Disease Detection on the Plant Leaves by Deep Learning

P. Goncharov[1(✉)], G. Ososkov[2], A. Nechaevskiy[2], A. Uzhinskiy[2], and I. Nestsiarenia[1]

[1] Sukhoi State Technical University of Gomel, Gomel, Belarus
kaliostrogoblin3@gmail.com
[2] Joint Institute for Nuclear Research, Moscow Reg, Dubna, Russia
auzhinskiy@jinr.ru

Abstract. Plant disease detection by using different machine learning techniques is very popular field of study. Many promising results were already obtained but it is still only few real life applications that can make farmer's life easier. The aim of our research is solving the problem of detection and preventing diseases of agricultural crops. We considered several models to identify the most appropriate deep learning architecture. As a source of the training data, we use the PlantVillage open database. During research, the problem with PlantVillage images collection was detected. The synthetic nature of the collection can seriously affect the accuracy of the neural model while processing real-life images. We collected a special database of the grape leaves consisting of four set of images. Deep siamese convolutional network was developed to solve the problem of the small image databases. Accuracy over 90% were reached in the detection of the Esca, Black rot and Chlorosis diseases on the grape leaves. Comparative results of various models and plants using are presented.

Keywords: Machine learning · Statistical models · Siamese networks
Plant disease detection · Transfer learning

1 Introduction

Increasing number of smartphones and advances in deep learning field opens new opportunities in the crop diseases detection. Probably, the most famous mobile application allowing users to send photos of sick plants and get the cause of the illness is Plantix (plantix.net). The application is developed by PEAT, a German-based AgTech startup. Currently Plantix can detect more than 300 diseases. Unfortunately, Plantix image database is closed and we cannot find any information about technologies used for disease detection. The quality of plantix detection is hard to measure, but we make a special study processing different types of images from our self-collected database. It allows us to conclude that Plantix identification of the plants type is rather good: 60 of 70 images (87%) were recognized as grapes. At the same time the

The reported study was funded by RFBR according to the research project № 18-07-00829.

B. Kryzhanovsky et al. (Eds.): NEUROINFORMATICS 2018, SCI 799, pp. 1–9, 2019.
https://doi.org/10.1007/978-3-030-01328-8_16

disease detection ability is rather limited. The most of the healthy leaves were identified as healthy or healthy was on the top of suggestions. 10 of 20 leaves with Chlorosis were also detected as healthy. Leaves with Esca and Black Rot were recognized as sick but the correct disease names were not on the top of suggestions. Only few images were detected correctly. Perhaps, our dataset does not match some requirements of the Plantix application. We used original images from Internet and preprocessed those in which problems were obvious, but the result was quite similar.

The aim of our research is to facilitate the detection and preventing diseases of agricultural plants by both deep learning and programming services. The idea is to develop multifunctional platform that will use modern organization and deep learning technologies to provide new level of service to farmer's community. However, in this paper we focus on the deep learning issues only. We would like to reach same functionality as Plantix but with better accuracy in the diseases detection. We will also going to provide an open access to our image database and share experience of our deep learning architectures.

The key issue for the implementation of the plant disease detection platform (PDDP) is an appropriate deep learning architecture. We considered different models used in related works to understand what the best option is. In [1] authors reached high accuracy in detection to 99.7% on a held-out test set. They have used PlantVillage well known public database of 86,147 images of diseased and healthy plant leaves. We have reproduced their experience using the same approach but different software. As a result, we obtained identical high accuracy when using PlantVillage data for training and testing, but the results obtained by applying our trained network on real-life images were quite unsatisfactory (for about 40% only). The problem lies in the type of the used images. PlantVillage photos were collected and processed under special controlled conditions, so they are rather synthetic and differ from real-life images, as it is shown in Fig. 1. Influence of the different image types on the accuracy of the diseases detection with field images is well shown at [2]. Some experiments were done with background modification and other optimizations [3], but it could improve the accuracy slightly.

