Autonomous End-to-end Detection of Northern Leaf Blight directly from Aerial Imagery

Harvey Wu[1]*, **Tyr Wiesner-Hanks**[2]*, **Ethan L. Stewart**[2], **Chad DeChant**[2],
Michael A. Gore[2], **Rebecca J. Nelson**[2, 3], **Hod Lipson**[4]
[1]Department of Computer Science, Columbia University
[2]Plant Breeding and Genetics Section, [3]Plant Pathology and Plant-Microbe Biology Section,
School of Integrative Plant Science, Cornell University
[4] Department of Mechanical Engineering and Institute of Data Science, Columbia University

Abstract

Northern leaf blight (NLB) is a foliar disease of maize that causes significant yield losses in North America. Current preventative measures against NLB require manually scouting fields to determine disease prevalence. In this work, we demonstrate an automated, high-throughput system for the detection of NLB in field images of maize plants. Using a small unmanned aerial system (sUAS) to acquire high resolution images, we train a convolutional neural network (CNN) model on lower resolution subimages, achieving 95.0 % accuracy on a test set of subimages isolated from the training process. The CNN model can then be used to create interpretable heatmaps of the original images, indicating where it believes lesions to exist.

1 Introduction

Northern Leaf Blight (NLB) is a fungal foliar disease of maize that has grown more severe in recent years. Between 2012 and 2015, annual estimated yield losses in the United States and Ontario rose sevenfold to 14 million metric tons [1], a loss worth roughly $1.9 billion. To breed maize with improved NLB resistance, plant breeders and geneticists need to accurately quantify infection in field trials. However, the current standard, human experts scoring plots by eye, is subject to high inter- and intra-rater variation [2]. Naturally, it is of both academic and economic interest to develop an quick, accurate, precision phenotyping solution to counter the effects of NLB. Prior work in image-based phenotyping [3] [4] [5] has demonstrated the potential to fulfill the desiderata above, but either at the cost of destructive sampling, or restricting image capture to homogeneous conditions. DeChant et al. [6] introduces an NLB phenotyping system that processes field images in natural condition using deep learning. A dataset of around 1000 manually captured images is used as dataset to train and evaluate an ensemble model of five convolutional neural networks (CNNs). However, capturing photographs with a handheld camera is still prohibitively time-consuming.

We develop an automated phenotyping system that combines the deep learning approach of [6] with sUAS-based imagery. Previous work [7] [8] has used sUASs to image large swaths of farmland rapidly and reliably. Inspired by such results, we tackle the data acquisition bottleneck of a machine learning phenotyping system with sUASs, largely automating the phenotyping process. With a trained model, our system requires almost no human input to perform end-to-end inference.

*equal contribution

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Related Work

Convolutional neural networks (CNNs) are currently state-of-the-art in many computer vision tasks, such as object classification and detection. Part of this success lies in a CNN's ability to perform automated feature extraction, as opposed to classical methods that may require hand-crafted features [9] [10]. We use a particular variant of a CNN, a residual network [11], that uses skip connections between layers to allow gradient information to propagate more effectively through many layers.

Figure 1: Four sample images from our dataset. The two images on the left were captured in August 2017; the right images are from September of the same year. Although the images of each column are visually similar, the top row contains no lesions while each image in the bottom row contains seven.

The work of Mohanty et al. [12] is an early example of deep learning applied to plant science; a CNN is trained on a dataset of 54,306 images to classify 14 different species of plants and 26 different diseases, achieving 99.35% accuracy on a held-out test set. Significantly lower accuracies (<50%), however, were reported when testing their model on images captured in natural conditions. A field image contains considerably more information than a controlled image of a leaf. A reliable machine learning phenotyping system operating on field imagery must learn to generalize between different lighting conditions and altitudes, and distinguish distractor objects such as tassels and weeds from the leaves of interest. [13] demonstrated 93% test accuracy using a transfer learning approach for cassava disease phenotyping from field images taken with mobile phones. This work is the most similar to our approach, with a notable difference: the pretrained model was treated purely as a feature extractor to input into three base models. In this work, a linear layer was appended to the model, and the weights of the pretrained model were trained together.

