Application of deep learning and machine learning workflows for field-scale phenotyping

Seyed Vahid Mirnezami

Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Recommended Citation
Mirnezami, Seyed Vahid, "Application of deep learning and machine learning workflows for field-scale phenotyping" (2020). Graduate Theses and Dissertations. 17839.
https://lib.dr.iastate.edu/etd/17839

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Application of deep learning and machine learning workflows for field-scale phenotyping

by

Seyed Vahid Mirnezami

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Program of Study Committee:
Baskar Ganapathysubramanian, Major Professor
  Patrick S Schnable
  Soumik Sarkar
  Asheesh K Singh
  Jin Tian
  Adarsh Krishnamurthy

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2020

Copyright © Seyed Vahid Mirnezami, 2020. All rights reserved.
DEDICATION

I dedicate this thesis to my wonderful wife, Mahsa, for her endless love, sacrificial care, unconditional support, and encouragement during the years of my Ph.D. to pursue and complete this research. This work is also dedicated to our loving parents and brothers (Farid and Mehrshad) to whom we owe everything and whose prayers kept us going.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td></td>
<td>x</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>xi</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xi</td>
</tr>
<tr>
<td>CHAPTER 1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>References</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 2. A HIGH-THROUGHPUT PHENOTYPING SYSTEM OF MAIZE TASSEL STRUCTURE CLASSIFICATION AND TRAIT ESTIMATION USING MACHINE LEARNING AND IMAGE PROCESSING APPROACHES</td>
<td>8</td>
</tr>
<tr>
<td>Introduction</td>
<td>8</td>
</tr>
<tr>
<td>Materials and methods</td>
<td>9</td>
</tr>
<tr>
<td>Tassel imaging</td>
<td>9</td>
</tr>
<tr>
<td>Image pre-processing</td>
<td>11</td>
</tr>
<tr>
<td>Tassel segmentation</td>
<td>11</td>
</tr>
<tr>
<td>Image skeleton</td>
<td>12</td>
</tr>
<tr>
<td>Manual measurements</td>
<td>13</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>13</td>
</tr>
<tr>
<td>Tassel Tip</td>
<td>13</td>
</tr>
<tr>
<td>Branch Points</td>
<td>14</td>
</tr>
<tr>
<td>Tassel length</td>
<td>15</td>
</tr>
<tr>
<td>Tassel Structure Classification</td>
<td>17</td>
</tr>
<tr>
<td>Central spike length</td>
<td>21</td>
</tr>
<tr>
<td>First lowest branch length</td>
<td>22</td>
</tr>
<tr>
<td>First lowest branch angle</td>
<td>24</td>
</tr>
<tr>
<td>Conclusion</td>
<td>25</td>
</tr>
<tr>
<td>References</td>
<td>26</td>
</tr>
<tr>
<td>CHAPTER 3. HIGH-THROUGHPUT PHENOTYPING OF FIELD-BASED IMAGES OF MAIZE TASSELS USING MACHINE LEARNING AND IMAGE PROCESSING ALGORITHMS</td>
<td>27</td>
</tr>
<tr>
<td>Abstract</td>
<td>27</td>
</tr>
<tr>
<td>Introduction</td>
<td>28</td>
</tr>
<tr>
<td>Materials and Methods</td>
<td>31</td>
</tr>
<tr>
<td>Image Acquisition</td>
<td>31</td>
</tr>
<tr>
<td>Data Preparation</td>
<td>33</td>
</tr>
<tr>
<td>Tassel Detection and Localization</td>
<td>36</td>
</tr>
<tr>
<td>Trait Extraction</td>
<td>37</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>38</td>
</tr>
</tbody>
</table>
CHAPTER 4. FIELD-BASED FLOWERING PATTERN RECOGNITION OF MAIZE TASSELS USING MACHINE LEARNING AND IMAGE PROCESSING APPROACHES

Abstract .................................................................................. 49
Introduction .............................................................................. 50
Materials and Methods ............................................................. 53
  Plant Materials .................................................................... 53
  Image acquisition .............................................................. 54
  Workflow overview ........................................................... 56
  Tassel Detection ................................................................. 56
  Tassel Classification .......................................................... 58
  Tassel tracking ................................................................. 60
  Tassel Segmentation ........................................................ 61
  Tassel Analysis ............................................................... 63
Results and Discussion ............................................................ 64
  Tassel Detection ................................................................. 65
  Tassel Classification .......................................................... 66
  Tassel Tracking ................................................................. 66
  Tassel Segmentation ........................................................ 67
  Tassel Analysis ............................................................... 68
Conclusion .............................................................................. 70
References .............................................................................. 70

CHAPTER 5. MAIZE LEAF APPEARANCE RATE MONITORING BY IMPLEMENTING DEEP CONVOLUTIONAL NEURAL NETWORK..................................................... 74
Abstract .................................................................................. 74
Introduction .............................................................................. 75
Material and Methods ............................................................. 78
  Field experiment .............................................................. 78
  Image Acquisition ........................................................... 78
  Counting method ............................................................ 79
  Dataset preparation ......................................................... 81
Results ..................................................................................... 84
  Dataset preparation and Turker results.............................. 84
  End-to-End Counting using the expert turker dataset ....... 87
Conclusion .............................................................................. 90
References .............................................................................. 90

CHAPTER 6. GENERAL CONCLUSION ................................................. 94
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Imaging setup for one sample tassel</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Different views of a sample tassel</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Image segmentation procedure. A) original image b) clustered color image c)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>biggest component</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Image-based measurement application. Desire points are selected by this</td>
<td></td>
</tr>
<tr>
<td></td>
<td>application and the traits are calculated based on the skeleton image on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>backend</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Endpoint definition. It has only one connection</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>Correlation between three different measurement of tassel length (Physical-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>based, automated, and image-based)</td>
<td>15</td>
</tr>
<tr>
<td>2.7</td>
<td>Comparison between Automated and image-based measurement with the tassels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>structure labels. Red, blue, and green dots are related to the close, open,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and medium tassel structure</td>
<td>16</td>
</tr>
<tr>
<td>2.8</td>
<td>Some examples of close and open tassel structure</td>
<td>17</td>
</tr>
<tr>
<td>2.9</td>
<td>Circles were drawn around the top most branch point. The density inside</td>
<td></td>
</tr>
<tr>
<td></td>
<td>them are important for tassel structure and to understand how dense a tassel</td>
<td>18</td>
</tr>
<tr>
<td>2.10</td>
<td>Density inside each circle at the different circle radius</td>
<td>19</td>
</tr>
<tr>
<td>2.11</td>
<td>Two rectangles around the branching zone</td>
<td>19</td>
</tr>
<tr>
<td>2.12</td>
<td>SVM classifier for open and close tassel classification</td>
<td>20</td>
</tr>
<tr>
<td>2.13</td>
<td>Scatter plot of predicted and image-based tassel length</td>
<td>21</td>
</tr>
<tr>
<td>2.14</td>
<td>Central spike length correlation between automated and image-based</td>
<td></td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td>22</td>
</tr>
<tr>
<td>2.15</td>
<td>First lowest branch length correlation between automated and image-based</td>
<td></td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td>24</td>
</tr>
<tr>
<td>2.16</td>
<td>First lowest branch angle correlation between automated and image-based</td>
<td></td>
</tr>
<tr>
<td></td>
<td>measurements</td>
<td>25</td>
</tr>
</tbody>
</table>
Figure 3.1. Aerial view of a maize field at Iowa State University........................................32
Figure 3.2. A sample image of 6 tassels taken in a maize field at Iowa State University .......33
Figure 3.3. Different weather type captured by one of camera if different days..................33
Figure 3.4. Sample images taken on September 9th, 2015............................................34
Figure 3.5. Sample images with overlap tassels ..........................................................35
Figure 3.6. A sample of blurred image ........................................................................36
Figure 3.7. Faster-RCNN network................................................................................37
Figure 3.8. Workflow of the entire process ...................................................................38
Figure 3.9. Different tassel architecture within an image and within cameras captured in one specific day.........................................................................................38
Figure 3.10. The area for all rectangle in the image a is less than the area of the biggest box in image b...........................................................................................................40
Figure 3.11. Blue, red, and black boxes are the representative of each turker. The green box was obtained based on the consensus method. A) The tassels have overlap b) blue turker was removed after filtering and does not affect the consensus c) The tassels are normally captured ........................................................................................................41
Figure 3.12. The mAP obtained after the training the images based on the consensus method.................................................................................................................................42
Figure 3.13. The mAP obtained after the training the images based on the consensus method.................................................................................................................................43
Figure 3.14. A sample procedure of converting raw image to binary image.....................44
Figure 3.15. A sample procedure of converting raw image to binary image.....................44
Figure 4.1. An aerial photo of the experiment ................................................................54
Figure 4.2. Two sample images of two different cameras ..............................................55
Figure 4.3. The first images taken after 10 AM on the mentioned dates for a specific camera...55
Figure 4.4. The steps of the entire end to end process.....................................................56
Figure 4.5. Different maize genotype creates various tassel structure ...............................57
Figure 4.6. Examples of discarded annotations ................................................................. 58
Figure 4.7. False positive predicted boxes by the trained RetinaNet .................................. 59
Figure 4.8. The classification model architecture ............................................................. 59
Figure 4.9. Sample images for training the classification model (left: tassel class and right: non-tassel class) ................................................................. 60
Figure 4.10. Tassel images with different background over times ...................................... 62
Figure 4.11. The segmentation model architecture ........................................................... 63
Figure 4.12. a) The red line is the tassel path from the starting point to the tassel tip and blue dots are the branch points. b) The main spike of the tassel was cropped from the top moat branch point ................................................................. 64
Figure 4.13. predicted and ground truth boxes are shown in red and blue colors, respectively .................................................................................................................. 65
Figure 4.14. the classification model predicts the left image as tassel and right image as non-tassel ................................................................................................................. 66
Figure 4.15. False positive predicted boxes by RetinaNet .................................................. 67
Figure 4.16. Accuracy and loss value changes with every epoch ....................................... 67
Figure 4.17. Two sample tassels predicted by RetinaNet (left side) and their corresponding binary images predicted by the segmentation model ................................................. 68
Figure 4.18. Flowering pattern of a tassel in a camera ..................................................... 69
Figure 4.19. Flowering pattern of a tassel in a camera ..................................................... 70
Figure 5.1. Six sample images taken by two different cameras (CAM176 and CAM563) in three different time points (June 23th, July 23th, and July 31th, 2019 at 10:00 AM) ................................................................................................................... 79
Figure 5.2. counting challenges (a): some plants are not germinated, (b) Occlusion with other rows, and (c) Occlusion within the row ................................................................. 80
Figure 5.3. Different weather during the experiment .......................................................... 80
Figure 5.4. Flowchart for counting number of leaves using an End-to-End deep learning model .................................................................................................................... 81
Figure 5.5. The web-based application given to each turker to annotate the leaf tips by a dot... 83
Figure 5.6. Finding the final bounding box based on consensus method of three turkers. Black, Blue and red boxes are related to the three turkers annotated the images. The yellow box was obtained based on the union of these three boxes. Green box is obtained after extending the yellow box by 100 pixels from all sides 85

Figure 5.7. Cropping the image based on the final box obtained in the previous step and dividing the cropped image to 6 pieces for the second campaign 86

Figure 5.8. Turker data depicted by yellow dots for each piece as well as cropped original image after stitching and converting the coordinates. Moreover, annotated dots from the turker for each piece were reconstructed to the original image 87

Figure 5.9. Comparison between the ground truth and predicted values by the model trained based on the expert turker data 89
LIST OF TABLES

Table 2.1. Confusion matrix of the testing the SVM model using the testing dataset .................. 20
Table 3.1. Camera specification .................................................................................................................. 32
Table 5.1. R² values between predicted and ground truth obtained by both expert and trained turkers for same images ........................................................................................................... 88
ACKNOWLEDGMENTS

First and foremost, I exalt God for everything and his care on my entire life. Without his blessings, this achievement would not have been possible.

I would like to express my special appreciation and thanks to my major supervisor Prof. Baskar Ganapathysubramanian for his guidance, encouragement, inspiration, and support. His insight and advice have always guided me in my life. In addition, thank you to my committee members, Professors Patrick Schanble, Danny Singh, Soumik Sarkar, Jin Tian, and Adarsh Krishnamurthy for their helpful suggestions.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience.

