Multifunctional platform and mobile application for plant disease detection

Alexander Uzhinskiy, Pavel Goncharov, Gennady Ososkov, Andrey Nechaevskiy

Joint Institute for Nuclear Research
auzhinskiy@jinr.ru

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

Symposium on Nuclear Electronics and Computing - NEC'2019
30 09 2019 - 4 10 2019, Montenegro, Budva, Becici

Introduction

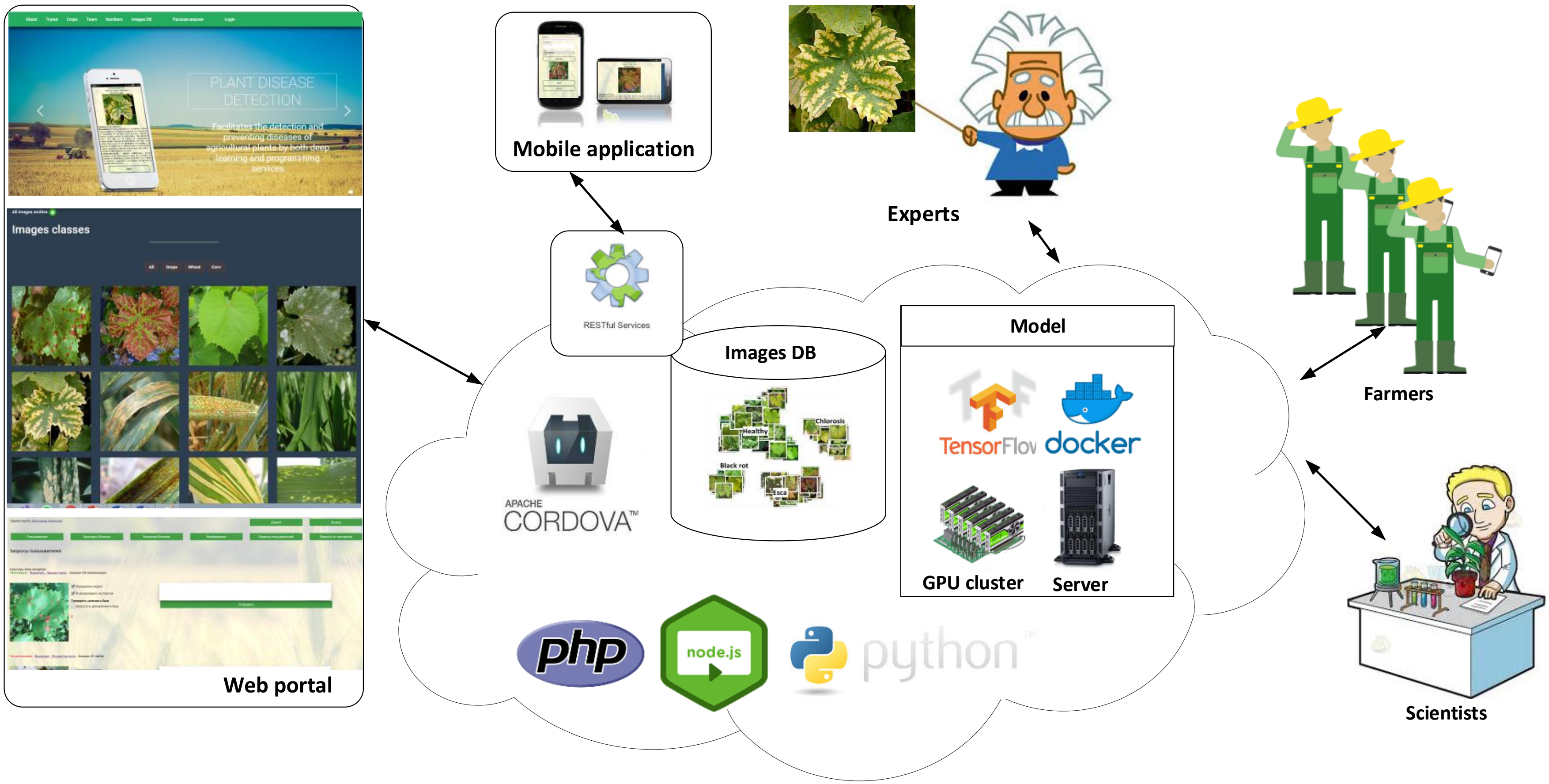
Crop losses are a major threat to the wellbeing of rural families, to the economy and governments, and to food security worldwide
USAblight (a national project on late blight on potato and tomato) says that (annual) global losses '**exceed US$6.7 billion**'.

Globally, about **16% of all crops are lost** to plant diseases each year.
Dr. Caitilyn Allen Department of Plant Pathology, University of Wisconsin–Madison

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.

PDDP Architecture



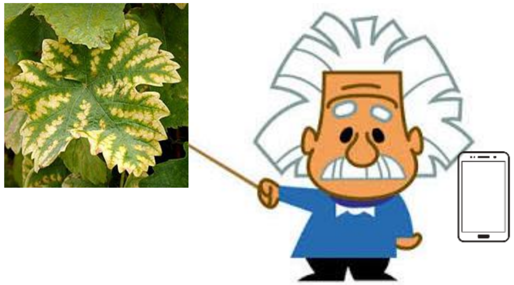
PDDP consists of a set of interconnected services and tools developed, deployed and hosted with the help of the JINR cloud infrastructure. Our web-portal (pdd.jinr.ru), was developed with the Node.js and PHP. It provides not only a web-interface but also the API for third-party services. We have the TensorFlow model in the Docker realized as a Tensorflow serving. The model can work at the virtual server, or at a GPU cluster.

We have a mobile App for Android that was developed using the Apache Cordova, so we could build it for iOS, and Windows.

PDDP abilities

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.