This proves that if we want a good result, we need a real-life database. Although many related successful studies are known using their self-collected databases [4–7], but there are no references to used databases, unfortunately. We collect our own database of the grape leaves from open source images and then reduce their size and extract only meaningful parts. Eventually, we have a set of 256 $\times$ 256 pixel images consisting of 130 healthy leaves, and 30–70 images with Esca, Chlorosis and Black Rot diseases. The number of images is very small but we are going to refill our database with users' images of correctly detected diseases when the public part of PDDP will be developed. The current database is available at http://pdd.jinr.ru/db. We used this database to test some models and to try some new approaches.

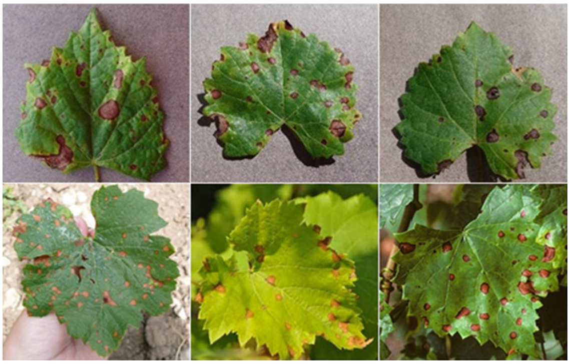

Fig. 1. Top three photos are from the PlantVillage database. Bottom photos are real photos of sick leaves from the Internet.

2 Unsuccessful Results of Our Study

2.1 Transfer Learning

Our first attempt was similar to [1], we applied transfer learning approach to train deep classifier on the PlantVillage images and then evaluated classifier on a test subset of images, collected from the Internet.

To find the most appropriate pretrained network (further base network) for the transfer learning we compared four models the weights of which were formed to solve the ILSVRC 2015 (ImageNet Large Scale Visual Recognition Challenge) [8], they are: VGG19 [9], InceptionV3 [10], ResNet50 [8] and Xception [11].

The comparative scheme of each classifier is to compose all layers of trained networks except final classification layer and to add the global average pooling operation [8] on the top of each base network to reduce the spatial dimensions of a three-dimensional output tensor. Further, we appended a densely connected layer with 256 rectified neurons with dropout having rate of 0.5. At the end of such network, softmax classification layer was utilized.

We froze all layers in the base networks and trained only last three layers using stochastic gradient descent (SGD) with learning rate equals to 5e−3, momentum 0.9 and weight decay with value of 5e−4 for 50 epochs. The best result of classification accuracy with the value of 99.4% on a test subset of the PlantVillage dataset was obtained using ResNet50 architecture. We applied this model to deduce the classification efficiency on a test subset collected from the Internet. Obtained results were very poor – 48% accuracy on a set of 30 images.

2.2 Advanced Transfer Learning and Data Augmentation

Supposing that pretrained on the ImageNet dataset network does not extract meaning features from leaves images, we decided to unfreeze more layers. There is no reason to train the whole network from scratch, as we do not have a dataset with the suitable size to train such a deep network. Authors in [1] got very low accuracy on the Internet images. Thus, we unfroze all layers except first 140 in the base network and trained the remaining 39 defrozen layers with the help of Adam optimizer [15] with learning rate equals to 5e−5 and weight decay with value of 1e−6 for 30 epochs. We used only three plant classes: Esca, Black rot and healthy. We cannot use Chlorosis, because there are no images of that class in the PlantVillage dataset.

Next, we proposed to apply a strong data augmentation by adding random transformations such as shifts, rotations, zooming etc., because the classification network overfits when we train more than 30 epochs. Also, we supposed that only central part of a leaf is required to recognize disease. Therefore, we tried to expand our dataset by using only parts of initial images. We started from 128×128 central square of each initial image and generate then at least 5 new parts around it (as shown in Fig. 2).

We supposed that this approach of data augmentation by splitting original images into parts allows also to decrease the influence of a background, so we cat bound parts of leaves using only their middle part for classification. However, accuracy on the test dataset of images collected from the internet was over 49% only. The network overfits after 10 epochs and even our strong augmentation does not help. Surprisingly, it works quite well for binary classification like recognizing of healthy and disease leaves – we obtained 85% accuracy on the test dataset.

