Architecture and basic principles of the multifunctional platform for plant disease detection

Pavel Goncharov[2], Gennady Ososkov[1], Andrey Nechaevskiy[1], **Alexander Uzhinskiy**[1]

1. Joint Institute for Nuclear Research
2. Sukhoi State Technical University of Gomel, Belarus.

The 8th International Conference "Distributed Computing and Grid-technologies in Science and Education", Dubna 2018

Introduction

Crop losses are a major threat to the wellbeing of rural families, to the economy and governments, and to food security worldwide

Quality of available data about the impact of plant diseases is variable, patchy and often missing, particularly for smallholders, who produce the majority of the world's food.

CIP, the international research center with an historical mandate for potato, estimates 15% production losses each year due to late blight.

USAblight (a national project on lb on potato and tomato) says that (annual) global losses 'exceed US$6.7 billion'.

As agriculture struggles to support the rapidly growing global population, plant disease reduces the production and quality of food, fibre and biofuel crops. Losses may be catastrophic or chronic, but on average account for 32% of the production of the six most important food crops.

Prof. Dr. David Guest. Faculty of Agriculture, Food and Natural Resources, The University of Sydney

Globally, about 16% of all crops are lost to plant diseases each year.

Dr. Caitilyn Allen Department of Plant Pathology, University of Wisconsin–Madison

Target

Increasing number of smartphones and advances in deep learning field opens new opportunities in the crop diseases detection.

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.

We are going to create the plant disease detection platform (PDDP) from the scratch and share our experience. We will describe used deep learning models and techniques and provide open access to our image database.

As end-product we are going to develop an mobile application allowing users to send photos and text description of sick plants and get the cause of the illness.

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

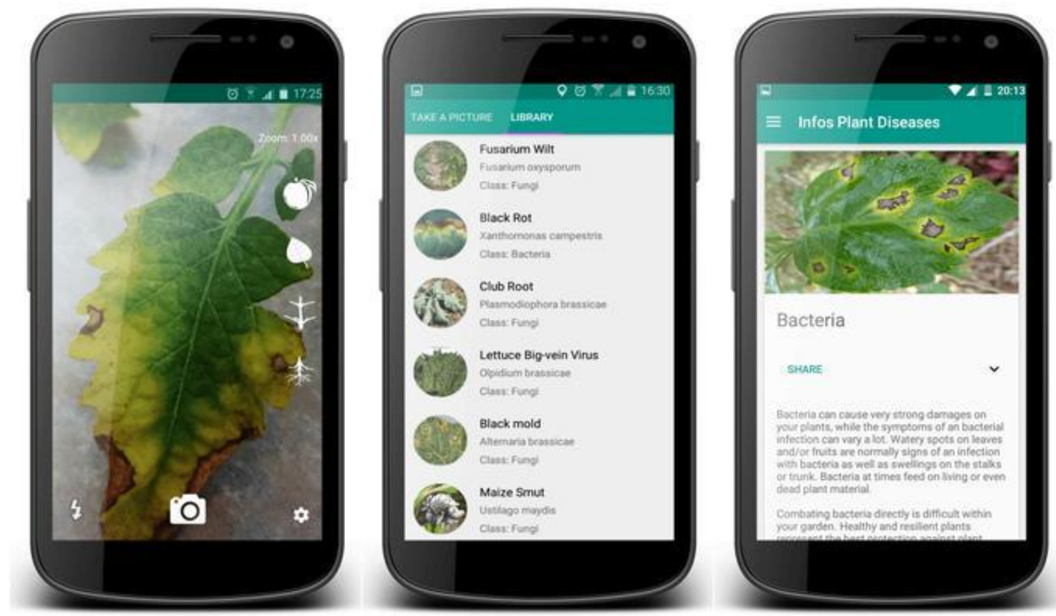
Sure, we are not the first

There are a lots of researches where deep learning used to identify plant diseases. Some of researches reports about a great detection level over 90%.
But there is luck of real application or open databases to reproduce an experiments.

Probably, the most famous mobile application for plants disease detection is Plantix (**plantix.net**). Currently Plantix can detect more than 300 diseases.

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 with grape diseases (Esca, Black rot, Chlorosis and Mildew).

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 disease detection ability is rather limited. Only few images were detected correctly. And for less than 10% right disease was at the top op suggestions.



Perhaps, our dataset does not match some requirements of the Plantix application. But, we used original images from Internet and preprocessed ones where problems were allocated clearly, but result was quite similar.

