High-Throughput Phenotyping of Populus Trichocarpa Using Computer Vision
SMC Data Challenge #1

Vivaan Singhvi, Langalibelele Lunga, Pragya Nidhi, Chris Keum, Varrun Prakash

Abstract

High-throughput phenotyping refers to the non-destructive and efficient evaluation of plant phenotypes. Machine Learning methods offer the prospect of addressing two major challenges in phenotyping plants: efficiency in handling large datasets and extraction of specific traits. Previous techniques, such as the application of Deep Neural Nets in tandem with automated cameras, to our best knowledge, refrain from using physically labeled plants to organize datasets. To accomplish this, we used Optical Character Recognition(OCR) to read the physical labels on the plants, image segmentation techniques(OpenCv, FacebookSAM) in order to apply Machine Learning algorithms(RandomForestClassifier, XGradientBoosting) for plant morphology classifications, and analyzed encoded EXIF tag information for the purpose of finding correlations between phenotypes. We used a Populus Trichocarpa plant dataset provided by Oak Ridge National Laboratory, with 1,672 image files of plants labeled with treatment(Control or Drought), block, row, position, and genotype. We found that OCR techniques, such as PaddleOCR, successfully read 91.3% of the labeled plants, allowing for labeled information to be accurately placed in a pandas Dataframe. The success of the OCR carried into our XGradientBoosting and RandomForestClassifier techniques, which classified leaf morpholgoies by leaf shape and color with an accuracy of 62.82%. Finally, we found [insert findings from step 4]. The use of Machine Learning in High-throughoutPut Phenotyping has shown to be effective in analyzing large plant datasets[Improve this sentence]. It allows for efficient and environmentally friendly classification of plants.

Keywords: computer vision, machine learning, high-throughput phenotyping

1. Introduction

Image-based phenotyping refers to a breakthrough technology used in plant biology and agriculture to examine and assess plants' "anatomical, ontological, physiological, and biochemical features" through the use of images. Previous studies have shown its potential as a noninvasive replacement for traditional on field techniques used to extract important phenotypic data. In recent years, this potential has been further intensified by coupling image-based phenotyping techniques with machine learning algorithms. This new approach, called high throughput phenotyping, has enabled the extraction of phenotypes from "complex" plant image datasets that were previously challenging to analyze efficiently. Furthermore, more recent studies have shown that the technology holds potential in identifying correlations between phenotype, genotype, and environmental metadata.

In this study, we aim to develop an accurate Machine Learning Model for High Throughput Phenotyping of leaves' morphological traits (e.g. leaf size, shape, color) in plant images; unlike datasets in previous research, the dataset used in this study contains physical labels within the images, each containing important information on treatment (Control or Drought), block, row, position, and genotype. As a result, this study will deal with a new aspect of image-based phenotyping namely, extracting data from the white labels to identify correlations between leaves' phenotypes and data embedded within the white labels. Given this information, the study aims to answer the following questions:

1. Is it possible to use optical character recognition (OCR) or machine learning techniques to "Read" the label on each tag and generate a spreadsheet containing the treatment, block, row, position, and genotype? Doing this would dramatically simplify data collection, as this information is usually collected manually.
2. Can machine learning differentiate and classify different leaf morphologies among genotypes by classifying leaf shape or color characteristics?
3. Can a predictive model be built using leaf morphology classifications that may indicate that a particular genotype was cultivated in a "drought" or "control" condition?
4. GPS and other camera information are encoded in EXIF tags. Can this data be used to determine characteristics such as leaf size? Can other data, such as soil maps, weather, etc. be used to find correlations among phenotypes?

2. Related Works

Traditionally, plant morphologies would have to be analyzed by hand, wasting time and potentially damaging the environment. Now, with breakthroughs in high throughput image phenotyping led by Machine Learning techniques, methods such as Automated Machine Learning (Koh et al., 2021), Computer

Vision with Deep Learning (Mochida et al., 2019), and Convolutional Neural Networks(CNNs) (Pound et al., 2017), have allowed for efficient and timely analysis of plant phenotypes. With these methods, researchers are able to extract features from plants and classify plant morphologies effectively with little effect on the plant environment.

