AGRISCIENCE AND TECHNOLOGY
Armenian National Agrarian University
ԱԳՐՈԳԻՏՈՒԹՅՈՒՆ ԵՎ ՏԵԽՆՈԼՈԳԻԱ АГРОНАУКА И ТЕХНОЛОГИЯ
International Scientific Journal
ISSN 2579-2822

Journal homepage: anau.am/scientific-journal
doi: 10.52276/25792822-2022.2-149

UDC 632.914.2

Usage of Nvidia Jetson Nano in Agriculture as an Example of Plant Leaves Illness Real-Time Detection and Classification

D.M. Galstyan
"Open Soft Consult" Company. LLC Armenia
E.A. Harutyunyan
"Krisp" Company, Armenia
K.H. Nikoghosyan
"Scylla" Company, Armenia
galstyandavid1998@gmail.com, eduard.harutyunyan.1999@gmail.com, karen.nikoghosyan.98@gmail.com

ARTICLE INFO

Keywords:
smart agriculture,
plant leaves illnesses,
Jetson Nano,
real time detection,
artificial intelligence

ABSTRACT

Early prediction of plant illnesses can reduce the spread of the disease through efficient resource management and treatment planning. The goal of this article is to integrate the Jetson nanomachine with software based on a pre-trained CNN machine learning model which has been trained via healthy and diseased plant image data. The created video server uses the RTSP protocol to send video data to the Detection Engine. Detected diseases of the host are sent through the SMTP server at an adjustable frequency to the emails of the users.

The developed system can be widely used in agriculture, making the fight against plant diseases more profitable and effective.

Introduction

Nowadays, the majority of the world's residential lands are used for agricultural purposes. In such an immense terrain it is impossible to avoid problems and complications related to crop diseases which cause a serious decline in the quantity and quality of agricultural products. The detection of these diseases in large farms requires a lot of time and human resources (www.bitrefine.group). Nowadays, Artificial Intelligence is becoming widespread due to its robust applicability in problems solution, especially those that cannot be handled by humanity. It achieves considerable progress in almost all fields, including agriculture. An automatic plant disease detection system offers obvious advantages in monitoring large fields, since this is the only approach that makes it possible to detect diseases at an early stage, enabling to prevent its spread and avoid financial losses (Gouravmoy, et al., 2018).

Jetson Nano: The Jetson Nano Developer Kit (Figure 1) is a small budget high performance AI device which provides new scope for a lot of compact, energyefficient AI systems. It also allows to parallelly run multiple neural networks for applications such as image classification, object detection, etc. It is an easy-to-use platform with low power consumption (up to 5V) and includes a complete Linux desktop with NVIDIA drivers, AI and artificial vision libraries, APIs and developer tools (www.ximea.com).

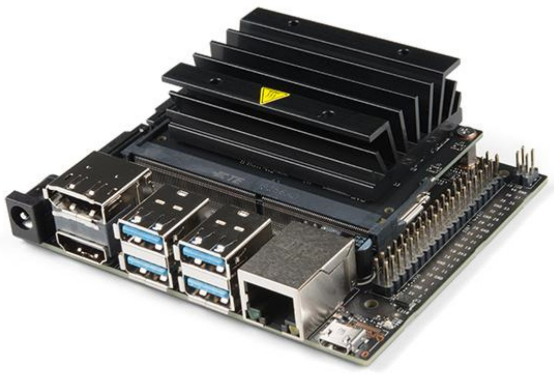


Figure 1. A view of the Nvidia Jetson Nano (www.developer.nvidia.com).

Table 1. Jetson Nano parameters*

Module	Parameters
Processor	128-core NVIDEA GPU, quad-core ARM Cortex-A57 1.43GHz processor
RAM	4 GB 64-bit LPDDR4 25.6 GB/s + SD card slot
Ports	Gigabit Ethernet, MIPI CSI-2 DPHY lanes video camera connector, HDMI 2.0 and eDP 1.4, 4x USB 3.0, etc.

*www.ximea.com

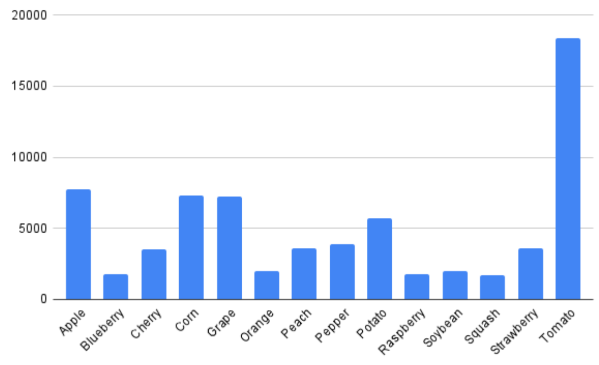


Diagram. Supported plants types and the number of their pictures in the dataset *(composed by the authors)*.

