep learning methodology

do not need a pre-processing phase to prepare the image and we use the raw image directly. To confirm the superiority of deep models against state of the art, we have used large dataset containing 14828 images and nine diseases of tomato.

In the rest of the paper, we present our methodology, followed by details and requirements of our approach. Finally, we give experimental results, showing the performance evaluation of our proposed approach based on deep models.

3. Methodology

The proposed approach, illustrated in Figure 1 contains four components as following:

1. Pre-taining phase: in this phase we train deep architectures on a large dataset like ImageNet using powerful machines. The objective of this phase is the initialization of network weights for the next phase.
2. Training (fine-tuning): we fine-tune output network from the first phase using images of tomato leaves affected by nine diseases. At this stage, we use a small dataset compared to the pre-training dataset. Also,

149 we replace the output layer of pertained networks (ImageNet contains
150 1000 classes) by new output layer having nine classes (nine diseases of
151 tomato).

152 Afterword, the developed deep model is deployed to users machines (Com-
153 puters, mobiles). The deployed models can be used in two modes:

- 154 3. Disease classification: in this mode, the user takes a picture of a leaf
155 and use the produced deep model to determine which disease affect
156 tomato plant.
- 157 4. Symptoms detection and visualization: after the disease classification
158 in the precedent mode, the user can visualize the regions that char-
159 acterize the disease. This symptoms visualization method helps the
160 inexperienced user by giving them more information about the disease
161 mechanism. Also, symptoms visualization give the user a tool to esti-
162 mate the spread of disease in the other tomato plants.

163 To assess the effectiveness of these contributions, we compare the re-
164 sults of deep models with results of shallow models based on hand-crafted
165 features. In this comparison, we have used two famous deep models in litera-
166 ture (AlexNet (Alex et al., 2012) and GoogleNet (Szegedy et al., 2015)). Our
167 choice of using existing deep architectures is motivated by their spectacular
168 results in computer vision challenge ImageNet. Furthermore, these models
169 allow us to fine-tune and transfer learning from a task where we have a large
170 labeled dataset to a particular task as disease classification. Finally, to eval-
171 uate the quality of symptoms detection, we compare the results regions with
172 ground truth extracted from agriculture books and expert descriptions. To
173 summarize, the present study includes the following three main points:

- 174 1. Deep models without features extraction vs. shallow models with hand-
175 crafted features: Comparison between performances of deep models
176 against shallow models, to measure the performance of the two ap-
177 proaches. This comparison is performed for validating the choice of
178 deep models as an alternative to shallow models. Also, this compari-
179 son evaluates the impact of automated features extractions embedded
180 in deep approach.
- 181 2. Deep Models with pre-training vs. Deep Models without pre-training:
182 Evaluation of the performance of pre-trained models and compare them
183 against those trained from scratch starting from a random configuration

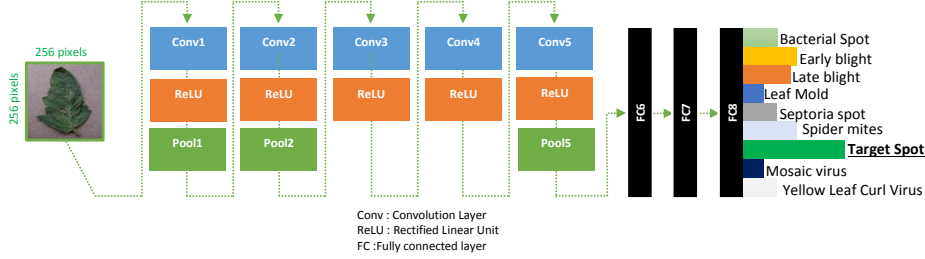


Figure 2: AlexNet for Tomato Disease Classification

of network weights. This comparison aims to exhibit the influence of transfer learning from another task to disease classifications.

3. Symptoms visualization: we use visualization methods after training stage to extract biological knowledge from models trained in a fully automated way. This automatic knowledge extraction can help users in understanding the plant disease and their symptoms. In other words, machine learning models of the hand-crafted features incorporate expert knowledge to facilitate classification. However, deep models can give biological knowledge extracted directly from raw data without expert helping.

4. Experiments

4.1. Pre-trained Models

For building a classifier for plant diseases without hand-crafted features, we use deep learning approach, especially convolutional neural networks (CNN). CNNs are directly trained using raw images. As a result, the proposed system learns to extract features from data without the intervention of human in features engineering. In other words, the training of the classifier includes automated features constructions.

