DEEP RESIDUAL LEARNING FOR TOMATO PLANT LEAF DISEASE IDENTIFICATION

[1]HABIBOLLAH AGH ATABAY

[1]Department of Computer, Gonbad Kavous University, Gonbad Kavous, Iran

E-mail: [1]atabay@gonbad.ac.ir

ABSTRACT

Deep Learning for plant leaf analysis has been recently studied in various works. In most cases, Transfer Learning has been utilized, where the weights of networks, which are stored in the pre-trained models, are fine-tuned to use in the considered task. In this paper, Convolutional Neural Networks (CNNs), are employed to classify tomato plant leaf images based on the visible effects of diseases. In addition to Transfer Learning as an effective approach, training a CNN from scratch using the Deep Residual Learning method, is experimented. To do that, an architecture of CNN is proposed and applied to a subset of the PlantVillage dataset, including tomato plant leaf images. The results indicate that the suggested architecture outperforms VGG models, pre-trained on the ImageNet dataset, in both accuracy and the time required for re-training, and it can be used with a regular PC without any extra hardware required. A common feature visualization and verification technique is also applied to the results and further discussions are made to imply the importance of background pixels surrounding the leaves.

Keywords: *Deep Learning, Convolutional Neural Network, Plant Leaf Disease, Tomato Disease*

1. INTRODUCTION

Plants are among the important elements of human life. Existing diseases disrupt the growth of plants and cause economic, social and ecological losses. Most of them, produce some form of manifestation in the visible portions of plants. Correct recognition of diseases when they first appear, is a crucial step for effective disease management. In most cases, human experts identify diseases visually, who may be efficient in the recognition and quantification of diseases, but, they are engaged with some difficulties that may harm their efforts. In this context, diagnosing diseases in an exact and timely way is of the great importance.

There are various types of diseases that harm the quality of tomatoes. Some of them have visible symptoms on the plant leaves such as Bacterial Spot [1]. The disease starts by touching the leaves and yields of the tomato plants and continues, resulting in complete defoliation and sun scalded fruits. As the disease extends leaves and parts appear as they were slightly burned, foliage turns yellowish and dies, with severe defoliation exposing fruits and stems. Other types of tomato diseases that affect leaves, include Early Blight, Late Blight, Septoria leaf spot and etc.

Hand-engineered features, including SIFT [2], HoG [3], SURF [4], have been the foundation of the traditional methods for image classification tasks. These features, after extraction from the images, are fed into a learning algorithm like support vector machines (SVM). The performance of these approaches heavily depends on the selected features. In disease recognition, these approaches have been applied for classification of tomato powdery mildew against healthy leaves using thermal and stereo images [5], detection of tomato yellow leaf curl virus by using a set of classic feature extraction steps, classified by SVM pipeline [6], recognition of greenhouse tomato disease [7] and etc. Smart phone based applications for shape and disease identification in plant leaves have been developed [8, 9]. The use of machine learning on plant leaf analysis has been discussed in [10].

Engineering and selecting efficient features is a complex and tedious process which needs to be revisited every time the existing problem or its associated dataset, changes considerably. Learning robust and invariant representation has been a long-standing goal in computer vision. Features learned by deep neural networks (DNNs) compared to, the hand-crafted visual features, have been proven more capable of capturing abstract concepts invariant to various phenomenon in visual world [11, 12].

In the recent past, Convolutional Neural Networks (CNNs), a specific type of DNNs, have emerged as a powerful framework for feature representation and recognition for a variety of image domains [13]. CNNs have been studied and applied in the field of computer vision for a long time. More than a decade ago, LeCun et al. [14] trained a multilayer neural networks with the back-propagation algorithm and the gradient learning technique, and demonstrated its effectiveness on the handwritten digit recognition. Deep learning has provide good generalization power in vision applications. In 2012, Krizhevsky et al. [11] achieved a breakthrough by outperforming the existing handcrafted features on Large Scale Visual Recognition Challenge (ILSVRC) [15]. The task includes recognition of 1000 object classes which was a very difficult problem to solve with the approaches of the time. Since 2012, CNNs have drawn a resurgence of attention in various tasks such as image classification [11, 16], semantic segmentation [17], object recognition [18], video analysis [19], etc. Recently the networks are going deeper from a depth of sixteen [41] to thirty [16] and with the Deep Residual Learning methodology [20] this is extremely facilitated even to 152 layers which won the first place in the ILSVRC 2015 classification competition.

