Image-based Plant Phenotyping: Localize and Detect Plants, Measure Projected Leaf Area and Perform Individual Leaf Segmentation

Arpit Mathur
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
a.mathur@student.unsw.edu.au

Asha Karki
School of computer Science and Engineering
University of New South Wales Sydney, Australia
asha.karki@student.unsw.edu.au

Gordon Chen
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
gordon.chen1@unsw.edu.au

Navya Vashisht
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
n.vashisht@student.unsw.edu.au

Zhihan Liu
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
zhihan.liu@student.unsw.edu.au

***Abstract*—Phenotyping is one of the observable characteristics of plant for efficient and sustainable agriculture. Image based plant phenotyping is a growing application area of computer vision in agriculture. The key tasks are to identify plants, their species, localize their position as well as segment leaves to observe the projected area and segment all of the individual leaves in images. This paper focuses on the work done on the above mentioned tasks. The first task involves localizing each plant and its position. In that task, we apply connecting components algorithm and the distance metric to detect the plant and draw the bounding box of the plant. From that, we were able to get 82.27% average precision score for Ara2013-canon data set which is the highest among all the data sets we were provided. The second task is to measure the projected leaf area, where we use foreground extraction method with histogram threshold and binary threshold to extract the leaf area. From that, we observe a score of 94.47% on Dice similarity coefficient and 89.51% on the intersection over Union metric on overall data tray sets of Ara2012 and Ara2013. The third task consists of segmenting individual leafs from a plant. This task is the most challenging aspect of this project since most of the leaves were overlapping due to similar appearance and shape characteristics. We attempt to solve it by using super-pixel segmentation and we then apply watershed segmentation to extract individual leaves. However, it wasn't the best, shown by our low metric score of 44.0% of Symmetric Best Dice on overall data plant set of Ara2012, Ara2013 and Tobacco. This task was challenging as most of the leaves were overlapping, as they do share the same appearance and shape characteristics.**

***Index Terms*—Plant phenotype, plant detection, projected leaf area calculation, plant segmentation, multi-instance segmentation**

I. Introduction

Plant phenotypes (from Greek pheno- 'showing', and type 'type') refers to the observable characteristics of an organism [1]. In the case of plants, traits such as height, biomass, leaf shape, seed germination rate and project leaf area would be good examples of the physical traits. We quantify such phenotypes to identify differences between structured populations in order to identify disease-ridden plants as well as to identify high yield sub-varieties of the species to further cultivate an ever increasing crop yield. Individual plants (for examples in the case of Leafy vegetables such as lettuce, spinach and cabbage) can be chosen to breed to increase the quantity and/or quality of the plant for further consumption. In this report, we showcase and compare methodologies to detect the plants as well as quantify the quality of biomass that specific plants have via their images presented to us via the Plant Phenotyping data set [2] [3] via supervised or un-supervised computer vision techniques.

1) Data set: The Plant Phenotyping data set is a collection of benchmark data sets in the context of plant phenotyping with annotated imaging data for plant/leaf segmentation, detection, tracking as well as classification and regression problems [4]. It contains RGB images in png format as well as csv files containing locations of the bounding box edges for segmented plants. The data set contains additional features such as leaf being segmented either via bounding box or different colours, each leaf centroid location and the outline of the whole plant. All of these features are stored in png format.

2) Metrics: For Task 1, we utilize the metrics of Average Precision to quantify the quality of the bounding boxes being developed while segmenting plants in an image. For Task 2, we utilize Dice Similarity coefficient and Intersection over Union while segmenting the images for individual plants while quantifying the Projected Leaf Area. For Task 3, we utilize Symmetric Best Dice measure to identify the quality of the multi-instance segmentation image by comparing our labelled image with given labelled image. Further details of our implementation is discussed in the method and experimental part.

II. Literature Review

Minervini, Fischbach, Scharr and Tsaftaris discuss the various problem statements which may be worked upon using the Plant Phenotyping data set [2] [3] such as plant detection and localization, plant segmentation, leaf segmentation, leaf tracking, boundary estimation, generic classification and regression problem statements.

In their thesis, Aich discusses the deep learning methodology to detect plants and their leaves [5]. They utilize SegNet [6] [7] [8], a Deep Convolutional Encoder-Decoder Architecture for object detection researched and developed by the University of Cambridge. They augment the data set by adding flipped, Gaussian blurred, sharpened and rotated images to the training sets. This generate 800,000 images from the 800 original images, which is later fed to a re-initialized SegNet model trained for an overall fifty epochs. In another network, they train the model to learn the number of leaves based on the segments (plants) of the image.

