Convolutional Neural Network for Studying Plant Nutrient Deficiencies

Rishav Bose[*]
Center for Playware, Technical University of Denmark, Building 326,
2800 Kongens Lyngby, Denmark

Henrik Hautop Lund
Center for Playware, Technical University of Denmark, Building 326,
2800 Kongens Lyngby, Denmark
E-mail: boserishav@gmail.com, hhl@playware.dtu.dk
https://www.elektro.dtu.dk/

Abstract

We discuss the development of a vision-based plant phenotyping system based on a novel type of robotic system called a food computer. The food computer used in this project is called the GrowBot. It has a host of sensors to help analyse the growth chamber including a Raspberry Pi camera. The project revolved around developing a system to segment the plant canopy from its background and analyse nutrient deficiencies from the images taken by the camera. The pilot project investigated how a segmentation model called U-Net could be used to study the images. One of the drawbacks of many existing vision-based plant phenotyping systems is that their convolutional neural networks (CNNs) were trained to analyse very ideal images of individual leaves. This pilot project tried to address that issue, while at the same time explored how to train the neural networks to learn segmentation from a small image dataset.

Keywords: Convolutional Neural Networks, Cyber Agriculture, Food Computing, Image Segmentation, Plant Health Monitoring, U-Net

1. Introduction

During the process of plant growth, plants need to be provided with various nutrients. Proper nutrition is a vital factor that strongly determines many aspects of a plant's life cycle, such as growth rate, flowering, fruit development, and fertilization. A lack of one or more nutrients causes diseases in plants that affect crop yield. Thus, identifying a nutrient deficiency or disease correctly when it first appears is a crucial step for efficient disease management. Nutrient deficiencies manifest themselves as unusual appearances on a plant - especially on its leaves (see figure 1). These symptoms can be identified by visual examination. However, such an analysis relies heavily on the presence of one or more individuals with domain expertise. It is also impractical in large-scale studies. Hence, a number of studies have made an effort to identify nutrient deficiencies and diseases by analysing images of the leaves using convolutional neural networks (CNNs).

A number of studies focused on performing disease classification using the PlantVillage dataset. In the PlantVillage dataset, the images were taken under ideal conditions with each image containing a single leaf, facing upwards, on a homogeneous background. The studies used both transfer learning and train-from-scratch approaches to learn how to predict the crop-disease pairs

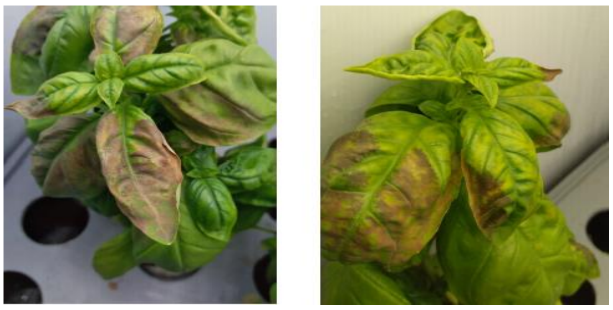
Fig. 1. K defiency in Italian basil (left), Mg defiency in Italian basil (right)

in the images from the PlantVillage dataset. It was reported that the ideal conditions that the diseased leaves were photographed were a drawback for real-world applications. The requirement of a more diverse set of training examples was stressed upon [1][3].
Apart from disease classification, deep learning models have also been applied for the automatic diagnosis of plant disease severity. A fine-tuned VGG16 network was used to classify images of healthy apple leaves and apple leaf black rot images from the PlantVillage dataset that were further annotated with severity labels - early-stage, middle-stage, or end-stage [2].

Marni Tausen et al. [4] and Ke Lin et al. [5] used a U-net-based plant segmentation model to extricate clover plants (Trifolium repens) from their background and to segment powdery mildew on cucumber leaf images, respectively.

2. GrowBot Hardware and Software

The GrowBot [6] (see figure 2) is a robotic system that can control the growth of natural life in the form of edible food plants. It allows its users to parameterize growth conditions for plants in terms of temperature, light cycle, the colour of grow light, fertilization, etc. Such a parameterization is called a recipe. Different recipes bring about different growth patterns in plants in the GrowBot.