It turns out that, when we crop images, we lose the information about location of spots and it produces very noisy data. Besides some parts of diseased leaves looks healthy and automatic split to the parts produces incorrect examples. We tried to use parts of images for multi-class classification, but experiments show that such approach is not effective. Using the images with their original sizes is more reliable and more suitable for real application.

The best accuracy of classification on the PlantVillage dataset was about 99%. On the test subset of 63 images collected from the internet we could reach 78% of accuracy.

3 Siamese Networks for Learning Data Embeddings

It becomes obvious that features extracted during training network models that we attempted to try, are not adequate for further classifying. These features cannot represent input images in a new multidimensional space, where diseased and healthy images should be separated into distinctive clusters. Probably, it depends on the synthetic nature of the available training set, so a question arises: how could we learn good features from very small amount of data collected from the Internet? We address this problem to so-called one-shot approach [12] offering a solution by siamese neural networks [12–14].

Siamese network consists of twin networks joined by the similarity layer with the energy function at the top. Weights of twins are tied, thus the result is invariant and in addition guarantees that very similar images cannot be in very different locations in feature space, because each network computes the same function. The similarity layer determines some distance metric between the so-called embeddings, i.e. high-level feature representations of input pair of images (Fig. 3).

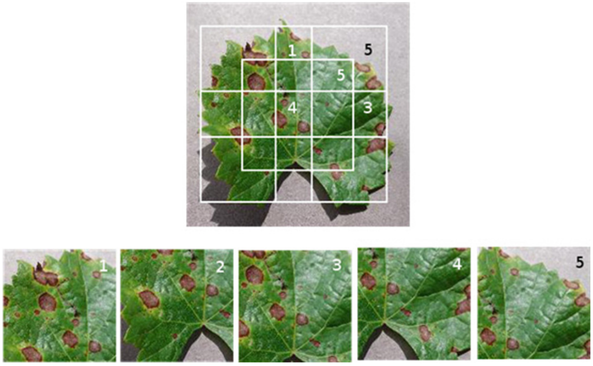

Fig. 2. Splitting original image into training parts

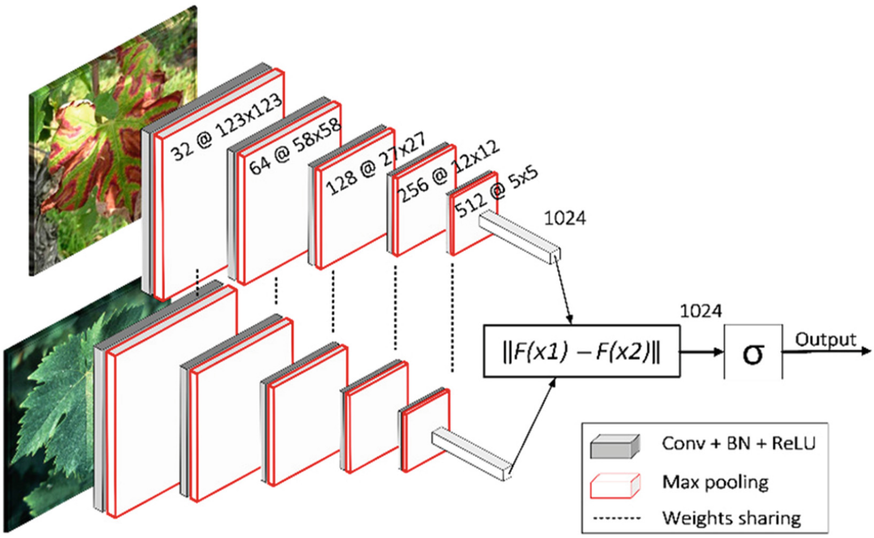

Fig. 3. Our best siamese convolutional architecture. «Conv» means the convolutional operation, «BN» is a batch normalization, «32 @ 123 × 123» – 32 feature maps with the particular size

Training on pairs means that there are quadratically many possible pairs of images to train the model on, thus making it hard to overfit. We can easily compute the number of possible pairs using the combinatorics formula of k-combinations. Thereby, for the smallest class with 31 images (Black rot) we have 1860 pairs, which is a great result.