Previous approaches to deep NLB phenotyping [6] use ensembles of CNNs to produce a heatmaps that are then fed into a final classification network. The CNN is trained from scratch on 224x224 subimages of the original images, and the resulting accuracy on a test set is 96.7%. Our contribution is twofold: one, using transfer learning to speed up training and improve accuracy; two, adjusting the subimage generation process to improve generalization; three, image capture using sUAS systems.

3 Methods

All images were taken in a planting of the Genomes to Fields Initiative. Plants were inoculated as described in [14]. Our dataset contains images of both infected and non-infected leaves between 22 and 84 days post-inoculation (DPI). Images were collected using a Sony alpha 6000 camera fitted with a Sony SEL55210 lense set to 210mm focal length. The camera was mounted to a DJI Matrice 600 UAV flown at a speed of 1.5 m/s and 6m above ground level. The UAV was programed to fly between way points set out in a serpentine fashion across the field and the cameras built-in intervalometer was used to capture an image approximately once per second. There was no overlap among images. For each image, we annotated the semimajor axis of each NLB lesion using a custom ImageJ macro. Images were first filtered automatically by Canny edge detection and discarded during annotation if they were out of focus, contained no maize leaves, etc. A total of 6,267 images are included in the dataset: 3,741 with lesions and 2,526 without lesions. We randomly divided each set of images (infected or non-infected) into training, validation and test sets by a ratio of 70:15:15. The test set was isolated from all aspects of model design, training, and hyperparameter tuning.

Our model is split into two stages: in the first stage, we trained a CNN to predict whether small subregions of an image contain lesions; the second stage uses the CNN as a sliding window over the whole image to generate a heatmap. Unlike [6], which uses a manually designed architecture, we used a Resnet-34 model [11] pretrained on ImageNet [15] as a base model for transfer learning, rather than training from scratch. We appended a linear layer of output dimension 2, and fixed all parameters of the ResNet-34 model besides those of the new linear layer. We trained the linear layer for one epoch, then unfixed the remaining parameters when training in subsequent epochs.

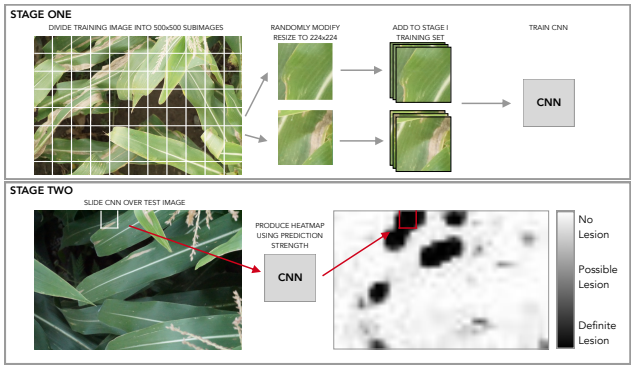


Figure 2: Our two stage pipeline for detection of NLB using deep learning. We randomly sample subimages out of our training images; these subimages are divided into those containing lesions and those without lesions, then used to train a CNN model using stochastic gradient descent. Using a sliding window approach, we take chunks of our original image (see small white square) and feed it into our CNN. The output of the CNN is used as the pixel value for that position in the heatmap. In this image, the final heatmap represents lesions as black and non-lesioned areas as white. The three main black areas in the heatmap (top left, bottom left, and bottom center) precisely overlap with the lesions in the image.

	Positive Prediction	Negative Prediction
True Positive	2790	597
True Negative	89	10261

Table 1: Confusion matrix for our CNN model on a hold-out test set of sub-images. There are 825 misclassifcations out of 13859 subimages. The number of false negatives is large, but in the heatmap generation setting, a model has many opportunities to make correct predictions when used as a sliding window.