We have used two CNN models (AlexNet (Alex et al., 2012) and GoogleNet (Szegedy et al., 2015)). These models are used previously with success in computer vision challenges like ImageNet. Our objective is to take advantage of these architectures to improve results in the diagnosis and detection of plant diseases. Figure 2 shows the architecture of AlexNet and the different stacked layers from the input image to the output probabilities of each disease. Input images of the network are resized to 256*256 pixels. Output results represent the probabilities of each disease. In our case, we have nine

Table 2: Workstation Specifications

Machine Type	Desktop workstations
RAM	128 GB
CPU	Intel Xeon Processor 2 CPU 2.00 GHz, 24 cores
GPU	Quadro K 5000 4GB and 1536 cores

diseases (Figure 1). To implement a convolutional neural network, we have used a dedicated hardware and software to accelerate the training.

4.2. Workstation Specifications and Deep learning framework

We have used workstation having the specifications that are summarized in Table 2. Parallelize the deep learning is the key to accelerating the training. Therefore, Graphics Processing Unit (GPU) is crucial for minimizing the learning time to few hours. Also, if the memory of GPU is small, we cannot use a large number of examples in each epoch of learning.

As deep learning framework, we have used DIGITS framework proposed by NVidia. DIGITS is a friendly environment that can help researchers to test deep networks for computer vision problems. In DIGITS, many deep learning frameworks are presented to design and train networks. In our study, we use the framework Caffe proposed by Breckly. Caffe offers more options than other frameworks according to the comparative study (Bahrampour et al., 2015). The default learning hyperparameters of Digits are used in all our experiments about CNN models.

4.3. Dataset

We have used a dataset recently published in (Goodfellow et al., 2016). This dataset is open access repository of images published online at website www.PlantVillage.org and contains more than 50 000 images of leaves. From this dataset, we extract only images of tomatos leaves. The Table 3 and Figure 3 give a summary about our dataset. The total number of images in our dataset is 14828 distributed in nine classes of diseases.

Table 3: Dataset Summary

Classes (Diseases)	# images
Tomato Yellow Leaf Curl Virus	4032
Tomato mosaic virus	325
Target Spot	1356
Spider mites	1628
Septoria spot	1723
Leaf Mold	904
Lateblight	1781
Earlyblight	952
Bacterial Spot	2127
Total	14828