Inside the growth chamber, there is room for a water tray with a lid with 16 holes arranged in a 4x4 grid (see figure 3). The seeds of the plants can be sown into small rockwool cubes that are laid down into cups. The cups are then placed into the aforementioned holes. On the front side of the chamber is a door, which can be tightened when closed to ensure an airtight space inside the chamber.

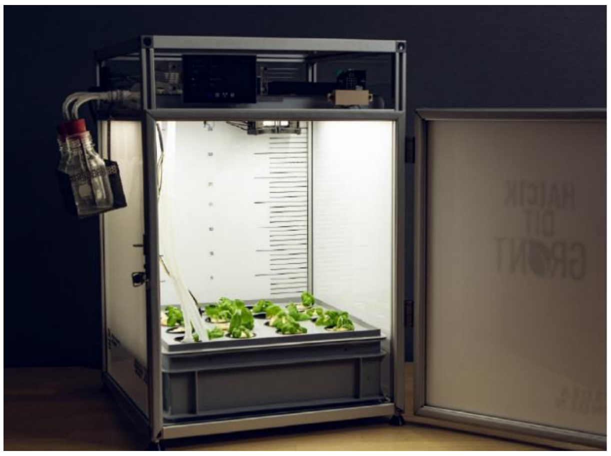
Fig. 2. The GrowBot

The GrowBot is equipped with the OV5647 CMOS image sensor, which is connected to the Raspberry Pi 4 - the main processor of the GrowBot.

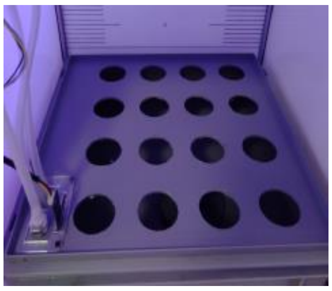
Fig. 3. Water tray in the growth chamber

3. Experiments

The first step of the experiments was to induce a combination of heat and nutrient stresses. For inducing heat stress, the growth chamber was maintained between 29ºC-35ºC. For inducing nutrient stress, nutrient solutions, each deficient in the individual elements, were created. The following stress experiments were performed:

- Heat and Magnesium (Mg) stress - Mg deficient solution was used
- Heat and Potassium (K) stress - K deficient solution was used
- Control I experiment - All nutrients were provided, and the temperature was maintained between 29ºC - 35ºC

- Control II experiment - All nutrients were provided, and the temperature was maintained between 24ºC - 29ºC

Images of the growth chamber were taken during all of these experiments. The training and test set split for the image data is shown in table 1. The train set had 89 images, while the test set had 42 images. The lighting combinations and intensities used while taking the images are shown in table 2. For each stress experiment, two plants were used. To record our dataset, we had placed the plants in unique locations for each stress test. In addition to the varying light intensities, changing the locations of the plants helped generate a lot of variety in the dataset. All the images used in the training, validation, and test sets were normalised, and rigid augmentations - horizontal flip, vertical flip, and random rotations were induced in the training images.

Table 1: Train and test set splits for the project

	Training set	Test set
Name of experiment	Number of images	Number of images
Heat and general nutrient stress	14	9
Heat and Mg stress	32	9
Heat and K stress	21	9
Control I	11	8
Control II	11	6

Table 2: Lighting setups used in the project

Type of coloured light	Intensities
Full spectrum	100%
Full spectrum	100%
Full spectrum	60%
Full spectrum and UV	50% (each)
Full spectrum and UV	30%, 45%

Our goal was to segment the plant canopy from its background and then study the variation of the hue values of the segmented image, as variation in hue has been linked to different nutrient deficiencies [4]. For performing the segmentation, we used a U-Net architecture which has an encoder-decoder structure (see figure 5). The encoder part of the U-net has a typical CNN-like architecture. This is the part of the U-Net that captures contextual information, while the decoder helps in localising the features [7].

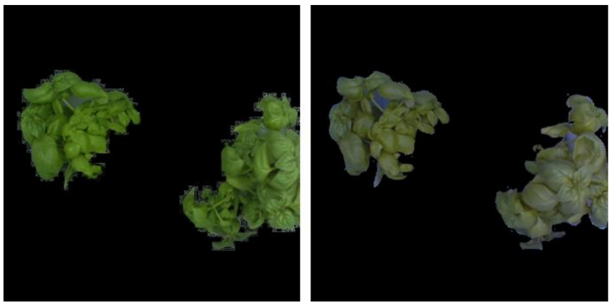
Fig. 4. Noisy images (left), denoised image (right)

The first step before training was to annotate the images, which was done using a tool called ImageJ [8]. One of the challenges with using ImageJ was that when the annotated masks were applied to the image, the resultant image in many cases was noisy around the edges of the plant canopy (see figure 4 (left)). So, the morphological opening operation was used to remove the noise along the edges of the plant canopy in the mask before applying it to the image. In figure 4 (right), we can see the results of applying morphological opening on a mask.