Basic architecture and images dataset are the key factors

We considered different models used in related works to understand what the best option is.
We also searched for the open database to reproduce researches experience.
We have concentrate on the "**Mohanty S.P., Hughes D.P., Salathé M.: Using Deep Learning for Image-Based Plant Disease Detection**" 2016 work.

They have used PlantVillage well known public database of 86,147 images of diseased and healthy plant leaves. Authors reach the great accuracy in detection 99.7% on a held-out test set. But results on the real-life images were quite unsatisfactory about 30% only.

Examples of the PlantVillage photos

But it was only one public database and we believes that we can improve their results.

Eventually, we have prepared the small test set of 256x256 pixel images consisting of healthy leaves, and images with Esca, Chlorosis and Black Rot diseases and started to work.

Examples of our test dataset images

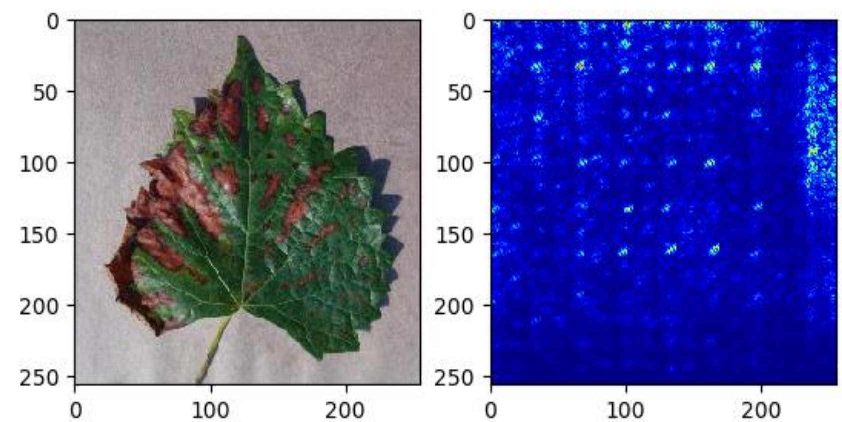
What we did, that did not work properly

We applied transfer learning approach to train the deep classifier on the Plant Village images and then evaluated our classifier on a test subset of images

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

The comparative scheme of each classifier is to compose all layers of trained networks except the final classification layer and to add the global average pooling operation on a 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 the softmax classification layer was utilized.

The best result of classification accuracy with the value of 99.4% on a test subset of the Plant Village 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 rather poor – 45% on a set of 30 images



Saliency map of the output of ResNet50

What we did, that did not work properly

Supposing that network pretrained on the ImageNet dataset network does not extract meaning features from leaves images, we decided to unfreeze more ResNet50 layers. 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 with learning rate equals to 5e-5 and weight decay with value of 1e-6 for 30 epochs.

Next, we propose 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 and we tried to expand our dataset by using only parts of initial images. Some other experiments were done with background modification and other optimization, but it could improve the accuracy only a little.

We believe that the problem is in type of the used images. PlantVillage photos were collected and processed under special controlled conditions, so they are rather synthetic and differs from real-life images

This proves that if we want a good result, we need a real-life database.

PlantVillage

Real life

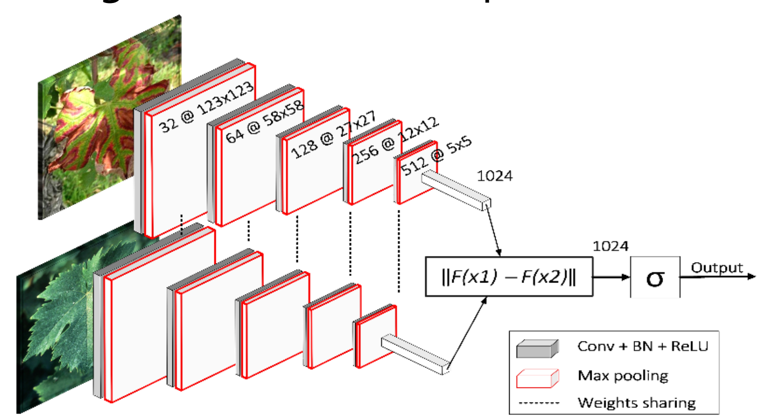
Siamese networks for learning data embeddings

We try to extend our test data set but success only a little. We can get some images from Internet: Healthy (130), Black Rot (31), Esca (73), Chlorosis (50), *Powdery mildew* (22). But how could we learn good features from very small amount of data?

We address this problem to so-called one-shot approach offering a solution by siamese neural networks [**Koch G., Zemel R., Salakhutdinov R. Siamese neural networks for one-shot image recognition //ICML Deep Learning Workshop. – 2015. – T. 2.**].

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 so-called embeddings, i.e. high-level features representations of input pair of images (see fig).