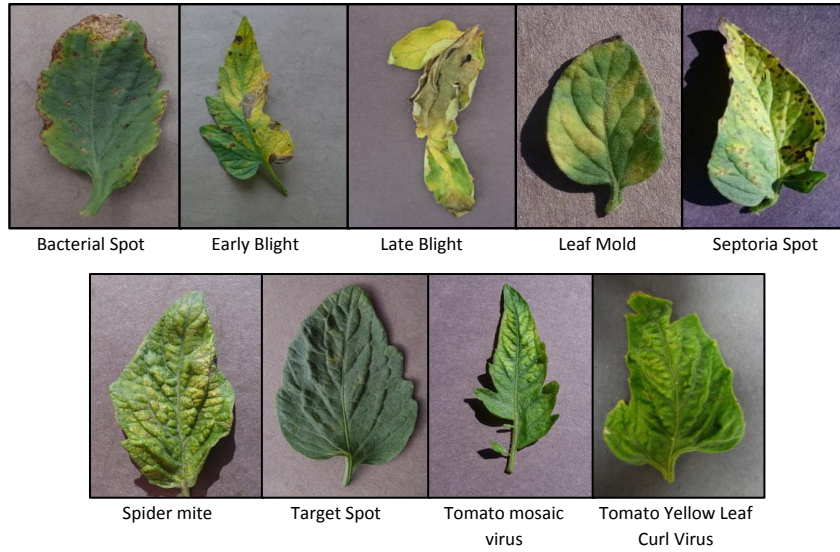


Figure 3: Extracted images from Dataset

Table 4: Measures of classification performance

<i>Measure</i>	<i>Formula</i>
<i>Accuracy</i>	$\frac{\sum_{i=1}^N \mathbb{1}_{\hat{Y}_i=Y_i}}{N}$
<i>Precision_i</i>	$\frac{\sum_{j=1}^N \mathbb{1}_{\hat{Y}_j=Y_j} * \mathbb{1}_{\hat{Y}_j=i}}{\sum_{j=1}^N \mathbb{1}_{\hat{Y}_j=i}}$
<i>MacroPrecision</i>	$\frac{\sum_{i=1}^C \textit{Precision}_i}{C}$
<i>Recall_i</i>	$\frac{\sum_{j=1}^N \mathbb{1}_{\hat{Y}_j=Y_j} * \mathbb{1}_{\hat{Y}_j=i}}{\sum_{j=1}^N \mathbb{1}_{Y_j=i}}$
<i>MacroRecall</i>	$\frac{\sum_{i=1}^C \textit{Recall}_i}{C}$
<i>F_i</i>	$2 \cdot \frac{\textit{Precision}_i \cdot \textit{Recall}_i}{\textit{Precision}_i + \textit{Recall}_i}$
<i>MacroF</i>	$\frac{\sum_{i=1}^C F_i}{C}$

233 5. Results

234 To measure the performance of models, we use cross-validation with five
 235 folds. We use four folds for training, and the remaining fold is used for
 236 the test. Furthermore, Models like SVM and Random Forest need tuning
 237 of hyper-parameters. For this reason, we have used grid search in hyper-
 238 parameter spaces to find the combination of parameters that maximize the
 239 performance in validation fold (one fold from the four folds of training). The
 240 Table 4 gives formulas of all metrics used in our experimental tests Where :

- 241 • $Y_i, (\hat{Y}_i)$: represents the index of real (predicted) classe of an image.
- 242 • N : number of images.
- 243 • C : number of classes (diseases).
- 244 • $\mathbb{1}_{condition}$: indicator funciton is one if the *condition* is satisfied, other-
 245 wise it is zero.

246 5.1. Shallow Models & Hand-crafted Features Vs Deep Models

247 To show the effectiveness of deep learning approach, we compare the re-
 248 sults of CNN with the state of the art approach shown in Figure 4. The

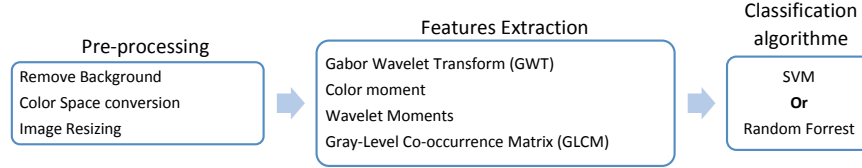


Figure 4: State of art approach in plant disease classification using leaf image

different approaches used in the other works are based on hand-crafted features, precisely texture and color features (Prasad et al., 2016; Mokhtar et al., 2015; Semary et al., 2015; Al Hiary et al., 2011; Sannakki et al., 2013). To compare the deep model with these existing studies, we have developed a classification model according to the indicated approach in Figure 4. Herein, the model uses three phases (1) Pre-processing (2) Features Extraction (3) Classification. In the preprocessing step, the background is removed and replaced with a black color, making the analysis focuses on the leaf. Also, color space conversion is used to avoid the dependency of RGB to camera (Prasad et al., 2016; Mokhtar et al., 2015; Semary et al., 2015; Al Hiary et al., 2011; Sannakki et al., 2013). Finally, we normalize the size of all images to 256*256. In Features Extraction, hand-crafted features used in previous works are combined to improve results. These features are extracted from color and texture (Gabor Wavelet Transform GWT (Prasad et al., 2016), Color Moment (Semary et al., 2015), Wavelet Moment (Semary et al., 2015), GrayLevel Co-occurrence Matrix (GLCM)(Mokhtar et al., 2015; Semary et al., 2015)). As classification algorithm, two shallow models are used: support vector machine (SVM) (Mokhtar et al., 2015; Semary et al., 2015; Dandawate and Kokare, 2015; Raza et al., 2015) and Random Forrest (RF). We have ignored classifiers having accuracy less than 90% like decision tree (DT) and k-nearest neighbors (KNN).

We notice from Table 5 and Figure 5, showing that deep models (CNN) dramatically overcome shallow models combined with hand-crafted features (SVM, Random Forrest). Indeed, the best accuracy in shallow models is 95.476 against 99.185 in deep learning. Also, Macro F1 measure that represents the harmonic mean of precision and recall is too high in deep models (98.518) comparing with macro F in shallow models (94.185). These results

show the power of deep models in constructing good features without human expert intervention. In other words, the features constructed automatically have a better performance than hand-crafted features.