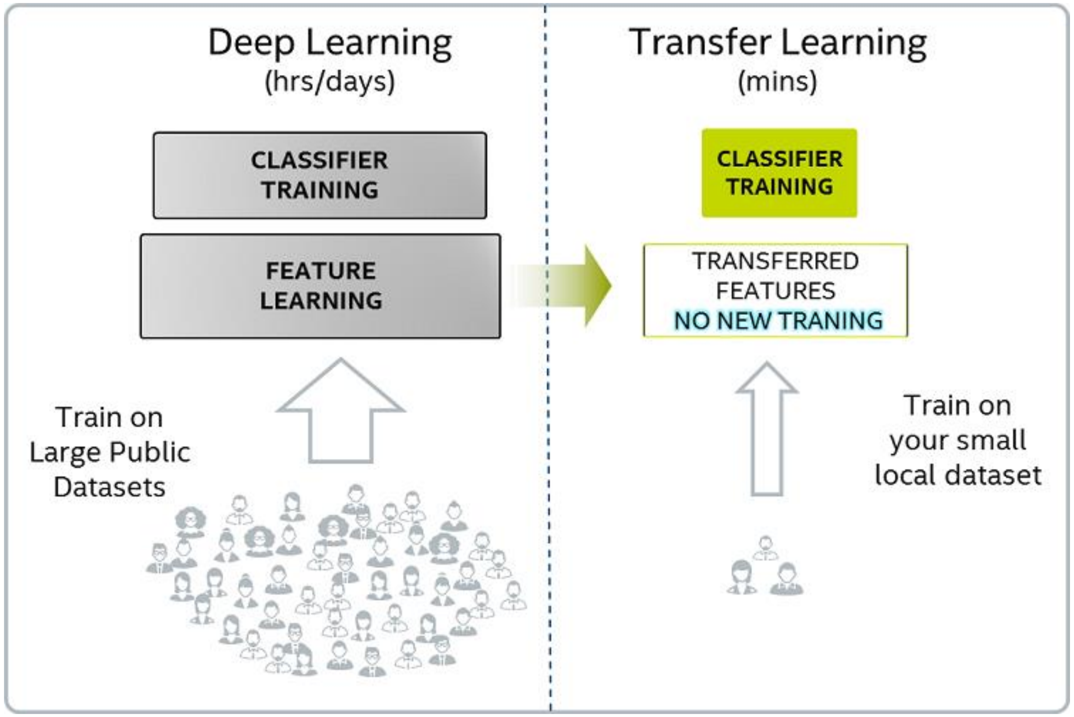
Experts can:
- browse user requests and verify recognition,
- request addition of their image or image from the user complain to the DB,
- request alter of the diseases description,
- request retraining of the model with new images.



Researchers can:
- download all or only part of the base,
- work with images data base through web-interface or API,

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

PDDP Why do we need the database?



Steps of general approach:
1. take a deep network pretrained on a big dataset;
2. fine-tune the chosen deep classifier on the huge images-database (PlantVillage);
3. evaluate it on a test subset of images, collected from the Internet.

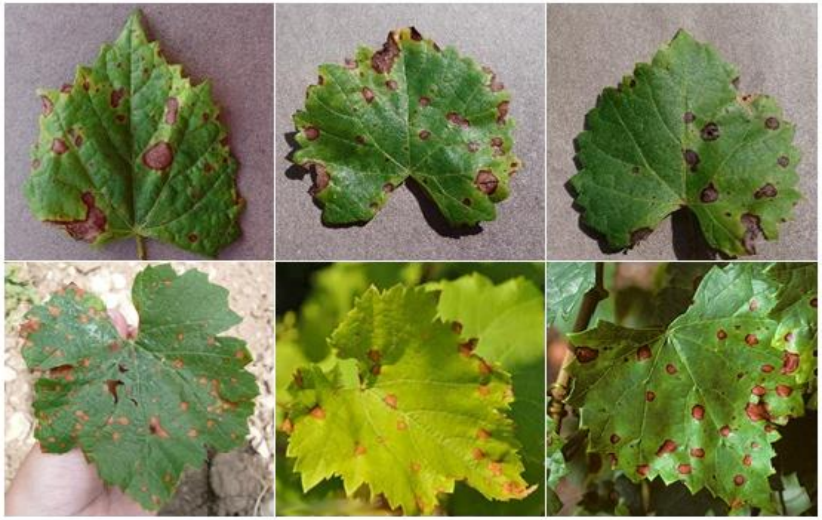
ResNet50 architecture showed the best result:

- accuracy on a test subset of the PlantVillage data – 99.4%;
- accuracy on 30 images collected from the Internet – 48%.

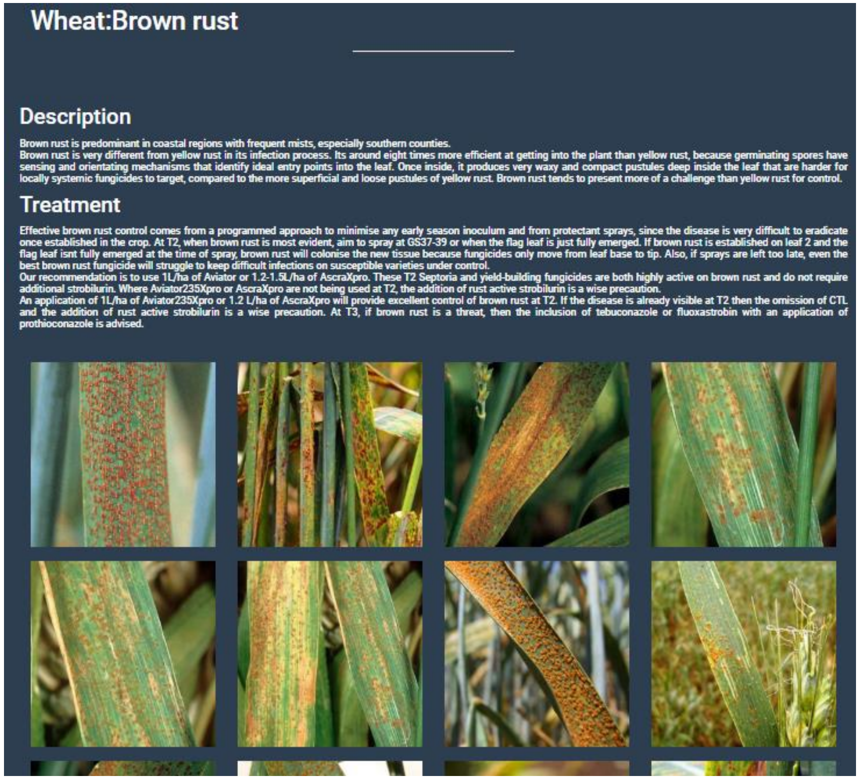
Transfer learning was not effective, but why?