Thank you to my beloved wife, Mahsa, our parents and brothers, who have supported me and shown me unconditional and continual love. Mahsa is my wonderful wife and friend who helped me and encouraged me to successfully finish my degree throughout all difficulties. In addition, I am deeply thankful and forever indebted to my parents for their endless love and support. Thanks to my father-in-law, mother-in-law, and brother-in-law, whose recommendations and experiences have flattened the path to reach my goals. Finally, thanks to my big brother, Farid, who dares me to follow my wishes.
ABSTRACT

Tassel is the male inflorescences organ of the maize plant that develops atop the plant. Coarse features of tassels, including shape and size, can influence shedding pollen, fertilization, and subsequently grain yield. Therefore, understanding tassel dynamics and characteristics as well as how it evolves during the plant growth can help the plant scientist community to increase the grain yield as a final goal. To do so, first, tassels were investigated in one time points. The tassels were cut in the field and their images were captured in a lightbox. Coarse features were measured using novel image processing approaches. 351 tassels with different genotypes were used for the experiment. Tassel length, first lowest branch length, and angle as well as central spike length were measured by applying image processing and machine learning techniques. Tassels were also classified to open and close structures to obtain accurate predictions for the traits. The results show that $R^2$ values for the tassel length and central spike length were 0.92 and 0.80, respectively. In addition, the $R^2$ values for the first lowest branch length and angle were 0.63 and 0.91, respectively. The $R^2$ values for the first lowest branch length was low compared to others because locating the first lowest branch point and its corresponding branch tip was hard due to branches occlusion. This study was done to create a robust algorithm for tassel phenotyping. Challenges were figured out for better tassel phenotyping in the field. Then, we looked at a diverse panel in the field, using stationary cameras to capture 6 tassels every 10 minutes for 8 hours per day during a month. Traditional approaches for phenotyping anthesis progression are time-consuming, subjective, and labor-intensive and are thus impractical for phenotyping large populations in multiple environments. In this work, we utilize a high throughput phenotyping approach that is based on extracting time-lapse information of anthesis progress from digital cameras. The major challenge is identifying the
region of the interest (i.e. location of tassels in the imaging window) in the acquired images. Camera drift, different types of weather, including fog, rain, clouds, and sun and additionally, occlusion of tassels by other tassels or leaves complicated this problem. We discussed the associated challenges for object detection and localization under noisy conditions. In addition, a framework was developed to utilize Amazon Mechanical Turk to allow turkers to annotate the images and evaluate them to create an object detection dataset. Finally, we illustrated a promising deep-learning approach to tassel recognition and localization that is based on Faster-RCNN which has shown the strong capability for detection and localization. This method was improved using a boosting method to improve the dataset. This approach is able to reliably identify a diverse set of tassel morphologies with the mAP of 0.81.

Tassel flowering pattern is the most important and complex trait. Tassel maize as a male structure is responsible to produce pollen for the silk as a female organ on the same plant. The amount of pollen and shedding time is important for the breeders as well as the biologists. This study introduced an automated end-to-end pipeline by coupling various deep learning, machine learning and image processing approaches. Inbred lines from both SAM and NAM panels were grown at Curtiss farm at Iowa State University, Ames, IA. A stationary camera was installed for every two plants. Tassels architecture, weather type, tassels and camera movements are the most important challenges of the research. To address these issues, deep learning algorithms were utilized. Tassel detection, classification, and segmentation. In addition, advanced image processing approaches were used to crop the tassel main spike and track the during tassel evolution. The results showed that deep learning is a powerful tool to detect, classify and segment the tassels. The mAP for the tassel detection was 0.91. The F1-score obtained for the tassel classification was 0.93. In addition, the accuracy of semantic segmentation for creating a
binary image from the RGB tassel images was 0.95. The width of the flowering was obtained using graph theory in image processing and the time and location of the flowering can be obtained from the width data over the main spike branch. In addition to tassel structures, crop growth simulation models can help farmers and breeders predict crop performance, and in maize, Leaf Appearance Rate (LAR) is an important parameter used in crop performance simulation models such as APSIM. Since breeders and biologists would like to minimize human involvement in monitoring LAR, this trait can be monitored by applying a high-throughput phenotyping system. Engineers have entered the picture in collaboration with plant scientists to establish different and robust phenotyping methods, and in this study, maize leaf appearance rate was investigated using high-throughput phenotyping approaches. We developed an imaging system for automatically capturing a time-series of images of maize plants under field conditions, with 380 RGB cameras were used to capture images from 380 rows. There were 6 plants with the same genotype in each row that had different genotypes differed row-by-row, and the images were taken for 9 hours daily at 20-minute intervals for more than one month during a growing season. An end-to-end deep learning method was then used to count the numbers of leaves in the images. The dataset for the deep learning algorithm, obtained using the Amazon Mechanical Turk platform, was created by one expert turker along with a well-trained turker. Results demonstrated that an end-to-end model with training based on the expert turker dataset performed very well, handling variation in images that included leaf occlusions and weather type. The $R^2$ between the ground truth obtained by the expert turker and predicted values was approximately 0.73, 0.74, and 0.95 for three testing cameras. The model’s prediction performance demonstrated that the number of
leaves increase with different slopes for different genotypes. The data can be used for further
genotypic analysis.
CHAPTER 1. INTRODUCTION

Maize (Zea mays L.), is primarily used to create biofuels, feed for livestock, food for human nutrition and thousands of other everyday products. Midwest region in the United States of America known as corn belt with the center Iowa State is the biggest corn producer. Iowa has rich soil with sufficient rain and the growing season in this state is long and warm enough for growing maize plants. Both male and female organs are in the same maize plants which introduce it as self-pollinated plants. Tassel is the male inflorescence that fertilized the silk which is the female part and produce corn on the ear.

Plant breeders are interested in crossing two maize pants with different genotype to create hybrid seeds. However, they are willing to obtain the high-quality seeds as well. They know that traits are genetically controlled, but they do not know the responsibility of each particular genes or genes. Therefore, measuring the plant traits for different genotype is valuable for breeders. Measuring the traits is called phenotyping. Plant phenotyping can be conducted manually by physically measuring. this task is time consuming, costly, and inconsistent. Semi-automated or fully automatically measuring the traits create the term high-throughput phenotyping. Hardware including imaging devices, storage equipment and processors units have helped improving the phenotyping. In addition, image processing, machine learning, and deep learning techniques accelerate the phenotyping.

Graph-based algorithms in which image processing and machine learning are coupled to one another for high-throughput phenotyping [1]. Image-processing algorithms have been used to automatically identify plant disease [2]. HTPheno is one of the image processing pipelines that includes region of interest definition, segmentation, and morphology trait extraction for high-throughput phenotyping [3]. Some researcher developed algorithms for
measuring coarse features of maize tassel and sorghum panicle in a lightbox [4,5,6] or soybean root in a vivo [7]. 3D image processing has recently been directed toward some specific traits whose study has been limited by 2D imaging. A 3D Canopy structure is one of the traits that can be quantified by reconstructing multi-view 2D images [8]. A researcher used machine learning and deep learning to identify apricot varieties [9]. Although 3D imaging provides more information, it is more expensive for breeders. Hyperspectral image processing has also been deployed for extracting some traits, including disease detection and water stress. Each band in these images has specific information because they have an electromagnetic spectrum with both visible and non-visible regions that can be used for calculating traits [10]. Hyperspectral images were also analyzed for early detection of drought stress, and image processing has become an inseparable part of FHTP [11]. Therefore, simple traits and clear images can be analyzed using image processing approaches.

Automatic measurement of plant traits at agricultural fields is still a bottleneck for breeders in the high-throughput phenotyping. Especially when the number of images and complexity of the traits are higher. In this situation, a concrete image processing approach cannot be taken for a so-called big data and complex traits. However, introduction of complicated machine learning, and deep learning methods known as computer vision has caused this bottleneck to become wider. The most common branches of computer vision are image classification, segmentation, and object detection. Different architectures including VGG16 [12] and ResNet [13] were introduced for image classification. In addition, Faster RCNN [14], RetinaNet [15], and Yolo have been implemented for object detection. These methods have jumped the accuracy for classification, segmentation and detection.
In the plant science community, these algorithms have been applied in crop classification, disease classification, image segmentation, fruit detection, weed detection. Moreover, a SVM image classifier was developed by [16,17] for classifying plants as healthy or unhealthy. Another researcher detected, classified, and measured plant stress using deep learning approaches [18]. StalkNet was developed by [19] with the purposes of both counting the number of stalks and measuring stalk widths. TasselNet also introduced for counting the number of tassels at field, and recently multi-task high-throughput phenotyping has been introduced by [20, 21]. An end-to-end deep learning approach was developed for detection of small particles identified as Soybean Cyst Nematode eggs [22]. In another study damage crops were detected in the field condition using three different powerful object detection methods [23]. Another researcher captured images of soybean roots grown in a growth chamber [24]. They used an auto-encoder model to consistently segment the root from the background for further analysis and measuring the roots traits. These are some of the ways deep learning facilitates FHTP.

Previous studies on tassel phenotyping have been of small scale and some were done in non-agricultural field settings such as growth chambers and greenhouses. In addition, manual measurement of pollen shedding has been performed in the field by [25] using passive pollen traps. These types of labor-intensive and low-throughput phenotyping methods are inconsistent and costly. Semi-automated methods have also been developed to calculate the traits by cutting the desired parts of the plant and capturing their images in a lightbox [4]. Since study of the shape characteristics of tassels must be measured directly in the field, engineers have come into the picture in collaboration with plant scientists to establish different and robust phenotyping methods. Using advancements in technology,
Thus, breeders and biologists would like to minimize human involvement in repetitive and time-consuming phenotyping tasks in the field.

Therefore, the aim of the first chapter is to automatically classify the tassel structure using binary machine learning classification approach as well as extracting various morphological traits. In addition, Genomic Wide Association Study (GWAS) was utilized to investigate the responsible gene(s). We Conceptualized, designed and engineered an open-source, fast, and extendable to 3D and efficient maize tassel phenotyping pipeline. This approach relies on image processing, graph-based and machine learning algorithms to cope with diverse tassel shapes and sizes.

After understanding the tassels morphologies and is phenotyping challenges, in the second chapter, we tried to phenotype the tassels in a field condition using computer vision and image processing. The required data for model training was obtained using Amazon Mechanical Turk. Therefore, we created a framework for turkers to draw a bounding box around each tassel. Developed a framework for evaluating turkers task. Tuned faster-RCNN object detection algorithm to detect the tassel using.

In the third chapter, we modified the experiment in the second chapter to phenotype a highly complex trait. This dataset enabled us to track tassel anthesis progression, together with the dynamics and characteristics, as well as how it evolved and developed during transition from vegetative growth to reproductive growth in maize and exploring the potential in yield improvement via modifying tassel structural variation for plant scientist.

Crop growth simulation models can help farmers and breeders predict crop performance, and in maize, Leaf Appearance Rate (LAR) is an important parameter used in crop performance simulation models such as APSIM. Since breeders and biologists would
like to minimize human involvement in monitoring LAR, this trait can be monitored by applying a high-throughput phenotyping system. Therefore, to assess genotypic variation of RLA, more efficient tools should be developed to meet the challenge posed by labor-intensive phenotyping. Then, in the fourth chapter, LAR for different genotype were automatically measured in the field condition.

References


CHAPTER 2. A HIGH-THROUGHPUT PHENOTYPING SYSTEM OF MAIZE TASSEL STRUCTURE CLASSIFICATION AND TRAIT ESTIMATION USING MACHINE LEARNING AND IMAGE PROCESSING APPROACHES

Seyed Vahid Mirnezami, Yan Zhou, Lakshmi Attigala, Patrick Schnable, Baskar Ganapathysubramanian

Introduction

Automation techniques development has embraced most of engineering and scientific fields including such as medicine and agriculture [1, 2]. Image processing and machine learning have revolutionized the agricultural industries [3, 4, 5]. Among all agricultural practices, High-Throughput Plant Phenotyping (HTPP) is one of the most crucial concern of breeders and biologists. Typical and human-based phenotyping tasks are time consuming, labor intensive, expensive, and inaccurate due to subjectivity. Furthermore, increasing the capability of computational and imaging devices have escalate the pace and precision of phenotyping. These drawbacks and benefits motivate biologists to ponder automatic plant phenotyping for the large scale of images.

Imaging pipeline systems, storage, data transferring, designing and implementing a robust algorithm are the challenges of plant phenotyping. Various morphological shapes of different genotypes of maize tassel intensify finding a robust approach for the tassel phenotyping. Several methods and software frameworks have been released for plant phenotyping. Most of them are specifically designed and created for particular species or distinct plant structure. They are also different in terms of input type, amount of manual work, and output type. GiARoots and DIRT are manual root-specific phenotyping tools which takes grayscale images [6, 7]. Full integrated imaging-analysis platforms like SmartRoot and Image-J were developed by [8, 9]. ARIA is a MATLAB-based application designed for root phenotyping for RGB images [10]. In this software, still a user is needed to supervise the results and conduct
some pre- or post-processing tasks. These are admirable software in the phenotyping area but none of they are solely designed and fully automated for maize tassel phenotyping. TIPS is the only available image-based phenotyping pipeline specifically for maize tassels phenotyping [11]. Tassel length and number of branches of tassel can be directly calculated by this method. However, biologist are interested in more traits.

Therefore, the aim of the present study is to automatically classify the tassel structure using binary machine learning classification approach as well as extracting various morphological traits. In addition, Genomic Wide Association Study (GWAS) was utilized to investigate the responsible gene(s). We Conceptualized, designed and engineered an open-source, fast, and extendable to 3D and efficient maize tassel phenotyping pipeline. This approach relies on image processing, graph-based and machine learning algorithms to cope with diverse tassel shapes and sizes. This method was evaluated 351 maize tassels genotypes. These data were then applied to a genome wide association study (GWAS) to detect marker-trait associations. The results of this study reveal that the proposed method is dependable conducting larger phenotyping experiments.