In the field of plant phenotyping, it is an essential step to segment plants from their background. Though very few researches focus on the projected leaf area (PLA), we can still get some inspiration from research works on other purpose, like leaf segmentation. Hanno Scharr et.al published a collation study on leaf segmentation, which introduced several methods for plant segmentation[7]. They used approach based on active contours on Arabidopsis images and colour-based approach to tobacco ones. Results were then manually refined with the help with graphics editing software to get the ground truth. The first segmentation method introduced is called IPK Gatersleben, which uses machine learning algorithms. 3-D histogram cube are built to calculate the probability for a pixel in the training set with certain colour of belonging to either the background or the foreground. The second method is based on the idea of super pixels. Images are first converted into Lab colour space, where they are over-segmented into super pixels. Then, seeded region growing algorithm is applied to extract the plants from the background. The third method uses Chamfer matching, which aligns one object instance in an image with a given template. It is originally designed for segmenting leaves in a video. A set of templates with different shapes, scales and orientations were used to make the matching algorithm adapted to the plant segmentation on images. The last method involves the use of deep learning algorithm via a supervised neural network. The network's features are the R-G-B values of each image pixel, excessive Green value (2G-R-B) and its text features like variance filtered green channel. For all images, at most 3000 pixels of each class are randomly chosen for the training.

Leaf segmentation of a plant is a challenging task especially when the canopy is highly crowded and leaves are highly overlapped. In their paper [9] on overlapping-free leaf segmentation method for plant point clouds, Dawei Li1 and Yan Cao1 present a new 3D joint filtering operator which uses Radius-based Outlier Filter (RBOF) and a Surface Boundary Filter (SBF) to segment occluded leaves. The plant is passed through multiple filters which include spatial region filtering, "RadiusOutlierRemoval" method from the PCL library, statistical k-nearest neighbor filter, color-based filter and lastly down-sampling method which removes the overlapped parts among the leaves. After obtaining the filtered regions, facet over-segmentation algorithm is applied which generates clusters of facets. The facets which get over-segmented are merged back into the pre-segmented leaf centers. Now, these unlabelled facets are labelled from the surrounding labels.

Adrian Rosebrock in his web resource [10] discusses how super pixel segmentation is beneficial to group local pixels as it achieves over segmentation which causes no loss of pixels and effective computation which lessens the complexity in the images. OpenCV library [11] provides super pixel segmentation functions to achieve segmentation. Watershed segmentation from PlantCV [12] provides algorithm for detecting boundaries of objects.

III. Methods

We utilize the below listed methods during work on the tasks described in this paper.

A. Identify green portions of image

In order to identify the green sections of the images, we convert the images from its RGB to HSV color-space and then extract pixels which are in 35 to 75 Hue ranges [13]. We utilize this method to identify the regions in the image of green Hue. By doing this in the first step, we are able to reduce the region of interest by a considerable margin as we remove pixels related to the background and retain the pixels of the plant in the foreground, as we may see in Fig. 1 and Fig. 2.

Fig. 1. Raw image

Fig. 2. HSV processed image

B. Morphological Transformations

We utilize the erosion and dilation methods of the opencv library [14]. Performed on binary images, these methods "erode" the boundaries of the pixels in the foreground (which in our case is in white color). Since we wish to remove the noise that we find in the HSV processed images from Method A, we will execute erosion and dilation consecutively which reduces the size of the foreground boundary and therefore "shrinks" the object in question. By dilating the image after the erosion, we are able to "reclaim" the eroded area, as well as reduce the noise in the image. An example of the same may be seen in Fig. 3 and Fig. 4, where the small flecks of white are removed after the Morphological Transformations.

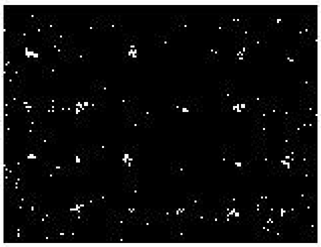


Fig. 3. HSV processed image

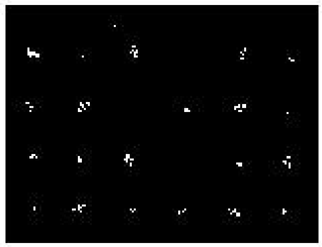


Fig. 4. After Morphological Transformations

C. Connected Components algorithm

The Connected Components algorithm takes in the noiseless image, where the foreground and background are allocated unique value. The algorithm parses the image to identify regions in the image separated from the others, therefore deriving the segments of the image. We see the execution of the algorithm via the Fig. 5. and Fig. 6. We observe in a few cases that the images generated have separately segmented leaves, therefore necessitating integration of multiple components [15].

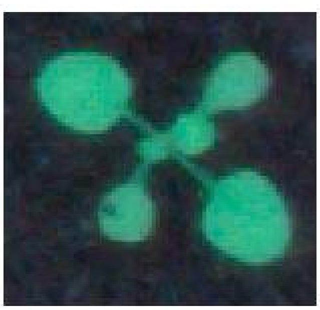


Fig. 5. Cropped image before connected components algorithm

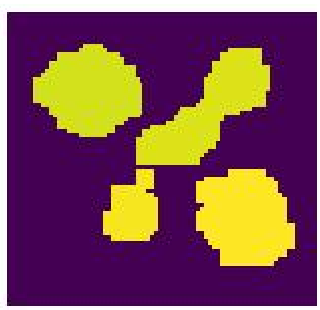


Fig. 6. Cropped image after connected components algorithm

D. Connected Components Merging

As we see in Figure 6, multiple leaves are shown as different components which potentially differs our count by quite a margin. In order to mitigate this, we utilize the centroids of the components to find the distance between that component and other component's centroid. By setting a threshold of minimum distance between 2 centroids allowed, if we find that the distance between 2 components' centroids are less than the threshold, we give both components the same number ID which indicates that those 2 components have 'merged'. From that, it decreases the number of components while ensuring the plant counts are correct.