Materials and methods

Model training and dataset software application for the detection of plant diseases has been developed, which is integrated into the Jetson nanomachine. The developed system is based on the Convolutional Neural Network (CNN) (Albawi, et al., 2017) deep learning model. The “Plant Village” dataset from Kaggle has been selected for the model training (www.kaggle.com). The dataset consists of about 87000 RGB images of healthy and ill crop leaves, which are classified into 38 categories. Keeping the directory structure intact, the dataset is partitioned into 80/20 training and validation sets. The test dataset contains 38 images. The dataset contains 14 types of plant images. The quantitative ratio of plant images available in dataset is given in Diagram.

Some examples of the dataset images are also introduced in Figure 2.

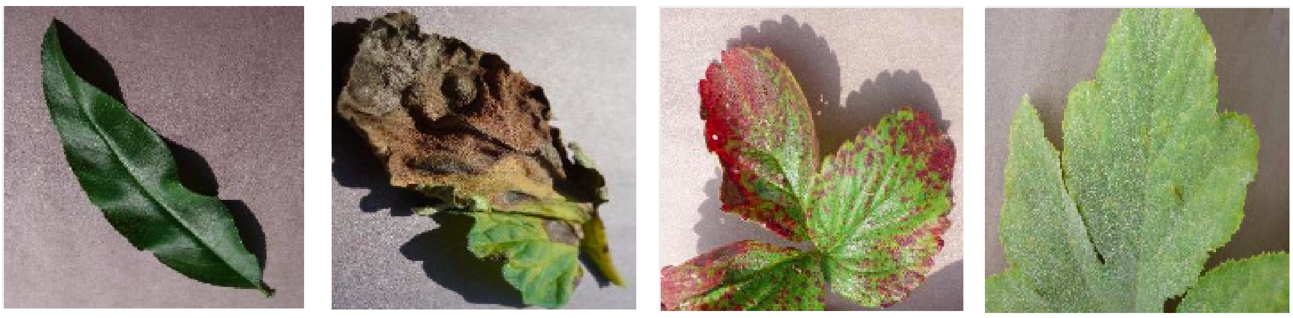


Figure 2. Examples of healthy and ill leaves from the dataset.

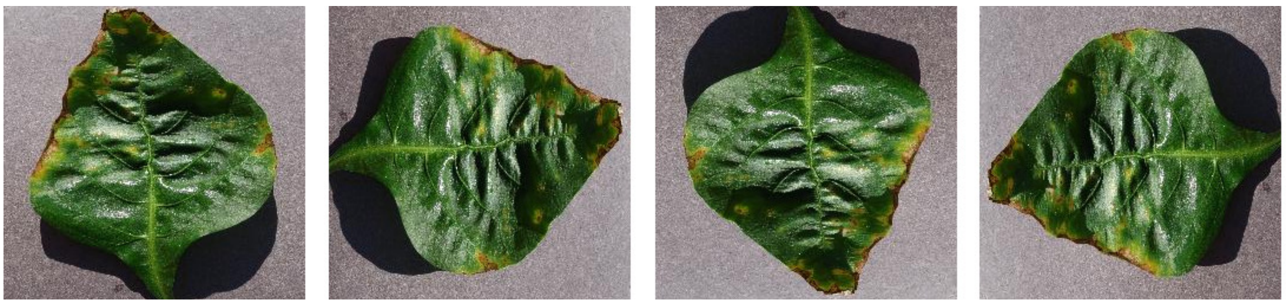


Figure 3. RandomRotation data augmentation example.

Data augmentation

When we have a limited number of elements in the dataset, we must apply data augmentation techniques in order to get a good performance out of the AI model (Figure 3). Moreover, the number of parameters required by your model depends on the complexity of the task it has to perform. We have applied RandomPerspective and RandomRotation methods from the Python PyTorch library to our dataset.

System workflow

The system can be divided into three main parts: RTSP Video Stream Server, AI-Based Detection engine for the disease detection, and SMTP mail server (Figure 4). For the uninterrupted work of system, the cameras used and the Jetson Nano device must be connected to the same network.

RTSP Video Stream Server: RTSP (Real-Time Streaming Protocol) is an application layer protocol designed to control the delivery of multimedia data (Schulzrinne, et al., 1998).

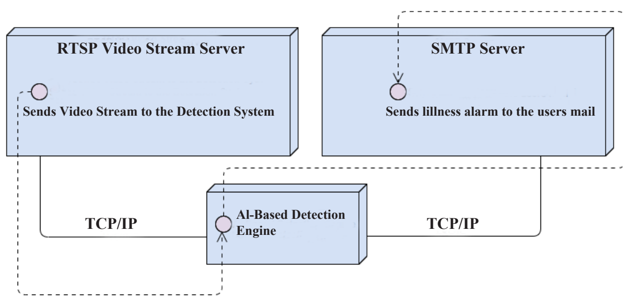

RTSP Video Stream Server
Sends Video Stream to the Detection System
SMTP Server
Sends lillness alarm to the users mail
TCP/IP
AI-Based Detection Engine
TCP/IP

Figure 4. Examples of healthy and ill leaves from the dataset *(composed by the authors)*.