**Materials and methods**

**Tassel imaging**

The tassels those at the appearance of anthesis were collected and were imaged indoor (barn close to the field). The image capturing setup was shown in Figure 2.1. A blue piece of fabric was used as the background of the image and was securely attached to the wall with no wrinkles. The tassels were mounted upright on a remote-controlled tassel holder, which was programmed to rotate $90^\circ$ with each rotation. The images were captured using a Canon EOS 5DSR camera with a Canon Macro 100mm lens. The camera was attached to a tripod and the
distance between the middle of the tripod to the wall where the blue fabric was attached was about 250 cm. The height of the tripod was about 122 cm. A remote control was used to control the shutter remotely to avoid camera movements.

Figure 2.1. Imaging setup for one sample tassel

Total of five images were taken for a given tassel. The tassel was attached to the tassel holder in a way that for the 1st image, the 1st lowest branch was always parallel to the imaging plane and avoid occlusion by other branches. This ensured that the 1st lowest branch and the angle of the 1st lowest branch with the main axis were always visible in the first image. Three more images were taken at 90° rotations. The 5th image was taken in a way that the 2nd lowest branch was always to the left and the angle between the 2nd lowest branch and main axis was clearly visible (Figure 2.2).
Image pre-processing

The images were transferred to a Microsoft surface laptop. For a given image, the starting point of the tassel was marked by a human using a stylus. A MATLAB application was used to draw a red rectangle to recognize the starting point of the tassel. The user needs to pin two points, the upper left and lower right side. Then a rectangle was created with red color inside it. We did this because some tassels are bending then we cannot distinguish where the starting point is. We used red color because this color is used nowhere in the image and it is easy to locate it. Then upper middle point of the box is assigned as starting point of the tassel.

Tassel segmentation

Next step is segmenting the tassel from the background containing the blue cloth, genotype sticker and yellow marker. Color-based clustering segmentation using K-Means algorithm [12] was used to cluster the colors in the image. These two layers were clustered by implementing a k-means clustering. K was assigned as indicating background and foreground. Foreground was tassel, sticker and the marker. Then the foreground was segmented as shown in Figure 2.3. After applying this segmentation method, the blue background was removed and the objects including the tassel, genotype sticker and the scaling marker were remained. As mentioned earlier, images were captured such that there is no connectivity between the tassels.
and other part. So, the biggest object of the image would be the tassel which can be easily segmented out.

Figure 2.3. Image segmentation procedure. A) original image b) clustered color image c) biggest component

Image skeleton

Then, the segmented tassel was converted to the binary image. After that, tassel skeleton was obtained by applying morphological operation on the binary image. Graph based algorithm is the most popular method using for analyzing the plant morphology. Therefore, a graph was constructed by using the skeleton image. Each node is the potential branch point and each leaf is the potential endpoint. The leaves were found since they are connected to only one vertex. So, these leaves are the possible endpoints of the branches. The proper branch point and end points were selected based on specific criteria which is described in the following sections. Then a shortest path between the branch point and respective endpoints could be driven from the graph. These paths are helpful for measuring the traits.
**Manual measurements**

Each tassel was separately measured by hand before imaging. The user measured each linear trait using a fixed ruler on a working table and angles with a protractor. All the measurements along with genotypes name were typed in an excel sheet. If a branch point is occluded or a tassel is dense, the user tried to find it. However, in image analysis, it would be impossible to find such points. This kind of problem could be diminished in 3D image processing. So, we decided to manually measure the traits by using the images themselves. An application was designed and implemented for a user to select the desire points. The points are tassel tip, topmost and first lowest branch points as well as the tip of first lowest branch. Finally, these points were mapped to the skeleton image of each tassel and shortest path between the points were assumed as traits. Figure 2.4 shows the application written in MATLAB R2017b.

![Image-based measurement application](image)

Figure 2.4. Image-based measurement application. Desire points are selected by this application and the traits are calculated based on the skeleton image on backend.

**Results and Discussion**

**Tassel Tip**

Two points including the tip of tassel and the first lowest branch point are required for calculating the tassel length. The tip of the tassel is easier to be located since it is always visible. This is also the further endpoint from the starting point of the tassel. Endpoint or leaf of skeleton
graph is defined as point where it has only one connection out of 8 possible connectivity (Figure 2.5). In addition, the nearest point from skeleton image to the middle top part of red box at the base of tassel is assigned as the starting point. Then all possible shortest paths were calculated, starting from the base of tassel to all other endpoints. The longest path is assigned as a major path of graph and tassel. Hence, the corresponding endpoint is the tip of tassel.

Figure 2.5. Endpoint definition. It has only one connection

**Branch Points**

A heuristic algorithm was deployed to locate the branch points of tassels. First, the major path was removed from the skeleton image, then the skeleton object is not a connected component anymore. One or several objects will be remained depending on the number of branches. The small components were removed by a threshold value. Ideally, each separate object would represent a branch provided that there is no overlap between the branches. However, in most cases there are overlaps between the branches. Hence, each object contains one or more branches. Then, we looped through all the objects to find the intersection points between the object and the major path. The crossed points were removed, and the major path was moved to the left or right depends on where the object is located. We kept doing the last step for 100 iterations or until only one crossed point is obtained. This process was done because several points might be connected to the major path due to occlusion or bending branch. So, those
connected points which are not the target branch points were easily removed. Then, the branch points for each set of branches were obtained.

**Tassel length**

Branch points for each set were combined and the lowest branch points was the representative of the first lowest branch point of the tassel. Then, shortest path was calculated from the first lowest branch point to the tip of the tassel. The number of pixels passed by the path was the tassel length.

The automated tassel length was between 10.3 and 68.58 cm. This range for ground truth and image-based measurement were 18 and 59, and 75.24 cm, respectively. The maximum value of tassel length among all views was considered as the true length. The maximum was taken in account because the tassel might be bent perpendicular to the surface. The linear correlation of tassel length for different type of measurements and structure are depicted in Figure 2.6. The $R^2$ between automated and physical ground truth is 0.79. However, the $R^2$ value between automated and image-based as well as physical ground truth and image-based were 0.8 and 0.87, respectively. It means that even manual image-based measurement does not correlate with the ground truth highly larger than fully automated.

![Figure 2.6](image)

*Figure 2.6. Correlation between three different measurement of tassel length (Physical-based, automated, and image-based)*
As can be seen in the Figure 2.6, there are some outliers in the image-based comparison with the automated measurement. We looked at the corresponding images and tassels were labeled into three different categories based on the physical visibility of topmost branch point by a human. The total number of samples in open, close and intermediate categories were 113, 187, and 51, respectively. Figure 2.7 shows the scatter plots of the measurements with the labels of the tassel’s structures. The red points are related to the close tassels. According to this we concluded that the tassel structure affects the result of phenotyping substantially. Sometime the first lowest branch is very close to the main spike or occluded by other branches. Therefore, it could be only detected by a user if the actual tassel is available. These two challenges cause the major portion of the error. Hence, a classifier is needed to first classify the tassels structure.

![Graph showing the comparison between Automated and image-based measurement with the tassels structure labels. Red, blue, and green dots are related to the close, open, and medium tassel structure.](image.png)
**Tassel Structure Classification**

As can be seen in the Figure 2.7 tassels structure affects the trait estimation. Open tassels phenotyping is more accurate than close ones. Therefore, a classifier is needed to classify the tassels as open or close morphology. Some example of close and open tassels were depicted in Figure 2.8. To do so, support Vector Machine (SVM) as a supervised classification learning algorithm was implemented to classify the tassel structures. This method has been widely used in plant science community. Moreover, SVM was used because it has shown powerful ability in a binary classification. We extracted some quantitative features to train the SVM model.

![Close Structure](image1)

![Open Structure](image2)

**Figure 2.8.** Some examples of close and open tassel structure

Only the first view image of each genotype was selected for the classification. Ellipse aspect ratio around the tassel, solidity, and convex hull area are the typical coarse features used for the classification. A tassel with low aspect ratio could be considered as a close tassel. Solidity and convex hull were assumed to be a representative of how dense a tassel is.

In addition, the tassel density inside a circle around the top-most branch point was considered as another feature. The top-most branch point was selected a center because the
central spike length depends on this point. In addition, if the tassel is close, most of the branches circulates around this point. A heuristic approach was taken to find the best radius for the circle. Firstly, the thickness of the base of tassel was measured using the marked red color and image processing tools. Then a circle was considered with the radius size of the thickness and center of the top-most branch point. After that, the radius was incremented by 50 pixels several times until the radius is not more than 500 pixels (Figure 2.9). This value was estimated based on the one tenth of width of the image which is 5792 pixels.

Figure 2.9. Circles were drawn around the top most branch point. The density inside them are important for tassels structure and to understand how dense a tassel is.

The ratio of the filled tassel area by the total circle area show how dense a tassel is around the top most branch point. Figure 2.10 shows the trend of this ration. It means that by increasing the circle diameter, the ratio decreased. Hence, the biggest circle was used another predictor.
Another feature is the tassel density inside two rectangles around the predicted branching zone path. The width of the first rectangle is two times of the thickness and this width was incremented by 50 pixels for the second rectangle (Figure 2.11). This was done because the branch points are located on the branching zone path and the tassel density inside the small rectangles can be a valuable representative of how dense a tassel is around the branching zone. These features were trained by SVM model using MTALAB R2017b.

The 60 extreme open and close tassels were selected by a human as training dataset. Other tassels were used as a testing dataset. All the features were trained by the SVM classifier and the hyperparameters were optimized by Bayesian Optimization (Figure 2.12).
The trained model was applied on the testing dataset to evaluate the model accuracy. The confusion matrix is shown in Table 2.1. The accuracy of the model was 0.82.

<table>
<thead>
<tr>
<th></th>
<th>Predicted Open Tassel</th>
<th>Predicted Close Tassel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Open Tassel</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>Actual Close Tassel</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

So, after applying the model, 234 images were predicted as open tassels. Then, the R^2 between of the tassel length between automated and image-based was 0.91 (Figure 2.13).
Central spike is a region between the topmost branch point and the tip of the tassel. Tassel tip and all branchpoints were already located. Hence, simply the shortest path between these two points represent the main spike path and number of pixels rows passed by this path is the length of the main spike. Hand measurement of tassel ranges between 2 and 37 cm while image-based and automated ranges from 2.1 to 42 and 10 to 37 cm, respectively. The image-based is higher because in the automatic measurement if a branch is connected to a central spike, then the intersection is going to be a top most branch point, however a lower point should have been selected. This problem was arisen from occlusion and causes the estimated central spike length is lower than the image-based. The $R^2$ between automated and physical measurement is 0.8 for the predicted open tassels (Figure 2.14).

In TIPS paper, tassel length was used as a representative of the main spike [13]. They did not present any specific algorithm for the main spike length because in the tassel length, finding the first lowest branch point location is the only problem and the tip of the tassel can be found...
easily. But in the central spike case, the probability of accuracy locating of topmost branch point is lower compared to the first lowest branch point due to occlusion and connection of other branches. A sample tassel with occlusion around the topmost branch point is depicted in figure. Therefore, the correlation between different measurements methods are lower than tassel length. In the image-based measurement, even the user cannot locate the topmost branch point precisely and consequently it is impossible to automatically locate the point. So, it causes an error between ground truth and both automated and semi-automated approaches. This conferred as a bottleneck of 2D imaging which could be eliminated by using 3d imaging and processing.

![Central Spike (R²=0.80205)](image)

Figure 2.14. Central spike length correlation between automated and image-based measurements

**First lowest branch length**

As discussed earlier, the skeleton image was divided to several objects in which including one or more branches. The corresponding object of the first lowest branch point was stored because it contains the first lowest branch. However, finding the endpoint of the branch and track it from the branch point are challenging. To address these problems, we located all possible endpoints for the object. Then, all the endpoints were assigned to zero except the ones which have intersection with the major path. Next, the end points were updated. We pruned the object
by keep doing the last two steps for 200 iterations or until one endpoint is remained. A hole might have emerged after removing a point. Hence, the objects holes were filled inside the loop before updating the endpoints. One of these endpoints are the target branch tip.

Then it is needed to find the corresponding endpoint to the branch point. To do so, if there is only one endpoint then there is only one path which is the first lowest branch path. Otherwise, the all possible paths were calculated from the branch point to all obtained tips. Each path was divided to serval chunk of size 100 pixels. The angle of each segment was computed. If there is not sudden change of direction, then the path was considered as the target path. The number of pixels passed by this path was the first lowest branch length. Since, 200 pixels were added to this length since the same number of pixels were initially removed to find the endpoints.

