Corn Tassel Detection Using Computer Vision Techniques

Amber Benbow

Department of Geographic Information Science, University of Minnesota

E-mail: petri142@umn.edu

Abstract

Through the accessibility of high-resolution remote sensing technologies such as unmanned aerial system (UAS), it is possible to use computer vision techniques to extract key insights about crops from the images. These insights can help to determine if crops such as maize are stressed, ready to tassel or in need of additional nitrogen. Applying supplemental irrigation or nitrogen at key phases in the maize life cycle can increase crop yields and make better use of increasingly scarce natural resources. From these high-resolution images, it is possible to detect the critical life cycle points for maize such as tasseling can be identified using computer vision methods such as supervised and unsupervised machine learning algorithms in conjunction with traditional methods such as thresholding and image segmentation. Exploring these different methods illustrates their varying effectiveness rates for detecting the emergence of maize tassels.

*Keywords*: Remote Sensing, KMeans, Principal Components Analysis (PCA), machine learning, morphology detection, UAS imagery, drone

Introduction

Gathering data in an agricultural environment has never before been easier. UAS technologies have become more readily accessible both in price and those with the certifications to fly them (Algoma University, Canada et al., 2018). UAS imagery can be used in conjunction with publicly available satellite imagery to make informed decisions in an agricultural setting. In today's shifting climate, it has become more critical than ever to better manage resources to ensure the best possible outcome for crops. For example, the megadrought in the North American West, ongoing for the past 19 years, has become the second driest period since the late 800s (Williams et al., 2020). The researchers warn anthropogenic stressors such as dryer soil and reduced water reservoirs can push a moderate drought into a megadrought even without the addition of non-anthropogenic climate variability (Williams et al., 2020). As such, with water resources becoming increasingly stressed, it is more important than ever to ensure that crops are receiving the right amount of support, whether that be through increased irrigation or additional applications of nitrogen, at precisely the right time during their development. With crops such as maize, we know that the crop is most vulnerable to fluctuation in water availability during the tasseling phase and while the cob is forming (Çakir, 2004). As Gao et al. observed "supplemental irrigation at tasseling significantly accelerated kernel sink establishment… result[ing] in high yield and high harvest index" (Gao et al., 2017, p. 1).

Making use of information gathered from sensors such as UAVs is an example of precision agriculture. Precision agriculture, as defined by Perakis et al., "is a technology-enabled, data-driven approach to farming management that observes, measures, and analyzes the needs of individual fields and crops" (Perakis et al., 2020, p. 1). Using this technological approach to analyze and make use of site-specific information increases crop yields, decreases costs and reduces the environmental impact of producing food (Perakis et al., 2020). The key is having on demand, site specific information with which to make critical agricultural decisions such as applying additional resources to some crops as opposed to others and even, within the same crop type, determining which fields need additional resources in order to increase yield.

Identifying these key phases within the complex field environment, using accessible technology and automatic methods can ensure the timely application of these critical resources. It is important to detect corn tasseling as early as possible to ensure that the crop does not later become stressed during this critical growing phase, thus negatively impacting yields.

Datasets

*Study Area*

Map

Description automatically generatedThe study area is a family farm located near Sleepy Eye, Minnesota. It has been in production for at least 10-15 years and is not specifically used for the purpose of research. The study area was chosen due to the data being readily available through a private agricultural insights company, Sentera, and as an opportunity to observe maize development under typical growing conditions. Data was collected by a PHX Pro UAV and Double 4K Analytics Camera. The purpose of the UAV flight was to retrieve general health information of the field, which included but was not limited, to the detection and estimation of corn tassel development.

Figure . Data collection occurred at a family farm outside of Sleepy Eye, MN.

Two sets of 27 RBG images were collected, one on 7-15-21 and then again on 7-27-21. These are dates when corn is tasseling in Minnesota. Each dataset had one image that was taken “out of bounds” of the crop field resulting in the images being deleted from the datasets, for a total of 52 useable images.

Background

*Computer Vision*

Being able to analyze, process, and extract meaningful information from digital images and video is known as computer vision (Kee & Compeau, 2019). Machine learning techniques are commonly employed to accomplish this task. Often, computer vision systems imitate human vision (Peters, 2017). Common computer vision problems are point of interest identification, image matching, segmentation, feature selection, edge detection, working with color spaces and identifying objects (Peters, 2017). Computer vision has some key concepts which are important to understand to gain useful information from images.