Attentively look on this picture. First row – PlantVillage images, second row – real-life images from the Internet.

Do you see anything strange?



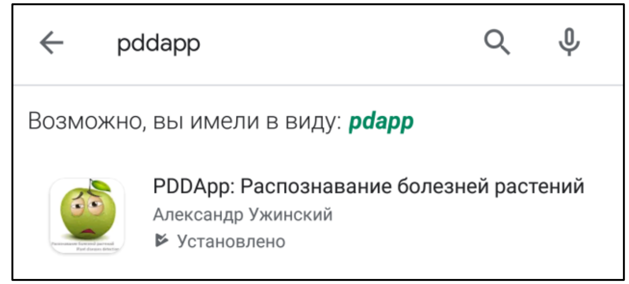
PDDP database (http://pdd.jinr.ru/crops.php)



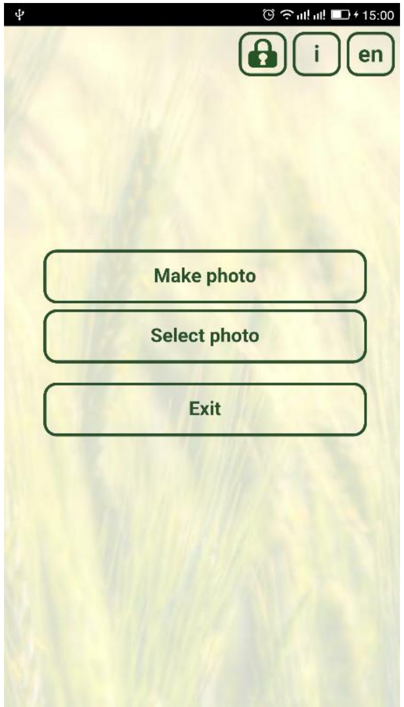
Grape:
Black rot (31) Esca (73) Healthy (121) Powdery mildew (22) Chlorosis (49)
Wheat:
Black chaff (30) Brown rust (31) Healthy (30) Powdery mildew (30) Yellow rust (30)
Corn:
Downy mildew (33) Eyespot (31) Healthy (34) Northern leaf blight (34) Southern rust (32)