One of the applications, which CNNs are recently introduced in, is plant leaf image recognition. In [21] deep learning was used to identify type of plants based on their leaf vein patterns. They classify three legume species, including white bean, red bean and soybean using CNNs having up to 6 layers. In [22], the well-known CNN architectures, AlexNet [11], GoogLeNet [16], and VGGNet [12], were experimented to identify the plant species captured in photographs. Using the plant task datasets of LifeCLEF 2015 [23] augmented by rotation, translation, reflection, and scaling, they applied Transfer Learning [24] to fine-tune these pre-trained models. Furthermore, the parameters of the networks were adjusted and different classifiers fused to improve the overall performance.

Recently the deep CNNs have been used to diagnose crop leaf disease [25] where the classification of 26 diseases in 14 crop species in 54,306 images of PlantVillage dataset [26], using two popular CNN architectures, AlexNet and GoogLeNet, was reported. In their work three different versions of the whole PlantVillage dataset were used; original colorful, gray-scale and background-removed. Similar to that work, but applied on the different dataset, is [27] where the authors used CaffeNet which is a slightly changed version of AllexNet, pre-trained on the ImageNet dataset [28] and available in Caffe [29] deep learning framework. The dataset used in the task contains numerous infected leaf images of various plants, downloaded from the Internet, including Peach, Apple, Grapevine and etc. CNNs also have been introduced in identification of rice diseases [30] where a dataset of 500 natural images of diseased and healthy rice leaves and stems was captured from rice experimental field. Another very recently published work in the literature is [31], where transfer learning of pre-trained models has been used to identify specifically tomato plant diseases of the PlantVillage dataset, similar to the experiments of this paper. In that paper, a common visualization method of deep features based on [32] was used, which is also adopted in this work.

In this paper, an application of Convolutional Neural Networks (CNNs) with a customized architecture, in the leaf disease recognition is experimented. Specifically a CNN architecture, based on the residual learning approach, is proposed to classify leaves of tomato plants, infected with various diseases, including Bacterial Spot, Early Blight, Late blight, Septoria leaf spot, Spider mites (Two-spotted spider mite), Tomato mosaic virus, Leaf Mold, Target Spot and Tomato Yellow Leaf Curl disease. The used dataset is a part of the PlantVillage image dataset [26], including 19742 images of tomato leaves. These images are resized to 280×180 pixels which is a fraction of the average size of all images. The size is chosen to minimize the amount of squash and stretch in images. 20% of the samples are firstly separated from the dataset and the remaining are divided into the training and validation sets. The accuracy on the validation set is tracked to choose the best state of the trained architecture and the parameters of the model. An issue with the residual networks is to prepare shortcut connections, in the cases of different sizes and number of filters. The appropriate decision is made in the design of the architecture.

Although the pre-trained deep models, applying on various problems have provided good results, fine-tuning them on a new dataset using a regular PC without GPU, requires a long time to complete. The questions that this paper considers include, how VGG networks perform in the task of identification of tomato plant diseases and is there exists a simpler network that works better in terms of accuracy and speed of training? The objective is to propose a CNN with a lighter and customized architecture, in comparison with the previously