*Object identification*

Object identification within computer vision has a few class types. The first is image classification where images are sorted into common sets based on what is contained within them. The classic example is distinguishing cat from dog images. The second is object instance segmentation (Liu et al., 2020). This is where each distinct object appearing in the image is detected and segmented from the others. Finally, there is semantic segmentation where the objects of a similar class are detected and segmented as a group (Liu et al., 2020). Additionally, within object detection there is a distinction between object detection models that only identify the object within an image by drawing a bounding box around it, also known as generic object detection, and pixel wise masks that closely follow the shape of the object detected segmenting it from the image itself (Liu et al., 2020). The second approach of using pixel wise masks can decern more precise information regarding the objects identified.

*Segmentation*

To isolate an identified object from an image, an image is divided into multiple parts called segmentation. There are different methods to achieve this. Region based segmentation is done by thresholding the pixel values. For example, within a gray scale image if a pixel value is 180 or higher, then the pixel is kept. This operation would produce a binary mask where each pixel has a value of either true or false. Combining masks with color or gray scale images would reveal all of the pixels, and possibly objects, that were isolated from that threshold. This operation can be done globally by defining one pixel threshold for the entire image or locally by specifying bins that pixels could fall within. Local thresholding is an example of semantic classification as it would classify all objects identified with certain criteria to one class. Thresholding can produce challenges, however, because sometimes the object of interest may fall within a narrow range or need additional information in order to properly segment it from the rest of the image.

Additionally, an alternative way to segment an image is by using edge detection. This is performed by utilizing specialized algorithms that run a filter window over an image to detect the edges within an image. Common algorithms are the Canny and Sobel operators where each operator views the maximum pixel value as a candidate edge point (Song et al., 2017). It systematically examines each point within an image and where there are the greatest differences, it infers what the edges of regions are in the image.

*Color Space*

Color space is where the extracted pixels from an image are assigned a value within a color space, whether that be the RGB, HSV or gray-scale, and then plotted in 2D or 3D space in order to ascertain to what degree the image displays contrasting regions of pixels (Klette, 2014). Plotting pixels in a color space assists in finding the visual patterns of the image or images. Finding clusters of these color spaces is another method for segmenting an image. Each color space offers different advantages when working with images. For example, if two objects in an image share a similar RBG color space, but their HSV color space displays distinct regions, then performing a segmentation on the HSV space would be desirable as that would properly extract the desired object from the image. It is important to examine color spaces in the exploratory phase of image extraction as it can help to identify a useful method for extracting the desired information.

*Supervised vs Unsupervised Machine Learning*

Computer vision also utilizes various machine learning algorithms to infer information from images. There are three primary types of machine learning; supervised, unsupervised, and reinforcement learning (Swamynathan, 2017). According to Swamynathan, supervised machine learning is where an algorithm is fed two sets of data (2017). One set is training data which teaches it what the desired class labels should be and the other is a validation set which would not include any class labels when being presented to the model (Swamynathan, 2017). Training data can take many forms whether that is a CSV file with class names associated with each row of data or a folder of images which include XML files containing the bounding box coordinates for each identified class object within the image. The fundamental hallmark of supervised learning is that the algorithm has an answer to extract patterns from with the goal of extrapolating the correct class label for the unmarked validation set (Swamynathan, 2017). Common supervised machine learning algorithms are Support Vector Machines (SVMs), neural networks (NNs), and convolutional neural networks (CNNs).

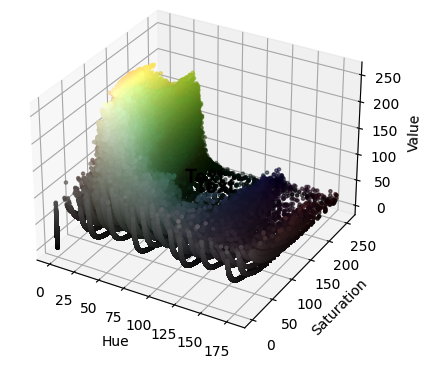
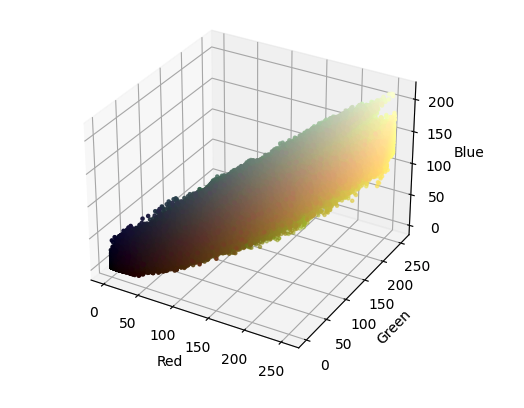
Unsupervised machine learning is where an algorithm finds patterns within a dataset, without class labels, and groups similar information together (Swamynathan, 2017)). Common unsupervised machine learning algorithms are regression analysis, K-means, K-nearest neighbors, and Principal Component Analysis (Swamynathan, 2017). Supervised machine learning offers greater accuracy but is expensive due to the labor-intensive nature of creating properly labeled training data sets. Unsupervised learning can be useful for exploring datasets, initial classifications, and discovering existing patterns within the data (Swamynathan, 2017).