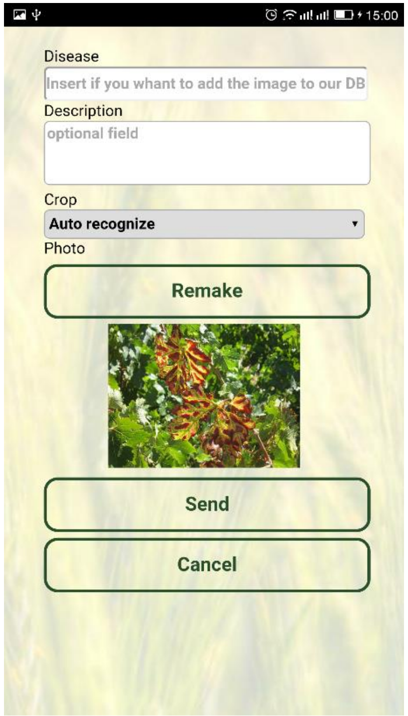
TOTAL: 611

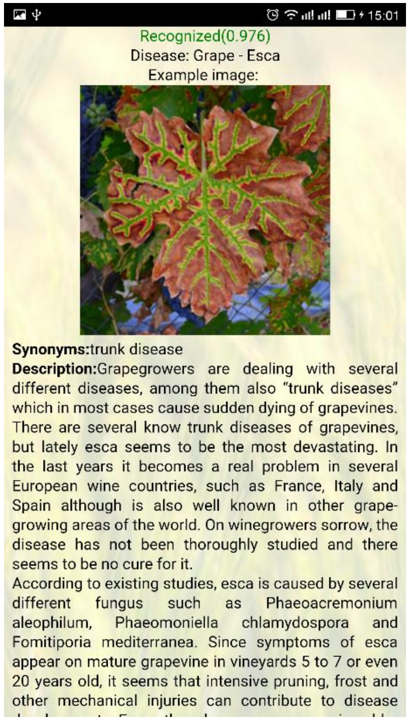
PDDP mobile App



Google play: “Pddapp”. * Only android version is available



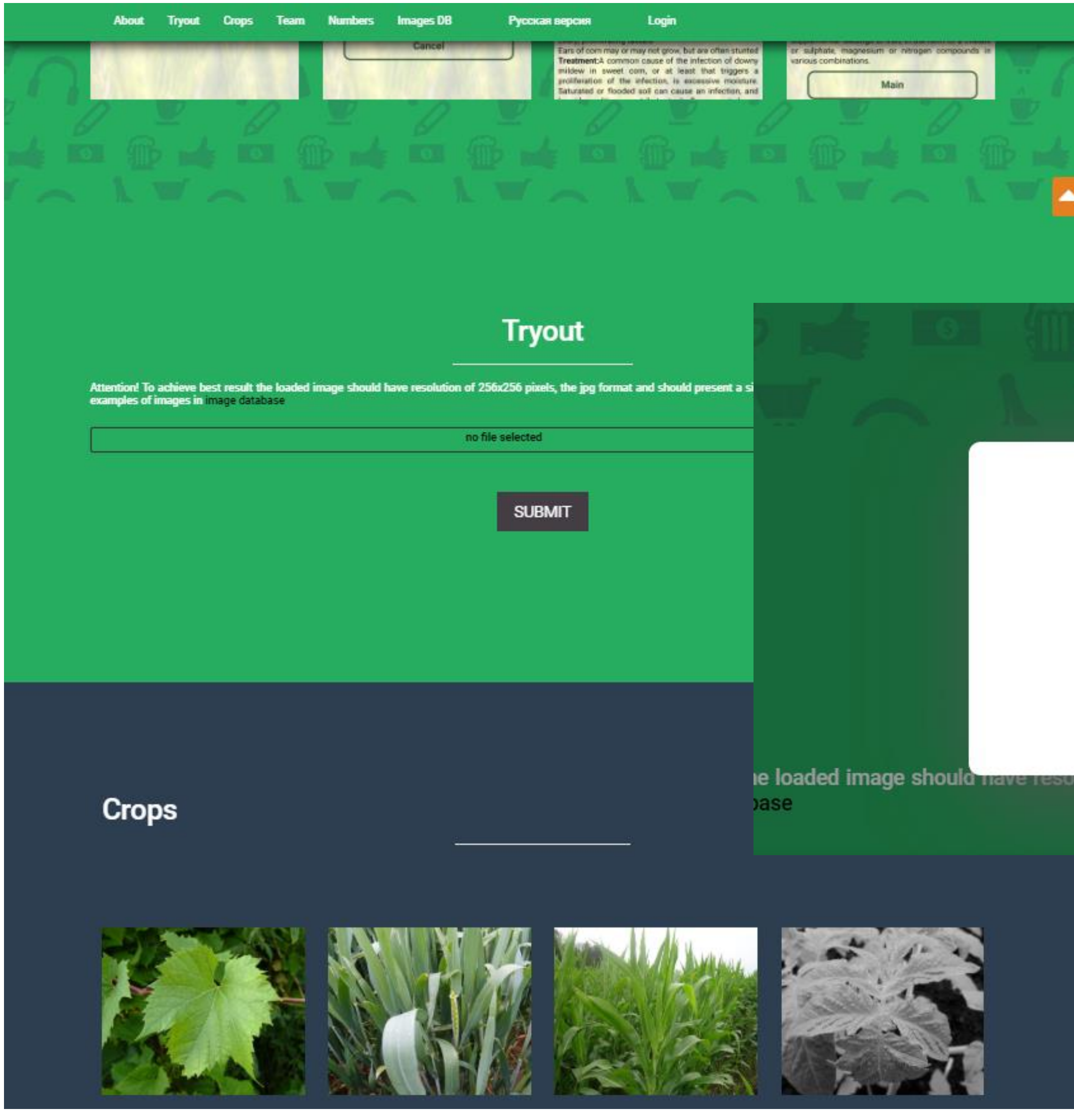




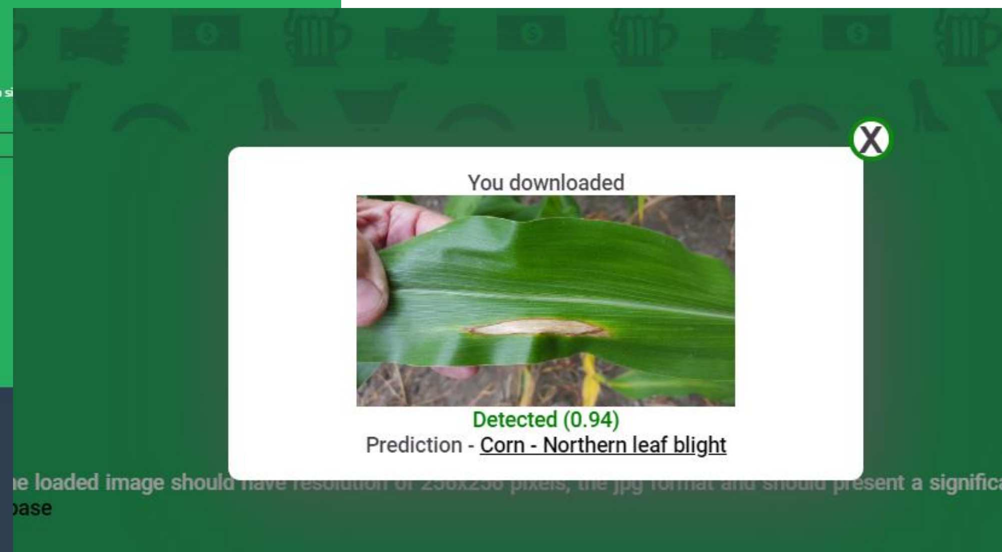
The user has the opportunity to photo a diseased plant and get a prediction for the disease and treatment suggestions. It is possible to download images in the absence of the ability to take a photo. The application requires access to the Internet to work.

We can run the model on the mobile device directly (we have tried it) but we will wait till images and disease description data settles down.