2.1. Advancements in High Throughput Phenotyping Techniques

Advancements in the field of plant phenotyping have been pioneered by technology. From using hardware, such as Raspberry Pi imaging systems (Tausen et al., 2020) to thermographical sensing techniques (Walter et al., 2015), research in the field of high phenotyping has led to an acceleration in working efficiency. Plants are now able to be analyzed in fast, effective, and accurate methods allowing for plant to be used in scientific discoveries. In our study, we rely on machine learning techniques to classify morphologies based on a phone image dataset provided.

2.2. Machine Learning Approaches for Phenotypic Analysis

Machine Learning has enabled researchers with computer vision, allowing them to create programs that "read" image information in real time. In addition, it increases the efficiency of obtaining leaf morphology data. However, researchers have found that the use of traditional classification techniques, such as RandomForrestClassifier, have a high performance, but do not generalize well across datasets (Pound et al., 2017). In order to increase accuracy of models and the generalization of the model, Deep Learning techniques have been introduced. With deep learning models, models train themselves iteratively until they reach a desired outcome. These techniques, such as Convolutional Neural Networks (Koh et al., 2021; Pound et al., 2017) have shown to be highly accurate in classifying plant structures while maintaining performance across different datasets. Convolutional Networks consist of several layers, which allow for more discriminative and detailed analysis of the plant image, leading to highly accurate classifications. In our study, we use traditional methods(GradientBoosting, RandomForest) to classify plant morphology, as our focus is on organizing the plant images into an easily readable dataset and performing classifications, instead of creating models highly transferable across datasets.

2.3. Applications of High Throughput Phenotyping with Machine Learning

Due to global events, such as climate change and global population increase, the ability to produce large amounts of healthy food will be crucial to society. With Machine Learning being used for phenotyping, researchers will be able to rapidly analyze food crops to maximize production and crop breeding (Arya et al., 2022; Shakoor et al., 2017). As climate change and global population increases, global crop yield will have to increase to provide for the growing populations. Further research has shown that climate change will continue to negatively affect crops and cause crop diseases (Newton et al., 2011), hence the need for effective and precise analysis of plants, which is done with Machine Learning models. In our study, we aim to show how Machine Learning can be used to organize plant data, dissect plant images into meaningful classifications, and allow for rapid investigation of plants for scientists in the field of plant phenotyping.

3. Methodology

Given the nature of the project, there are many aspects requiring obtaining large files, such as the dataset itself or large models. Therefore, all scripts needed to fully set up the project (downloading large files or setting up a large virtual environment), are viewable in the `scripts/` folder in the GitHub repository, linked in Section 6.

All solutions to the challenge were implemented in Python 3.9/3.10. Several notable libraries used throughout the project include: OpenCV (Bradski, 2000), the open-source computer vision library; Scikit-Learn (Pedregosa et al., 2011), Python implementations of dozens of machine learning algorithms and data processing tools; Pandas (McKinney et al., 2010), a data manipulation library featuring the powerful `DataFrame` object; and NumPy (Harris et al., 2020), an array manipulation library crucial for fast operations on images.

3.1. Reading Labels with Optical Character Recognition

To read the text, a pre-trained Optical Character Recognition (OCR) model seemed like the optimal choice. However, there are several models high-performing models available for use. The three candidates chosen for this project were `PyTesseract` (Lee, 2017), `EasyOCR` (JaidedAI, 2020), and `PaddleOCR` (Du et al., 2020).

The three models were tested using metrics of performance and efficiency, which were measured through their accuracy score and time taken on a sample dataset of nine images. The results of the testing can be viewed in Appendix A.

`PaddleOCR` was superior in speed, being significantly faster than `EasyOCR` and marginally faster than `PyTesseract`. It is also more accurate in both measurements, having a higher accuracy score with and without null values.

In order to fix the aforementioned null values from the OCRs, images went through augmentation, with subsequent attempts being made to read text during each step. First, images are rotated 45 degrees in order to fix potential orientation issues. Then, the original image is thresholded using OpenCV's `adaptiveThreshold` to amplify the edges. Finally, if nothing works, the image is both rotated and thresholded.