The lowest value for automatically estimating this trait was 1.7 whereas the highest predicted length was 56. This value was higher than the maximum values of both manual and image-based which were 44.5 and 45.3, respectively. The reason is related to locating the endpoints. Sometimes, the branches are overlapped and cause extension of actual length. Furthermore, the end of one branch could be connected to another one which cause misplacing of the endpoint. The minimum length of both manual and image-based were 3 and 0.13. The overall linear correlation was not high because accurately locating the endpoint and tracking it from the first lowest branch is a difficult problem. However, the algorithm worked reasonably well for the predicted open tassels by SVM model. The correlation for the predicted open tassels between automated and image-based was 0.63 (Figure 2.15).
First lowest branch length correlation between automated and image-based measurements

**First lowest branch angle**

Since both the first lowest branch and the tassel main branch path are calculated, then it is possible to compute the angle between these two paths at the first lowest branch point. The correlation of the first lowest branch angle was higher compared to the length. Because for the length, the tip of branch should be located accurately. However, if the tip is misplaced, the angle can be still calculated since we used a lower part of the branch. The minimum predicted and measured of this trait for all type of measurement was very close to zero. However, the maximum for automated, image-based and physical measurement were 172, 169, and 120, respectively. The minimum measured angle was zero for all cases. The linear correlation between automated and image-based was very high for predicted open tassel which was 0.91 (Figure 2.16). In this case, finding the branch point was the only problem. There was no need to locate any endpoint and estimating the direction was not a hard challenge because the branch was tracked up to one third of estimated length. We did not compare the image-based and
automated with manual measurement since the tassels were held in their natural position while, this trait was not physically measured in the same position.

Figure 2.16. First lowest branch angle correlation between automated and image-based measurements

**Conclusion**

351 tassels with different genotypes were used for the experiment. Tassel length, first lowest branch length, and angle as well as central spike length were measured by applying image processing and machine learning techniques. Tassels were also classified to open and close structures to obtain accurate prediction for the traits. The results show that $R^2$ values for the tassel length and central spike length were 0.92 and 0.80, respectively. In addition, the $R^2$ values for the first lowest branch length and angle were 0.63 and 0.91, respectively. The $R^2$ values for the first lowest branch length was low compared to others because locating the first lowest branch point and its corresponding branch tip was hard due to branches occlusion. This study was done to create a robust algorithm for tassel phenotyping. Challenges were figured out for better tassel phenotyping in the field. To recapitulate, a given image can be classified into open
or close tassel, then the algorithm works well for the open tassels which can be used by biologists for further analysis.

References


CHAPTER 3. HIGH-THROUGHPUT PHENOTYPING OF FIELD-BASED IMAGES OF MAIZE TASSELS USING MACHINE LEARNING AND IMAGE PROCESSING ALGORITHMS

Seyed Vahid Mirnezami¹, Yan Zhou², Cheng-Ting Yeh², Jung Yong Seok³, Srikant Srinivasan⁴, Patrick Schnable², Baskar Ganapathysubramanian¹,⁵

¹Department of Mechanical Engineering, Iowa State University, Ames 50011, Iowa, USA
²³Department of Agronomy, Iowa State University, Ames, Iowa 50011, USA
⁴School of Computing and Electrical Engineering, Indian Institute of Technology Mandi, Kamand, Himachal Pradesh 175005, India
⁵Department of Electrical Engineering, Iowa State University, Ames 50011, Iowa, USA

Abstract

Tassel is the male inflorescences of the maize that develops atop the plant. Many coarse features of tassels, such as shape and size, would determine the number of florets that one tassel could carry. Thus, it influences the pollen shedding, fertilization and subsequently grain yield. In addition, understanding tassel dynamics and characteristics as well as how it evolves during the plant growth can help the plant scientist community to increase the grain yield as a final goal. In this study, we examined the maize tassels at a diverse panel in field condition, using stationary cameras to capture 6 tassels every 10 minutes for 8 hours per day across a month. Traditional approaches for phenotyping anthesis progression are time consuming, subjective, and labor intensive and are thus impractical for phenotyping large populations in multiple environments. In this work, we utilize a high throughput phenotyping approach that is based on extracting anthesis progress from time-lapse images. The major challenge is identifying the region of the interest (i.e. location of tassels in the imaging window) in the acquired images. We discussed the associated challenges for object detection and localization under noisy conditions, such as camera drift, different types of weather, tassel occlusions. In addition, a framework was developed to utilize Amazon Mechanical Turk to allow turkers to annotate the images and
evaluate them to create object detection dataset. Finally, we illustrated a promising deep-learning approach to tassel recognition and localization that is based on Faster-RCNN which has shown strong capability for detection and localization. This method was improved using a boosting method to improve the dataset. This approach is able to reliably identify a diverse set of tassel morphologies with the mAP of 0.81.

Keywords: Tassel; Phenotyping; Faster-RCNN; Amazon Mechanical Turk

Introduction

Maize (*Zea mays* L.), is primarily used as feed for livestock as well as for human nutrition. Tassel is the pollen-bearing male inflorescences that develops atop the maize plant. This organ consists of hundreds to thousands of spikelets the number of which would be responsible for pollen production and thus could affect the efficiency of fertilization [1]. Properties of tassels, including shape and size, can influence shedding pollen, fertilization, and subsequently grain yield [2]. In contract to continuing yield improvements, it’s reported tassel size of commercial hybrids planted in North American corn belt had been gradually reduced during past decades [3]. Therefore, understanding tassel dynamics and characteristics can potentially help the plant scientist community in tuning the yield improvement as a final goal.

Although advances in next-generation sequencing technologies have improved our ability to generate genotypic data, large-scale high-throughput phenotypic experiments are still developing. Labor-intensive and low-throughput phenotyping methods are inconsistent and costly. Semi-automated methods have also been developed to calculate the traits by cutting the desired parts of the plant and capturing their images in a lightbox [4]. There have also been some semi-automated studies on maize phenotyping in agricultural fields [5,6] and images developed by [7] have revolutionized semi-automated phenotyping as well. However, in such cases tassels
are fragile and exposed to death and deformation after only a couple of hours. Hence, researcher have been trying to use remote sensing in agricultural fields to remove these kinds of problems.

Recently, field-based high-throughput phenotyping (FHTP) is a useful platform for quantifying phenotypic traits required by biologists and consequently breeders [8,9] and it could lead to acceleration of the selection of high yield varieties by unraveling their genetic combinations. Field internet of things (IOT) and sensors [10,11], RGB cameras [12,13], hyperspectral cameras [14,15], and 3D laser scanning [16,17] are crucial and non-destructive tools that can simplify the FHTP. The images and their embedded data are crucial for the FHTP, yet their numbers are challenging. So, researchers are trying to improve the situation by eliminating resource constraints and applying scientific and engineered solutions.

One of the challenges in FHTP is detecting target objects in images. Different object detection methods were introduced using image processing algorithms. Viola-Johns and bag-of-feature were the popular traditional algorithms for detecting objects such as human face in the images [18,19]. Introducing deep learning and creating powerful algorithms have been dramatically improved the object detection and localization technique and the accuracy, and a target object in such images can be segmented using deep learning methods [20]. Faster Region-based convolutional neural network (Faster-RCNN) [21], You Only Look Once (YOLO) [22], Single Shot Detector (SSD) [23] are the most popular methods for object detection. Faster-RCNN has been extensively used by the plant science communities [24]. Deepfruits was introduced as a deep learning network concept to detect and localize fruits by implementing Faster R-CNN on both (RGB) and Near-Infrared (NIR) images information [25].

Although deep learning has dramatically boosted FHTP, it requires cumbersome work in preparing the dataset. PASCAL VOC, MNIST, CIFAR 10, ImageNet are large-scale publicly-
available datasets for image classification [26–29]. COCO is another dataset that can be used for object detection, image classification, and segmentation [30]. These datasets have been created by many users around the world to increase the number of sample and diversity. Object detection algorithms also require a dataset for training a model. A bonding box around target objects in the images are required by object detection algorithm. Amazon Mechanical Turk (AMT) is one of the methods for creating a large-scale dataset in which a tedious, repeatable, and massive task can be launched into the AMT. This platform allows researcher to obtain a low cost dataset in a limited time [31]. Several regular or master turkers from different countries are asked to complete the task. In this way, numerous turkers can participate in parallel, each performing a small portion of a task, and their results merged together to complete the campaign. Launching a campaign for data annotation is popular in plant science community [5,32]. Therefore, this can be useful for creating a dataset for phenotyping the plants.

All these above-mentioned methods have been applied to measuring different plant traits while aiming at high-throughput phenotyping at a field. So far, there is no platform for extracting tassel properties at a field condition. Most have been conducted in small scale facilities like greenhouses or manual monitoring at single time point. In addition, study of the shape characteristics of tassels must be measured directly in the field. However, breeders and biologists would like to minimize human involvement in repetitive and time-consuming phenotyping tasks in the field. Therefore, in this study, we looked at a diverse panel in the field, using stationary cameras to capture 6 tassels every 10 minutes for 8 hours per day during a month. Determination of high probability of occlusion, different types of tassel architecture, and tassel movement are challenges in the study. In addition, a collection of features were utilized for identifying and locating the maize tassels in images captured in agricultural fields [33]. A deep learning
approach, Faster-RCNN, was implemented for detecting and localizing tassels, because since it has demonstrated strong capability for detection and localization. The dataset for training the model was annotated turkers on the AMT. Dealing with turkers is hard. After detecting the tassel different image proceeding algorithm developed by other researchers in a light box may be applied on the indiviudal images to calculate coarse features and prominent traits. We followed three main targets in this paper:

- Create a framework for turkers to draw a bounding box around each tassel
- Developed a framework for evaluating turkers task
- Tuned faster-RCNN object detection algorithm to detect the tassel

**Materials and Methods**

**Image Acquisition**

A large-scale maize tassel phenotyping system was designed and deployed during summer 2015 (Figure 3.1). A maize field containing a Nested Association Mapping (NAM) [34] population was planted at Iowa State University by Schnable Lab members. This population was used because its genetic architecture can be characterized by an extensive variety of traits [35–37] The genotypes were organized in the ascending order of height in the entire field from north to south. In this case, minimum interference occurred in the tassel backgrounds. One camera was installed at the south end of a row facing north to avoid direct sunlight into the camera. In addition, there was a spacing of 60 rows in the north-south direction. Each row was laid out in the east-west direction and 12 of them constituted a range. A total of 19 ranges with east-west direction were planted.
Figure 3.1. Aerial view of a maize field at Iowa State University

A total of 455 cameras simultaneously captured the images every 10 minutes between 8 AM and 5 PM. The experiment was conducted over a two-week span during August 2015. Each group of 24 cameras were connected to an inexpensive commercial camera (Table 3.1) through cheap raspberry pi microprocessors. The resulting huge database of images were automatically stored in the field. Cameras were powered using solar panels optimally placed at the field. Storage volume required for each JPG image was about 8 MB, and image size was 5152 pixels high and 3864 pixels wide, so more than 300,000 images with a total data storage requirement size of approximately 4 Terabytes were collected. Figure 3.2 shows a sample image captured by one of the cameras, with the images reflecting different weather condition, occlusion and overlap.

<table>
<thead>
<tr>
<th>Table 3.1. Camera specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
</tr>
<tr>
<td>Type</td>
</tr>
<tr>
<td>Image format</td>
</tr>
<tr>
<td>Lens mount</td>
</tr>
</tbody>
</table>
Figure 3.2. A sample image of 6 tassels taken in a maize field at Iowa State University

**Data Preparation**

Recall that the first stage is tassel detection and localization. Blurred images can be caused by camera movement or weather type. Some images may have blue sky or white and dark cloudy background. Figure 3.3 shows different images taken by one camera on different days.

Figure 3.3. Different weather type captured by one of camera if different days

All images were taken between 9.30 to 10.30 AM on September 9\textsuperscript{th}, 2015, and about 4 images per camera per day were selected for annotation. This particular day was chosen because extracting the traits at a single time point was desired, the weather that day was sunny, and the blue background can be helpful for segmenting the tassels. There was a of total 2201 images.

Figure 3.4 shows four sample images from the selected day.
The deep learning faster-RCNN method requires a dataset containing images as input and rectangular coordinates around the target object as labels [21]. The target object in this study was the maize tassel. A rectangle should be drawn around each visible and complete tassel. Annotating 2201 is a painful job for one person. The AMT platform was therefore used to recruit regular turkers for annotating the images using Javascript code. The images were uploaded to the Amazon Server., and we constructed a tool to allow Amazon Mechanical Turkers to annotate images. This tool was deployed on our website using HTML, CSS, Javascript, and PHP in the front, with annotation results saved on a MySQL back-end database.

There was an instruction on the first page of the survey to show the turkers how to annotate the images. For this task, each turker was given a set of images containing at most six corn tassels and directed to draw a tight bounding box around each of the tassels on the frontmost row of plants in the image. The boxes were to be drawn around the tassels as long as they could be completely seen in the image. Since the tassels were planted next to each other and windy weather could cause overlapping between two tassels, the bounding boxes could overlap. Figure 3.5 shows sample images with complete visible tassels and overlap samples.
Figure 3.5. Sample images with overlap tassels

For each target tassel, a bounding box should be drawn from left to right on the image by positioning a cursor at the top-left coordinate of the tassel to establish the starting point of a bounding box, after which a blue dot recorded the position on the image. The cursor then should be positioned on the bottom-right coordinate of the tassel to establish the end of the bounding box; a drawn bounding box would be automatically based in red on the top-left and bottom-right points.