We took a different approach to producing training data for our Stage 1 CNN than [6]. A key distinction between hand-held images and drone images lies in the altitude from which the pictures are taken. Whereas the individual lesions in hand-held images may occupy a large proportion of the pixels in an image, at heights of a few meters each lesion will be significantly smaller. We found that when taking subimages of 224 by 224 pixels (the input size for the ResNet-34 model, and the size used in [6]), it was sometimes impossible for even an expert to tell whether the subimage contained a lesion or not, simply because there was not enough context in the image. We modified the procedure by taking a 500x500 selection from our original image of 4000x6000, and labeling it according to whether the center-most 224x224 portion contained a lesion. We then introduced a random variable X sampled from a discrete uniform distribution [-50, 50], and sampled subimages with dimension $500 + X$ by $500 + X$ instead of 500 by 500. After applying other post-processing steps (flips, rotations), we scaled down our modified 500x500 selection to 224x224 and added it to our subimage training set.

In Stage 2, a sliding window of 500x500 is applied over the image, scaled down to 224x224, and fed into the trained Stage 1 CNN. The output of the Stage 1 CNN determines the strength of the region of the heatmap. The step size for the sliding window was fixed at 40.

We performed our experiments on a Google Compute Engine instance with 16GB RAM and an NVidia P100 GPU, as well as a local machine with 16GB RAM and an NVidia 1080 GTX. The ResNet model was imported from the PyTorch Model Zoo. Training was performed using stochastic gradient descent (SGD) with a batch size of 80. After training, we used our model to generate heatmaps on the original 4000x6000 images. We ran the CNN as a 224x224 sliding window over our image and applied the softmax function on the outputs, normalizing them so that they represent a probability distribution. Collecting the component of the output corresponding to "with lesion", we generated a heatmap based on the matrix of these selected values. Each pixel of the heatmap represents an associated portion of the original image - the intensity, or the strength of the softmax output, denotes the CNNs "degree of belief" in lesion presence. [2]

[2]Code will be released.

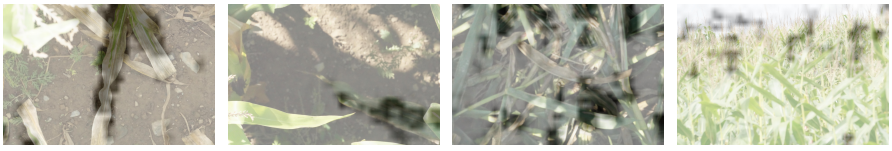
	Lesions	No Lesions
Training	17324	56528
Validation	3730	10404
Test	3384	10350

Table 2: Statistics for subimage dataset. This dataset was used to train our final model.

4 Results

6267 images of maize leaves, comprising of 3741 images containing NLB-infected leaves and 2526 NLB-free images were analyzed in our work. On average, each image of infected leaves had 6.28 labeled lesions, totaling 25508 lesions. Not all senesced leaf tissue in our dataset was due to NLB - other causes included physical damage, natural lower leaf senescence, nitrogen deficiency, corn flea beetle feeding, and other foliar diseases such as northern corn leaf spot. Lesions present due to inoculation were comparable to those caused by natural infection in the noninoculated batch, with similar color and shape. We then reshuffled the data split, and changed the generation process such that only one image per lesion was emitted, and many more negative samples. This model, despite having a lower accuracy of 95.0%, created much more interpretable heatmaps. A confusion matrix for this model is shown in Table 1; dataset statistics are shown in Table 2.

Figure 3: A comparison between an initial CNN model (left, trained with 1:1 lesion to nonlesion ratio), and our final CNN model (right, trained with ratio of around 1:4). The original image is shown in the middle. In both cases, the first model fails to detect many smaller lesions, while the final model is much more sensitive.



(a) (b) (c) (d)

Figure 4: The darker areas are where the model believes there to be a lesion; these areas indeed contain lesions. **(a), (b):**Two examples of our model identifying erroneous labels in the dataset, or "beating the experts". **(c), (d):**Two examples of out-of-distribution inputs on which our models do not achieve good performance: a pile of dead leaves, and a picture of the field taken from a horizontal rather than vertical viewpoint.