Table 5: Experimental results

		Without Pre-training	With Pre-training
Deep Models	AlexNet	Accuracy	97.354 ± 0.290
		Macro Precision	96.566 ± 0.388
		Macro Recall	96.266 ± 0.414
		Macro F	96.368 ± 0.262
	GoogleNet	Accuracy	97.711 ± 0.149
		Macro Precision	96.989 ± 0.506
		Macro Recall	96.783 ± 0.350
		Macro F	96.582 ± 0.161
Shallow models	SVM	Accuracy	94.538 ± 0.301
		Macro Precision	93.317 ± 0.752
		Macro Recall	92.917 ± 0.461
		Macro F	93.067 ± 0.303
	Random Forrest	Accuracy	95.467 ± 0.004
		Macro Precision	94.628 ± 0.004
		Macro Recall	93.808 ± 0.006
		Macro F	94.185 ± 0.005

5.2. AlexNet Vs GoogleNet

The results of Table 6 show that GoogleNet network overcomes the result of AlexNet in plant disease classification. Although the size of GoogleNet is small (36.6 MB) comparing to AlexNet (201 MB) size, but results of GoogleNet are more accurate than AlexNet results. The accuracy of pertained GoogleNet is 99.185 and macro F1 is 98.518 while AlexNet has an ac-

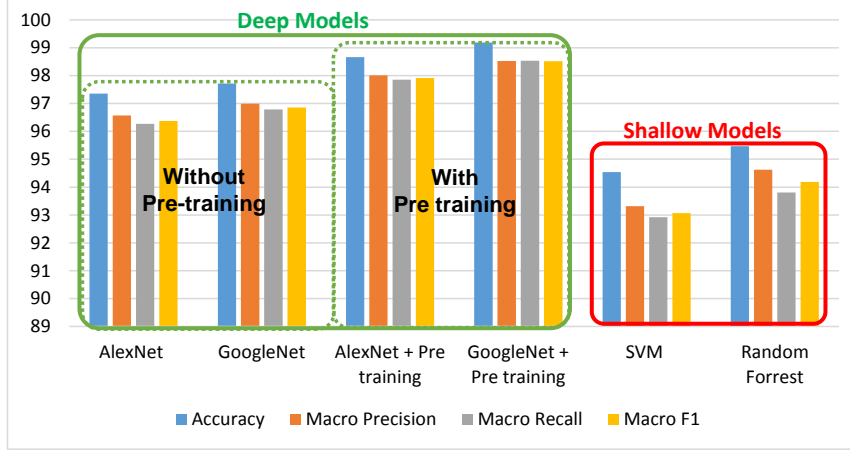


Figure 5: comparison between Deep models and shallow models

285 curacy equal to 98.660 and macro F1 = 97.911. This superiority of GoogleNet
 286 is due to the new architecture that used in this network to increase nonlin-
 287 earity without an explosion of the number of weights (Szegedy et al., 2015).
 288 GoogleNet uses inception module that inspired from the architecture Net-
 289 work in Network (NIN) proposed in (Lin et al., 2014). Inception module
 290 uses convolutions with filters one by one (1*1), leading to decreases in the
 291 depth of input volume. Moreover, the number of weights decreases, without
 292 losing much information. In other words, convolution using filters of size 1*1
 293 plays the role of filtering information along the layers and this reduces the
 294 size of the network.

295 5.3. Deep Models with pre-training Vs Deep Models without pre-training

296 Also from our experiments (Table 5), we observe that fine tuning of per-
 297 tained models improves the results of CNN. Indeed, fine tuning pre-tained
 298 models improve the accuracy of GoogleNet from 97.711 to 99.185 and accu-
 299 racy of AlexNet from 97.354 to 98.660. The effectiveness of fine tuning is
 300 explained by the ability of the network to reuse and transfer learned features
 301 from one problem to another. The network learns features in one domain
 302 or a large dataset and, afterward, reuses these features in another task with
 303 a small modification, especially in the last layers. Also, fine-tuning is very
 304 suitable in cases when the number of training examples is small, and we can-
 305 not train deep models from scratch because deep models are very greedy to