Journal of Theoretical and Applied Information Technology
31st December 2017. Vol.95. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

known deep models, to be easier to retrain (in order to further learning from newly presented data) by means of a regular PC, without any special hardware requirements. The suggested CNN architecture is specifically designed for the mentioned size of colorful images. The accuracy of the proposed model is compared with the well-known VGG models pre-trained on ImageNet and fine-tuned on the considered dataset. The results show that, training such a network from scratch can provide the same accuracy as a large pre-trained models in less training time, thus the network can be rapidly retrained as the size of the dataset grows. The results are investigated with a visualization approach to answer the questions: Which part of the image is more effective to distinguish diseased leaf images? The visualizations reveals the importance of background pixels even if they contain simple patterns. The visualization method, used in this paper, is originally suggested by [32] and further applied on leaf image analysis in [21] and [31]. The main contributions of this paper are summarized as follows:

- Application of pre-trained VGG models fine-tuned on the dataset to classify tomato leaf images according the disease it infected.
- Proposing a simplified CNN architecture based on the residual learning the competes with VGG architectures in terms of classification accuracy and speed of training
- Analysis of results using an occlusion based visualization method

This paper is mainly inspired by [25] where well-known CNNs were used to classify diseases on all plant leaf images of the PlantVillage dataset. But in this paper, the dataset is restricted to the specific type of plants, tomato, and aims to suggest a classifier with a simpler architecture, specific to the dataset and re-trainable by means of a regular PC in CPU mode. Although a similar work has been published [31] very recently, the experiments of this paper were done separately, and this work has some extra experiments including training of a plain and residual CNN architectures from scratch, and some argues about the visualization results.

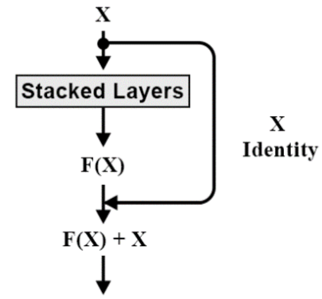

Figure 1. A building block of residual learning

The rest of the paper is organized as follows. Section 2 briefly describes CNNs and the proposed architectures. In section 3, the experimental results and discussions are presented. Section 4 concludes the paper.

2. CONVOLUTIONAL NEURAL NETWORKS

A convolutional neural network (CNN) is a type of deep neural networks (DNNs), inspired by the human visual system and designed for image understanding. A CNN is constructed of a series of alternatively stacked convolutional layers and spatial pooling layers. The convolutional layer is devised to extract feature maps by linear convolutional filters followed by nonlinear activation functions (e.g., rectifier, sigmoid, tanh, etc.). The spatial pooling layer is used to aggregate the local features extracted from spatially adjacent

Figure 2. Samples of the datasets

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

pixels. This layer is typically added to improve the robustness to slight deformations of objectives. These levels build deeply abstract representations of the input pattern: the first might be sensitive to edges, the second to corners and intersections and so on. These representations become more and more abstract and invariant as the pattern data goes deeper into the CNN. The output of the last stage is usually a vector (not an image) that is fed to a multi-layer perceptron (MLP) that produces the final output of the network, usually a class label.

Various CNN architectures have been proposed to be used in object recognition. Among them LeNet [14] and AlexNet [11] are considered as a baseline for various tasks. The former, was used to classify tiny images of handwritten digits (28×28 pixels) while the latter is more complicated and has been considered as a breakthrough in object recognition tasks in colorful medium sized images (256×256 pixels). AlexNet was designed in the context of ILSVRC 2012 [15] for the ImageNet dataset [28].

Deep residual learning [20] framework, provides a solution to vanishing or exploding gradients [33] occurred in very deep networks and allows the networks to have very deep architectures such as a network with 152 layers. The idea is to let the layers fit to a residual mapping (*F(x)*) rather than the original one (*H(x)*):

$$F(x) = H(x) - x. \tag{1}$$

It has been shown that for a very deep model it is easier to optimize *F(x)* rather than *H(x)* [20]. The realization of (1) is carried out by shortcut connections as illustrated in Figure 1.