PDDP web-portal (pdd.jinr.ru) public part

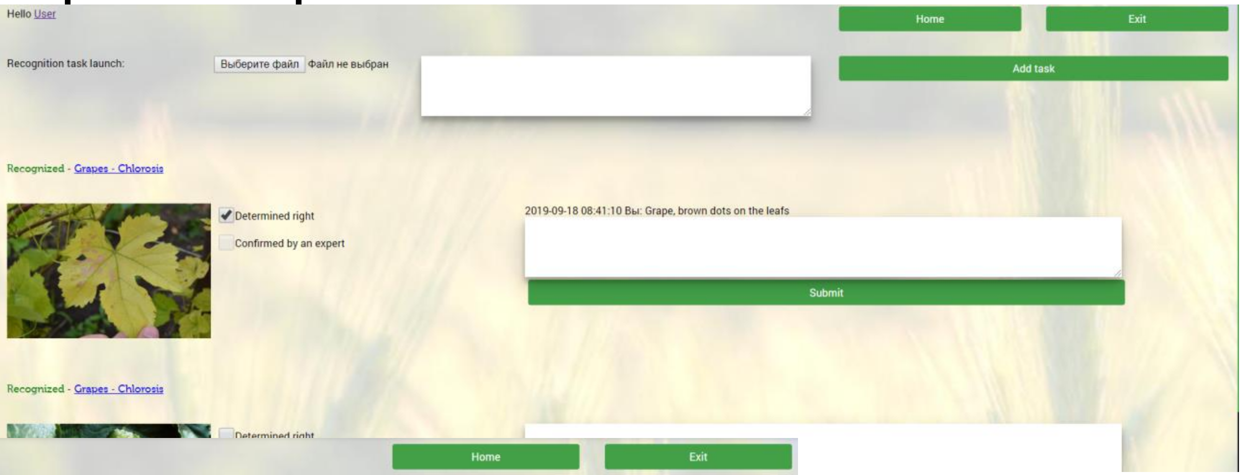


pdd.jinr.ru

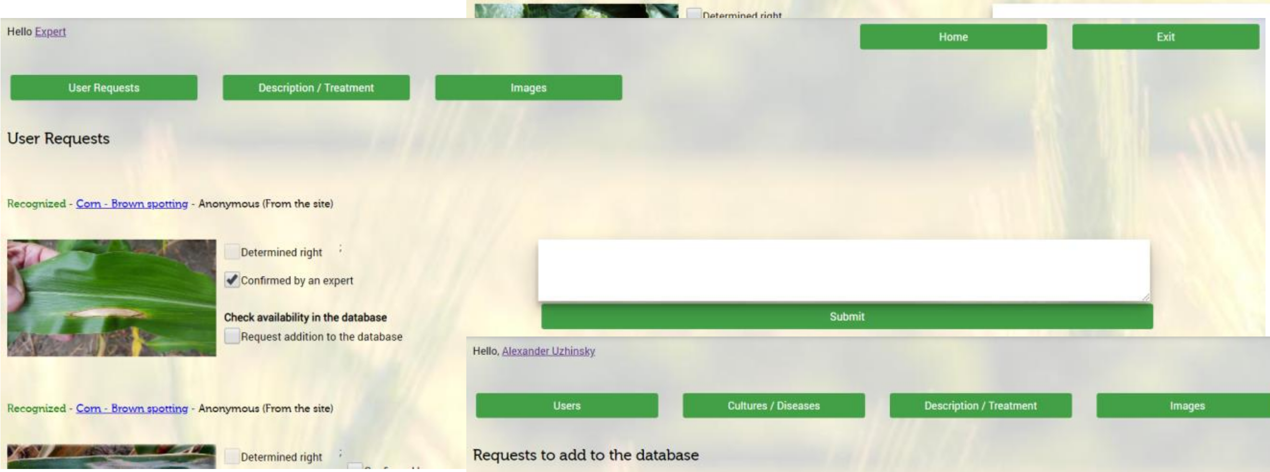


Detection results

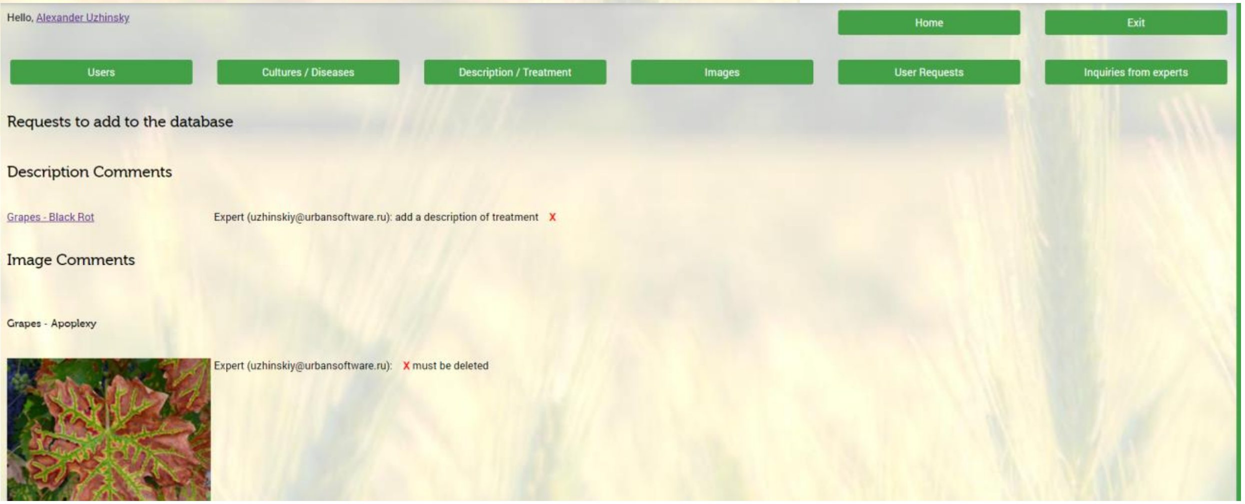
PDDP web-portal private part



User interface



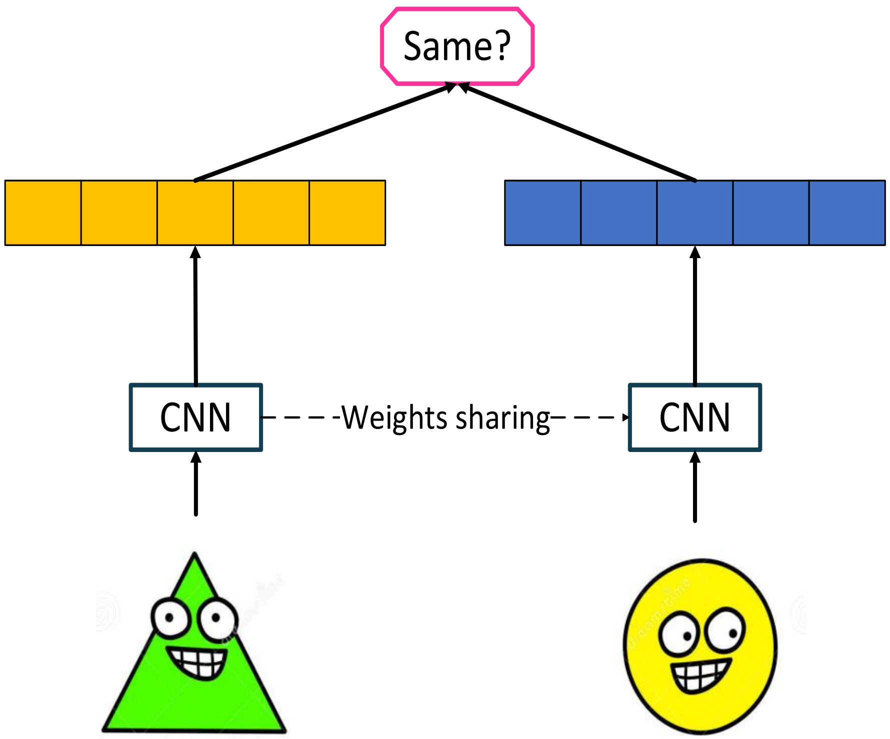
Expert interface



Administrator interface

PDDP What is behind

The only way to train a deep neural network on a small data is one-shot learning, in particular Siamese networks



Siamese network consists of twin networks joined by the similarity layer with energy function at the top.