In the event of a need for re-drawing a bounding box, a turker would click on the redraw button and repeat these two steps. There was an option for skipping the tassels either if they were occluded from the viewing angle or there were difficulties preventing drawing of a bounding box. The turkers could also reject the image by writing a reason, e.g., if no tassel could be seen or the image was blurred. (Figure 3.6Figure 3.2). Approximately one minute was provided to draw boxes around at many as 6 tassels per image. While the maximum time that a turker was allowed to work on this campaign was 2 hours. In addition, the maximum number of images delivered to a turker is 80 images.
Figure 3.6. A sample of blurred image

**Tassel Detection and Localization**

While the tassel detection and localization described above was the first step in image analysis, the Faster-RCNN method has been used in several fields for phenotyping [25,38]. In this study, while only one object, the maize tassel, needed to be detected and located, other objects such as leaves, maize plants, and rod might be presented in the image. A Faster-RCNN approach including both a classifier and a localizer inside its network has been used in several research papers. Such a network requires an image with one or more bounding boxes around the target object, the tassel in the current study. The Faster-RCNN network is shown in Figure 3.7. MATLAB R2017a was selected for Faster-RCNN deployment because implements several built-in applications as well as powerful image processing and deep learning modules.
Typical image processing approaches including binary and skeleton conversion were next implemented. In this case, the skeleton image was used for obtaining morphological traits such as convex hull area, total branch length, and total area. All of the processing approaches were implemented and run on the CyEnce high performance computing cluster at Iowa State University. Cyence consists of 248 SuperMicro servers, each with 16 cores, and 128 GB of memory per Node. The workflow of the entire fully-automated process is given in Figure 3.8.
Results and Discussion

Different maize genotypes create various tassel architectures, with tassel density, number of branches, and tassel convex hull the major differences between the different tassel structures shown in Figure 3.9; this model must be robust enough to deal with such different image types of.
**Amazon Mechanical Turk (AMT)**

This Amazon turk batch lasted for 3 days, 22 hours, 17 minutes, starting from June 26 at 12:07 PM, with a total of 130 turkers participating in the campaign. While the platform was designed such that each image was annotated by three turkers, sometimes the images were annotated by more than three turkers. This happened because users were required to submit a verification code generated after annotating the images into the amazon mechanical turk to complete the hit, and because the website was independent of amazon, there is a possibility that multiple users were working on the same hit. For this reason, when two turkers submitted their annotations on our website, they were both given verification codes even though only one code would be used on amazon’s hit.

Since not all the turkers performed equally well, and hits were accepted or rejected based on the accuracy of the bounding boxes. 6978 hits were recorded after finishing the campaign. 315 of them rejected for image annotating because of blurry frames or not fully visible tassels. Those 315 hits contained 274 different images, with 270 of them both rejected and accepted by different turkers. So, we decided to check all 274 images visually to see which turkers had carefully performed the task, resulting in finding that two of the images were correctly rejected by all turkers, representing 7 hits. Since the other 271 images should have been annotated but were rejected by turkers, a total of 308 hits among 315 were not paid for due to cheating. One turker also annotated a blurry image that was also rejected for payment.

Because height and width of each box should be less than a threshold value, the height and width of each box was also checked for each hit. We did not use the area of each box because there could be some boxes that are big and correctly annotated (Figure 3.10). Since if the entire set of boxes for one hit is too long, that hit would be rejected, 565 hits were removed due
to this filtering. A total of 873 hits were therefore rejected and not paid, resulting in an approval rate was 0.87, resulting in a total cost of $132.06.

![Image](image_url)

Figure 3.10. The area for all rectangle in the image a is less than the area of the biggest box in image b

**Tassel detection and localization**

**Consensus-based for the best annotation**

A total of 6105 hits were accepted after checking and filtering the boxes. The total number of images for all these hits was 2196. All boxes annotated by all turkers for each image were compared by considering only those images annotated by at least three turkers, i.e., 2192 images. A consensus-based method was deployed to find the best box among all the boxes drawn by the turkers. If at least three boxes had overlap of more than 30%, their union was considered instead of each box separately.
A consensus-based method was deployed to find the best box among all boxes drawn by the turkers. Since the union of two boxes with overlap more than 30% replaced those two boxes, several union boxes were created. These boxes were compared two by two and their union considered if they had more than 30% overlap. Finally, all the boxes obtained were compared with each box annotated by the turkers, and if more than three boxes had overlap with the union, that union was recorded as that of the correct box. Only images with 6 boxes were used in the dataset since the experiment was designed such that six tassels should be inside the frame. The images for the obtained boxes were visually checked and 80 of them selected as the testing dataset, with all others forming a training dataset. Different turkers results were shows in Figure 3.11.

**Faster-RCNN**

Approximately 3000 maize plants with different genotypes of the NAM population were planted and captured, and each of the maize tassels had various structures and traits. Difference in tassel architecture can result in different size, weight, number of branches, and length. Each such trait is important for biologists and breeders in obtaining desired plants, and machine learning should detect and locate all such tassels with one model. Faster-RCNN is powerful
enough to detect different structures. Figure 3.9 illustrates different tassel morphologies ranges from very dense to very sparse.

The Faster-RCNN model was used to train the 353 images used for training and 88 images used for the testing. About six hours of model training was required using the CyEnce cluster at Iowa State University. Images were resized to 387×516 pixels, and various hyperparameters were tried to obtain the best model based on evaluation using mean Average Precision (mAP). The mAP for the best model was 0.78. Figure 3.12 shows the recall-precision plot for the model.

![Figure 3.12](image)

Figure 3.12. The mAP obtained after the training the images based on the consensus method

The number of images with 6 boxes in the image, using the consensus-based method of three turkers, was 441. However, since only two turkers agreed on 1565 images, to increase the size of the training dataset, a boosting method was applied to salvage images with two-turker agreement. To accomplish this, the model was used to predict where tassels were located on all other images not used in the testing and training dataset. We then assumed that, if the boxes predicted by the model had overlap with those for the other two turkers, the probability of its
being a tassel inside the union of those two turkers would be high. We obtained 1565 images with two turkers agreeing on predicted boxes. The model obtained was then used as a pre-trained model and these unseen images were fed into the model to be trained. The mAP was 0.81, higher than the other model since it was a pre-trained model and more images were used (Figure 3.13).

Figure 3.13. The mAP obtained after the training the images based on the consensus method

No one has previously used a similar method of tassel identification based on such a very high mAP. A collection of features were used to detect the tassels using the similar images [33] Such a collection of features is useful for image classification, and they obtained high accuracy with respect to tassel versus non-tassel classification.

**Calculating tassel traits**

The first six predicted boxes with highest probability were chosen, and they were cropped to obtain isolated tassels, with the boxes expanded to ensure that complete tassels were enclosed by the rectangle. Since the next step was to remove the image background from these cropped images, the cropped tassels were segmented out using color-based clustering [39]. This could be done because the cameras were installed in such a way that the sky served as image background,
permitting very good separation of tassel from sky. These images were then converted into binary images, as shown in Figure 3.14.

![Figure 3.14. A sample procedure of converting raw image to binary image](image)

Since there are some samples in which the tassel might have been deformed by external objects such as a rod or other tassels, and also some of the tassels may not be completely included in the rectangle because some parts of tassel might be out of the frame of the original image, a filtering method was applied as post processing to whole images to remove undesired binary images. To do this filtering, the areas of non-zero pixels and borders of cropped images were investigated. 440 images were removed out of a total of 2280 cropped tassels. Figure 3.15 shows examples of removed images. Morphological operations were deployed on the remained images to extract the morphological traits of the tassels. Convex hull area, total root length, tassel area, and perimeter were calculated for each tassel.

![Figure 3.15. A sample procedure of converting raw image to binary image](image)
Conclusion

In this study, we looked at a diverse panel in the field, using stationary cameras to capture 6 tassels every 10 minutes for 8 hours per day during a month. The main goal of this study was to show the feasibility of detecting tassel in images captured using an easy-to-access RGB images. The major challenge is identifying the region of the interest (i.e. location of tassels in the imaging window) in the acquired images. Camera drift, different types of weather, including fog, rain, clouds and sun and additionally, occlusion of tassels by other tassels or leaves complicated this problem. Computer vision tool and deep learning algorithms can assist to identify tassels. To create the dataset, AMT was utilized to annotate the images required for training the computer vision algorithm. Therefore, a framework was developed to annotate the images and evaluate the turkers. Then, Faster-RCNN was customized and trained to identify the tassels in different images. After that, a boosting method was implemented to improve the dataset annotated by turkers. This approach is able to reliably identify a diverse set of tassel morphologies with the mAP of 0.81. The detected tassels can be segmented, and then morphological operation can be applied on the binary images to calculate the coarse features.

References


33. Lee N. High-throughput phenotyping of above and below ground elements of plants using feature detection, extraction and image analysis techniques. 2016;


CHAPTER 4. FIELD-BASED FLOWERING PATTERN RECOGNITION OF MAIZE TASSELS USING MACHINE LEARNING AND IMAGE PROCESSING APPROACHES

Seyed Vahid Mirnezami, Yan Zhou, Patrick Schnable, Baskar Ganapathysubramanian

Abstract

The tassel in the maize plant is responsible for producing pollen for subsequent capture by the silk or female organ. Both the amount of pollen and the shedding time are physiological traits that impact yield, and hence are important to both breeders as well as biologists. This study describes an automated end-to-end pipeline that combined deep learning, and image processing approaches to extract tassel flowering patterns from time-lapse camera images of plants in field conditions. Inbred lines from both SAM and NAM panels were grown at the Curtiss farm at Iowa State University, Ames, IA, with ~500 stationary cameras installed across the field that captured images of plants every 10 minutes for a two weeks period in the summer of 2016.

Extracting data from imaging performed under field conditions is challenging due to variabilities in weather, illumination and tassel diversity. To address these issues, deep-learning algorithms were used for tassel detection, classification, and segmentation. Image processing approaches were then used to crop the tassel main spike and track tassel evolution. The results demonstrated that deep learning with well labeled data is a powerful tool for detecting, classifying, and segmenting tassels. Our end-to-end workflow exhibited the following metrics: mAP for the tassel detection was 0.91, F1-score obtained for tassel classification was 0.93, and accuracy of semantic segmentation in creating a binary image from the RGB tassel images was 0.95. This work flow was used to determine spatiotemporal variations in the thickness of the main spike – which serves as a proxy for anthesis progression.
### Introduction

Flowering time in plants is highly geographically adapted. After a plant has been introduced into a new environment, flowering time can vary greatly [1,2]. For field crops such as maize, date of flowering is crucial for yield. Therefore, altering reproduction time to achieve better adaptation to local environments and different climate conditions has become a major task in plant breeding. For example, breeders in the corn-planting regions of north central United States have found late maturity to be associated with higher yield, although over-delayed maturity might lead to yield loss caused by frost in early autumn [3,4]. It is understood that proper flowering time selection will help maximize yield gains and avoid loss from unfavorable weather during preharvest time.

For maize, the flowering activity, or anthesis, first starts from pollen shed from the topmost branch of the main rachis of its male flower, the tassel [5]. Pollen grains then land on the silks of female flower, the ears, to complete fertilization. Since one ear can bear hundreds of fertilized seeds, pollination efficiency is directly affected by the amount of pollen a tassel can carry and the length of time over which anthesis occurs. Understanding the genetic basis for anthesis provides a way to improve pollination efficiency. This calls for monitoring anthesis progression under field conditions across genetic diversity (SAM, NAM panels) to tease out the genetic basis. However, monitoring this process simultaneously on hundreds or thousands of tassels under field conditions is a challenging task – especially if data is collected manually. Studies seeking to understand maize flowering patterns were manually done by [6], who developed a dispersion model for predicting maize pollen, predicting total pollen shed by maize tassels for each male fertility treatment. Pollen traps were used to measure maize tassel pollen shed each day [7]. Measuring the pollen shed using such methods is very time-consuming and
resource intensive, and is not feasible to deploy for large scale experiments involving thousands of plants.