In this work, leaf disease classification using four CNN architectures is experimented. The first two are the well-known VGG architectures with 16 and 19 layers, pre-trained on ImageNet, and the remaining two are customized simpler architectures, one with residual shortcut connections (Residual CNN) and the other without (Plain CNN). The proposed models are aimed to be simplified and easier to retrain with a regular PC without degradation of the accuracy. The details of the models are described in the following section.

3. EXPERIMENTS AND RESULTS

In this section, the CNN-based classifiers are tested on a subset of the PlantVillage leaf diseases dataset, including tomato plant leaf diseases. The dataset consists of 9 leaf diseases of tomato plant, including Bacterial Spot (2,127 samples), Early Blight (2,579 samples), Late Blight (3,575 samples), Septoria Leaf Spot (1,771 samples), Spider Mites (Two-Spotted Spider Mite) (1,676 samples), Tomato Mosaic Virus (373 samples), Leaf Mold (952 samples), Target Spot (1,404) and Tomato Yellow Leaf Curl disease (5,357). Adding healthy tomato leaf images, the used dataset contains 19742 images in 10 categories. A few samples of these datasets are depicted in Figure 2.

The preliminary preparation and augmentation are applied to the dataset. The images of the dataset are resized to fit into 280×180 dimensions which are chosen to be relatively small and close to a fraction of the average size of all images. After excluding 20% of the images as test set, the remaining images as training set are augmented, in order to reduce overfitting, by adding horizontally flipped copy of the images, then a portion of these images is further separated as the validation set.

Several CNN models are used in this paper to identify tomato disease from their leaves, including VGG architectures with 16 and 19 layers (VGG16 and VGG19), pre-trained on ImageNet and fine-tuned on the dataset, and the proposed CNN architecture with and without residual learning. Firstly, the pre-trained VGG models, are fine-tuned on the dataset to be considered as a baseline for comparison. Then a simplified CNN architecture is proposed and trained with and without the residual learning framework (residual and plain CNN) to compare the results. The details of the models are illustrated in Figure 3.

VGG16 architecture contains a number of convolutional layers with filter size of 3×3 and padding 1 (Conv3s), max-pooling layers with size of 3×3 and stride 2 (Pool2s), and fully-connected layers (FCs). After each Conv layer a ReLU nonlinearity function is applied. The architecture is divided into 6 blocks. First two blocks contain the two Conv3s following by a Pool2. Each of next three blocks contain three stacked Conv3s followed by one Pool2. The last block contains three FCs. Number of units in Conv3 layers in the first to fifth blocks are 64, 128, 256, 512, 512, accordingly. Two FC layers before the last layer (output) contains 4096 units. Input data (images) are of the size 254×254 color pixels and the output layer contains 1000 units, according to the number of target categories of ImageNet. VGG19 is similar to VGG16 but with additional 3 layers scattered in third to fifth blocks as shown in figure 3. In the experiments of this paper, the input and output

Journal of Theoretical and Applied Information Technology
31st December 2017. Vol.95. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

layers are replaced according to the selected input size and existing target classes. In order to transfer learning, the fully-connected layers of the models are replaced with two FC layers with 1024 and 10 units, respectively, and for fine-tuning the last convolutional block of the models are allowed to update during training.

The proposed CNN architecture consists of four blocks. In the first block a convolutional layer with filter size 11×11 and stride 3 (Conv11) followed by a max-pooling layer with filter size 3×3 and stride 3 (Pool3) is utilized. The next three blocks contain two Conv3s followed by a pooling layer each of filter size of 5×3 and stride 2 (Pool5_2) except the first one which has stride 3 (Pool5_3). After each Conv layer a ReLU nonlinearity function is applied. The outcome of the hierarchy after getting flattened is fed into the two FC layers. The number of filters in Conv11, Conv3s and FCs are 64, 128 and 1024, respectively, except for the last FC which is 10, corresponding to the number of diseases.