E. Gaussian Blur Algorithm

We use a Gaussian Filter to remove the Gaussian noise prevalent in the image using and appropriate kernel [16]. We utilize Gaussian Filter to smooth images in order to remove strong edge which is prevalent in Task 3.

F. Intensity Thresholding

Intensity thresholding is an intuitive method for image segmentation. We use a threshold T in gray scale colour space, and apply a piece wise function f to each pixel in the images.

$$f(x) = \begin{cases} 0 & x < T \\ 255 & x \geq T \end{cases} \tag{1}$$

G. Median Filter

Another algorithm to remove noises in the images is called median filter, which is a spatial filter method. The main idea for median filtering is to replace the value of a pixel with the average value considering the neighbouring pixels.

Much like Gaussian Filter, we also utilize Median Filter to remove strong edge from images.

H. Support Vector Machine

Given a set of features variables and its labels, SVM is a supervised machine learning algorithm which aims to find a hyper plane that splits the sample space consisting of feature variables such that the distance to the nearest element of each label type is the largest. The key advantage of utilizing SVM is its ability to map features in high dimension space via the use of kernel functions such as Gaussian radial basis function. SVM is utilized in this project to solve binary classification between background and plant pixels.

I. Super pixel method

Super pixel method is defined as a group of the pixels that share common characteristics such as colour. The main benefit of super pixel is a group of pixels carries more information about the object of interest than a single pixel. Another benefit is super pixel is convenient and compact representation of image which will be useful for task that involves more processing such as leaf segmentation in Task 3.

Super pixel segmentation algorithm is similar to the k-means based approaches as it defines the distance function to be the balance among the boundaries and the intensities belonging to one super pixel [17]. Super pixel can be used as image processing technique before other segmentation algorithm are utilized.As in task 3, we used super pixel as prepossessing part which help us to find the boundaries of leaves in another way different from task 1 and 2.

J. Watershed Segmentation Method

Water Segmentation method is used for separating and extracting different objects that are overlapping each other in images such as multiple leaves highlighted in task 3. Given a marker array which indicates the local maxima of the image, the watershed algorithm treats the pixels in the input images as local elevation by flooding the pixel, starting from the markers. The flooding continues until the valleys of different markers meet each other, hence resulting in a mixture of different objects. As a result, an accurate marker array is absolutely important to ensure that the watershed segmentation perform at its best, a problem evident in task 3.

IV. Experimental Setup

The data sets for all three tasks were used from Plant Phenotyping Data sets 2015 [2] [3]. The programs run on windows 10 environment with Python 3.5 with the python libraries of OpenCV 4.2.0, Numpy 1.18.1 and Scikit-Image 0.17.2.

A. Task 1 (localize and detect plant)

In the first task, our task involves detecting and localizing plants in an image of multiple plants. We are given a subset of the Plant Phenotype data set [2] [3], totalling 70 files. Our aim is to draw bounding boxes around each plant and then evaluate the performance of our process via the Average Precision metric.

1) Method 1: Base Method: Since the plants in our images are depicted as green areas, we convert the image into HSV and choose the green area, as depicted in section III.1. We then remove the salt and pepper noise via Morphological Transformations as depicted in section III.2, which is then passed through the connected components algorithm as per section III.3. To ameliorate the components, in case they are close together but are labelled differently that indicate that they are not connected, we re-label the ones which are closer than a threshold to connect them.

2) Method 2: Median Filter implementation: Similar to the first method, we extract the green regions and remove the salt and pepper noise via median filter, depicted in section III.6. We again pass the image through the connected components algorithm and re-labeling components which are closer than a threshold we set.

3) Method 3: Median and Gaussian Filter implementation: Similar to the first method, we extract the green regions and remove the salt and pepper noise via median filter, depicted in section III.6 and the Gaussian Noise via the Gaussian Blur method, depicted in section III.5. We again pass the image through the connected components algorithm and re-labeling components which are closer than a threshold we set.

B. Task 2(Projected leaf area measure)

In the second task, our task is to calculate project leaf area of multiple plants in an image. We were given a subset of Plant Phenotype data set [2] [3], totalling 16 files for Ara2012 and 27 files for Ara2013-Canon. Our aim is to extract the