Our Siamese network is built in the same way as in [12] – it unites the twins within L1 distance layer followed by sigmoid activation in order to train the net with cross-entropy objective. The opposite approach includes application of special contrastive loss, which was first introduced in [13]. Unfortunately, the using of contrastive loss requires a special way of image pairing, otherwise one could face with very low testing accuracy. Our experiments with weighted contrastive loss in tandem with random guessing proved this statement – the maximum value of testing accuracy was 52% only. Other research [14] also confirms the correctness of our conclusions.

3.1 Architecture

Our classification model is the siamese network with convolutional twins, which ties weights between themselves. Each of twins processes one image from input pair of samples to extract a vector of high-level features. Then a pair of embeddings is passed through lambda layer, actually computing the simple elementwise L1-distance (Fig. 3). We present connection a single sigmoid neuron to the distance layer, thus it becomes possible to train the model with binary cross-entropy loss.

We use exclusively rectified linear (ReLU) units in the twins except the last densely-connected layer with sigmoid activations. The initial number of convolutional filters is 32 and doubling each layer. After the last pooling layer, we added a flatten operation to squeeze convolutional features into vector followed by densely-connected layer with 1024 sigmoid neurons.

We also experimented with adding L2 regularization to each of convolutional layers but have not received visible effect. In addition, we have been trying to vary the size of the embedding layer from 256 to 4096. Out best architecture is illustrated on Fig. 3.

In [16] authors evaluated the trained network on new images in a pairwise manner against the test image. The created pairs consist of test image in each pair and one sample for every class. Then the pairing with the highest score according to the trained siamese network is then awarded the highest probability for the one-shot task. We, in turn, offer to use K-nearest neighbors (KNN) algorithm to solve the classification task in test phase. We set K parameter to 1 nearest neighbor, so it is equivalent to one-shot learning task, except one moment – we utilize all training data as support set instead of random picking from dataset. For the distance metric we can use any of manhattan distance or euclidean distance, but as we have used the absolute distance in the lambda layer of the siamese network, we preferred to manage the manhattan distance.

We apply data augmentation by adding rotations in range of 75°, random shifts in all dimensions, including shift within channel in range of 0.2. In addition, we use little zooming, vertical and horizontal flips.

We have trained the siamese network with the proposed augmentation for 100 batches size 32 per one training epoch for 35 epochs. As the method of optimization we used Adam [15] with learning rate value of 0.0001. For the loss function we utilized a binary cross-entropy loss.

3.2 Classifying by Nearest Neighbors

After training the one-shot model, we left only the encoder network represented as «shoulder» of the one-shot model or so-called twin. We further use this part of network as feature extractor. After that, we take the training subset of the database, collected by us from the Internet. This database includes four classes of images: healthy images and 3 types of diseases – Esca, Black rot and Chlorosis with 133, 73, 31 and 42 images per class respectively. We split data on train and test with the ratio near 75:25, where 75 means 75% of images of particular class. Accordingly, we have 208 training images and 71 testing images. Then we pass these sets of images through our feature extractor to obtain data embeddings.

The training subset of images is then utilized as training data for the KNN. For verification we used the remaining test set.

3.3 Results

In the Table 1 one can see the confusion matrix[1] of the KNN on the test subset of embeddings data.