The most distinctive parts of the residual learning framework are shortcut connections. Adding data from previous layers to the current (using equation (1)) when its dimensions are changed causes difficulties. This occurs when either the size of the filters or the number of them are changed. It is a common practice that when data goes deeper into the architecture, the filter sizes are decreased and the number of them is increased. For the prior issue we can zero-pad the smaller data to be the same size of the larger, and for the latter, one can use a Conv layer with filter size 1×1 (Conv1) to propagate the historical data to a new group of units [20, 32]. As shown in the figure 3, the proposed residual CNN uses shortcut connections so that they are not disturbed by the filter size changes, but in the case of change on the number of filters, a Conv1 is used. The Conv1 layer is just used one time because this layer adds more parameters for learning thereby increasing the time. Therefore the

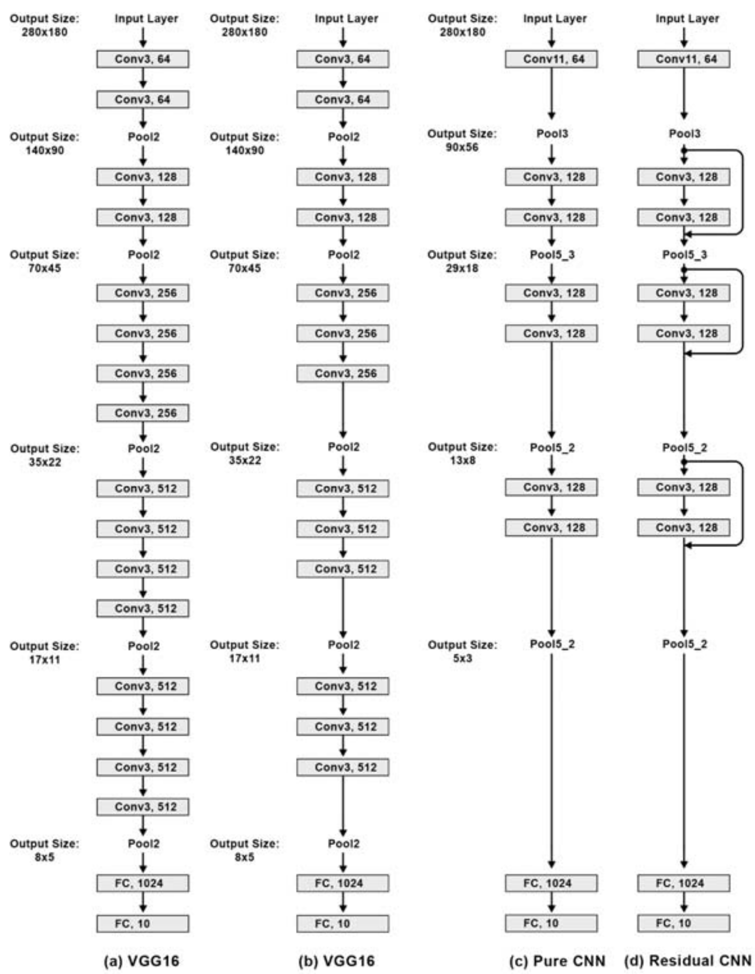

Figure 3: Details of the CNN architectures

Table 1. The classification accuracy of the experimented CNNs (%)

Models	Pre-trained VGG16	Fine-tuned VGG16	Fine-tuned VGG19	The Proposed Pure CNN	The Proposed Residual CNN
Top 1 Accuracy	94.31	96.55	97.50	96.78	97.53
Top 3 Accuracy	99.42	99.89	99.84	99.84	99.89

Table 2. Classification accuracy on each class of the dataset and on average (%)

Class	Bacterial Spot	Early Blight	Late Blight	Septoria Leaf Spot	Spider Mites	Tomato Mosaic Virus	Leaf Mold	Target Spot	Yellow Leaf Curl	Healthy
Top 1 Accuracy	95.75	95	96	98.33	95.67	100	98	91.5	99.8	99.33
Top 3 Accuracy	99.75	100	99	100	100	100	100	99	100	100

number of filters in all Conv3 layers is left unchanged.