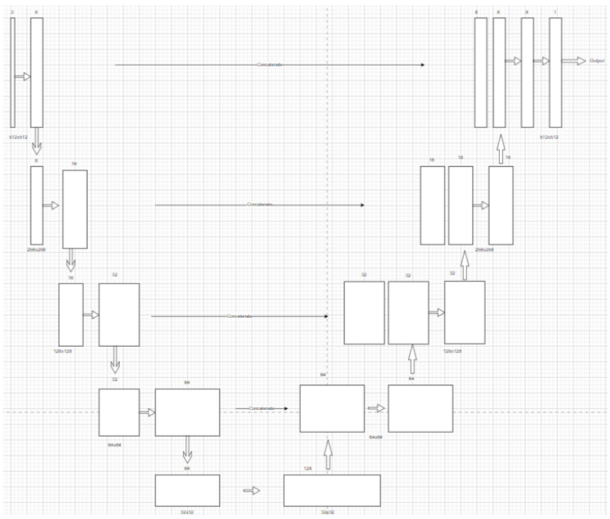
Fig. 5. Structure of the U-Net used in this project

4. Results and conclusion

We tested three models of the U-Net, see Table 3, and the best results on the training and validation sets were obtained with model 2. The training-validation split was

75% and 25%, respectively. Stochastic gradient descent (SGD) was used as the optimiser with a batch size of 1.

Table 3: Details of the three models used

	Model details
Model	Architecture
M1	8x-16x-32x-64x-128x
M2	16x-32x-64x-128x-256x
M3	64x-128x-256x-512x-1024x

The other parameters and the training and validation losses are shown in table 4. Dropout with a probability of 50% was used as the regularization technique to counter over-fitting. An epoch corresponds to the number of passes completed over the entire training set. Dice loss was used to compute the training and validation losses. Dice loss measures the relative overlap between the prediction and the ground truth. The best results on the test set were obtained on the heat and general nutrient test set. Table 5 shows the metrics of the results on the test set. The Jaccard index is a similarity coefficient to gauge the similarity and diversity of sample sets. F1score, recall, and precision take into consideration false positive and false negatives predicted by the algorithm. Lastly, pixel accuracy is the percentage of pixels the algorithm classified correctly.

Table 4: Parameters of the best performing model

Learning rate	Epochs	Dropout	Train loss	Validation loss
1e-4	20	50%	0.6298	0.6179

Table 5: Metrics on the best performing test set

Jaccard Index	F1 score	Recall	Precision	Pixel accuracy
0.4235	0.5914	0.5922	0.5983	0.7997

Figure 6 (left) shows the prediction on an image from the heat and general nutrient test set, and the ground truth is shown on the right. We can see that in the prediction, the algorithm predicts a part of the growth tray as the plant. This is because the ImageJ annotation tool used splines to enclose and annotate the region of interest. In many images, there were parts of the plant canopy that had gaps through which the water tray or the growth chamber walls were visible. Unfortunately, the tool did not have the resources to keep these parts out of the annotation (see figure 7). This hints towards the fact that the

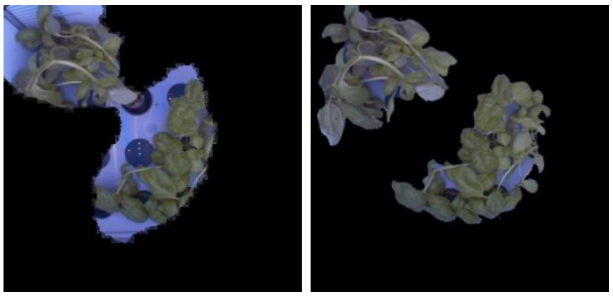

Fig. 6. Prediction (left), Ground truth (right)