Pre-processing

The acquired image data set was pre-processed prior to using it for the supervised and unsupervised method. After looking at all of the images, the "out of bounds" image was immediately deleted from the dataset. The images from both sets were also renamed using a set convention and which removed any special characters from the file names.

Following the methodology of Zan et al, a 75% central crop of each image using the TensorFlow Python library (2020). This reduced any distortion which may have been present at the edges of the images. Following this, data exploration was performed by experimenting with global thresholding of the pixels. For several images, there was a sweet spot around the 180-to-190-pixel value of the gray scale image which identified tassels with the least amount of background noise. This method alone was not sufficient, however. While tassel regions were clearly visible on the binary mask, there was a great deal of interference by the brighter pixels from the maize leaves.

Figure RGB and HSV color spaces for a single image.



Next a few images were selected for color space examination. Each image was converted to both RGB and HSV values. Each set of these values were then extracted and plotted in 3D space using Matplotlib to inspect them for visual patterns. There was some clustering in the RGB space which looked to be a golden or yellow color (Matplotlib can apply the color of the pixel onto the 3D space to help the user infer the patterns). Again, there was some clustering in the HSV color space, but not as distinctly as compared to RGB. Possibly combining both RGB and HSV as feature values would assist in being able to extract the tassels alone from the images.

Method I: Supervised Transfer Learning with YOLOX

After initial exploration, the first method to explore automatically identify maize tassels was via transfer learning of the YOLOX (You Only Look Once X) model in Google Colab. YOLOX requires that images be 640 by 640 for training. All images were re-sized to conform to the size required by the model. Next, the dataset was divided into training and validation image folders by a 70/30 split. Both the validation and training sets contained a mix of both flight dates to ensure good model generalization of differing lighting conditions.

To prepare the training set, a Python tool called LabelImg was downloaded and employed. This tool allows the user to drag the mouse over the image to create a bounding box and assign a custom class to each bounding box. A user may include as many bounding boxes as desired. The three classes chosen for training were: “leaf”, “tassel”, and “shadow”. These were the prominent features of the images. While some images did show bare soil or grass, these features were not of interest and thus not tagged. Each image received approximately 120 tags, where each class was about evenly represented in the image with a preference given to tassels. The LabelImg tool saves the training data to an XML file to be later retrieved by the model. Each image in the training folder was tagged and a correspondingly named XML file was saved with it.

Creating a good training set can be challenging. Generally, when creating bounding boxes, it is best to include the desired object fully. When there boxes overlap, include where the desired object would extend to behind another object. Classes should have approximately the same representation of objects in an image with a preference for the desired class. While this was not tested, it is suspected that mimicking known patterns in the training set would help the model determine that pattern is a factor in determining the class. In this example of maize tassels, since crops are planted in a uniform row, using that regular row pattern by identifying full rows of tassels could help the algorithm determine distinguish between a true and false positive (i.e. tassel from background noise).

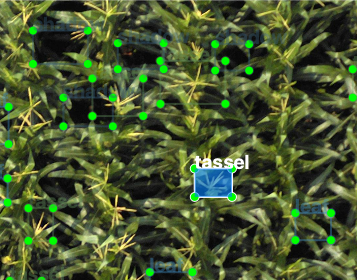


Figure 3. Example of the training set as viewed within LabelImg.

Training of the YOLOX model was performed in Google Colab to utilize Google's computing power. All needed dependencies were installed, the training and validation datasets were mounted onto Google Drive, the custom labels were specified, and then a branch of YOLOX was made. Training was performed several times in an attempt to improve training results. Each attempt was made with modifications made to YOLOX's epochs (100, 200, and 300, respectively). Training took between two to three hours on Google Colab’s GPUs depending on the number of training epochs chosen. More epochs increased the training time.