The hyper-parameters of the CNNs are set as follows. Networks are trained by stochastic gradient descent with 0.9 momentum. The learning rate is set to be 0.001. The weight decay parameter is 1e-6. Initial weights are selected randomly from standard normal distribution, multiplied by the factor of 0.01. The mini-batch size is set to 100.

Keras [34], a Python based deep learning library, with Theano [35] backend, is used in the experiments. Keras is a wrapper for another deep learning framework aimed to facilitate working with those. It currently supports Theano and TensorFlow [36], but it is rapidly extending to include others. One of the helping features of Keras, especially when working with low equipped systems, is processing images without loading the whole dataset in RAM, which saves the system from a crash during processing large number of images.

The experiments are done using a PC with 8GB RAM and Core i5 CPU. The trained models are selected according to the best validation accuracies through 25 epochs, except for the pre-trained VGGs that the FC layers were trained until 50 epochs. For the proposed models the training process took almost two days and for the fine-tuned model this time was doubled. The proposed model contains about 2.8 million trainable parameters, but that of the VGG16 model (three bottom FC layers substituted with two FC layers in order to transfer learning) is about 28 million parameters, thus the proposed model is much lighter than the original VGGs.

The comparison of the average classification accuracy of the models on the test set is presented in Table 1. The detailed results of classification of leaf disease by the proposed residual CNN are shown in Table 2. The results are shown as top 1 and top 3 classification accuracy. The model better classifies Tomato Mosaic Virus, Yellow Leaf Curl disease and healthy images. The best and the worst classification accuracy belongs to Tomato Mosaic Virus and Target Spot, respectively.

In order to understand that the proposed CNN is sensitive to which part of the input samples, an experiment based on the idea of [32], is performed similar to [21] and [31]. In this approach different parts of an input image are occluded with a gray or black patch and the variation in the output

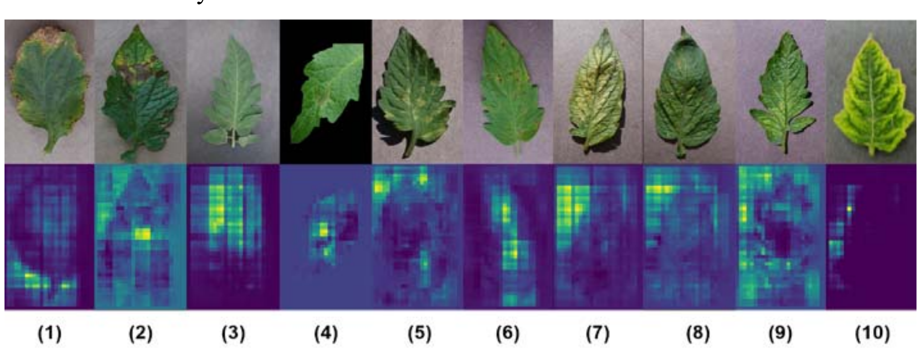

Figure 4. Samples of the dataset and the corresponding visualization by sliding 7×7 black patch.

Journal of Theoretical and Applied Information Technology
31st December 2017. Vol.95. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

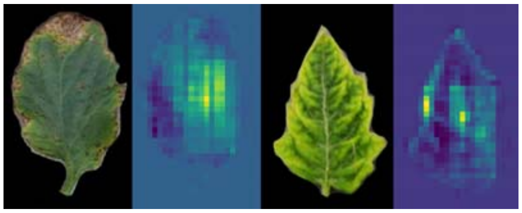

Figure 5. The background removed samples and their visualizations

probability of the correct class is traced. In this work the size of the occlusion patch is selected as 7×7 pixels, but this size can be arbitrarily changed. The result is a heat map indicating the locations of the image where the output is more sensitive to. The procedure is described in [31] as: "… we slide black rectangle over an image, and afterward, we run CNN to calculate the probability of correct classes of current image $PC_{i,j}$. The indexes (*i, j*) indicate the occlusion rectangle position in the image. Then, we visualize the negative log likelihood ($-\log(P_{i,j})$) of these probabilities using heat map." Figure 4 shows some randomly investigated example images of each category. Yellow colored regions correspond to the decrease in the output probability.