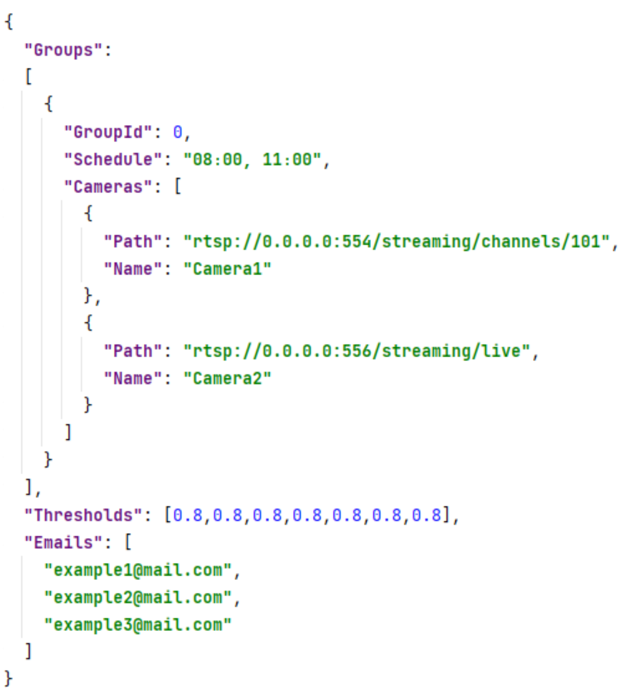
A video stream server using the RTSP protocol was created in this work that will send video data to our detection engine. We have used the OpenCV library (Bradski & Kaehler, 2000) for creating precise stream servers. All cameras that are located in the same network with Stream Server can be used in the system.

SMTP Server: SMTP protocol (Hoffman, 2002) was used to send the alarms to the user. With the help of this protocol, alarms are sent to users in the form of emails. An SMTP Server was created which is responsible for sending the alarms from the engine to the user. The email list is received by the server via the configuration file described in the following chapters.

AI-Based Detection Engine: The AI model described above is the unit responsible for detecting and recognizing plant diseases. In its turn it is divided into two parts. The first part is the Detection, which tries to find parts of the images that resemble some kind of disease. The other part is Classification, the input of which is the crop images obtained after the detection phase. It tries to recognize the type of disease on these crops, and, in case of success, sends collected information as an alarm to all users registered in the system.

System Configuration: A configuration file format has been developed (Figure 5) which contains:

- *Camera groups* - characterized by a unique identification number (GroupId), work schedule (Schedule), additional camera data (Path, Name)
- *Detection Engine* - is an array of numbers in the range [0, 1] that regulates the accuracy of the model. The size of this module should correspond to the number of diseases recognizable by the model
- *Email Addresses* - the addresses to which alarms should be sent in the form of emails.



```
{
  "Groups":
  [
    {
      "GroupId": 0,
      "Schedule": "08:00, 11:00",
      "Cameras": [
        {
          "Path": "rtsp://0.0.0.0:554/streaming/channels/101",
          "Name": "Camera1"
        },
        {
          "Path": "rtsp://0.0.0.0:556/streaming/live",
          "Name": "Camera2"
        }
      ]
    }
  ],
  "Thresholds": [0.8,0.8,0.8,0.8,0.8,0.8,0.8],
  "Emails": [
    "example1@mail.com",
    "example2@mail.com",
    "example3@mail.com"
  ]
}
```

Figure 5. Configuration file *(composed by the authors)*.

When uploading this file into the system the required working data are received.

Results and discussions

The most important parts of any AI-based system are the Detection and Classification models. In the current research work state-of-the-art, efficient artificial intelligence algorithms have been used to obtain a fairly accurate working model. As an activation function for the hidden layers, the ReLU (Rectified Linear Unit) has been chosen. ReLU (Schmidt-Hieber, 2020) has several advantages the most important being the ease of computation and scale invariance. Besides, for networks with randomly initialized weights, as it happens when network only starts the training, only about half of hidden neurons are being activated. So, the addition of the ReLU layer helps to add some non-linearity to the network.

As an activation function for output layers the Sigmoid function (Marreiros, et al., 2008) has been chosen. The main reason for using sigmoid function is that it exists between [0 to 1]. Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, Sigmoid is the right choice.

As an optimization algorithm, Keras Adam (Diederik and Jimmy, 2015) optimizer was used. The full list of the hyperparameters of the model is provided in Table 2.