Compared to the Stage 1 ensemble network in [6], which made use of individual CNNs with accuracies as low as 81%, we attribute our high accuracies to transfer learning and the modified subimage generation procedure. We hypothesize that the decrease in validation accuracy for the latter is simply due to the fact that those training images do not have enough information - human experts could not identify whether those subimages contain lesions or not. "Enlarging" the image eases the burden on the CNN. While examining the heatmaps produced on the test set, we realized that some of the mistakes made by our model, especially false positives, were actually mistakes in the dataset. Another category of misclassifications belonged to out-of-distribution data, such as images of dead leaves, or different irregular viewpoints (Figure 4).

5 Acknowledgements

This work was supported by the U.S. National Science Foundation National Robotics Initiative grant number 1527232 (M. A. Gore, R. J. Nelson, and H. Lipson).

References

[1] Daren S Mueller, Kiersten A Wise, Adam J Sisson, Tom W Allen, Gary C Bergstrom, D Bruce Bosley, Carl A Bradley, Kirk D Broders, Emmanuel Byamukama, Martin I Chilvers, et al. Corn yield loss estimates due to diseases in the united states and ontario, canada from 2012 to 2015. 2016.

[2] Jesse A Poland and Rebecca J Nelson. In the eye of the beholder: the effect of rater variability and different rating scales on qtl mapping. *Phytopathology*, 101(2):290–298, 2011.

[3] CH Bock, PE Parker, AZ Cook, and TR Gottwald. Visual rating and the use of image analysis for assessing different symptoms of citrus canker on grapefruit leaves. *Plant Disease*, 92(4):530–541, 2008.

[4] Ethan L Stewart and Bruce A McDonald. Measuring quantitative virulence in the wheat pathogen zymoseptoria tritici using high-throughput automated image analysis. *Phytopathology*, 104(9):985–992, 2014.

[5] Céline Rousseau, Etienne Belin, Edouard Bove, David Rousseau, Frédéric Fabre, Romain Berruyer, Jacky Guillaumès, Charles Manceau, Marie-Agnès Jacques, and Tristan Boureau. High throughput quantitative phenotyping of plant resistance using chlorophyll fluorescence image analysis. *Plant Methods*, 9(1):17, 2013.

[6] Chad DeChant, Tyr Wiesner-Hanks, Siyuan Chen, Ethan Stewart, Jason Yosinski, Michael Gore, Rebecca Nelson, and Hod Lipson. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. 107:1426–1432, 06 2017.

[7] Scott C Chapman, Torsten Merz, Amy Chan, Paul Jackway, Stefan Hrabar, M Fernanda Dreccer, Edward Holland, Bangyou Zheng, T Jun Ling, and Jose Jimenez-Berni. Pheno-copter: a low-altitude, autonomous remote-sensing robotic helicopter for high-throughput field-based phenotyping. *Agronomy*, 4(2):279–301, 2014.

[8] Sindhuja Sankaran, Lav R Khot, Carlos Zúñiga Espinoza, Sanaz Jarolmasjed, Vidyasagar R Sathuvalli, George J Vandemark, Phillip N Miklas, Arron H Carter, Michael O Pumphrey, N Richard Knowles, et al. Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review. *European Journal of Agronomy*, 70:112–123, 2015.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[10] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[12] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[13] Amanda Ramcharan, Kelsee Baranowski, Peter McCloskey, Babuali Ahmed, James Legg, and David P Hughes. Deep learning for Image-Based cassava disease detection. *Front. Plant Sci.*, 8:1852, 2017.

[14] Tyr Wiesner-Hanks, Ethan L. Stewart, Nicholas Kaczmar, Chad DeChant, Harvey Wu, Rebecca J. Nelson, Hod Lipson, and Michael A. Gore. Image set for deep learning: field images of maize annotated with disease symptoms. *BMC Research Notes*, 11(1):440, Jul 2018.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[16] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.