In the recent similar work [31], it was stated that the areas highlighted by visualization method are directly matched with an expert's descriptions about the diseases, that is the highlighted points in the visualized image show the diseased area and the background information is ignored. But the experiments of this paper show that this is not always true because it depends on the areas of the image that differentiate it from others. As you can see in Figure 4, in most samples, significantly in 1 and 10, the highlighted points in the visualized images, are falling in the background of the image. And when the background is replaced with a solid color like black, the highlighted points, direct us to the infected area (as in sample 4). In Figure 5 background of some samples replaced by black color.

It is observed that the background information can help the classifier to choose the right class. For example, in sample 2 (in Figure 4), where the classifier's confidence (the value of corresponding output unit) is low (less that 0.1), if we replace the background with black color, the classifier cannot determine the correct class, or in sample 10 where the classifier's confidence is rather high (over 0.9), the classifier can still find the correct class but with lower confidence. As you can see in Figure 5, the overall colors involved in the visualized images in lighter (low confidence) and this time (compared with the corresponding sample in Figure 4) the area within the leaf is highlighted. Although the background pattern in most samples of the dataset is simple, but this background has an important role in differentiating the overall shape of the leaf. These examples, suggest the importance of the background pixels in categorizing images using CNNs.

4. CONCLUSION

In this paper, based on deep residual learning methodology, a CNN architecture was suggested to apply to tomato disease classification task. The results show that the proposed model can compete with VGG networks pre-trained on the ImageNet dataset, with the advantage of lighter training time on a PC with regular hardware abilities. This paper also confirms the usefulness and effectiveness of Transfer Learning on the task of tomato plant disease identification. In comparison with the recent similar work, in addition to fine-tuning existing deep models, in this work a new CNN architecture is proposed and trained from scratch to classify potato diseases and also more investigations of the results are done. With help of the visualization technique, it's discussed that the measure for the network to correctly classify leaf disease is not only the correct recognition of diseased area, but sometimes is the diagnosis of clues in the background and other not obviously related parts of the leaf image. The reason is that the important thing for the network, is to find distinctive parts of images, not actually infected regions. However, in some cases the distinctive parts of the images match the infected regions.

This work improves the current state-of-the-art in terms of proposing a model for the new area of plant leaf disease classification, lighter than current general object detection models, specifically VGGs, both in time of training and classification accuracy. Setting appropriate parameters to define CNN architectures is still a challenging issue, to tackle the real-world problems. Therefore, instead of empirically assessing different CNN settings, future works can be focused on devising methods to automate this process.

REFRENCES:

[1] D.L. Borges, S.T.d.M. Guedes, A.R. Nascimento, P. Melo-Pinto, Detecting and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

grading severity of bacterial spot caused by Xanthomonas spp. in tomato (Solanum lycopersicon) fields using visible spectrum images, Computers and Electronics in Agriculture, Vol. 125 No. 2016, pp. 149-159.

[2] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision, Vol. 60 No. 2, 2004, pp. 91-110.

[3] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE, 2005, pp. 886-893.

[4] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Computer vision and image understanding, Vol. 110 No. 3, 2008, pp. 346-359.

[5] G. Prince, J.P. Clarkson, N.M. Rajpoot, Automatic detection of diseased tomato plants using thermal and stereo visible light images, PloS one, Vol. 10 No. 4, 2015, pp. e0123262.

[6] U. Mokhtar, M.A. Ali, A.E. Hassanien, H. Hefny, Identifying two of tomatoes leaf viruses using support vector machine, in: Information Systems Design and Intelligent Applications, Springer, 2015, pp. 771-782.