Plot, showing results achieved during training, is introduced in Figure 6. It indicates accuracies achieved during the training phase. Model was trained with 50 epochs. The best accuracy obtained during training phase was found on epoch 15 and equals to about 98.7%. It's clear that practically, such a result is fairly suitable for the system use on a daily basis in different agricultural farms.

Table 2. The hyperparameters of the AI model used for detection and classification tasks*

Hyperparameter		Value
optimizer	batch size	64
	optimizer	Adam
	leraning rate	0.001
hidden layers count		18
hidden layers activation function		ReLU
output layers count		3
output layers activation function		Sigmoid

*Composed by the authors.

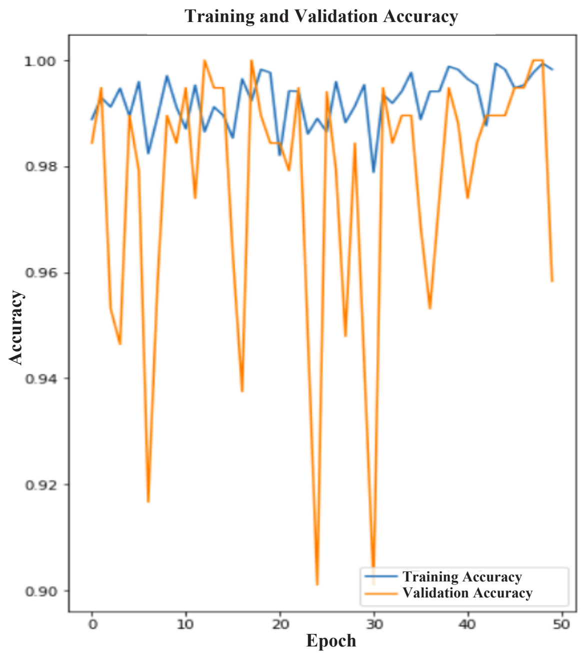


Figure 6. Graphical representation of model accuracy *(composed by the authors)*.

Conclusion

In the result of current research work, the possibility of creating AI-based system has been introduced that will promote automation of disease detection in gardens, greenhouses and other agricultural structures using quite a few financial resources. The proposed system is cost-effective, since it requires Jetson Nano devices, which cost about $340. This is much more profitable and efficient than the computers ensuring similarly smooth and efficient operation of such systems.

In order to improve the system and make it easier for the user, it is planned to create a mobile app and website in the future, where the user can follow his plants and trees in real-time. It is planned to create a new dataset in which images of the most common plants and trees in Armenia should be added. It will enable to improve the efficiency of the AI model and enhance its working capacities.

References

1. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a Convolutional Neural Network. In 2017 International Conference on Engineering and Technology (ICET), - pp. 1-6. https://doi.org/10.1109/icengtechnol.2017.8308186.
2. Bradski, G., & Kaehler, A. (2000). OpenCV. Dr. Dobb's Journal of Software Tools, 3, 2.
3. Diederik, P. Kingma, Jimmy, Ba (2015). Adam: A Method for Stochastic Optimization. 3rd International Conference for Learning Representations, San Diego.
4. Gouravmoy Bannerjee, Uditendu Sarkar, Swarup Das, Indrajit Ghosh (2018). Artificial Intelligence in Agriculture: A Literature Survey. West Bengal, India.
5. Hoffman, P. (2002). SMTP Service Extension for Secure SMTP over Transport Layer Security. RFC 3207, February. https://doi.org/10.17487/rfc3207.
6. https://developer.nvidia.com/embedded/jetson-nano-developer-kit (accessed in March, 2022).
7. https://www.ximea.com/support/wiki/apis/Jetson_Nano_Benchmarks (accessed in March, 2022).
8. Marreiros, A. C., Daunizeau, J., Kiebel, S. J., & Friston, K. J. (2008). Population Dynamics: Variance and the Sigmoid Activation Function. Neuroimage, 42(1), - pp.147-157. https://doi.org/10.1016/j.neuroimage.2008.04.239.
9. Plant Disease Detection: https://bitrefine.group/industries/precision-agriculture/88-industries/agriculture-food/agriculture-solutions/184-plant-disease-detection (accessed in March, 2022).
10. Plant's Dataset: https://www.kaggle.com/arjuntejaswi/plant-village (accessed in March, 2022).
11. Schmidt-Hieber, J. (2020). Nonparametric Regression Using Deep Neural Networks with ReLU Activation Function. The Annals of Statistics, 48(4), - pp. 1875-1897. https://doi.org/10.1214/19-aos1875.
12. Schulzrinne, H., Rao, A., & Lanphier, R. (1998). Real Time Streaming Protocol (RTSP). https://doi.org/10.17487/rfc2326.

Accepted on 25.03.2022
Reviewed on 13.04.2022