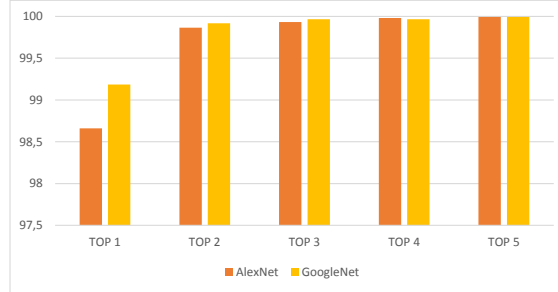


Figure 6: Top accuracies of deep models (k= 1, 2, 3, 4, 5)

306 labeled examples. In this case, we can use model trained on a large dataset in
 307 the scale of ImageNet and train it in a dataset that contains a small number
 308 of examples. Finally, fine-tuning helps to train deep models in machines that
 309 do not have big memory in GPU. For example, our machine used in the ex-
 310 periments have only four GB of GPU memory and using pre-trained models
 311 in ImageNet we have reached the accuracy of 99.185 with small batches in
 312 training.

313 5.4. Top k Accuracy

314 Top K accuracy evaluates the power of classifier in ordering predictions.
 315 Top K accuracy divides the number of images having the correct class in top
 316 k list of predicted classes by the number of all images. When good classifier
 317 cannot find the right class of image, it is useful to put the right class in
 318 a good position in the predictions list (sorted according to probabilities).
 319 Results presented in Figure 6 shows that Top 5 Accuracy is close to 100. In
 320 other words, the user of the classifier can exclude four diseases from nine and
 321 focuses his efforts only on five diseases with an empirical confidence equals to
 322 one. Also, the risk is not very important when the user attention is focused
 323 on the two first predictions because Top 2 accuracy is 99.864 in AlexNet and
 324 99.918 in GoogleNet.

325 6. Symptoms and disease regions detections using CNN

326 The drawback of using old neural networks is the difficulty of interpreting
 327 how it works. In deep learning, many studies proposed the visualization of
 328 learned features, in order to explain how classifier gives the final result and

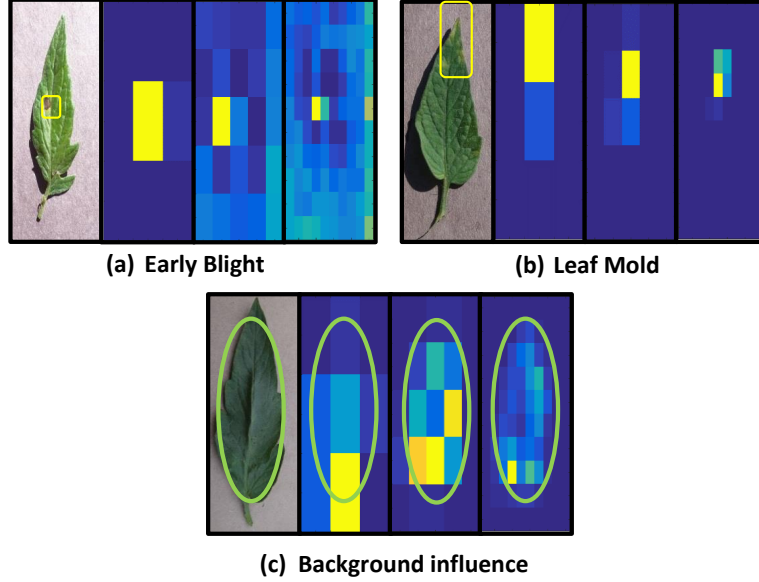


Figure 7: Symptoms visualization of CNN

329 how features are constructed (Yosinski et al., 2015; Zeiler and Fergus, 2014;
 330 Grün et al., 2016). In our work, the visualization of features helps users to
 331 understand the plant disease and subsequent its symptoms. We have used
 332 the occlusion method as previously described in (Zeiler and Fergus, 2014)
 333 to understand how network classifies the disease. We hide some part of
 334 the image and we calculate predictions of this image. If the hidden part is
 335 important, then this occlusion will decrease the probability of the correct
 336 class. However, if the occlusion does not decrease the probability of correct
 337 class, then this part does not participate in the classifier decision. Herein, we
 338 slide black rectangle over an image, and afterward, we run CNN to calculate
 339 the probability of correct classes of current image $PC_{i,j}$. The indexes (i, j)
 340 indicates the occlusion rectangle position in the image. Then, we visualize
 341 the negative log likelihood $(-\log(P_{i,j}))$ of these probabilities using heat map.
 342 We can control the resolution of this heat map by changing the number of
 343 rectangles. For example, in Figure 7 there are three used resolutions 3×3 , 5×5
 344 and 10×10 . The intuition behind this technique is as follows: if the probability
 345 $PC_{i,j}$ corresponding to occlusion rectangle (i, j) is low, then this region is
 346 important. If this region is important it will be yellow in heat map because
 347 the value of $-\log(P_{i,j})$ will be high. On the other hand, if $PC_{i,j}$ is high then