Table 1. Confusion matrix on the test subset of internet images

	Black rot	Chlorosis	Esca	Healthy
Black rot	**7**	0	1	1
Chlorosis	0	**11**	0	1
Esca	0	0	**20**	1
Healthy	0	0	0	**29**

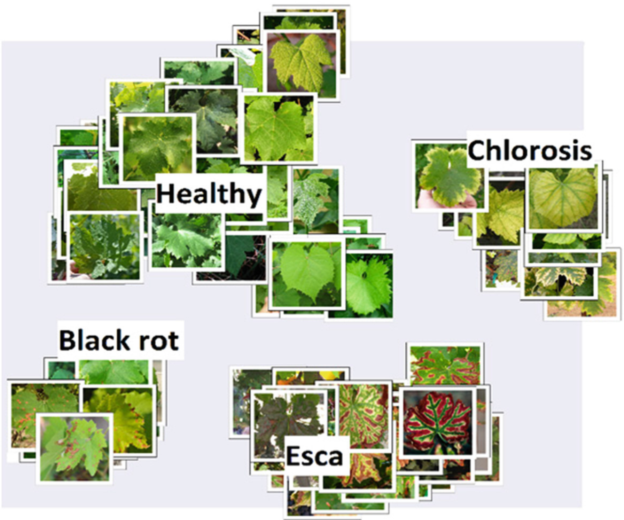

Fig. 4. T-SNE visualization of the high-level features extracted by the siamese twin

[1] This matrix shows how many class i objects were recognized as class j objects [17].

It is simple to deduce that the classification accuracy equals to 94.3% which is acceptable. Besides, we tried to mix the real-life data with the images from the PlantVillage database within train and test subsets. Siamese networks allow to generalize input data to latent high-dimensional embedding space, even for unseen images. The obtained accuracy with the value of 92% (our best result) proves this consideration.

As the low testing accuracy shows, our previous models cannot separate input images into clearly distinguishable clusters of classes. Therefore we use the prepared embeddings to train the T-SNE method [18], which is the common technique to visualize high-dimensional data. We extract two components to plot them in 2D space (Fig. 4). One can see that there are four separate clusters – one per each class. We have signed each cluster with the name of particular class. Although, there are a few points, which wrongly got into the different set (see the Table 1), but it is not detractive.

4 Conclusion and Plans

Siamese neural networks are very perspective research field. We are going to use them as a basic deep learning architecture for PDDP and since their power lies in seeking for differences between classes, we are going to add more classes to the train dataset soon. It is clear that unequivocal detection of the diseases is unsolvable task especially at first stages of a plant illness. The list of the most suitable variants should be provided to the users so they could try different treatment schemas. Quality of the database is extremely important for the results so the mechanism of fulfilling DB with images of correctly detected diseases should be developed. We keep on working on PDDP and are going to present web-interface prototype by the end of 2018.

References

1. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Front. Plant Sci. **7** (2016). Article: 1419
2. Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. **145**, 311–318 (2018)
3. Cortes E.: Plant Disease Classification Using Convolutional Networks and Generative Adverserial Networks. Stanford University Reports, Stanford (2017)
4. Fuentes, A., Yoon, S., Kim, S.C., Park, D.S.: A robust deep-learning-based detector for real-time tomato plant diseases and pest recognition. Sensors **17**(9), 2022 (2017)
5. Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D.: Deep learning for image-based Cassava disease detection, frontiers in plant science. Front. Plant Sci. **8**, 1852(2017)
6. Lua, J., Hua, J., Zhaoa, G., Meib, F., Zhanga, C.: An in-field automatic wheat disease diagnosis system. Comput. Electron. Agric. **142PA**, 369–379 (2017)
7. Ronnel, R., Atole, A.: Multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **9**(1) (2018)
8. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
10. Szegedy, C. et al.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
11. Chollet, F.: Xception: deep learning with depthwise separable convolutions. arXiv preprint (2016)
12. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
13. Kotikalapudi, R.: contributors (2017). keras-vis (2017)
14. Ososkov, G., Goncharov, P.: Shallow and deep learning for image classification. Opt. Mem. Neural Netw. **26**(4), 221–248 (2017)
15. Ososkov, G., Goncharov, P.: Two-stage approach to image classification by deep neural networks. In: EPJ Web of Conferences, vol. 173, p. 01009. EDP Sciences (2018)
16. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop, vol. 2 (2015)
17. Powers, D.: Evaluation: from precision recall and F-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)
18. Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)