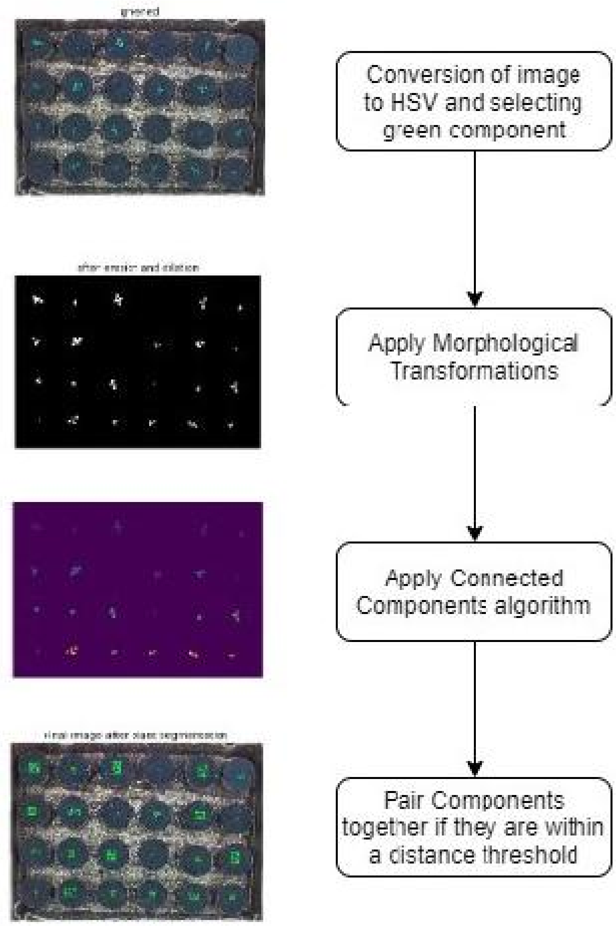


Fig. 7. Image processing flow for Task One

plant from the background and evaluate the performance of our process via Dice Similarity Coefficient and Intersection Over Union metrics which both compare our processed image with the correct final image given in the data set.

1) HSV Colour Space Method: Much like task 1, the plants in our images are depicted as green areas so we use HSV to extract plant pixels from the background. We first convert the image into HSV form and extract green pixels as shown in IV-B1b. In the HSV colour space, we use an upper bound (H_1, S_1, V_1) and a lower bound (H_2, S_2, V_2) to remove the background. We then apply a median filter to remove minor noises left in the image before converting it to gray scale where we use intensity threshold to separate foreground and background to create the final binary image. For the intensity threshold, we used two methods to convert gray scale image to binary image. One method is using CV2 binary threshold from the OpenCV library while the other methods involves the use of intensity histogram.

a) Method 1 (Using intensity histogram threshold): In this method, we want to find the values of $H_1, S_1, V_1, H_2, S_2, V_2, T$ to best match the labelled images given in the data set. After some variations, the best upper and lower bound we found are $(H_1, S_1, V_1) = (30, 60, 75)$

and $(H_2, S_2, V_2) = (100, 255, 255)$. The main appeal of this method is the discovery plants images have similar histogram shape with different amounts of pixels. Hence, the intensity with the most pixels (except 0) is retrieved where we have 2 further cases to determine which threshold to use. If pixel values are below 5000, then we uses the lower threshold of 70 to retain small plants in the image. Otherwise, the large threshold of 1000 will be used to retain larger plants and remove any small plants or noises.

b) Method 2 (Using binary threshold): For this method we used the cv2 threshold function to convert into binary and for this method we tried the upper bound and lower bound that are $(H_1, S_1, V_1) = (35, 30, 30)$ and $(H_2, S_2, V_2) = (90, 255, 255)$.

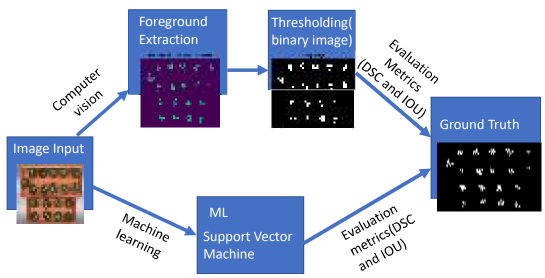


Fig. 8. Different stages in Task Two

2) Machine Learning Method: In this method, we treat each pixel as an input data. Features are the R, G and B values of the pixel. Label of the pixel is 0 if it is background and 1 if foreground. Hyper parameters of the SVM classifier are set to default. Only 1% of the data set have chosen as train set, other 99% are all test set. It takes too long to train the model if we do the normal 80:20 split.

3) Evaluation Methods: We use Dice similarity coefficient (DSC) and Intersection over Union (IOU) to evaluate the performances in task 2. For class i in a classification problem, let TP represent true positives, which are pixels correctly labelled as c_i, FP represent false positives, which are pixels wrongly labelled as c_i, and FN represent false negatives, which are pixels of class c_i wrongly labelled as other classes. Then we have Dice similarity coefficient

$$DSC = \frac{(2 * TP)}{(FP + 2 * TP + FN)} \tag{2}$$

Intersection over Union (IOU) is the value of TP(intersection) over (FP+TP+FN) (union). So we have,

$$IOU = \frac{TP}{(FP + TP + FN)} \tag{3}$$

C. Task 3 (Individual leaf segmentation)

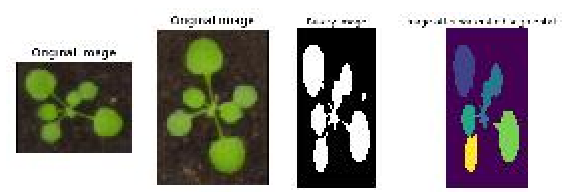
In the task three, our task is to segment the individual leaves. For this task, three sets of data was used which are Ara-2012, Ara-2013 and Tobacco, all of which are available on [2].