348 the occluded region does not affect the correctness of network result. In this
 349 case, this region is not important and the heat map, becomes blue because
 350 the value of $-\log(P_{i,j})$ is small. This visualization can be used by agriculture
 351 expert to identify new symptoms from analyzing heat maps. Observing the
 352 obtained heat map, the user can understand model behavior, and he can
 353 also gain some insights into disease and infected regions. For instance, in
 354 Figure 7(a), the gray concentric ring is used by the classifier as a symptom
 355 of Early Blight disease. To be sure, we have compared this observation
 356 with the expert-defined symptoms (Koike et al., 2007). Indeed, as reported
 357 by Koike et al. (Koike et al., 2007), gray concentric rings are described
 358 as distinguishing characteristic of Early Blight. Also, changing resolutions
 359 in Figure 7(a) helps us to locate the region of disease accurately and focus
 360 on a small region. In Figure 7(b), a yellow region on the top of the leaf is
 361 activated in the heat map. To be sure that this region represents the symptom
 362 of leaf mold, we have consulted the website of Maine Organic Farmers and
 363 Gardeners Association and the conference of fall 2012. In this specialized
 364 association conference, they describe the symptoms of leaf mold as follow:
 365 "It shows up as yellowing on top of the leaf and, on the underside of the
 366 leaf, as a little speck of gray fuzz under each spot". Therefore, the automatic
 367 detection of leaf mold symptom is identical to an expert description. Finally,
 368 all heat maps seen in (Figure 7(C)), show that the background of leaf does not
 369 affect the results (blue color). The independence of results to the background
 370 is an excellent property of CNN. Indeed, the model focuses on leaf and ignores
 371 its background without complicated algorithm in preprocessing for removing
 372 this background.

373 7. Conclusion and Further Research

374 In this study, we have proposed deep models approach to build a classi-
 375 fier for disease classification. Our result shows that deep models and CNN
 376 dramatically overcame the state of the art of this field. In addition, our ex-
 377 periments demonstrate the benefit of using pre-trained model, especially if
 378 the number of examples used in training is not very large, unlike that used
 379 in disease classification context. In this study, we have also proposed the
 380 using of occlusion techniques to localize the disease regions, helping human
 381 to understand the disease.

382 In the future, our objective is to reduce the computation and the size
 383 of deep models for small machines like mobiles. Besides, visualization of

384 features is a hot topic in deep learning, and it could be used to understand
385 plant diseases.

386 References

- 387 S. Nechadi, F. Benddine, A. Moumen, M. Kheddam, Etat des maladies
388 virales de la tomate et strat{é}gie de lutte en Alg{é}rie, EPPO Bulletin
389 32 (2002) 21–24.
- 390 FAOSTAT, Food and Agriculture Organization of the United Nations
391 Database, 2016.
- 392 I. M. Hanssen, M. Lapidot, Major Tomato Viruses in the Mediterranean
393 Basin, in: G. Loebenstein, H. Lecoq (Eds.), *Advances in Virus Research*,
394 volume 84 of *Advances in Virus Research*, Academic Press, 2012, pp. 31–
395 66.
- 396 D. Blancard, *Tomato Diseases*, Academic Press The Netherlands, 2012.
- 397 H. Al Hiary, S. Bani Ahmad, M. Reyalat, M. Braik, Z. ALRahamneh, Fast
398 and Accurate Detection and Classification of Plant Diseases, *International*
399 *Journal of Computer Applications* 17 (2011) 31–38.
- 400 A. Akhtar, A. Khanum, S. a. Khan, A. Shaukat, Automated Plant Disease
401 Analysis (APDA): Performance Comparison of Machine Learning Tech-
402 niques, in: 2013 11th International Conference on Frontiers of Information
403 Technology, {IEEE} Computer Society, 2013, pp. 60–65.
- 404 U. Mokhtar, N. El-Bendary, A. E. Hassenian, E. Emary, M. A. Mahmoud,
405 H. Hefny, M. F. Tolba, U. Mokhtar, A. E. Hassenian, E. Emary, M. A.
406 Mahmoud, SVM-Based detection of tomato leaves diseases, in: *Advances*
407 *in Intelligent Systems and Computing*, volume 323, Springer, 2015, pp.
408 641–652.
- 409 S. S. Sannakki, V. S. Rajpurohit, V. B. Nargund, P. Kulkarni, Diagnosis and
410 Classification of Grape Leaf Diseases using Neural Networks, in: *Com-*
411 *puting, Communications and Networking Technologies (ICCCNT)*, 2013
412 Fourth International Conference on, IEEE, pp. 3–7.
- 413 A. Breitenreiter, H. Poppinga, T. U. Berlin, F. N. Technik, Deep Learning,
414 *Nature* 521 (2015) 2015.