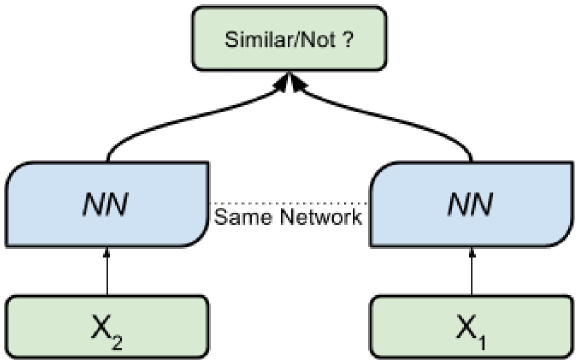
Weights of twins are tied (the same), thus the result is invariant and in addition guarantees that very similar images cannot be in very different locations in features space.

The similarity layer determines some distance metric between so-called embeddings, i.e. high-level features representations of input pair of images.

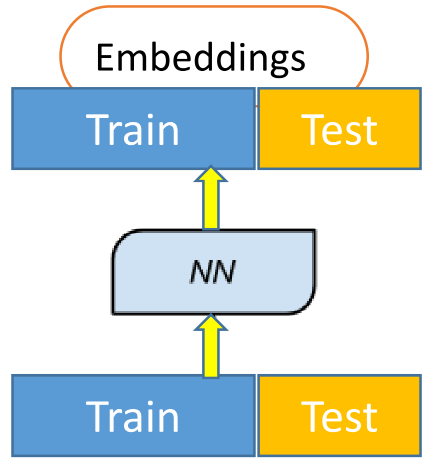
Training on pairs is more beneficial since it produces quadratically more possible pairs of images to train the model on, making it hard to overfit.

PDDP What is behind

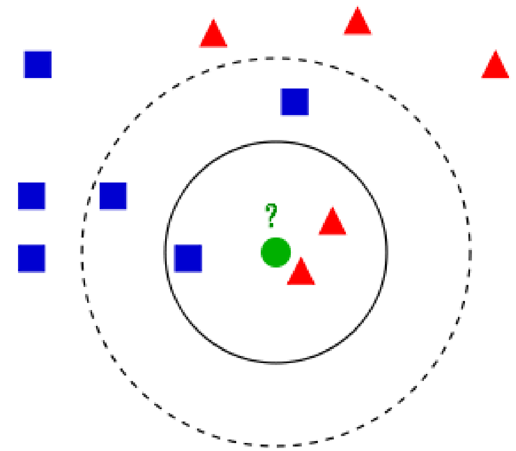
Step 1: Train the siamese network



Step 2: Use one of twins to get high-level features



Step 3: Train and evaluate KNN with 1 neighbor



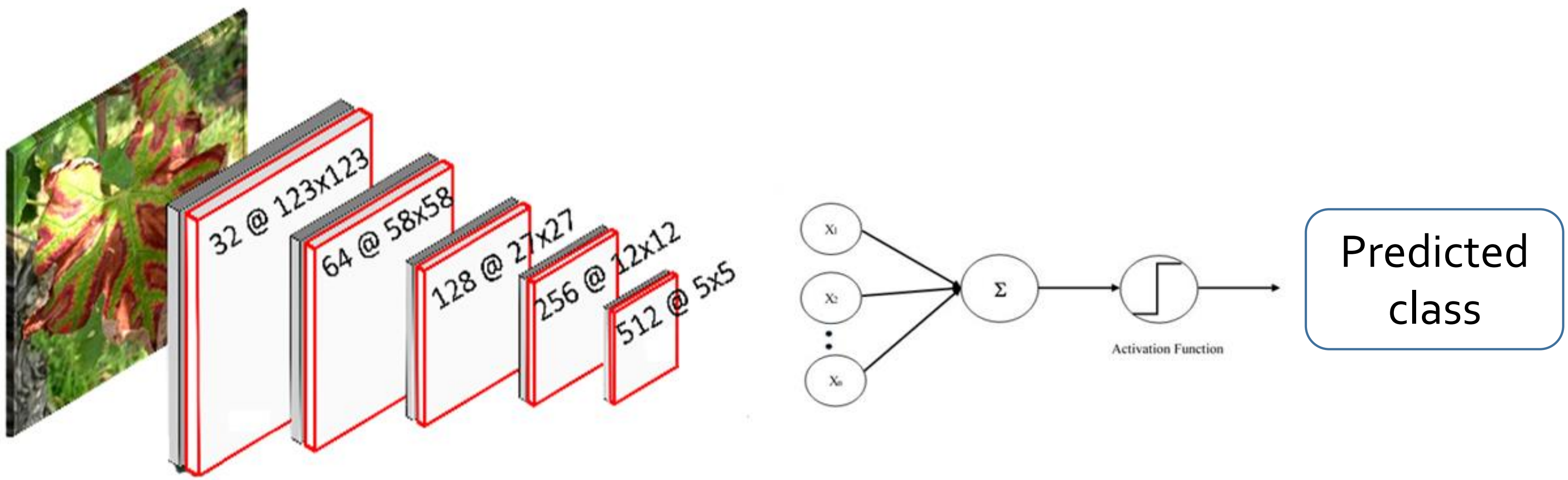
Instead of using classical KNN we formulate the problem of finding nearest neighbors as matrix multiplication of the two l2-normalized matrices of train and validation features.

It works good on 5 classes. **Accuracy equals to 96%.**
But on 15 classes model reached only **86% accuracy** being trained for 150 epochs!

Probably, such a decrease in accuracy value may be caused by using **KNN as a classifier**. It is a well-known fact that KNN suffers from hubs when working with high-dimensional data. A hub is a node which tends to have much more in-going edges than the other nodes. To deal with hubs one can reweight all distances using special scaling parameters, or simply **replace KNN** with another classifier

PDDP What is behind

To improve the test classification accuracy we made a special comparative study of different types of estimators including logistic regression, support vector machines with cosine similarity as a kernel, decision tree, random forest, gradient boosting and a simple perceptron with one input and one output layer ending with softmax activation.



The single-layer perceptron being trained for 100 epochs with Adam optimizer allows us to obtain the classification accuracy equals to 95.71% on the test subset of images. Consider that we add **on top of our trained feature extraction** the **fully-connected neural network layer** with the softmax activation and train the whole model except the fact that the weights of the feature extractor are frozen. It is very **similar to the transfer learning** method,but in our case, the base network was trained directly on our domain data.

Is there something like this already?