1) Method 1 (Using Watershed segmentation): As our feature in task three is the leaves, we need to extract the plant from its background much like the previous 2 tasks. So for the background removal, we convert the image to HSV form where we used particular range highlighted in task 2 to extract green plant pixels from the image.

Then we apply watershed segmentation to label and segment individual leaf of the plant as in Figure 9. To do so, we need to find the centroid of each leaf. To achieve this, we first converted the plant image to its binary version as highlighted in Figure 9. We then used Euclidean Distance Map which calculate for each non-background pixel the shortest distance to the nearest background pixel. The centroid of the leaf are mostly the point where it has a high Euclidean distance and is the highest among of its neighbour. This is achieved by finding the local maxima in the gray-scale image.

The main problem that we can have a lot of local maxima where multiple maximas represent the centroid of the same leaf of the plant. To deal with that problem, we set a custom threshold of the minimum distance between any 2 centroids. So, if 2 maximas are too close together, we remove one of them and continue. Hence, we ensure that there are only 1 maxima that represents the centroid of the leaf. All of those maxima are stored in an array or marker for the next step. [18].

Once we got the centroids in an array, we can then apply watershed transformation between the basins in the leafs area as shown in the Figure 9.



(a) (b) (c)

Fig. 9. (a) An Arabidopsis image and resize image, (b) its image after threshold, (c) its image after watershed segmentation.

2) Method 2 (using super pixel and watershed method: All tasks has the first step of extracting the plant from its background via HSV. However, after some experimentation, we found that super pixel with segmentation gives a satisfactory result, if not even better.

In super pixel segmentation method, the first step is to generate super pixels within our image via S.L.I.C. (Simple Linear Iterative Clustering). In term of amounts of super pixels generated, we experimented and determined that 1000 super pixels are enough to segment plant from its image for all test set given.

With those super pixels, the next steps is to convert the image to Lab colour space where we extract the 'a' range of image's Lab colour space as 'a' has the value of green which we wanted. With a custom threshold, we calculate the mean 'a' value of each super pixel and compare to the custom threshold.

If the mean value is lower than the custom threshold, it is determined to be a plant super pixel and hence, retain that super pixel. Otherwise, it is determined to be a background super pixel and hence remove that super pixel. The output should be the same as method 1 which is an image of just the plant.

After that, we then apply watershed segmentation explained before. The flow for task 3 is in shown in IV-C2.

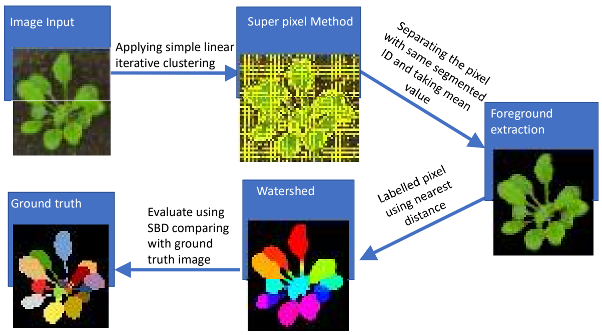


Fig. 10. Different stages in Task Three

3) Evaluation Metrics: For task three, we used symmetric best dice metric to compare our individual leaf segmentation with the ground truth from the data set. Symmetric average dice is also used to estimate the average leaf segmentation accuracy. This is achieved by checking all the leaves with the ground truth label which yields the maximum Dice which is taken as the result. Best dice is defined as [18].

$$BD(L^a, L^b) = \frac{1}{M}\sum_{i=1}^{M} \max_{1\le j\le N} \frac{2 \mid L_i^a \cap L_j^b \mid}{\mid L_i^a + L_j^b \mid} \qquad (4)$$

where $\mid . \mid$ denotes leaf area (number of the pixels),and L_i^a for $1 \le i \le M$ and L_j^b for $1 \le j \le N$ are sets of leaf object segments belonging to leaf segmentation's L^a and L^b respectively SBD between L^{gt}, the ground thruth and L^{ar}, the algorithmic result, is defined as

$$SBD(L^{ar}, L^{gt}) = \min\{BD(L^{ar}, L^{gt}), BD(L^{gt}, L^{ar})\} \quad (5)$$

For our case in task 3, in individual leaf labelling, all of the label leaves that is foreground are set to 1 while background is set to 0, similar to the ground truth images. This allows us to calculate and find the symmetric best dice for each plant and all data sets.