[7] Y. Chai, X. Wang, Recognition of greenhouse tomato disease based on image processing technology, Pattern Recognition and Simulation, Vol. 9 No. 2013, pp. 83-89.

[8] S. Prasad, P.S. Kumar, D. Ghosh, An efficient low vision plant leaf shape identification system for smart phones, Multimedia Tools and Applications, Vol. 76 No. 5, 2017, pp. 6915-6939.

[9] S. Prasad, S.K. Peddoju, D. Ghosh, Multi-resolution mobile vision system for plant leaf disease diagnosis, Signal, Image and Video Processing, Vol. 10 No. 2, 2016, pp. 379-388.

[10] A. Singh, B. Ganapathysubramanian, A.K. Singh, S. Sarkar, Machine learning for high-throughput stress phenotyping in plants, Trends in plant science, Vol. 21 No. 2, 2016, pp. 110-124.

[11] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097-1105.

[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, Vol. No. 2014, pp.

[13] S. Yu, S. Jia, C. Xu, Convolutional neural networks for hyperspectral image classification, Neurocomputing, Vol. 219 No. 2017, pp. 88-98.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, Vol. 86 No. 11, 1998, pp. 2278-2324.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Imagenet large scale visual recognition challenge, International Journal of Computer Vision, Vol. 115 No. 3, 2015, pp. 211-252.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[17] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, H. Larochelle, Brain tumor segmentation with deep neural networks, Medical image analysis, Vol. 35 No. 2017, pp. 18-31.

[18] L.A. Alexandre, 3D object recognition using convolutional neural networks with transfer learning between input channels, in: Intelligent Autonomous Systems 13, Springer, 2016, pp. 889-898.

[19] H. Xue, Y. Liu, D. Cai, X. He, Tracking people in RGBD videos using deep learning and motion clues, Neurocomputing, Vol. 204 No. 2016, pp. 70-76.

[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[21] G.L. Grinblat, L.C. Uzal, M.G. Larese, P.M. Granitto, Deep learning for plant identification using vein morphological patterns, Computers and Electronics in Agriculture, Vol. 127 No. 2016, pp. 418-424.

[22] M. Mehdipour Ghazi, B. Yanikoglu, E. Aptoula, Plant identification using deep neural networks via optimization of transfer learning parameters, Neurocomputing, Vol. No. pp.

[23] H. Goëau, P. Bonnet, A. Joly, LifeCLEF Plant Identification Task 2015, in: CLEF2015 Working Notes. Working Notes for CLEF 2015 Conference, Toulouse, France, September 8-11, 2015, CEUR-WS, 2015.

[24] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[25] S.P. Mohanty, D.P. Hughes, M. Salathé, Using Deep Learning for Image-Based Plant Disease Detection, Frontiers in Plant Science, Vol. 7 No. 2016, pp.

[26] D.P. Hughes, M. Salathé, An open access repository of images on plant health to enable the development of mobile disease diagnostics, CoRR abs/1511.08060, Vol. 1 No. 2015, pp.

[27] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovic, Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification, Computational Intelligence and Neuroscience, Vol. 2016 No. 2016, pp. 11.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 248-255.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675-678.

[30] Y. Lu, S. Yi, N. Zeng, Y. Liu, Y. Zhang, Identification of Rice Diseases using Deep Convolutional Neural Networks, Neurocomputing, Vol. No. 2017, pp.

[31] M. Brahimi, K. Boukhalfa, A. Moussaoui, Deep Learning for Tomato Diseases: Classification and Symptoms Visualization, Applied Artificial Intelligence, Vol. No. 2017, pp. 1-17.

[32] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818-833.

[33] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249-256.

[34] F. Chollet, Keras, URL http://keras. io, Vol. No. 2015, pp.

[35] T.T.D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, Theano: A Python framework for fast computation of mathematical expressions, arXiv preprint arXiv:1605.02688, Vol. No. 2016, pp.

[36] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467, Vol. No. 2016, pp.