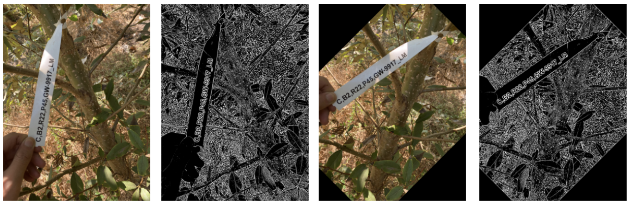

Figure 1: Different versions of images tried for optical character recognition, in order

After the text was read by the OCR, regular expressions (RegEx) were used to find the precise details mentioned in the dataset: treatment, block, row, position, and genotype. A subjective analysis of the images in the dataset showed that the treatment was either a ‘C’ or a ‘D’, and the remaining features followed the RegEx patterns below (note: `\d` represents a digit, anything between 0-9) :

```
Block:      B\d+
Row:        R\d+
Position:   P\d+
Genotype:   [A-Z]{2,}(-\d+)+(_\d+)*(_[A-Z]+)?
```

After all the text is read and processed, it is converted to a Pandas `DataFrame` and saved to an Excel file.

3.2. Classifying Leaf Morphologies with Image Processing

In order to be able to determine the morphological characteristics of the plants, it is necessary to perfectly segment the leaves from each image. This proved to be a challenging endeavor due to the complexity of the background of the images, other datasets, such as the Komatsuna dataset, (Uchiyama et al., 2017) had ’cleaner’ image backgrounds.

To obtain perfect masks for leaves in each image, the Segment Anything Model (SAM) (Kirillov et al., 2023) is the obvious choice. Being a relatively new and groundbreaking model, this paper is one of the first to utilize a pre-trained model of its caliber.

The `AutomaticMaskGenerator` provided generates accurate masks for everything in an image. However, it has two issues: it may generate too many masks (especially for images as cluttered as those in the dataset), and is computationally expensive, taking too long to be a plausible approach for all 1672 images in this dataset. Through a `SamPredictor` in conjunction with an ONNX (Bai et al., 2019) `InferenceSession`, masks would generate much quicker, and only from desired points. However, to obtain these desired points, the images would have to be processed to approximate the locations of the leaves.

The first step was to hide non-leaf elements in the image. A popular technique called HSV filtering was employed, used in a similar project by Szachowicz (2021). This involves filtering all pixels that are not between an HSV-encoded color range, helping eliminate many background objects such as wood, dirt, and of course, the label.

Then, OpenCV’s implementation of the `Canny` algorithm was used to highlight significant edges in the image, which were present around the boundary of leaves but were minimal inside of them. However, some leaves had boundary edges with gaps, causing most contours detected (using OpenCV’s `findContours`) to be sporadic. Therefore, the edges were dilated in multiple iterations by using a large kernel, causing the boundaries to be closed. After contours were found on the resulting image, they were filtered by pruning those that had a height or width (found using the `boundingRect` function around each contour) too small compared to that of the image. Additionally, contours with not enough green inside them (namely less than $\frac{100}{255}$ green) were removed.

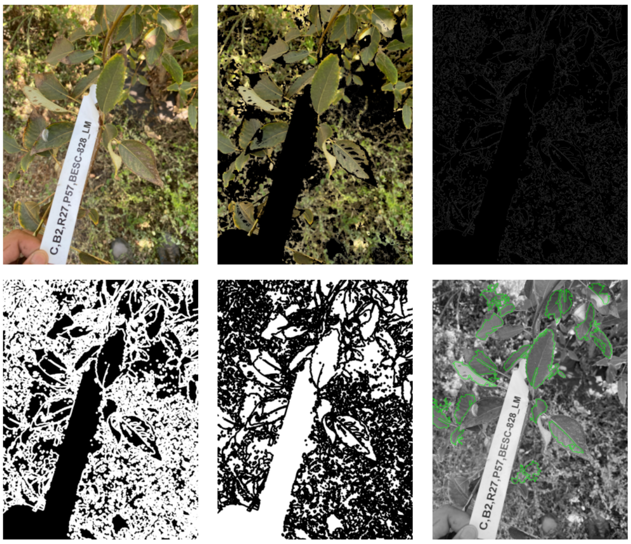

Figure 2: Pipeline of image transformations for leaf approximation, read left to right

To find the approximate points of these leaf contours, the midpoint was calculated using this simple formula, with x, y, w, h being the outputs of OpenCV’s `boundingRect` function:

$$\text{midpoint} = (x + \frac{w}{2}, y + \frac{h}{2}) \tag{1}$$

By using these points as target points for the SAM/ONNX mask predictor, leaves could reliably be obtained from the image. Masks could then be saved for each image by converting each of an image’s masks to a random uniform grayscale value and combining them all into an image. While very accurate, however, some leaves detected by the program were not appropriate for use in morphology analysis. For example, some leaves could be cut off by another object, or, in rare cases, be green wood misinterpreted as leaves.