V. Results and Discussion

A. Task 1 (localize and detect plant)

1) Method 1: After executing the first task's flow, we observe that the bounding boxes fit the plants quite closely. We observe that the tendency of the bounding boxes to "cling" a bit closer than expected, which can be led to the fact that we have used methods to remove salt and pepper noise, which, in turn has reduced the area per bounding box by a few pixels for each plant, reducing the metric by a significant amount. A

TABLE I
Task 1 Average Precision Metrics

	Method 1	Method 2	Method 3
Ara2013-Canon	0.7442	0.7714	0.8278
Ara2013-RPI	0.6964	0.7073	0.7964
Ara2012	0.6964	0.6646	0.7256

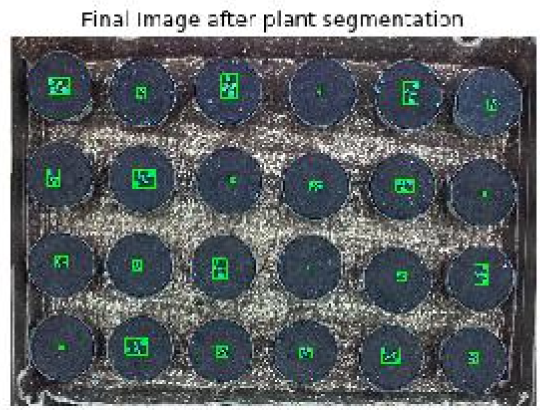


Fig. 11. Plant identified Image

sample tray image, after the plant segmentation is shown in Table I

We see the Average Precision metric for method 1 doing quite good - with a worst value being 0.587 for image, as showcased in Fig. 10. We observe that we have patches of green areas, which cause the segmentation algorithm to either increase area, or to have multiple segments for the same plant. On the best case that we see, we get an Average Precision metric of 0.8225, where we observe that the projected bounding boxes are slightly smaller in size than the ones provided in the data set. This causes the metric to go down ever so slightly to the value we see here.

Fig. 12. The best segmented image

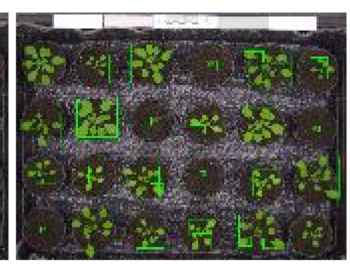


Fig. 13. The worst segmented image

2) Method 2: After executing the variation of the flow, we observe that the bounding boxes for images in data set 2012 are slightly worse off, implying that the median filter has seemingly had a detrimental effect to the plant boundaries, due to a reduction in the size of the plant. On the other hand, we observe a slight increase in the metric, showcasing the presence of Gaussian noise, and the the subsequent reduction of the same due to the pre-processing. As we may observe from Fig. 9 and Fig. 10., the worst metric image has multiple splotches of green areas which were not taken care by with the median filter algorithm. We see overlapping bounding boxes as well as unnecessarily large boxes depicting small splotches of green pixels (noise).

Fig. 14. The best segmented image

Fig. 15. The worst segmented image

3) Method 3: We observe that the addition of the Gaussian Blur method to the methodology increases the metric quite significantly across the board. Most of the miscalculation is because of the noise removal process deteriorating the boundaries of the plants. We observe that in one of the plants of the worst image (Fig. 14), separate leaves are identified as separate plants. We also observe that one plant has been identified by two plants. These miscalculations result in the AP metrics value being 0.7084, a value still higher than the average AP metric for the previous methods.

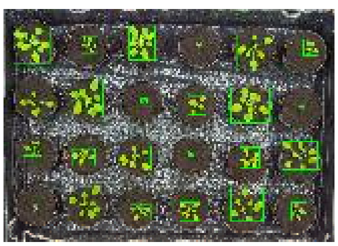


Fig. 16. The best segmented image

Fig. 17. The worst segmented image

B. Task 2 (Projected leaf area measure)

We apply intensity histogram threshold in our computer vision method to all the images and get results as follows in Table II. From the result shown, it's clear that the computer vision methods work better with using median blur filter for our given methods.

TABLE II
TASK 2 DICE SIMILARITY COEFFICIENT (DSC) AND INTERSECTION OVER UNION (IOU) USING HISTOGRAM THRESHOLD IN OVERALL DATA SET

Method for Task 2 for all data sets (Ara2012 and Ara2013)		Dice Similarity Coefficient	Intersection over Union
Computer vision Method	With using Median blur Filter	0.9447	0.8951
	Without using Median blur Filter	0.9389	0.8848

The average score of DSC and IOU are positive. As we can see, the usage of median filter has a higher performance on both DSC and IOU, with about 0.02 more.

TABLE III
TASK 2 DICE SIMILARITY COEFFICIENT (DSC) AND INTERSECTION OVER UNION (IOU) USING BINARY THRESHOLD

Method for Task 2 using binary threshold		Dice Similarity Coefficient	Intersection over Union
Computer vision Method with using median blur	Ara2012	0.8909	0.8035
	Ara2013	0.8885	0.8001

From the Table III, our process works effectively for Ara2012 with a DSC score 0.8909 which is slightly greater than Ara2013/Canon. However, the scores are pretty low compared to the intensity threshold method highlighted before. The main reason behind it is due to the green mask range, highlighting the importance of extracting appropriate green range.

For machine learning method, we have encountered a big problem which is the inability to apply SVM to the whole data set despite the fact that it has the potential to achieve a better result than the computer vision method. The main cause is the huge amounts of data from the data set sent to our SVM model. Given the fact that we are sending every pixel in every image to our SVM model one by one, this results in over 3 billion inputs to both store and make prediction on it. This creates a hardware requirement that none of our devices are unable to afford.