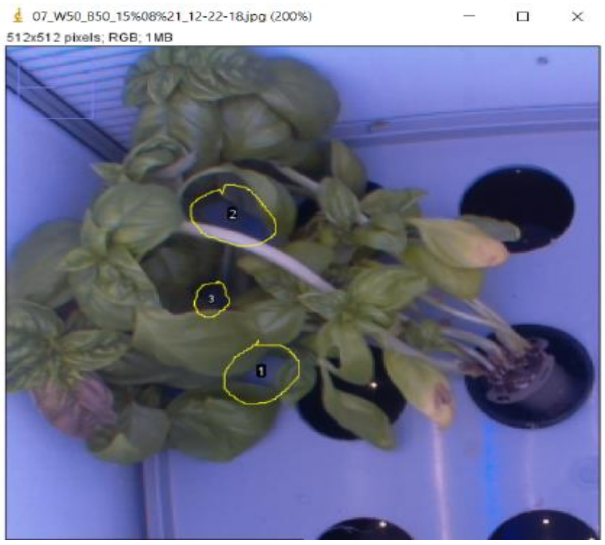

Fig. 7. Drawback of ImageJ

annotation tool was not appropriate for our problem. Unfortunately, due to the fact that the segmentation was recognizing parts of the growth chamber walls and the growth tray as a part of the plant, the analysis of the hue values of the segmented regions were erroneous. However, this problem can be tackled by focusing on the development of a more appropriate annotation tool. Once a proper segmentation is achieved, the hue values of the plant canopies need to be computed, after which we can plot how the hue varies chronologically. This is because variation of hue has been linked to nutrition deficiencies [4].

A remarkable thing to be noted is the time efficiency of the U-Net. It took only 1.0812s to compute the predictions on 7 images on the CPU. This indicates a path towards plant phenotyping on smartphones or low-powered devices. In the future, transfer learning or knowledge distillation can be used to improve accuracy. Additionally, GANs (generative adversarial networks) can be used to generate more data points instead of relying on image augmentation alone.

Acknowledgements

We would like to thank the project partners at DTU Food, DTU Skylab, and Københavns Professionshøjskole. We also express our gratitude towards the funding bodies - Nordea Foundation and Novo Nordisk Foundation.

References

1. Sharada P Mohanty, David P Hughes, and Marcel Salathé. "Using deep learning for image-based plant disease detection". In: Frontiers in plant science 7 (2016), p. 1419.
2. Guan Wang, Yu Sun, and Jianxin Wang. "Automatic image-based plant disease severity estimation using deep learning". In: Computational intelligence and neuroscience 2017 (2017).
3. Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. "Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers". In: Plants 9.10 (2020), p. 1319.
4. Marni Tausen et al. "Greenotyper: Image-based plant phenotyping using distributed computing and deep learning". In: Frontiers in plant science 11 (2020), p. 1181.
5. Ke Lin et al. "Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network". In: Frontiers in plant science 10 (2019), p. 155.
6. Lund, H. H., Exner, M., Jensen, N. E., Leggieri, M., Outzen, M., Ravn-Haren, G., Sehested, M. v., Væring, A., Andersen R. (2022) Robotics for Growing Life. ibid
7. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: International Conference on Medical image computing and computer-assisted intervention. Springer. 2015, pp. 234–241.
8. Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. "NIH Image to ImageJ: 25 years of image analysis". In: Nature methods 9.7 (2012), pp. 671–675.

Authors Introduction

Rishav Bose

Rishav Bose is a Research Assistant at the Center for Playware, Technical University of Denmark (DTU). He completed his masters degree in Electrical Engineering from DTU in October 2021, where he wrote his master's thesis on Optimizing Plant Growth Recipes in Growbots with Machine Learning. His main research interests are food computing, cyber agriculture, and computer vision.

Henrik Hautop Lund

Professor, Ph.D. Henrik Hautop Lund, Center for Playware, Technical University of Denmark, is World Champion in RoboCup Humanoids Freestyle 2002, and has more than 200 scientific publications and 5000 citations. He was awarded the Most Outstanding Healthcare Innovator in the World in 2019. Over the last year, he received international awards in Tokyo, Singapore, and London. He has developed shape-shifting modular robots, presented to the emperor of Japan, and has collaborated closely on robotics and AI with companies like LEGO, Kompan, BandaiNamco, Microsoft, Mizuno. He is the inventor of the Moto Tiles (www.moto-tiles.com), which are used by seniors all around the world.