In this kind of detection model, the evaluation criteria is called Intersection over Union (IoU). This compares the overlap between the bounding boxes that the model created and which were provided as ground truth through the validation images. This IoU number ranges from 0-1 and can best be thought of in terms of percentages. The final run of the YOLOX model achieved an IoU of 0.50. In other words, when the model made a prediction its bounding box overlapped with the validation image's bounding box by approximately 50%. This is a decent overlap, but not enough where the model could be put into production. The overall mean average precision for all three classes was 0.30. This meant that the model was able to accurately predict each class about 30% of the time.

When fed the validation images, the model did not perform as well as expected. Ideally, the model would have enough training data to extrapolate the criteria of a tassel versus a leaf, especially because of their similar gray scale pixel value and then apply that criteria to the entirety of each validation image. Instead, the validation images with the class predictions looked similar percentage wise to the training images. There were around 20 of each class identified on each image, but not thoroughly across the image. To improve results using this method of transfer learning, a more robust training set would be needed, where every single instance of each class is identified in each training image. Using this kind of approach is incredibly labor intensive to produce. Producing approximately 4,200 bounding boxes for the 35 images in the training set took 8 hours of manpower.

Figure 4. Validation image after classification was performed by YOLOX.



Method II: K-means and Principal Components Analysis

Given how laborious the supervised training method is, it was decided to explore isolating the maize tassels using unsupervised methods. A popular algorithm for unsupervised training is K-means. This algorithm clusters similar pixels based on the number of clusters as defined by the user. This machine learning algorithm is often used in conjunction with Principal Components Analysis (PCA). PCA is an algorithm that determines the features which have the greatest amount of variation between them and thus are more likely to have influence on the prediction outcome of models such as K-means. When K-means is fed only the most important features, it is able to better group like pixels. K-means struggles when clusters are of differing sizes and densities or if there are outliers within the dataset that skew results (Swamynathan, 2017).

For this method, each image was converted to both RBG and HSV. The RBG pixel values which range from 0-255 were normalized so that the pixel values would be on the same scale as HSV which ranges from 0-1. The two image arrays were combined using cv2's addWeighted function. Then each pixel containing both sets of values was extracted, appended to a list, and then later converted to a NumPy array. This array was then fed into PCA to determine the three most important components of the array.

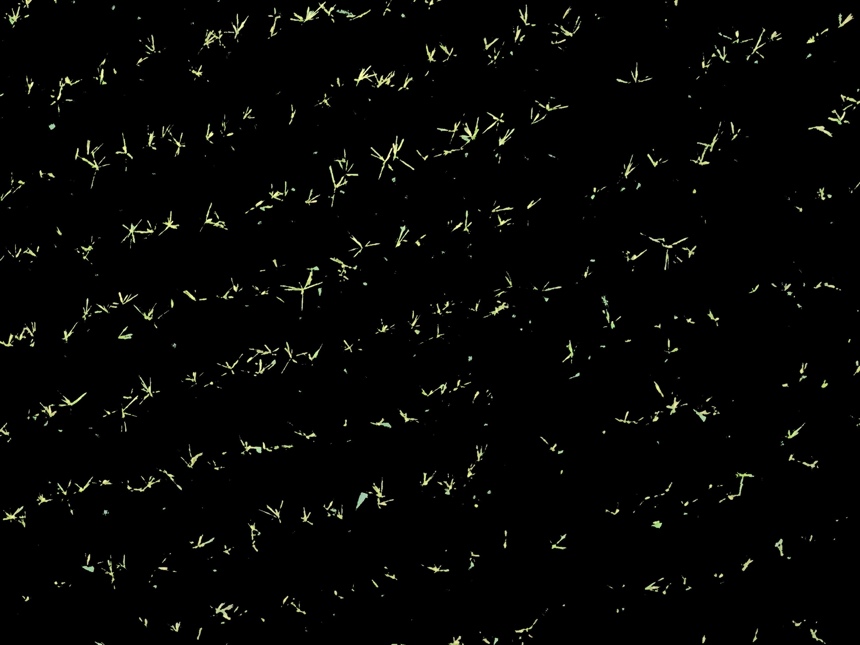


Figure . Successful K-Means segmentation. Results include tassels nearly exclusively with little interference from the leaves.