Thus, it was necessary to train a machine learning model that was able to accurately detect these kinds of ’bad’ leaves. Training data was generated by showing randomly selected leaves and getting human input regarding their suitability, extracting their features, and storing the data in a file. The model was implemented using Scikit-Learn’s `RandomForestClassifier`. The model had an accuracy of 90.91% on testing data generated by Scikit-Learn’s `train_test_split`.

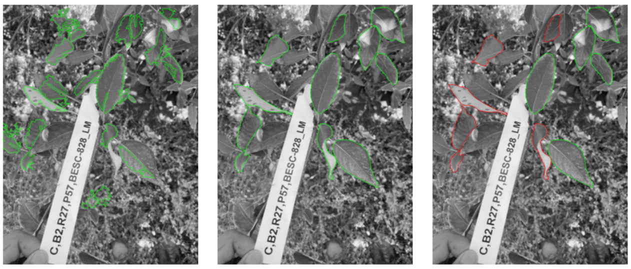

Figure 3: The process for filtering segmentations through the model: from contour approximations to generated masks to filtered masks

Then, each leaf could be cropped and rotated, which, when combined with masking every pixel outside the its border, effectively highlighted and isolated it for visual purposes. By repeating this process for each image in the dataset, random leaves could then be selected to generate data for training the morphological classification model.

By using a slightly modified version of the aforementioned feature extraction function in conjunction with human labeling of leaves regarding their shape, color, and level of brown splotches, a `MultiOutputClassifier` was able to be implemented with Scikit-Learn. More specifically, the `XGBoost` algorithm was used for each of the three predicted features in the `MultiOutputClassifier`.

TO BE CONTINUED (REWRITING NOT FINISHED YET).

3.3. Treatment Predictions using Morphological Classifications

The task for this step was to build a model to predict if a leaf was grown in a control or drought environment, given morphological classifications from the previous step.

The challenge for using classification data as features to train a model with is that they are categorical, rather than quantitative variables; thus, extra transformations must be done to properly represent the data.

To properly represent the classifications, the method of one-hot encoding (MAYBE FIND SOURCE SAYING ITS GOOD) was implemented using the Pandas `get_dummies` method. This is a process in which a feature is split into multiple columns, with each row having a one if it matches the column or a zero if it doesn't.

leaf_color	light_green	dark_green	yellow_green	yellow
light_green	1	0	0	0
yellow_green	0	0	1	0
dark_green	0	1	0	0
light_green	1	0	0	0
yellow	0	0	0	1

Table 1: Example transformation of a column with one-hot encoding, with the original column on the left and the encoded columns on the right

This process was done with the leaf color and shape variables. However, since the level of brown splotches is ordinal, it was assigned levels 0, 1, 2, and 3, corresponding to 'none', 'low', 'medium,' and 'high.' Now, with all the data numerical, it could be used to train and test the model.

3.4. Using EXIF Data to Assess Leaf Size and Analyze Correlations Between Morphologies and Environments

Firstly, we aimed to measure the size of a leaf through metadata, encoded in EXIF Tags, which are embedded within each of the images. These EXIF tags typically give important information about the images, including the type of camera utilized, the distance of the camera from a photo, and photo dimensions (in pixels). In order to initiate the extraction of such metadata from the images, a package called "exif" was used and gave a variety of retrieval options.

Out of all the metadata extracted from the EXIF tags, the most notable pieces of information included ResolutionX/Y tags and focal length. Although these pieces of information are crucial in developing a mechanism to measure leaf size in the dataset, the information provided simply does not suffice in determining leaf size; we needed information such as how far the leaf was from the camera, information that was not given from EXIF tags alone. This lack of information rendered the metadata as inadequate for direct measurement of leaf size.

Another approach we considered to measure leaf size included using the white labels, held up by the researcher in each image, as a known reference object. Although this method seemed promising, there were certain limitations in most images in the dataset that prevented a consistent measurement. Firstly, the white label in the image was often obstructed by the researcher's hands, as shown in Figure 1. This obstruction would ultimately interfere with alignment of the white label, making the label unreliable as an object of reference. Another limitation to this method came from the fact that the labels were often tilted, not in the same plane as the leaves. The tilt of the labels caused inaccuracies in measurement and would lead to misrepresentation of the true length. Along with that, there was no way of finding out how far forward or behind the leaf was relative to the tag, meaning the method would result in an inaccurate representation of the leaf.