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.

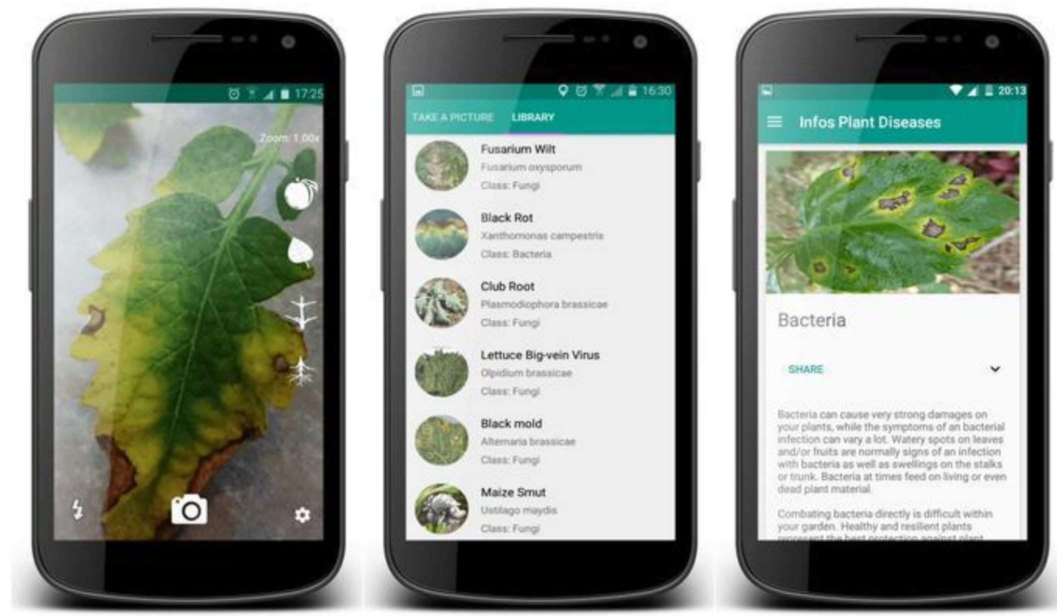
Probably, the most famous mobile application for plants disease detection is Plantix (**plantix.net**). Currently Plantix can detect more than 300 diseases.

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.

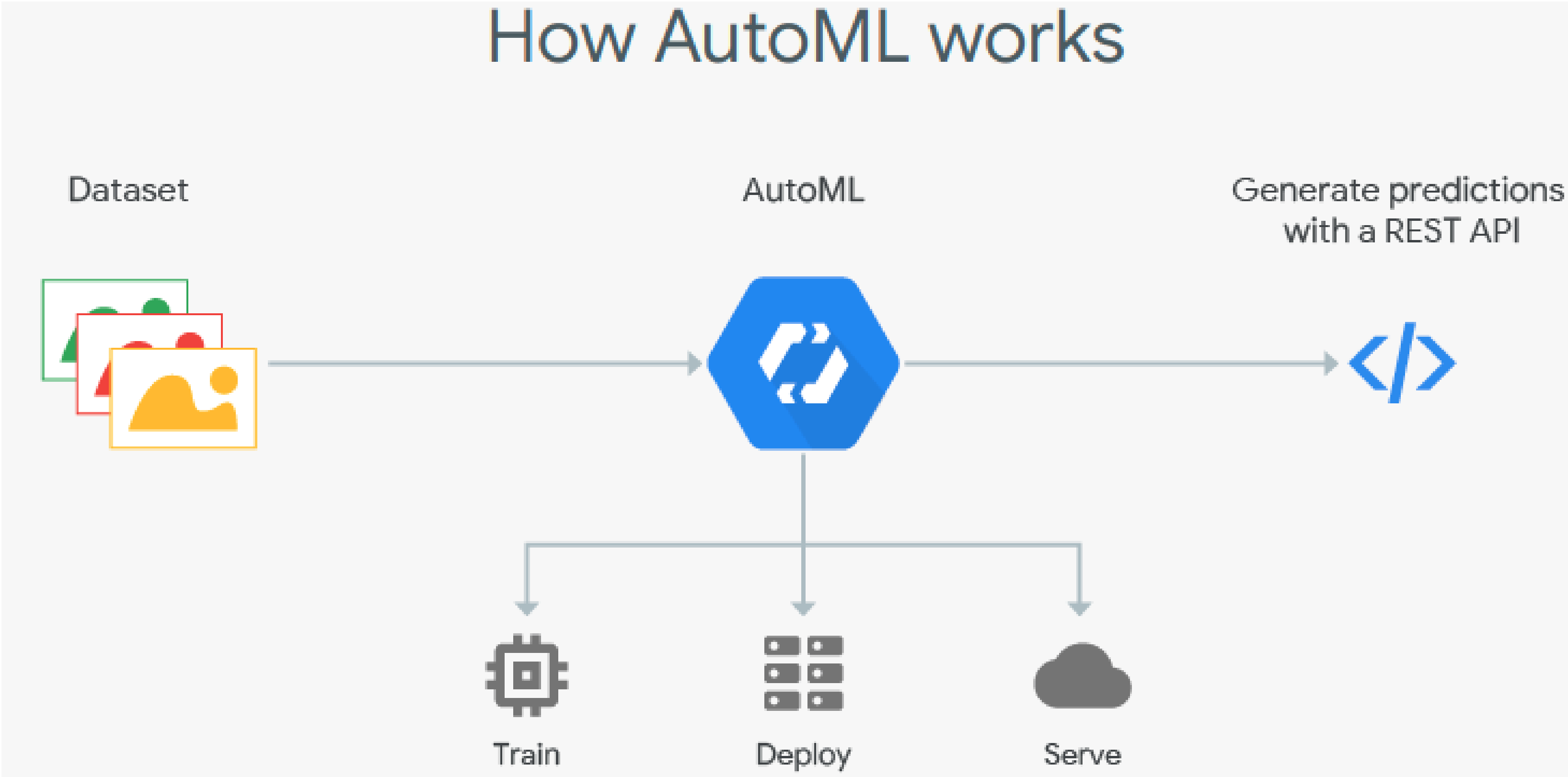
Plantix models evolves a lot last **year**!

A year ago the accuracy of the detection **was over 15%**
Now it's **about 50%**

Fortunately there is no information about their models.