The availability of light-weight, robust and cheap imaging devices and associated developments in image processing has created the possibility of phenotyping faster and more accurately. Wheat-flowering stages and ear emergence were monitored using computer-vision in [8]. Similar methods have been implemented in automatically detecting flowering using time-series RGB images taken in rice fields [9]. Image-processing algorithms for extracting information from digital RGB images on color image segmentation in HSI color space were developed and used to investigate lesquerella flowering [10]. In these studies, flowering was divided into three stages: fully, partially, and non-flowering, and advanced image processing approaches were used [11]. Corn tassels were detected under outdoor field conditions using image-processing methods [12]. Another researcher used an image-based bag-of-features to detect and locate maize tassels, and also used advanced image-processing approaches to extract features of individual tassels [13]. Some tassel traits, including tassel and main spike length as well as tassel weight, were estimated using a heuristic image processing method called TIPS [14,15] measured sorghum panicle length and width using lightbox images. These examples all demonstrate that using cameras and image processing tools allows vastly improved plant phenotyping. However, image processing techniques exhibit a lack of robustness, especially when used on image data from field experiments (in contrast to lab experiments with tightly controlled imaging conditions). This is because natural factors, including weather, temperature, humidity, and wind speed, significantly affect image quality in the field, in addition to occlusion and large signal to noise. This large variability makes it difficult (if not impossible) for purely image processing based approaches to perform consistently.
In this context, the advent of deep learning (DL) approaches has revolutionized feature extraction from image data, specifically image classification [16], object detection [17], and segmentation [18]. Additionally, DL methods exhibit enhanced robustness and processing speed compared to conventional image processing. DL approaches have been successfully applied to an increasing array of problems in the plant sciences including precision agriculture [19], fruit detection [20,21] fruit counting [22] weed detection [23], disease classification [24], disease detection [25]. More importantly, there have been several plant science specific developments of deep learning workflows. For instance, a network called TasselNet was able to accurately count the number of maize tassels in a field [26]. A study demonstrated a phenotyping system for automatically identifying Northern Blight-infected maize plants based on field image data [27]. Moreover, deep learning was implemented for estimating soybean-leaf coverage in the field [28]. Stalknet was developed to measure the sorghum stalk count and its width using Faster-RCNN and semantic segmentation [29]. The Panicle-SEG model developed a robust segmentation algorithm for sorghum panicle using CNN and clustering super-pixels [30]. CNN-based architectures was used to develop a wheat-disease diagnosis system in a field environment [31]. These developments have allowed breeders and biologists to collect and efficiently analyze image data from large fields and consider more complex phenotypic traits [32].

The careful integration of large field experiments with DL approaches open up the possibility of quantitatively extracting spatiotemporal patterns associated with maize anthesis. This is the primary motivation of the current work. This is a challenging problem, with very limited studies -- at the field scale -- on temporal-spatial extraction of traits. We specifically focus on designing workflows to extract the spatiotemporal dynamics of flowering starting position and periodical flowering patterns on tassels based on a large scale field experiment
involving 500 consumer cameras that image specific group of plants every 10 minutes. We describe development of a high-throughput system for recognizing the flowering patterns of different maize-tassel genotypes, including design, development, and implementation of an automated end-to-end pipeline for monitoring such patterns. In the pipeline, various computer-vision algorithms and heuristic image-processing approaches were implemented.

**Materials and Methods**

**Plant Materials**

Leveraging availability of inexpensive 20 mega-pixel cameras integrated with advances in Internet of Things (IoT) that allows coordination of such cameras via cheap raspberry pi microprocessors, the Schnable group deployed a large scale plant image capture experiment in the summer of 2016 (Figure 4.1). Field phenotyping of the 185 inbred lines from a shoot apical meristem panel (SAM) [33] and 932 recombinant inbred lines from a nested association mapping population (NAM) [34] was conducted using 456 cameras simultaneously, each camera imaging a set of 2 plants. The cameras were powered by solar panels strategically placed between the rows of plants. Each camera took an image every 15 minutes during a three week growing period in August 2016. The population that was imaged consists of about 5,000 recombinant inbred lines (RILs) created by crossing 25 diverse inbred lines with a common parent, B73. This population has been extensively used to define the genetic architecture of a wide variety of agronomic traits. The plants were organized in 6 ranges with each line containing one plant. Each range contained 14 rows at a 1.52 m row spacing. In each row, 16 plants were planted in an east-to-west direction at a 38 cm spacing, with border plants at the same spacing planted at the side of each row. The genotypes were arranged in the descending order of height from south to north. This was done to minimize interference from background tassels in the captured images.
The genotypes within each row were typically of similar height and were therefore ordered randomly.

Figure 4.1. An aerial photo of the experiment

Image acquisition

In each row, for every two plants a Nikon Coolpix S3700 camera was deployed at a position 60 cm south of the row. The camera was mounted facing the midpoint between two plants in the row and the height of the camera was adjusted to enable image capture of both tassels. For cases in which two tassels had emerged at different times or at different heights, the camera was adjusted so it could at least monitor the first emerged tassel’s anthesis. The camera used for imaging each set of plants was placed south of the row with the camera facing north, to prevent overexposure due to the direct incidence of the sun on the camera.

Every set of four cameras were connected to one Raspberry Pi2 processing unit powered by one solar panel and controlled by custom-written Python codes (Schnable Lab, unpublished result) to produce images at 10-minute intervals from 7am in the morning until 7pm in the afternoon, starting after the first tassel had emerged from a flag leaf and terminated when the anthesis processes of both plants monitored by a given camera was finished. Since tassels evolved every day, the images were checked daily to ensure the tassels are not out of frame.
About 500 stationary cameras were installed in front of every two tassels with the sky as the background. Figure 4.2 shows sample images from two different cameras, and a sample of evolution images from one camera is shown in Figure 4.3.

![Sample images from two different cameras](image1)

Figure 4.2. Two sample images of two different cameras

![Sample evolution images from one camera](image2)

Figure 4.3. The first images taken after 10 AM on the mentioned dates for a specific camera
Workflow overview

We next describe the end-to-end pipeline used to analyze the tassels and monitor the flowering patterns. We divide the workflow into several steps, as illustrated in Figure 4.4. The first step is to detect a tassel in each image, and track its location across the time sequence of image data. After detecting the tassel, the next step is to segment the tassel from the background to create a binary image (i.e. black-white image). Finally, this time series of binary images are analyzed using image-processing approaches to extract physiologically meaningful traits. Each of these steps are described in detail in the following sections.

![Figure 4.4. The steps of the entire end to end process](#)

Tassel Detection

Images captured over the growing season in Ames, Iowa, exhibited a wide variability and diversity. The first reason for such variability is weather -- with foggy, rainy, sunny, and cloudy conditions impacting the image quality. Secondly, the imaging protocol was designed to contain two tassels in each camera's imaging window. However, due to wind and ensuing occlusion, images contained two, one or no tassels. Moreover, the presence of other objects in the imaging window, including leaves and other cameras, makes detection more difficult. Thirdly, each genotype exhibits a distinct (and diverse) tassel architecture. Finally, tassels developed during the imaging process, and the first image of a tassel was sometimes completely different from the last image of the same tassel. Figure 4.5 illustrates this using a small set of images captured from
different cameras, making the case for a robust detection method to accurately detect and locate individual tassels.

![Image](image.jpg)

**Figure 4.5.** Different maize genotype creates various tassel structure

We train and deploy a deep-learning based detection method called RetinaNet, which is a powerful object-detection method described in [35]. The output of the model is a set of box-shaped coordinates surrounding the target object, a tassel in the current study. While this model is very similar to the Faster-RCNN [36] model, the loss function in this model has been modified and its implementation optimized for better tassel detection.

This method, like other supervised deep-learning methods, required a training dataset. Images were first randomly selected from each camera, with a total of 3600 images selected from among more than 500,000 images. These images were annotated by turkers using the Amazon Mechanical Turk tool. Turkers were asked to draw a box on the two biggest tassels (i.e. the foreground) on the images, with an evaluation algorithm applied to the annotation to ensure accuracy and discard incorrect annotations (Figure 4.6). Quality control of the annotated boxes included removal of very small boxes (measured in terms of pixel area), along with a check to see if the width and height of annotated boxes were within a priori defined bounds. Subsequently, images that passed these automated quality check were visually checked. The final, quality assured, annotated dataset consisted of a total of 2911 images for analysis by
RetinaNet, with 2619 of those used for training and 292 for testing. The model was trained using the dataset on the Nova cluster at Iowa State University using a GPU NVIDIA Tesla V100 with 32GB memory. The details of the model as well as hyperparameter tuning are discussed in the supplementary information. The mean Average Precision (mAP) metric is used to evaluate the accuracy of the model for object-detection. The mAP – which is a standard metric for such problems -- quantifies the precision of the model at different levels of recall.

![Figure 4.6. Examples of discarded annotations](image)

**Tassel Classification**

We perform additional quality assurance on the results produced by the tassel detection model. This is because, due to the wide diversity of images and imaging conditions, some false positives boxes are predicted by the model (usually before tassel emergence because imaging may have started before emergence of tassels). Figure 4.7 illustrates this by showing some of false positive boxes predicted by the model. Instead of utilizing strategies involving manual quality control, we rely on an automated approach where we train and deploy another (simple) model to differentiate between a box that contains a tassel from a box with no tassel. This tassel classification model then identifies the false positive boxes predicted by the tassel detector and removes these data.
Figure 4.7. False positive predicted boxes by the trained RetinaNet

We train and deploy a CNN based binary classifier. Examples over the past few years suggest that CNN classification models produce robust classification models [16], especially for highly variable data sets characterized by a diversity of tassel shapes, weather as well as camera and leaf movements. Figure 4.8 shows the architecture of the binary classification model using CNN. The input to the model is the image within the boxed region identified by the tassel detection step (Section 2.4), and the output of this model is a binary output (1 if the image is a tassel, 0 if it is not).

Figure 4.8. The classification model architecture

The training dataset needed to train the model was created by using 299 images from the boxes predicted by the object-detection model. These images represent a diversity of tassel and leaf images, with 158 of the images belonging to the tassel class and 141 of the images belonging to the non-tassel classes, with 284 and 15 images selected for training and testing, respectively. Sample images for both classes are shown in Figure 4.9.
Since this is a binary classification, a confusion matrix based on the results of applying the trained model on the testing dataset was obtained and used to quantify the evaluation metrics, including precision, recall, accuracy and F1 score. While “precision” quantifies how many of the detected boxes are actually tassels, “recall” quantifies how many of the tassels in the image have been identified. Model performance and robustness were decided based on the accuracy of F1 score. These quantities are defined as follows:

\[
\text{precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}
\]

\[
\text{precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}
\]

\[
\text{precision} = \frac{\text{True Positive}}{\text{all number of samples}}
\]

\[
F1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

**Tassle tracking**

Once tassels had been detected (and boxed), the next step is to track individual tassels within a given camera across the time series of images to enable extraction of temporal characteristics of anthesis. As mentioned earlier, the experiment was designed such that there
were two tassels per image, one at the left and the other at the right side of the frame. But due to occasional movement of tassels due to wind, occlusion by leaves or movement of camera position, the number of tassels and their relative locations were not consistent over time, requiring implementation of a tracking method to track and tag individual tassels in a camera across time. We utilized a simple strategy of associating the detected tassels in each image to a time series dataset corresponding to a genotype. For all the images from a specific camera that make up a time series of data, the centers of the two (biggest) detected boxes were considered (since there were two tassels imaged per camera and the tassels of interest were closed to the camera and hence had the largest size) followed by K-means clustering to cluster the points into two groups (k=2). Boxes in first cluster were associated with the fist tassel, and boxes in the second correspond to the second tassel. This approach – while seemingly simple – worked surprisingly well in tracking individual tassels across the time horizon.

**Tassel Segmentation**

Following detection of the tassels as a function of time, the next step is segmenting out the tassel morphology from the image. We train and deploy a deep-learning based segmentation model, as previous work [28] has shown the robustness of such approaches compared to regular thresholding approaches. Conventional thresholding strategies produced inconsistent results due to the diversity of backgrounds that includes sky, leaves, cameras, and tassels from the other rows. This is illustrated in Figure 4.10 that shows a sequence of time lapse images tracking a single tassel, and it is clear that the background (as well as illumination) varies substantially. This representative image sequence shows effects of foggy, cloudy, sunny, or rainy weather.
In segmentation, an RGB image is mapped into a binary image pixel by pixel. We train and deploy a semantic segmentation method using convolutional autoencoder. Specifically, a deep-learning method called Convolutional-AutoEncoder (CAE) was utilized. This type of model consists of two symmetrical parts, the encoding section and the decoding section. The encoding section compresses the input image into a latent feature representation. The decoding section then uses this latent representation to map to a segmented image. The architecture of the CAE model is shown in Figure 4.11.
The model takes an RGB image as an input and produces the corresponding binary image as an output. We utilize a semi-automated approach to create the annotated dataset to train the model. Basically, we utilized four standard image segmentations approaches based on Otsu, HSV and LAB color spaces [37]. A human then selected the segmented image that most closely represented the true segmentation. This results in 1135 total images that were used for training and 142 used for testing.

**Tassel Analysis**

Once the segmented images are obtained, standard image processing methods were used to analyze the images. We are primarily interested in the flowering patterns (anthesis progression) of the central (or main) spike of the tassel. We define the main spike as the longest branch of the tassel. Since the main spike is sometimes occluded by other tassel features, we seek to extract the visible part of the main spike, which enables us to track anthesis progression. We divide the feature extraction into two steps: in the first step we identify the longest branch in the tassel; in the second step, we identify the part of this longest branch which is (the visible part) of the main spike.