Training on pairs is that there are quadratically many possible pairs of images to train the model on, making it hard to overfit. We can easily compute the number of possible pairs using the combinatorics formula of k-combinations. for our small dataset from the internet with size of **279** images for **4** classes we have $N_{pairs} = \mathbf{38781}$, which is very good. Now it is **48828** pairs for 313 images!



Our best siamese convolutional architecture. «Conv» means the convolutional operation, «BN» is a batch normalization, «32 @ 123x123» – 32 filters with the particular size

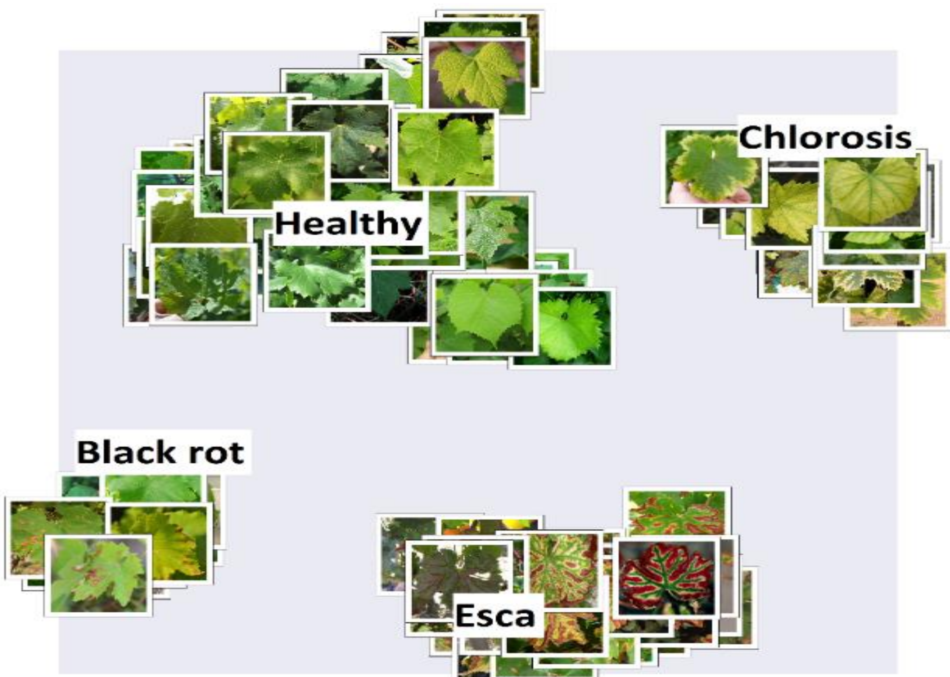
Results

Our Siamese network unites the twins within L1 distance layer followed by sigmoid activation in order to train the net with cross-entropy objective. Detailed information can be found at [**Disease detection on the plant leaves by deep learning, Springer**]

In the table 1 one can see the confusion matrix of the K-nearest neighbors algorithm on the test subset of embeddings data.

	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

It is simple to deduce that the classification accuracy equals to 94.3%.



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

We extract two components to plot them in 2D space. One can see that there are four separate clusters – one per each class.

Although, there are a few points, which wrongly got into the different set (see the table), but it is not detractive.

After we have finished with architecture we can keep on going with other parts of the platform.