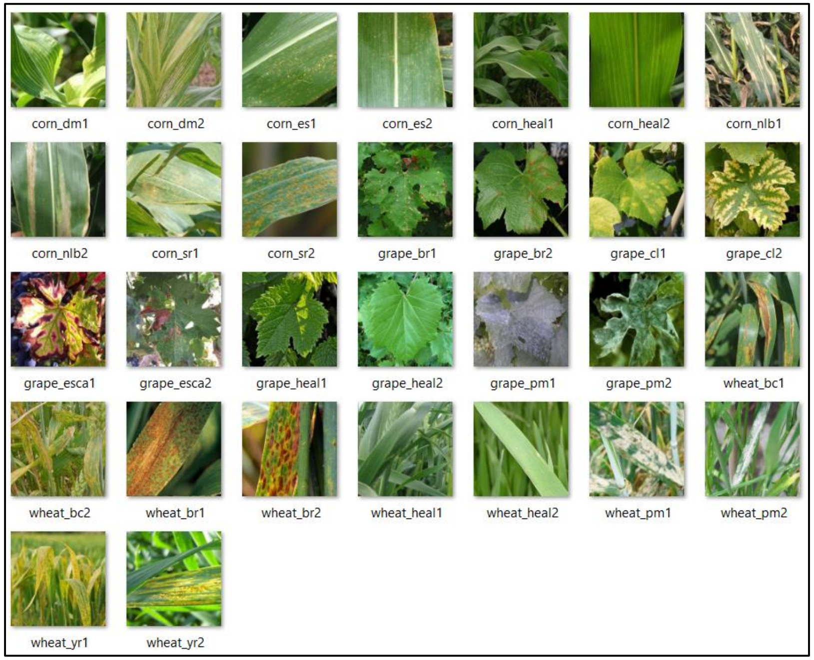


AutoML

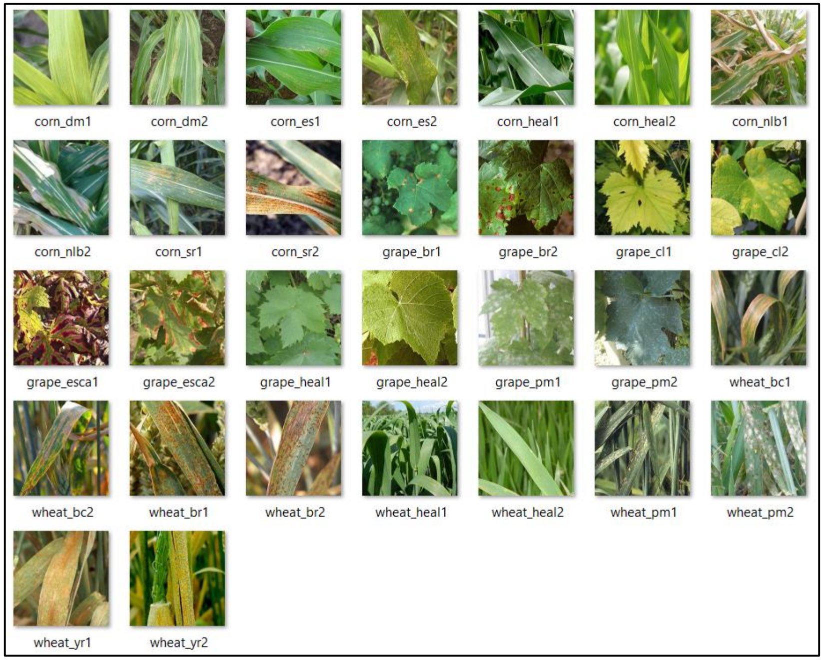


Automated machine learning (AutoML) and **Neural Architecture Search** (NAS) provides methods and processes to make Machine Learning available for **non-Machine Learning experts**, to **improve efficiency** of Machine Learning and to accelerate research on Machine Learning

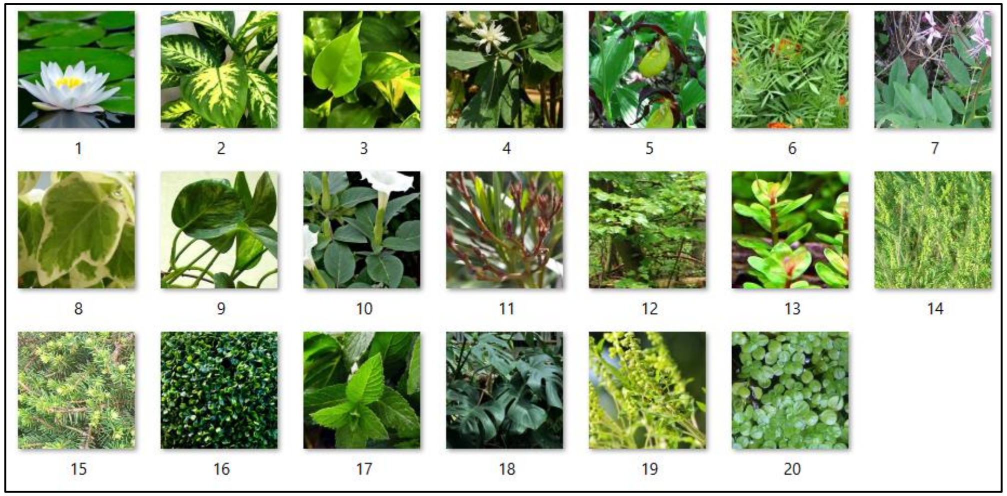
Test dataset



known images (30)



unknown images (30)



not ill plants at all (20)

Our model VS Auto ML (results)

	PDD Old model	PDD New model	Google Cloud Vision	Microsoft Custom Vision	IBM Watson Visual Recognition
Known (30)	27	29	28	29	29
Unknown (30)	20	24	22	25	25
Not ill at all (20)	0	5	1	7	2

We have compared our old and new model with some AutoML system.
In cells, there is a number of correctly classified images.
At the last raw number of images that were detected as a plant with known disease

Plans

We are going to:

- increase the images DB
- improve the mobile App and the Web-portal
- tune and improve our model
- compare our model with other AutoML solutions
- create model for classification by text-description
- localize the App and the portal (Arabic)

Thank you for your attention