In the first step, we skeletonize the image [38], thus converting it into a graph that can be easily analyzed. The bottom tip of the graph is assumed to be the lower end of the tassel, after which all the possible endpoints (i.e. the end points of individual tassel branches) as well as paths
were detected based on morphology methods [39]. The longest of these paths contains the main spike, and the corresponding end point to this path is the tassel tip.

In the second step, we identify the branching points (i.e. locations along the longest path where secondary spikes start, see Figure 4.12). The main spike of the tassel is then the part of the path between the top most branch point and the tip of the tassel. The tassel was finally cropped between the topmost branch point and the tassel tip. This object represents the visible part of the main spike. The width of this object at each pixel was obtained based on the method described in [15].

![Figure 4.12. a) The red line is the tassel path from the starting point to the tassel tip and blue dots are the branch points. b) The main spike of the tassel was cropped from the top moat branch point](image)

**Results and Discussion**

We show results and discuss the results of each step separately. This will enable nuanced assessment of each step. We finally illustrate the full pipeline for two representative camera data sequences.
Tassel Detection

Transfer learning was used to train the RetinaNet object detection model to reduce the number of training epochs needed. The model was initialized with the ImageNet weights, and several training campaigns were deployed with different hyperparameters. The average time for training the model was approximately five hours, and the best model was chosen based on the higher mAP value obtained by testing the trained model on the unseen dataset, because the better-trained model explored features rather than memorizing the pattern. This model uses two loss functions, a regression and a classification loss function [35]. The first function is for bounding box regression, with value 0.078 after 100 epochs. The other is object classification to identify whether or not the object is tassel, and its value was 0.0006 after 100 epochs. The mean average precision values were used to check the robustness of the trained model on the testing dataset. RetinaNet returns the locations and probability of each detected box for each image, and Mean average precision (mAP) was used for evaluating the model, producing a mAP value of 0.91. Figure 4.13 shows sample predicted boxes in red and ground truth in blue, with mAP showing a notable match between the model and human annotation.

Figure 4.13. predicted and ground truth boxes are shown in red and blue colors, respectively
Tassel Classification

After an exhaustive hyperparameter search of various convolutional neural network architectures, we chose the network shown in the Figure 4.8. RGB images with sizes $387 \times 516 \times 3$ pixels were used for inputs, and $28 \times 3$ by $3$ filters were used for each of the convolutional layers. After each convolutional layer, there was a max-pooling layer to reduce the computation load of the network and monitor important features. The activating function was Rectified Linear Unit (ReLU). The 2D arrays were flattened to enable SoftMax to classify the images as tassel or non-tassel. A dropout [40] was also added to the model to prevent overfitting. The precision, recall, and F1-scores of the model were 1, 0.875, and 0.93 respectively. Based on the results obtained, this model can be satisfactorily used for ensuring whether or not the box detection by the RetinaNet model is accurate. Figure 4.14 shows two images predicted by RetinaNet and identified by the classification model as tassel and non-tassels.

![Figure 4.14. The classification model predicts the left image as tassel and right image as non-tassel](image)

Tassel Tracking

Ideally, there would be two tassels per image captured by a camera. After tassel detection and classification, we must track the same tassel within a camera over the time period that the cameras are active. The cameras were checked each day to ensure that the tassels were completely located within the frame. Figure 4.15 is a scatter plot of the center points of tassels in
a camera. These points were categorized using a k-mean clustering method with two clusters. The points in a cluster (yellow points) were grouped as the same tassel and the images were cropped based on the box coordinates, and the same processing was done for the cluster of blue points.

![Figure 4.15. False positive predicted boxes by RetinaNet](image)

**Tassel Segmentation**

Extensive hyperparameter exploration was performed to identify a good CCN based segmentation model. The final accuracy of the model was 0.95. Figure 4.16 shows the accuracy and loss values over training for different epochs.

![Figure 4.16. Accuracy and loss value changes with every epoch](image)
The trained model was used to segment tassels from the background, producing as output a binary tassel image where the tassel is white and the background is black. A sample output image predicted by the model is shown in Figure 4.17.

![Image](image1.png)

Figure 4.17. Two sample tassels predicted by RetinaNet (left side) and their corresponding binary images predicted by the segmentation model

**Tassel Analysis**

Image processing and morphological operations were performed after the binary images of tassels are obtained. The main spike was cropped based on the approach described in the method section. The lengths of the cropped main spike varied image-by-image because a branch’s movement might occlude the main spike branch and make top-most branch point
detection harder, so the width was calculated from the tip to the bottom of the cropped main spike to enable comparison with the width at the same branch location.

**Flowering Pattern:** After removing the outliers, a 3D surf plot to monitor the width over time for one of the tassels in the camera is plotted, as shown in Figure 4.18. The Y axis represents images at different time points, the X axis is the pixel location of the main spike, and the Z axis represents the width of the main spike. As can be seen in the figure, greater width indicates flowering. Notice that the location of the flowering began from the middle of the tassel and continued toward the bottom and the top of the main spike. This flowering happened over 4 days.

![Flowering pattern of a tassel in a camera](image)

Figure 4.18. Flowering pattern of a tassel in a camera

The flowering pattern of the other tassel captured by the same camera is shown in Figure 4.19 as a 3D surface plot. For this figure the flowering took 3 days.
Conclusion

The goal of the paper was to develop an automated end-to-end pipeline for investigating maize-tassel flowering. We show that a workflow comprising several deep-learning and image-processing methods provides a robust end-to-end pipeline for this purpose. Tassel detection, classification, and segmentation were successfully trained. Future work consists of performing genetic analysis of the collected data, as well as linking the anthesis progression patterns with the yield.

References


37. Lee N. High-throughput phenotyping of above and below ground elements of plants using feature detection, extraction and image analysis techniques. 2016;


CHAPTER 5. MAIZE LEAF APPEARANCE RATE MONITORING BY IMPLEMENTING DEEP CONVOLUTIONAL NEURAL NETWORK

Seyed Vahid Mirnezami¹, Stefan Hey², Cheng-Ting Yeh¹, Patrick S Schnable², Baskar Ganapathysubramanian¹,³

Affiliations: ¹ Department of Mechanical Engineering, Iowa State University, Ames, IA, USA; ² Department of Agronomy, Iowa State University, Ames, IA, USA; ³ Department of Entomology, Iowa State University, IA, USA.

Abstract

Crop growth simulation models can help farmers and breeders predict crop performance, and in maize, Leaf Appearance Rate (LAR) is an important parameter used in crop performance simulation models such as APSIM. Since breeders and biologists would like to minimize human involvement in monitoring LAR, this trait can be monitored by applying a high-throughput phenotyping system. Engineers have entered the picture in collaboration with plant scientists to establish different and robust phenotyping methods, and in this study maize leaf appearance rate was investigated using high-throughput phenotyping approaches. We developed an imaging system for automatically capturing a time series images of maize plants under field conditions, with 380 RGB cameras were used to capture images from 380 rows. There were 6 plants with same genotype in each row that had different genotypes differed row-by-row, and the images were taken for 9 hours daily at 20-minute intervals for more than one month during a growing season. An end-to-end deep learning method was then used to count the numbers of leaves in the images. The dataset for the deep-learning algorithm, obtained using the Amazon Mechanical Turk platform, was created by one expert turker along with a well-trained turker. Results demonstrated that an end-to-end model with training based on the expert turker dataset performed very well, handling variation in images that included leaf occlusions and weather type. The $R^2$ between the ground truth obtained by the expert turker and predicted values was
approximately 0.73, 0.74, and 0.95 for three testing cameras. The model’s prediction performance demonstrated that the number of leaves increases with different slopes for different genotypes. The data can be used for further genotypic analysis.

**Introduction**

The staple crops that humans rely upon to meet their caloric demands have experienced remarkable increases in productivity over the last century ([http://www.fao.org](http://www.fao.org)). Plant breeding has drastically reshaped crop species to achieve better performance using selection based on genetic and phenotypic properties that are pre-requisite to yield enhancement [1,2]. Crop growth simulation models can help farmers and breeders in making predictions about crop performance [3,4]. To support credible simulation, crop growth simulation models require robust Eco-physiological functionality of various genotype × management × environment (G × M × E) combinations [5].

Plant leaves are the major organs in which photosynthesis takes place and as such they are of utmost importance for transforming light and atmospheric CO\(_2\) into biomass. In maize, while Leaf Appearance Rate (RLA) is an important parameter in crop performance simulation models such as APSIM [5], due to the labor-intensive nature of RLA assessment, genotypic variation in this trait has been widely ignored. Therefore, to assess genotypic variation of RLA, more efficient tools should be developed to meet the challenge posed by labor-intensive phenotyping.

Automatic plant phenotyping has become a vital tool for breeders and researchers and wider, and advances in hardware such cameras, data storage, and internet of things provide a platform for capturing a huge volume of high-resolution images in the form of so-called big data, and associated improvements in processing systems have supported efficient analysis of such big
data. Typical image processing algorithms along with computer vision methods implemented through machine learning are the most important current approaches to tackling plant-phenotyping problems.

Many researchers have contributed to phenotyping of plant leaves. In addition to automatic leaf counting, image processing has been used to in many studies to perform leaf detection, leaf segmentation, and leaf disease detection. Multiple leaf segmentation methods, including distance transform via optimal template selection and Chamfer matching, were compared in [6] using Arabidopsis plant images. Two studies concluded that an Guided Active Contour algorithm can be implemented for leaf segmentation [7,8] [9]. In another study simple thresholding was used within the HSV color space to count aphids on soybean leaves [10]. Another researcher developed an automated high-throughput phenotyping pipeline for capturing images of Arabidopsis and measured phenotypic traits such as leaf length and rosette area in 2D images [11]. A graph-based method and circular Hough transforms [12] were used by [13] to count Arabidopsis leaves. Altogether, numerous researchers have performed detection of leaf disease using image-processing algorithms [14].

Although use of image-processing approaches has accelerated the pace of leaf phenotyping by introducing graph-based algorithms [15], some drawbacks remain. These methods are (i) not generalizable to other plants, (ii) unable to handle leaf occlusion, and, finally, (iii) require long analysis. To address such barriers, machine learning has been coupled with image processing,.e.g., a Support Vector Machine (SVM) [16,17] model using features obtained by image processing approaches was used to detect tea-leaf disease [18,19]. Edge classification and an artificial neural network were implemented to count leaves. Such studies have proven that machine learning can assist image processing to obtain improved results [20].
Recently, a deep convolutional neural network has been widely implemented to improve accuracy in leaf phenotyping. One researcher created a mobile application that counted the numbers of leaves on orache plant images captured in a greenhouse by training a deep-learning model [21]. The number of leaves were counted in a CVPPP 2019 non-annotated dataset using an unsupervised adversarial domain [22]. In a counting application, a researcher applied a deep-learning model in combination with 3D image-processing approaches to count the number of grapes [23]. Generally, a number of studies have reported high-accuracy leaf-counting methods based on segmentation [24], detection [25], or direct counting [26–28].

Previous studies on leaf phenotyping have been of small scale, with some performed in non-agricultural field settings such as growth chambers and greenhouses. In addition, studies on maize-leaf counting and associated growth-crop models are rare. In one study, maize leaves of greenhouse images were counted by applying CNN networks [29], although only late stages of maize plants was used, so investigation of time-series data ranging from early stage to late stage is lacking. In the study reported here, we automatically monitored numerous genotypes of maize plant growth under actual field conditions by taking RGB images captured from early stages up to the time that tassels emerged. For these images the number of leaves were estimated by applying a deep-learning method that used CNN networks.

The goals of the study were to:

Develop an imaging system to automatically capture time-series images of maize plants under field conditions.

Create an application to assist the Amazon Mechanical Turkers in annotating the images.

Train a CNN network to use an end-to-end process to count the number of maize leaves at all stages.
Material and Methods

Field experiment

A large-scale maize plant experiment was conducted by the Plant Science Institute at Iowa State University during summer 2018. Maize plants selected from the SAM panel grown in an Ames, Iowa, field were organized in 380 rows (or plots) laid out in an East-West direction. There were six plants per row with each row having the same genotype, with genotypes arranged in the randomized split plot design.

Image Acquisition

An automated imaging system leveraging 380 commercial and cheap Nikon 20-megapixel cameras was designed and deployed. The cameras were shaded with attached mylar reflective-covering shields to help reduce the possibility of their being overheated by sun rays. Each camera used for imaging each set of six plants was located south of the row with the camera facing north, an orientation chosen to avoid direct sun exposure to the camera lens. Each set of 24 cameras was connected to an inexpensive raspberry pi microprocessor integrated with the internet of things (IOT). A solar-based system was used as the power source, and battery-charging, power-on, image-capture, and power-off steps were automatically performed.