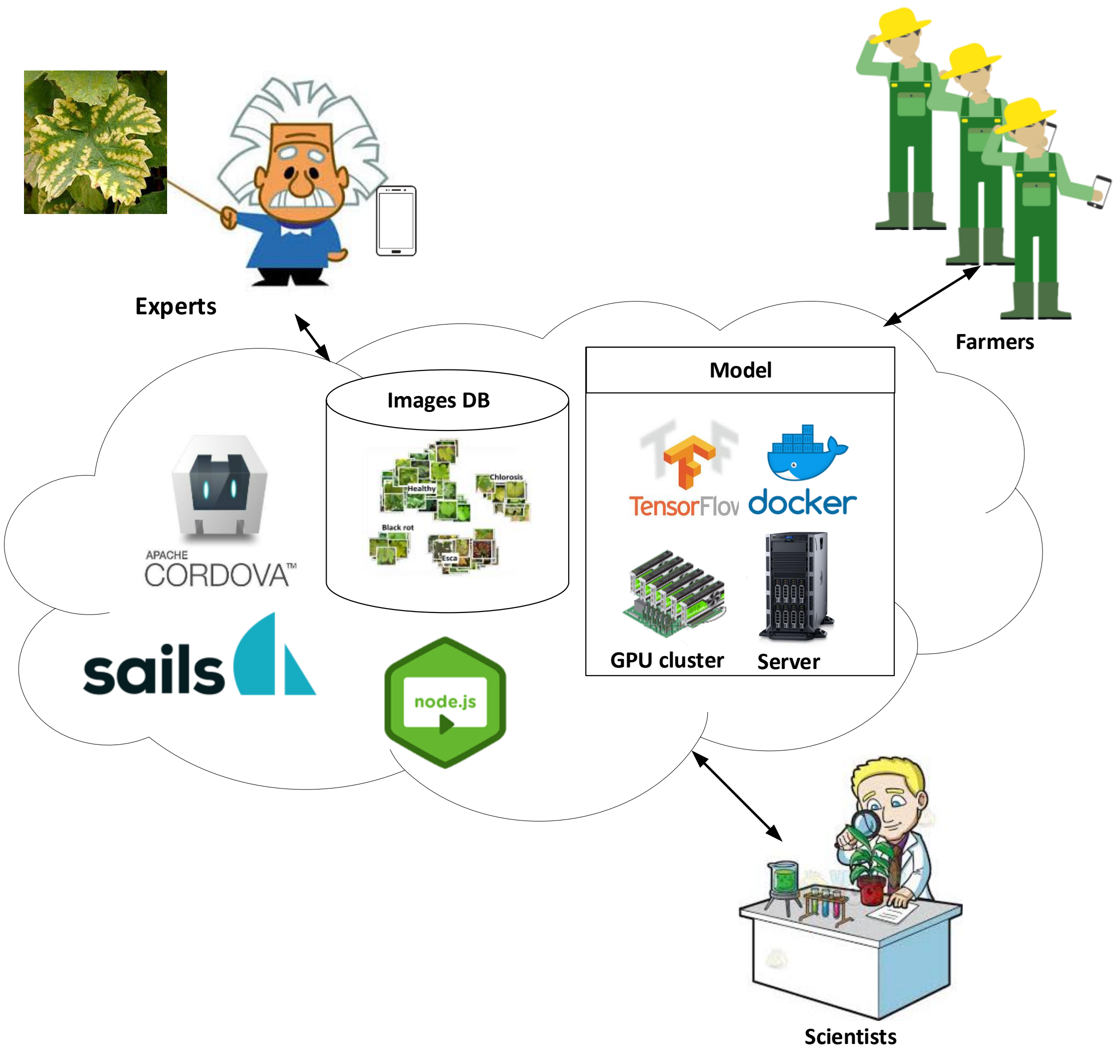
PDDP Architecture

PDDP consists of a set of interconnected services and tools developed, deployed and hosted in the JINR cloud infrastructure.

Our web-portal pdd.jinr.ru is developed with Node.js (Sail.js). It provide not only web-interface but also an API for third-party services.

We have TensorFlow model in in Docker realized as a service. The model can work at the virtual server or at GPU cluster.

Right now we are storing images directly on the local drive but if their number will increase dramatically we will use cloud storage (disk.jinr.ru).

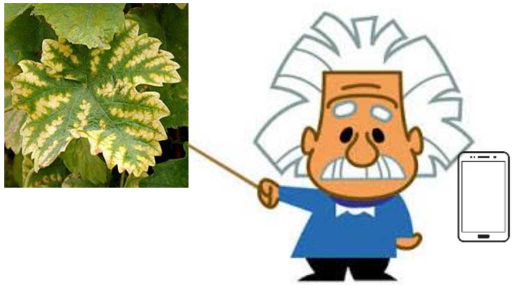


We will use Apache Cordova to create Mobile App for Android, IOS and Windows platforms.

PDDP basic principals

Users can:
- send photos and text description of sick plants through web-interface or mobile application and get the cause of the illness,
- browse through diseases description and galleries of ill plants,
- verify that requested disease was recognized right and treatment helps.

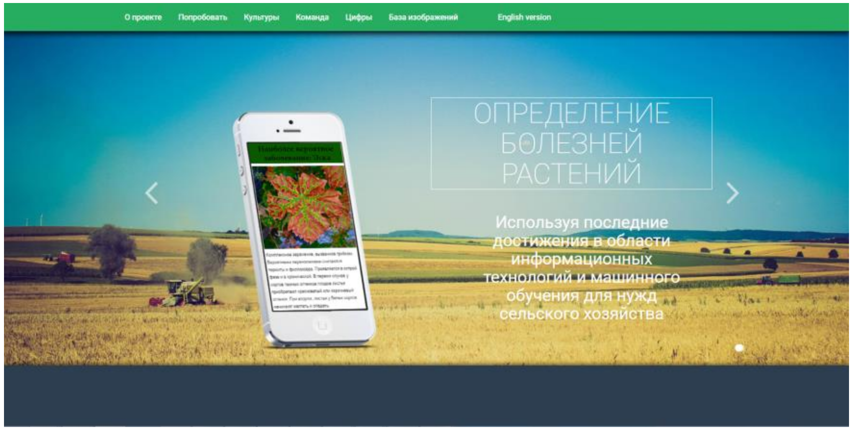
Experts can:
- browse user requests and verify recognition,
- request addition of their image or image from the user complain to the DB,
- request alter of the diseases description,
- request retraining of the model with new images.