In the second part of challenge 4, we aimed to find correlation among phenotypes using geographical features including soil type and weather conditions. Using the EXIF tags, we were able to retrieve the coordinates where the image was taken. We noticed all of the images contained were taken in approximately the same area, meaning weather would be the same for all plants. Along with that, most soil map APIs contain information about the soil in an area, so it would be unable to distinguish miniscule changes in locations. For all other potential geographical changes, due to the close proximity all images are taken in, we were unable to find out if there were any correlations. Not only was the location the same, the images were taken in a short time frame, within a week. This gave us the same limitations as the close location had.

4. Results

4.1. Reading Labels with Optical Character Recognition

4.2. Classifying Leaf Morphologies with Image Processing

4.3. Treatment Predictions using Morphological Classifications

4.4. Using EXIF Data to Assess Leaf Size and Analyze Correlations Between Morphologies and Environments

Overall, the metadata embedded in these specific EXIF tags did not prove to be enough for direct measurement of leaf size. Despite two alternative approaches being used, there were always absences in crucial information. Future studies could include additional calibration features to enable the development of a more feasible method for leaf size assessment. For the second part of the challenge, we needed larger differences in location or times that the images were taken in order to study how the conditions in those periods affected the phenotypes.

5. Conclusions and Significance

High throughput phenotyping has gained prominence due to its potential in solving a wide variety of agricultural problems. With the world population projected to reach 9.3 billion people by 2050 and a need to produce 60% more food (Silva, 2012), it will become crucial for researchers to discover a method for identifying productive genotypes for plant breeding. With machine learning methods, such as Optical Character Recognition (OCR), classifiers such as RandomForest, and image segmentation techniques, productivity in the field of plant/crop production will increase rapidly, along with the accurate analysis of plants.

By successfully applying an OCR to read labels, our research shows potential for automating a data extraction process. This process will significantly reduce the manual labor needed to organize information and allow for larger datasets to be handled with ease.

In addition, the use of XGradientBoosting and the RandomForestClassifier for the classification of plant phenotype allows for precise examination of plant morphologies. Although the achieved accuracy of our RandomForestClassifier, 62.82%, shows room for improvement, it highlights the potential the method holds in classification. Future optimization may enable higher accuracy whilst still providing a feasible method for non-invasively extracting valuable information about plant morphologies from onsite images.

Finally, the analysis of EXIF tags provided by the dataset shows promise in estimating the dimensions of an image as well as identifying correlations between plant phenotypes. Although the EXIF tags did not prove sufficient for these specific tasks, we have outlined a few crucial data points that may be helpful/necessary in order to assess dimensions and identify correlations in future studies.

6. Code Availability

All code is publicly available under the MIT License on GitHub here: `https://github.com/vivaansinghvi07/smoky-mountain-data-comp`.

Appendix A. Results of OCR Model Testing

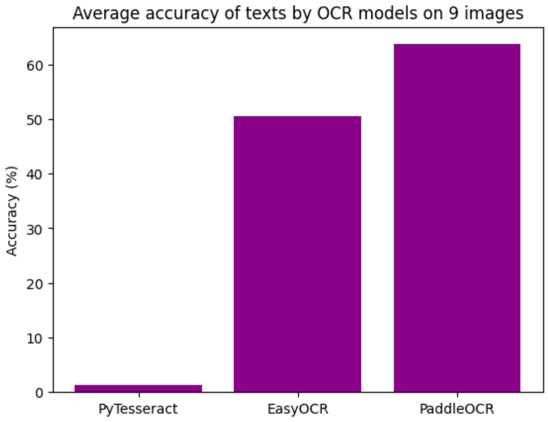

Figure A.4: Average accuracy score for reading the sample dataset for the three OCR models