Regardless of the issue, the DSC and IOU performances on one single image (ara2012_tray01_rgb) are 0.9810 and 0.9627 respectively using the machine learning algorithm. Comparing result with the computer vision method, we find that machine learning approach has worked well if we are looking at single image rather than computer vision method, despite the huge hardware requirement.

(a) (b) (c)

Fig. 18. (a) An Arabidopsis Original Image, (b) Its image after threshold, (c) Binary image.

The Figure 18 are the results for the projected leaf area measure for the Arabidopsis plant which are in the tray.The final result show good results.

C. Task 3 (Individual leaf segmentation)

The task was challenging, the metric score in task 3 for both methods is very low with a below-average score compared to other tasks.

1) Method 1 (Using Watershed): The HSV method is shown and proven to be very effective in extracting the plant from the background from task 1 and 2. With that in mind, it's fair to question our watershed segmentation algorithm as the main reason for the downfall of the metric score.

TABLE IV
TASK 3 EVALUATION METRICS SYMMETRIC BEST DICE

Dataset	Symmetric Best Dice
Ara 2012	0.41
Ara 2013-Canon	0.56
Tobacco	0.32
Overall	0.44

There are a number of factors that affects the performance of watershed segmentation. A big factor could be the marker array which contains all of the local maxima representing the centroid of each leaf. We can control the amounts of local maxima via setting the minimum distance required for two maxima to be apart from each other. However, we can't settle on a minimum distance that gives the best result on all test data.

Another factor could be lack of use of filters such as Gaussian or Median filter. However, we experimented with it and determined that it has little to no impact on overall performance of the whole data set.

2) Method 2 (Using super pixel with watershed): We have suggested and shown that the watershed segmentation requires more improvement. In term of super pixel segmentation, there are some noticeable increases in Symmetric Best Dice score when comparing this method with the other method.

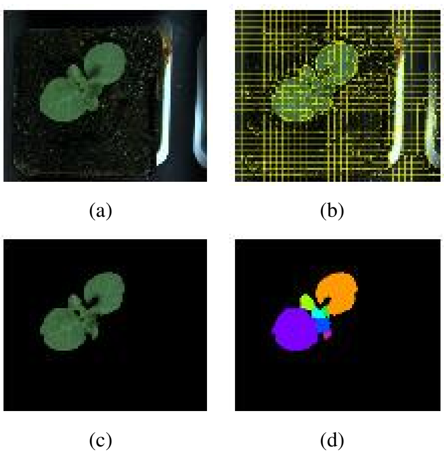

(a) (b) (c) (d)

Fig. 19. (a) Tobacco Original Image, (b)Image after using Superpixel, (c) Removing background according to Superpixel ID, d)Watershed image

The Figure 19 is the result after using super pixel method with watershed method in one of the tobacco plant.The result shows it has labelled prominently and distinguish the leaf properly.

VI. CONCLUSION

A. Task 1 (localize and detect plant)

From our three pre-processing method variations, we observe that the best way includes the removal of the salt and pepper noise via Median filter as well as the Gaussian Noise via Gaussian Filter. By implementing the connected components algorithm and merging the close by components, we are able to identify and place bounding boxes over plants with a reasonable degree of certainty.

B. Task 2 (Projected leaf area measure)

As a conclusion of task 2, we think the best approach is to remove the background of images in HSV colour space and then convert images into gray scale colour space, where we apply intensity threshold to label each pixel.If we use maximum range in green range, it was removing the feature that we are interest and if we used the low range it gives salt and pepper noise.Though we tuned within the medium range.The background of images contain pot, moss.If we can eliminate those noise it gives good result thus in our case histogram threshold was working properly rather than binary threshold. Using support vector machine algorithm to classify each pixel has a better result but requires enormous space and very long time. In the future, we can try machine learning method again with better computation power.

C. Task 3 (Individual leaf segmentation)

We tried different segmentation method such as watershed, super pixel segmentation, canny edge detection, finding centre of leaves, calculating distances between each of the leaves. But in most of the cases, it doesn't perform well. The hardest part in this task is to find the leaves centre as the leaves for different area has different color range, in centre it has brightest of all,and most of the leaves are overlapping to.Due to this reason must of the methods were failing.The satisfactory results we found was with the super pixel methods with watershed segmentation.For most plants, the challenging part was to locate the centre of the leaves as leaves were in various colour map and highly overlapping which make us undesired result from the segmentation method. Only super pixel with watershed gives satisfactory result still the score are pretty low. For future work, we want to try to use the deep neural network like Convolutions Neural Network in hope that it can finds individual leaves better than our current method.