Researchers can:
- work with images data base through web-interface or API,
- download all or only part of the base,
- obtain and API-key to submit recognition tasks to the platform.

Supervisors can:
- add new images to the data base,
- initiate retraining of the model,
- get different statistical metrics about portal users.

What we have now



Site with general information (**pdd.jinr.ru**)

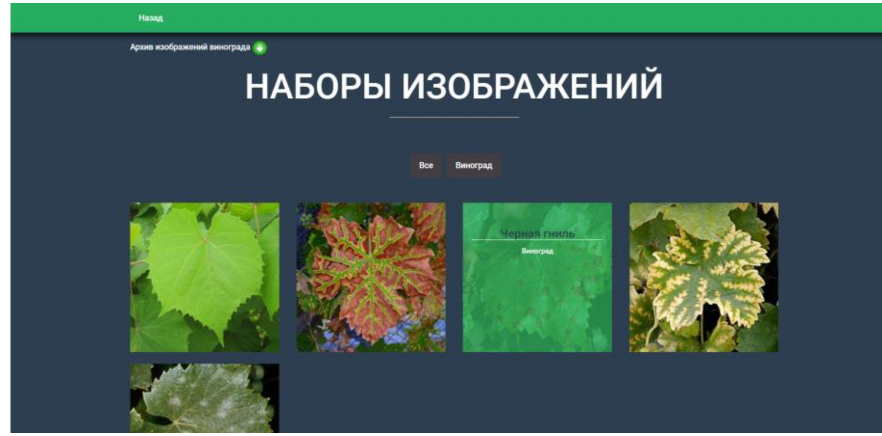
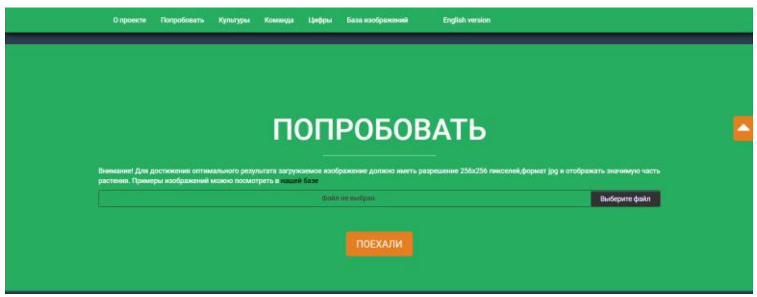
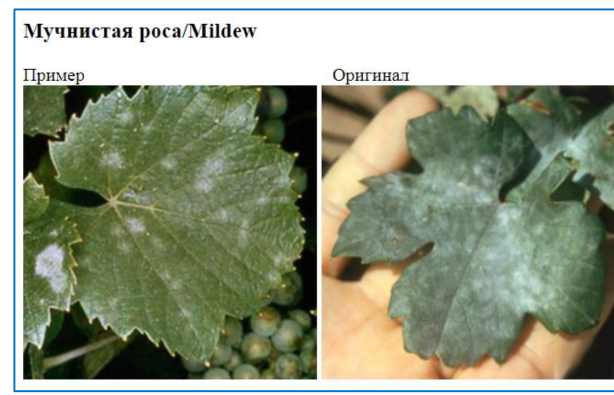


Image database



Ability to submit disease detection jobs at site

Model in Docker running at the virtual server



and get the results

Conclusion and plans

- It is not enough to have lots of images to recognize diseases. Quality of the images database is extremely important for the results of detection.
- Siamese neural networks are very perspective research field for plant disease detection projects.
- It is clear that unambiguous detection of the diseases is unsolvable task especially at first stages of a plant illness.

We are going to use siamese neural networks as a basic deep learning architecture for PDDP and since their power consists in seeking for differences between classes, we are going to add more classes to the train dataset soon.

We will use another deep learning model to define disease by text description. We believes it could improve the detection.

We continue to develop web-portal. Section for users, experts and researchers will be open in first half of 2019

We are going to present draft mobile application in second half of 2019

Thank you for your attention