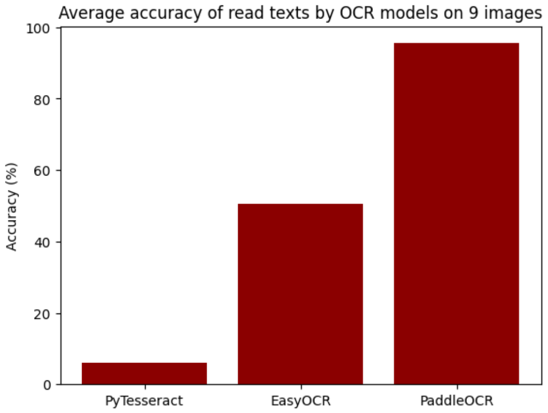

Figure A.5: Average accuracy score only including non-zero values (only images that were able to be read)

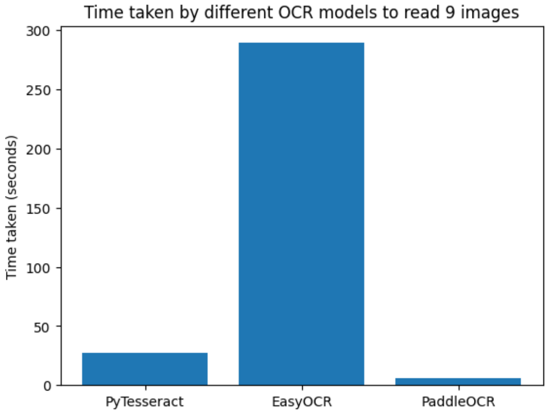

Figure A.6: Total time taken for the three OCR models on the sample dataset

References

Arya, S., Sandhu, K.S., Singh, J., Kumar, S., 2022. Deep learning: As the new frontier in high-throughput plant phenotyping. Euphytica 218, 47.

Bai, J., Lu, F., Zhang, K., et al., 2019. Onnx: Open neural network exchange. URL: `https://github.com/onnx`.

Bradski, G., 2000. The OpenCV Library.

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., Wang, H., 2020. Pp-ocr: A practical ultra lightweight ocr system. URL: `https://github.com/PaddlePaddle/PaddleOCR`, `arXiv:2009.09941`.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. URL: `https://doi.org/10.1038/s41586-020-2649-2`, doi:`10.1038/s41586-020-2649-2`.

JaidedAI, 2020. Easyocr. URL: `https://github.com/JaidedAI/EasyOCR`.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R., 2023. Segment anything. URL: `https://github.com/facebookresearch/segment-anything`, `arXiv:2304.02643`.

Koh, J.C., Spangenberg, G., Kant, S., 2021. Automated machine learning for high-throughput image-based plant phenotyping. Remote Sensing 13, 858.

Lee, M.A., 2017. Python-tesseract. URL: `https://github.com/madmaze/pytesseract`.

McKinney, W., et al., 2010. Data structures for statistical computing in python, in: Proceedings of the 9th Python in Science Conference, Austin, TX. pp. 51–56.

Mochida, K., Koda, S., Inoue, K., Hirayama, T., Tanaka, S., Nishii, R., Melgani, F., 2019. Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective. GigaScience 8, giy153.

Newton, A.C., Johnson, S.N., Gregory, P.J., 2011. Implications of climate change for diseases, crop yields and food security. Euphytica 179, 3–18.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Pound, M.P., Atkinson, J.A., Townsend, A.J., Wilson, M.H., Griffiths, M., Jackson, A.S., Bulat, A., Tzimiropoulos, G., Wells, D.M., Murchie, E.H., et al., 2017. Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. Gigascience 6, gix083.

Shakoor, N., Lee, S., Mockler, T.C., 2017. High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field. Current opinion in plant biology 38, 184–192.

Silva, J.G.D., 2012. Feeding the world sustainably. The Future We Want? 49.

Szachowicz, J., 2021. The komatsuna dataset with python and opencv. URL: `https://github.com/julzerinos/python-opencv-leaf-detection`.

Tausen, M., Clausen, M., Moeskjær, S., Shihavuddin, A., Dahl, A.B., Janss, L., Andersen, S.U., 2020. Greenotyper: Image-based plant phenotyping using distributed computing and deep learning. Frontiers in plant science 11, 1181.

Uchiyama, H., Sakurai, S., Mishima, M., Arita, D., Okayasu, T., Shimada, A., Taniguchi, R.i., 2017. An easy-to-setup 3d phenotyping platform for komatsuna dataset, in: Proceedings of the IEEE international conference on computer vision workshops, pp. 2038–2045.

Walter, A., Liebisch, F., Hund, A., 2015. Plant phenotyping: from bean weighing to image analysis. Plant methods 11, 1–11.