Each image represented 6 plants with the same genotype, and between June 20th and August 10th during summer 2018 images were collected between 8 AM and 5 PM each day at 20-minute intervals, so that images were taken representing all stages of plant growth. Since each camera captured approximately 32 images per day, approximately 1000 images were collected by the entire camera system every day, and the total number of images was more than 380000 at the conclusion of the experiment. The size of each image was 5152x3864 pixels stored in JPEG format, corresponding to about 10 MB of storage space for each image, representing a total
storage requirement of about 2 Terabytes. Figure 5.1 shows 6 sample images, each taken by two cameras at time points of three different stages: early, middle, and late, showing that each genotype’s leaf densities differed significantly.

![Figure 5.1](image1.png)

**Counting method**

There were several challenges to counting the number of leaves shown in these images. First, there was a possibility that a plant from one row was not germinated (Figure 5.2-a), in which case one camera might acquire images from fewer than 6 plants per row. Another challenge was occlusion, and images could differ with respect to occlusion both within a row and among rows (Figure 5.2-b and c).
(a) some plants are not germinated  (b) Occlusion with other rows  (c) Occlusion within the row

Figure 5.2. counting challenges (a): some plants are not germinated, (b) Occlusion with other rows, and (c) Occlusion within the row

Weather was another variable, and this field experienced rainy, cloudy, sunny, and foggy weather, as shown in Figure 5.3. Considering all these above-mentioned challenges, we analyzed the images using an end-to-end deep convolutional network.

Sunny  Cloudy  Rainy  Windy

Figure 5.3. Different weather during the experiment

**End-to-End machine learning**

While this method used no typical image processing, a supervised end-to-end regression deep learning architecture was investigated for efficiently and accurately estimating the number of maize leaves in the images. The term “end-to-end” means that, given an RGB image, the network would estimate a regression number representative of the number of maize leaves. This approach: inspired by a redundant counting method called count-ception [35], used an Inception
model, [36] and offered several advantages by handling: (i) weather variation, (ii) occlusion, and (iii) different stages. Figure 5.4 shows the flow chart for this method.

![Flowchart](image)

**Figure 5.4. Flowchart for counting number of leaves using an End-to-End deep learning model**

This counting method requires a dataset for training the model. The input images were the RGB images produced by the cameras, and the label is dotted images, with each dot representative of a leaf tip, so the total number of dots is the number of maize leaves in the images.

**Dataset preparation**

**Amazon Mechanical Turk**

Training deep-learning models require a large volume of data and preparing such datasets can be very tiresome work. The Amazon Mechanical Turk (AMT) is a platform that uses humans to perform jobs that computers are unable to do. There are numerous turkers located all around the world who can complete specified tasks for payment.

In this experiment, we used Amazon Mechanical Turk crowd-sourcing platform to annotate the images from 366 maize-plant genotypes from the SAM diversity panel collected between June 23 and July 31, 2018 (38 days), each image containing a single row of six (6) plants, all of the same genotype. From each genotype, we then selected one image per day
(captured closest to 10 o’clock in the morning) for a total of 10,872 total images to be annotated through the Amazon Mechanical Turk platform in two separate campaigns.

In the first campaign, the size of images was shrunk from 5152x3864 pixels to 1030x773 pixels. We then randomized the order of the images and grouped them into 159 tasks, each containing on average 68 images. Each worker in the Amazon Mechanical Turk platform were presented with one image at the time and asked to draw a bounding box by identifying the top-left and bottom-right corners of the smallest box that would enclose the row plants in the foreground. The process was repeated on all the images within each group by the same worker, and a total of 3 workers were assigned to each task.

After obtaining the bounding box coordinates of each image from each worker, a consensus bounding box for the row of plants in the image was obtained by intersecting the annotations provided by all 3 workers on the same image and expanding them by 100 pixels on all sides. These consensus bounding boxes were then horizontally sub-divided into six (6) smaller segments for the second Amazon Mechanical Turk campaign. The second one was accomplished by only one expert turkers for 12 random cameras. An expert turker is a person who has deeper knowledge of the experiment and has spent more significant time on creating the dataset. In this campaign, each segmented image was cropped from the original image size because we wanted the turker to work with the original size of the image to avoid using an image-zoom option. Since it was felt better to make the turker’s tasks as easy as possible, asking them to create a dataset using the segmented images was deemed more logical [37].

The turker was then asked to place dots on all the leaf tips present in each segmented image, Figure 5.5 shows the web-based application for this purpose. For the second campaign, a
total of 2464 segmented images were randomized and grouped into 9 tasks containing around 273 segmented images each.

Figure 5.5. The web-based application given to each turker to annotate the leaf tips by a dot

The coordinates of the leaf tips in each segmented image were then converted into coordinates relative to the bounding boxes and the original images for downstream analyses., 369 annotated images were ultimately obtained from Amazon turkers.
Creating a Small Dataset by a Trained Turker

In addition, images from three cameras were provided to another turker and asked him to exactly follow the instruction. His dataset was then used for comparison with results from the expert turker for evaluating him.

Results

The goal of the paper was to count the number of maize leaves found in the images under field conditions. The images were analyzed to determine the most reliable method for this purpose. The following sections describe the methods considered.

Dataset preparation and Turker results

A deep-learning algorithm called countception was used to directly count the number of leaf tips in the original RGB images. The dataset for this approach was created by launching two campaigns using Amazon Mechanical Turk. In the first campaign three turkers per image were asked to draw bounding boxes around all 6 plants from the foreground row, i.e., 10,872 images were divided into 159 tasks containing 68 images on average. Each of these tasks were given to three turkers to help make sure they had done a good job by confirming one another. Also, if someone were to cheat and wrongly annotate, we have results from two other turkers as a check, so on average 3 turkers per task, i.e., 159 x 3 = 477 tasks were launched. The average time that each turker spent on annotation was 21 min, 55 seconds, and the entire campaign lasted for 1 day, 2 hours, and 8 minutes. Figure shows a sample image annotated by three turkers, with black, red, and blue bounding boxes related to the different turkers. A consensus-based method was implemented to obtain the best among all three boxes. To do this, the union of all the boxes was
considered to represent a best box (yellow box). The final box was also extended by 100 pixels in all four directions to ensure that all 6 plants were included (Figure 5.6).

Figure 5.6. Finding the final bounding box based on consensus method of three turkers. Black, Blue and red boxes are related to the three turkers annotated the images. The yellow box was obtained based on the union of these three boxes. Green box is obtained after extending the yellow box by 100 pixels from all sides.

The images were then cropped using the best box and divided into 6 pieces to retain the original size of the image so that the expert turker could more easily find the leaf tips and can locate a dot on each one (Figure 5.7).
As mentioned earlier 9 tasks were launched in the campaign. The entire campaign lasted for three days and the turker spent about two hours on each given task. The sub-images were stitched together to create the original images and dotted points were converted into the original size (Figure 5.8).
Figure 5.8. Turker data depicted by yellow dots for each piece as well as cropped original image after stitching and converting the coordinates. Moreover, annotated dots from the turker for each piece were reconstructed to the original image.

The expert turker annotated 2464 pieces cropped from 417 images. He spent additional time to accurately annotate the images. No further analysis was done by this turker. In addition, three cameras

**End-to-End Counting using the expert turker dataset**

The code used in the count-ception paper was used as the base code. 417 images were split into training, testing, and validation datasets. Images from 8 cameras were used for training, one camera for validation, and three cameras were used for testing. To create labels the annotated dots for each original image were inserted into a black image of the same size as the original image. In addition, both input and label images were resized from 5152x3864 pixels to
192x256 pixels based on the height/width ratio of the original image. The images were trained using the training images. 272, 19, and 107 images were used for training, validation, and testing, respectively. The testing dataset had the images captured by three different cameras, and both the input and the labels were obtained based on the method described in the regular turker training section. The $R^2$ values demonstrate that the model can predict very satisfactorily based on the data obtained by the expert turkers. The $R^2$ obtained for comparing the ground truth and predicted values for CAM197, CAM176, and CAM 278 were 0.73 and 0.74, and 0.95, respectively, meaning that the model trained based on the expert turker works better. In addition, the data obtained based on the trained turker was used to validate the model and demonstrated that the model is not biased towards the expert turker. The $R^2$ values for CAM197, CAM176, and CAM 278 were 0.6 and 0.61, and 0.78, respectively based the trained turker. Table 5.1 shows the $R^2$ values for both turkers. It means that the model is robust enough to predict the leaf rate appearance.

<table>
<thead>
<tr>
<th>Table 5.1. $R^2$ values between predicted and ground truth obtained by both expert and trained turkers for same images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert turker</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>0.73</td>
</tr>
<tr>
<td>Trained turker</td>
</tr>
</tbody>
</table>

Figure 5.9 shows plots comparing the model results to the data based on the expert turker, and trained turker. According to the plots, all predicted and turkers value trends are similar within each camera. Moreover, the mode predicts the number of leaves less than the data obtained by the expert turker. However, the increasing trends are quite similar. This end-to-end
method can also handle heterogeneous in-field variations. The model can predict number of leaves in the images with occlusion and/or foggy weather (b) with occlusion of the leaves.

Figure 5.9. Comparison between the ground truth and predicted values by the model trained based on the expert turker data
Conclusion

This paper describes two methods applied to counting the number of maize leaves using time-series images captured under field conditions. An end-to-end deep-learning model was used to estimate the leaf rate appearance. While the machine learning algorithm also could not segment the images under all conditions, including plants in the late stage and different weather types, the end-to-end algorithm was very promising. Models were trained based on the datasets obtained by the expert turker. The $R^2$ values for the model trained by data annotated by the expert turker were 0.74, 0.73, and 0.95 for three different cameras in the testing dataset which is quite satisfactory for field condition. In addition, the increasing trend is retained by the model and it can also handle variations in images. The estimated number of leaves can be used for further genotypic analysis. The experiment could be improved by having four plants per row, and it also would be better to ask turkers to annotate leaf tips on the slice of images while they can observe the entire image.

References


21. Valente J, Giuffrida MV. Leaf counting from uncontrolled acquired images from greenhouse workers.


CHAPTER 6. GENERAL CONCLUSION

Coarse feature of tassels, including shape and size, can influence shedding pollen, fertilization, and subsequently grain yield. Therefore, understanding tassel dynamics and characteristics as well as how it evolves during the plant growth can help the plant scientist community to increase the grain yield as a final goal. 351 tassels with different genotypes were used for the experiment. Tassel length, first lowest branch length, and angle as well as central spike length were measured by applying image processing and machine leaning techniques. Tassels were also classified to open and close structures to obtain accurate prediction for the traits. The results show that $R^2$ values for the tassel length and central spike length were 0.92 and 0.80, respectively. In addition, the $R^2$ values for the first lowest branch length and angle were 0.63 and 0.91, respectively. The $R^2$ values for the first lowest branch length was low compared to others because locating the first lowest branch point and its corresponding branch tip was hard due to branches occlusion. This study was done to create a robust algorithm for tassel phenotyping. Challenges were figured out for better tassel phenotyping in the field.

Then, we looked at a diverse panel in the field, using stationary cameras to capture 6 tassels every 10 minutes for 8 hours per day during a month. The main goal of this study was to show the feasibility of detecting tassel in images captured using an easy-to-access RGB images. The major challenge is identifying the region of the interest (i.e. location of tassels in the imaging window) in the acquired images. Camera drift, different types of weather, including fog, rain, clouds and sun and additionally, occlusion of tassels by other tassels or leaves complicated this problem. Computer vision tool and deep learning algorithms can assist to identify tassels. To create the dataset, AMT was utilized to annotate the images required for training the computer vision algorithm. Therefore, a framework was developed to annotate the images and evaluate the
turkers. Then, Faster-RCNN was customized and trained to identify the tassels in different images. After that, a boosting method was implemented to improve the dataset annotated by turkers. This approach is able to reliably identify a diverse set of tassel morphologies with the mAP of 0.81. The detected tassels can be segmented, and then morphological operation can be applied on the binary images to calculate the coarse features.

The goal of the next experiment was measuring the most complex traits in the tassel which is monitoring the flowering pattern in the function of time. We developed an automated end-to-end pipeline for investigating maize tassel flowering. We neatly connected the deep learning and image processing methods to create a novel end-to-end pipeline for the purpose. Tassel detection, classification, and segmentation were successfully trained. Moreover, the width of the main spike was tracked such that we are able to detect when and at which location the flowering is started.

In addition to tassel structures, crop growth simulation models can help farmers and breeders predict crop performance, and in maize, Leaf Appearance Rate (LAR) is an important parameter used in crop performance simulation models. This study described two methods applied to counting the number of maize leaves using time-series images captured under field conditions. An end-to-end deep-learning model was used to estimate the leaf rate appearance. While the machine learning algorithm also could not segment the images under all conditions, including plants in the late stage and different weather types, the end-to-end algorithm was very promising. Models were trained based on the datasets obtained by the expert turker. The $R^2$ values for the model trained by data annotated by the expert turker were 0.74, 0.73, and 0.95 for three different cameras in the testing dataset which is quite satisfactory for field condition. In addition, the increasing trend is retained by the model and it can also handle variations in images.
The estimated number of leaves can be used for further genotypic analysis. The experiment could be improved by having four plants per row, and it also would be better to ask turkers to annotate leaf tips on the slice of images while they can observe the entire image.