415 Y. Dandawate, R. Kokare, An automated approach for classification of plant
416 diseases towards development of futuristic Decision Support System in
417 Indian perspective, in: 2015 International Conference on Advances in
418 Computing, Communications and Informatics, ICACCI 2015, IEEE, pp.
419 794–799.

420 S. Prasad, S. K. Peddoju, D. Ghosh, Multi-resolution mobile vision system
421 for plant leaf disease diagnosis, *Signal, Image and Video Processing* 10
422 (2016) 379–388.

423 N. A. Semaary, A. Tharwat, E. Elhariri, A. E. Hassanien, Fruit-based tomato
424 grading system using features fusion and support vector machine, in: *Ad-
425 vances in Intelligent Systems and Computing*, volume 323, Springer, 2015,
426 pp. 401–410.

427 T.-L. Le, N.-D. Duong, V.-T. Nguyen, H. Vu, Complex Background Leaf-
428 based Plant Identification Method Based on Interactive Segmentation and
429 Kernel Descriptor, in: *2nd International Workshop on Environmental Mul-
430 timedia. In Conjunction with ACM Conference on Multimedia Retrieval
431 (ICMR) 2015.*, ACM, pp. 3–8.

432 C. Xie, Y. Shao, X. Li, Y. He, Detection of early blight and late blight
433 diseases on tomato leaves using hyperspectral imaging., *Scientific reports*
434 5 (2015) 16564.

435 C. Xie, Y. He, Spectrum and Image Texture Features Analysis for Early
436 Blight Disease Detection on Eggplant Leaves, *Sensors* 16 (2016) 676.

437 S. E. A. Raza, G. Prince, J. P. Clarkson, N. M. Rajpoot, Automatic detection
438 of diseased tomato plants using thermal and stereo visible light images,
439 *PLoS ONE* 10 (2015) e0123262.

440 A. S. M Schikora, Image-Based Analysis to Study Plant Infection with Hu-
441 man Pathogens, *Computational and structural biotechnology journal* 12
442 (2014) 1–6.

443 K. Alex, I. Sutskever, G. E. Hinton, Imagenet classification with deep con-
444 volutional neural networks, in: *Neural Information Processing Systems
445 (NIPS)*, pp. 1097–1105.

- 446 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan,
447 V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Pro-
448 ceedings of the IEEE Computer Society Conference on Computer Vision
449 and Pattern Recognition, volume 07-12-June, pp. 1–9.
- 450 S. Bahrampour, N. Ramakrishnan, L. Schott, M. Shah, Comparative Study
451 of Caffe, Neon, Theano, and Torch for Deep Learning, ArXiv 2 (2015)
452 1–14.
- 453 I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
- 454 M. Lin, Q. Chen, S. Yan, Network In Network, arXiv preprint
455 arXiv:1312.4400 (2014).
- 456 J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding Neu-
457 ral Networks Through Deep Visualization, International Conference on
458 Machine Learning - Deep Learning Workshop 2015 (2015) 12.
- 459 M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Net-
460 works arXiv:1311.2901v3 [cs.CV] 28 Nov 2013, in: Computer VisionECCV
461 2014, volume 8689, Springer, pp. 818–833.
- 462 F. Grün, C. Rupprecht, N. Navab, F. Tombari, A Taxonomy and Library for
463 Visualizing Learned Features in Convolutional Neural Networks, Proceed-
464 ings of the Workshop on Visualization for Deep Learning at International
465 Conference on Machine Learning (ICML) 48 (2016).
- 466 S. T. Koike, P. Gladders, A. O. Paulus, Vegetable Diseases: A Color Hand-
467 book. Ed, 2007.