VII. CONTRIBUTION OF GROUP MEMBERS

TABLE V
CONTRIBUTION OF GROUP MEMBERS

	Arpit	Asha	Gordon	Navya	Zhihan
Task 1	✓			✓	
Task 2		✓	✓		✓
Task 3	✓	✓	✓	✓	
Report	✓	✓	✓	✓	✓
Presentation	✓	✓	✓	✓	✓

A. Arpit Mathur

- Designed and Developed the First Task code
- Tried implementing Deep Learning methods for the third task
- Worked on the presentation and report (Introduction, Literature Review, Methods, Results, Conclusions and Discussions and Further Work

B. Asha Karki

- Designed and developed the second and third task code
- Worked on Report(Abstract, Methods, Experimental Setup, Results, Conclusions and Discussions (Task 2 and Task 3),added in Further Work components and reviewed the whole report)
- Worked on presentation

C. Gordon Chen

- Worked on second and third task code and wrote its report
- Organised, designed and created demo and presentation
- Final reviewed the whole report before submission

D. Navya Vashisht

- Designed and Developed the First Task code
- Developed methods to solve task 3(Grab cut and watershed algorithm with segmentation)
- Worked on report for task1, task3 and presentation

E. Zhihan Liu

- Wrote codes for Task 2 , including both methods
- Wrote most part of the report for Task 2
- Prepared the demo and presentation for Task 2

VIII. Further Work

For the Machine Learning approaches we can also augment the data set being fed into the model by rotating, skewing, mirroring, adding noise and sharpening the images in the data set [19]. We could also utilize Conditional Generative Adversarial Networks to artificially generate training images for out model [20].

We could further work on detecting the plants in the images via the YOLOv5, a real-time object detection model which would enhance the work done in all tasks [21]. By utilizing this model to draw bounding boxes around the plants themselves, we segment the images into individual plants, for them to be passed onto the methods depicted above.

In order to ensure generalized of our scripts, we can also implement them on Crop/Weed Field Image Dataset [22]. The data set contains field images, segmented masks as well as plant type annotations.

Also we can use Deep Neural network,in the plant phenotype.Computer Neural network (CNN) can be used for leaf count,for bounding boxes which gives good accuracy and other than we can used Deep plant phonemic platform,this platform provides deep learning functionality for plant phenotype [23]. This platform has the predefined and pretrained models to lead in image classification,object detection,localization,image segmentation multi threading instances. Beside that PlantCV method can be used.PlantCV is composed of modular functions in order to be applicable to a variety of plant types and imaging systems [24]. We could work using PlantCV packages for segmenting,localize, multi instances problem in future.

References

[1] "Phenotype," Oct 2020. [Online]. Available: https://en.wikipedia.org/wiki/Phenotype

[2] M. Minervini, A. Fischbach, H. Scharr, and S. Tsaftaris, "Plant phenotyping datasets," 2015. [Online]. Available: http://www.plant-phenotyping.org/datasets

[3] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," *Pattern Recognition Letters*, pp. –, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865515003645

[4] "Introduction." [Online]. Available: https://www.plant-phenotyping.org/datasets-home

[5] S. Aich, "Object counting with deep learning," Ph.D. dissertation, University of Saskatchewan Saskatoon, 2019.

[6] A. Kendall, V. Badrinarayanan, , and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[8] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[9] D. Li, Y. Cao, G. Shi, X. Cai, Y. Chen, S. Wang, and S. Yan, "An overlapping-free leaf segmentation method for plant point clouds," *IEEE Access*, vol. 7, pp. 129 054–129 070, 2019.

[10] W. Baig, Luca, A. Rosebrock, Deven, C. Viehoff, R. Garcia, J. Marotta, Brian, Ovi, David, and et al., "Segmentation: A slic superpixel tutorial using python," Apr 2020. [Online]. Available: https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/

[11] "Superpixels." [Online]. Available: https://docs.opencv.org/3.4/df/d6c/group__ximgproc__superpixel.html

[12] P. D. Team, "Watershed segmentation." [Online]. Available: https://plantcv.readthedocs.io/en/stable/watershed/

[13] [Online]. Available: https://docs.opencv.org/master/df/d9d/tutorial_py_colorspaces.html

[14] "Morphological transformations." [Online]. Available: https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html

[15] "Structural analysis and shape descriptors." [Online]. Available: https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html

[16] "Smoothing images¶." [Online]. Available: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html

[17] L. X. G. X. Z. C. Zhang, Yongxia, "A simple algorithm of superpixel segmentation with boundary constraint," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. pp, 2016. [Online]. Available: 10.1109/TCSVT.2016.2539839

[18] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, X. Yin, and S. A. Tsaftaris, "Leaf segmentation in plant phenotyping: a collation study," *Machine Vision and Applications*, vol. 27, no. 4, pp. 585–606, 05 2016.

[19] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020, online; accessed 01-Feb-2020.

[20] Y. Zhu, M. Aoun, M. Krijn, J. Vanschoren, and H. T. Campus, "Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants." in *BMVC*, 2018, p. 324.

[21] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[22] S. Haug and J. Ostermann, "A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks," in *Computer Vision - ECCV 2014 Workshops*, 2015, pp. 105–116. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-16220-1_8

[23] J. R. Ubbens and I. Stavness, "Deep plant phenomics: A deep learning platform for complex plant phenotyping tasks," *Frontiers in plant science*, vol. 8 1190, 2017. [Online]. Available: doi: 10.3389/fpls.2017.01190

[24] [Online]. Available: https://plantcv.readthedocs.io/en/latest/multi-plant_tutorial/