PCA found that the RBG pixels had the most variation within array. Once the principal components were determined, an array of only the RGB pixels were extracted and fed into K-means for image segmentation. K-Means produced a segmented image displaying each cluster. This segmented image was then looped through to isolate the cluster which was most effective at isolating only the maize tassels from the image. At least one cluster within each run was effective at segmenting the tassels without too much interference from leaves. Images from the 7-15-21 set had more even and consistent lighting and thus produced crisper segmentations. Comparatively, the images from 7-27-21 appeared to have more sunshine resulting in a kind of interference where K-Means was not as able to distinguish leaves from tassels as well.

Discussion & Conclusion

Between the supervised transfer learning and unsupervised learning methods, it is clear from visual results and time invested that the unsupervised method was superior. The transfer learning model did not generalize the task at hand well enough by tagging every single instance of the object classes (leaf, tassel, shadow) and both evaluation metics, IoU and mean AP were too low to put into production for reliable object detection. The results for transfer learning could be improved by increasing the number of objects annotated in the training images, by increasing the number of images included within the training set, or by attempting to train a model from scratch.

For the unsupervised learning method, results could be improved by running a large array containing all of the pixels for each image within the training set into K-Means several times to produce a more generalized number for the centers. These ideal centers could then be used on individual images to better isolate only the tassels from the images. This method could address the lighting inconsistences noticed between images. Additionally, including both the RGB and HSV features may improve the performance of K-Means. A combination of both approaches could be employed to progress future results.

Acknowledgements

Many thanks to Sentera and Terrance Tauer for generously sharing their images and allowing this work to be possible.

References

Algoma University, Canada, Zhang, C., Kovacs, J. M., Nipissing University, Canada, Walters, D., & Nipissing University, Canada. (2018). The use of unmanned aerial systems (UASs) in precision agriculture. In Silsoe Solutions, UK & J. Stafford (Eds.), *Burleigh Dodds Series in Agricultural Science* (pp. 107–128). Burleigh Dodds Science Publishing. https://doi.org/10.19103/AS.2017.0032.04

Çakir, R. (2004). Effect of water stress at different development stages on vegetative and reproductive growth of corn. *Field Crops Research*, *89*(1), 1–16. https://doi.org/10.1016/j.fcr.2004.01.005

Gao, Z., Liang, X.-G., Lin, S., Zhao, X., Zhang, L., Zhou, L.-L., Shen, S., & Zhou, S.-L. (2017). Supplemental irrigation at tasseling optimizes water and nitrogen distribution for high-yield production in spring maize. *Field Crops Research*, *209*, 120–128. https://doi.org/10.1016/j.fcr.2017.04.015

Kee, K., & Compeau, T. J. (2019). *Seeing the Past with Computers: Experiments with Augmented Reality and Computer Vision for History*. University of Michigan Press. https://doi.org/10.3998/mpub.9964786

Perakis, K., Lampathaki, F., Nikas, K., Georgiou, Y., Marko, O., & Maselyne, J. (2020). CYBELE – Fostering precision agriculture & livestock farming through secure access to large-scale HPC enabled virtual industrial experimentation environments fostering scalable big data analytics. *Computer Networks*, *168*, 107035. https://doi.org/10.1016/j.comnet.2019.107035

Peters, J. F. (2017). *Foundations of Computer Vision* (Vol. 124). Springer International Publishing. https://doi.org/10.1007/978-3-319-52483-2

Song, R., Zhang, Z., & Liu, H. (2017). Edge connection based Canny edge detection algorithm. *Pattern Recognition and Image Analysis*, *27*(4), 740–747. https://doi.org/10.1134/S1054661817040162

Swamynathan, M. (2017). *Mastering Machine Learning with Python in Six Steps*. Apress. https://doi.org/10.1007/978-1-4842-2866-1

Williams, A. P., Cook, E. R., Smerdon, J. E., Cook, B. I., Abatzoglou, J. T., Bolles, K., Baek, S. H., Badger, A. M., & Livneh, B. (2020). Large contribution from anthropogenic warming to an emerging North American megadrought. *Science*, *368*(6488), 314–318. https://doi.org/10.1126/science.aaz9600

Zan, X., Zhang, X., Xing, Z., Liu, W., Zhang, X., Su, W., Liu, Z., Zhao, Y., & Li, S. (2020). Automatic Detection of Maize Tassels from UAV Images by Combining Random Forest Classifier and VGG16. *Remote Sensing*, *12*(18), 3049. https://doi.org/10.3390/rs12183049